

De Kracht van een Framework



Machinebouw

12 december 2023, Congrescentrum 1931

GAIN

Automation
Technology

Even voorstellen

Rudi Schilder

[Rudi Schilder | LinkedIn](#)



Marco Conrads

[Marco Conrads | LinkedIn](#)



Machinebouw

12 december 2023, Congrescentrum 1931

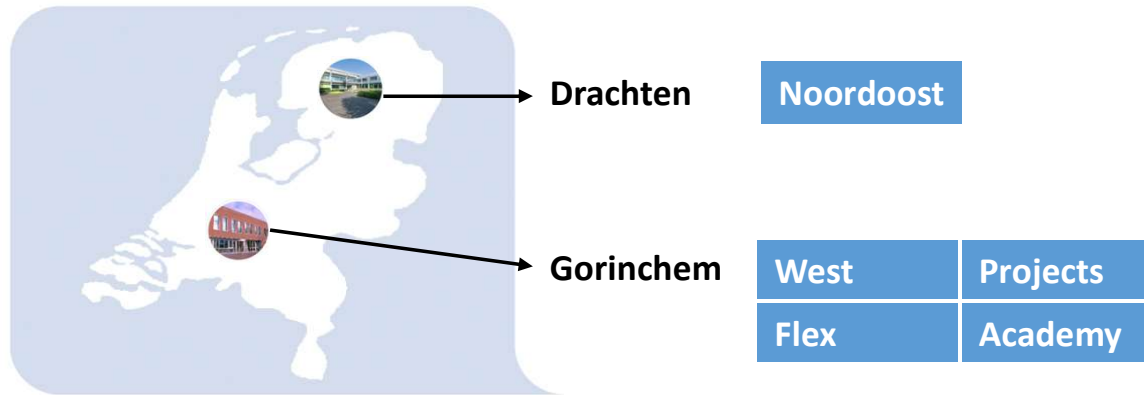
GAIN

Automation
Technology

Introductie Gain

Missie:

Gain verbetert de productie- en projectresultaten van haar opdrachtgevers door oplossingen en diensten te leveren op het gebied van technische en industriële automatisering.



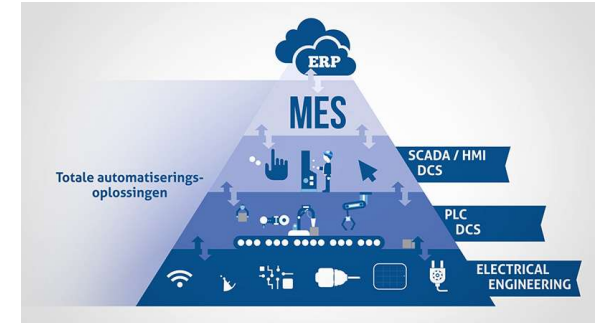
Machinebouw

12 december 2023, Congressentrum 1931

Introductie Gain

Markten

- Machinebouw
- Voeding, Farma & Fijn Chemie
- Water, infra & energie
- Olie, gas & chemie
- Maritiem, scheepvaart & offshore



Diensten

- Hardware Engineering
- Besturingssoftware
- Visualisatie & bediening
- Productie informatisering



Machinebouw

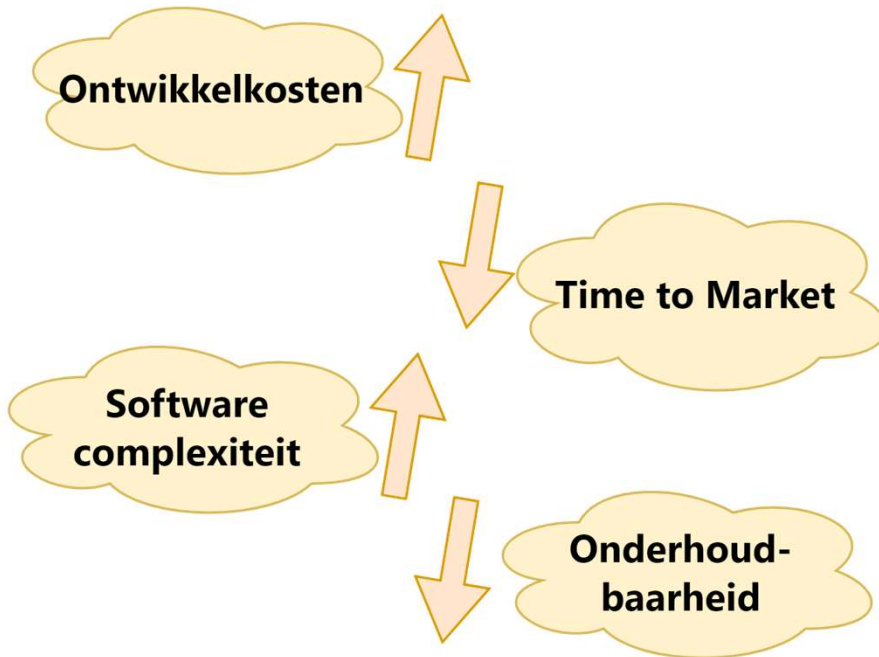
12 december 2023, Congrescentrum 1931

gain

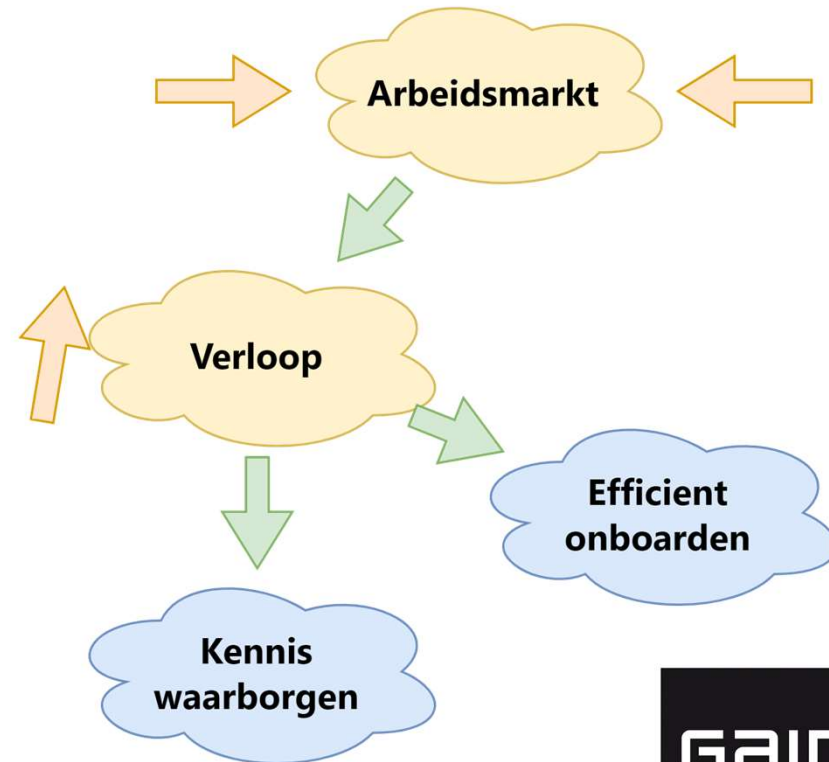
Automation
Technology

Welke uitdagingen zien we?

Product



Medewerkers

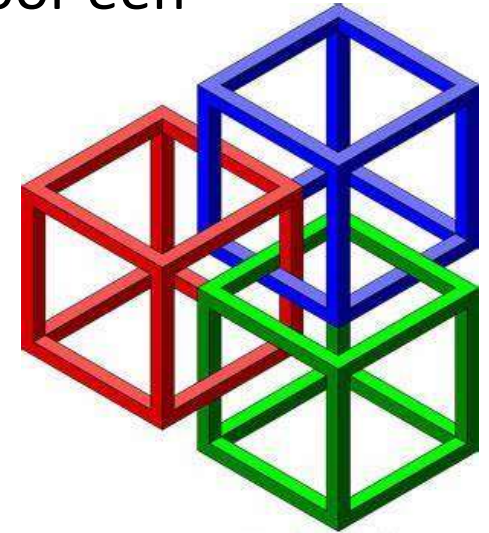


De oplossing

“Een framework is de omgeving dat bedoeld is als basis voor een software applicatie”

"Niet telkens het wiel opnieuw uitvinden."

"Dwingt tot werken volgens standaarden"



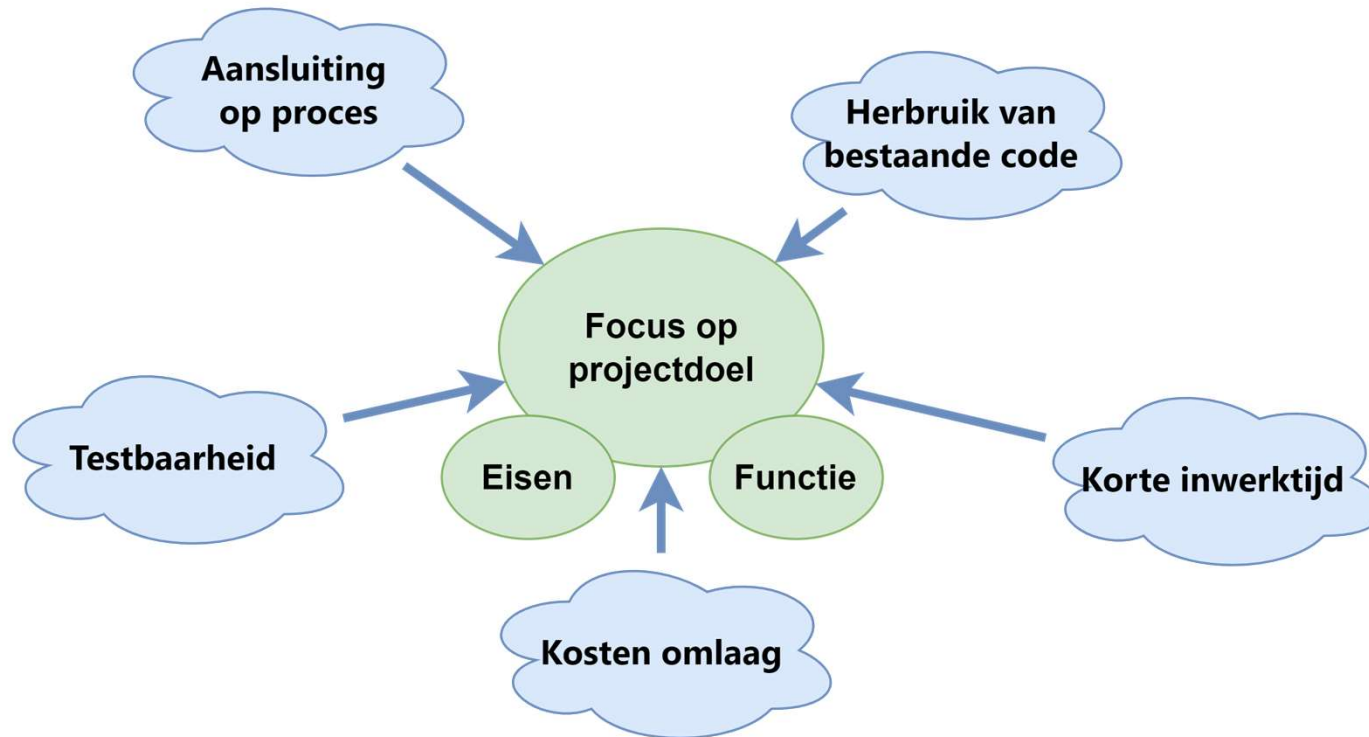
Machinebouw

12 december 2023, Congrescentrum 1931

GAIN

Automation
Technology

Voordelen framework



Machinebouw

12 december 2023, Congrescentrum 1931

GAIN

Automation
Technology

Aandachtspunten framework

- Eerste opzet en ontwikkeling
- Onderhoud
- Complexiteit
- Versiebeheer
- Documentatie



Machinebouw

12 december 2023, Congrescentrum 1931

GAIN

Automation
Technology

Framework componenten

Denk aan:

- Control modules/drivers
- Receptuur- / parameterbeheer
- Alarmmanagement
- Software architectuur/raamwerk
- Veel gebruikte algoritmen
- Koppeling met HMI, databases etc.
- Documentatie



Machinebouw

12 december 2023, Congrescentrum 1931

GAIN

Automation
Technology

Framework ontwikkeling

- Opstellen naming conventions
- Duidelijke afspraken maken over coding guidelines en documenteren
- Kiezen voor een architectuur bijvoorbeeld bijvoorbeeld PackML of ISA-88
- Introductie van Git / versiebeheer
- Bestaande code herstructureren/herschrijven
- Libraries maken en beheren



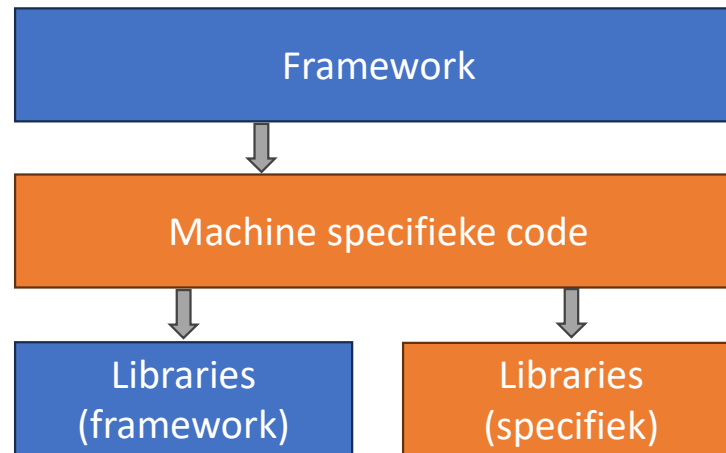
Machinebouw

12 december 2023, Congrescentrum 1931

GAIN

Automation
Technology

Voorbeeld Gain framework



Machinebouw

12 december 2023, Congrescentrum 1931

gain

Automation
Technology

Geschiedenis Gain framework

- Ooit begonnen met gezamenlijke stijlguide
- Toepassen bestaande standaarden (PackML, PLCopen, ISA88)
- Hergebruik van blokken die daarop aansloten
- Verzamelen in libraries
- Documenteren
- Opzetten versie beheer tooling (libraries -> SVN -> GIT)
- PackML templates ontwikkelen
- Onderhoud



Machinebouw

12 december 2023, Congrescentrum 1931

gain

Automation
Technology

Uitgangspunten Gain framework

- Focus op machinebouw
- PackML als standaard
- Versiebeheer: Git
- Technologie: Beckhoff OOP



OMAC
PackML



Machinebouw

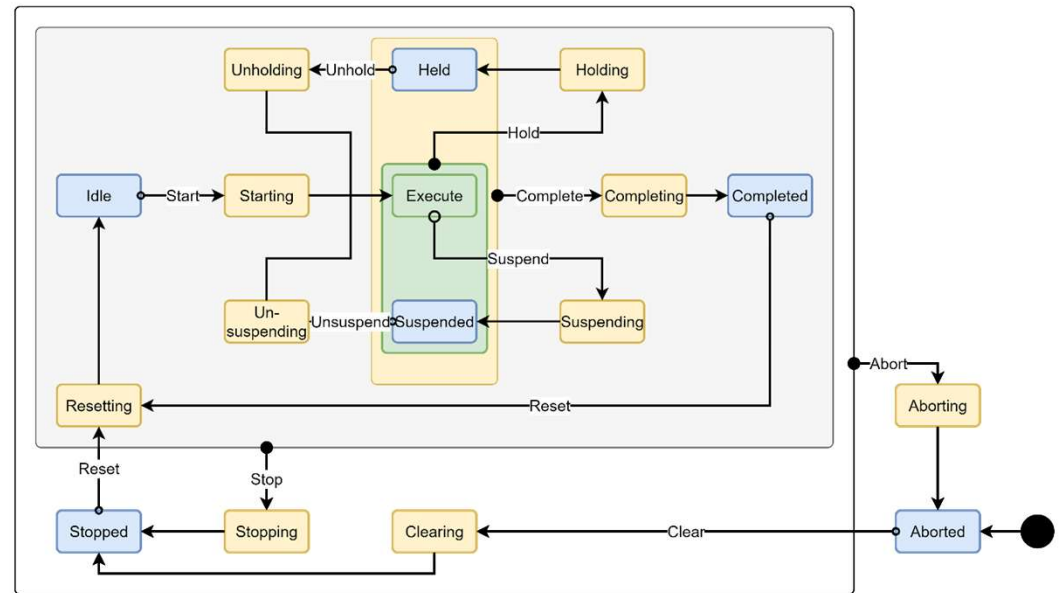
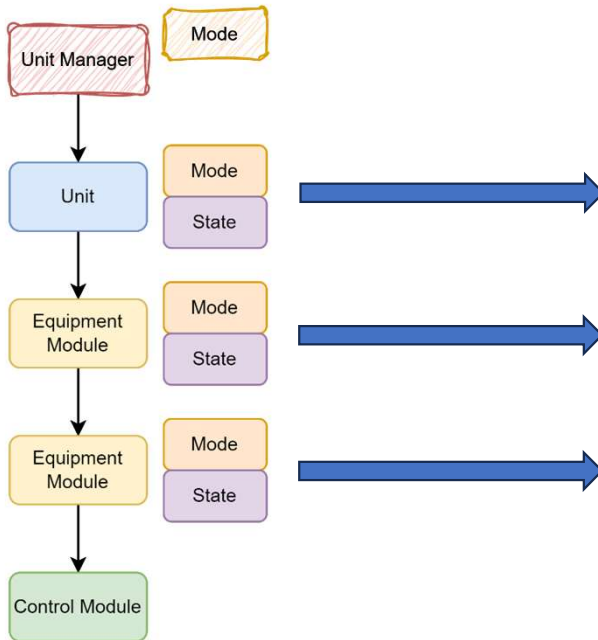
12 december 2023, Congrescentrum 1931

gain

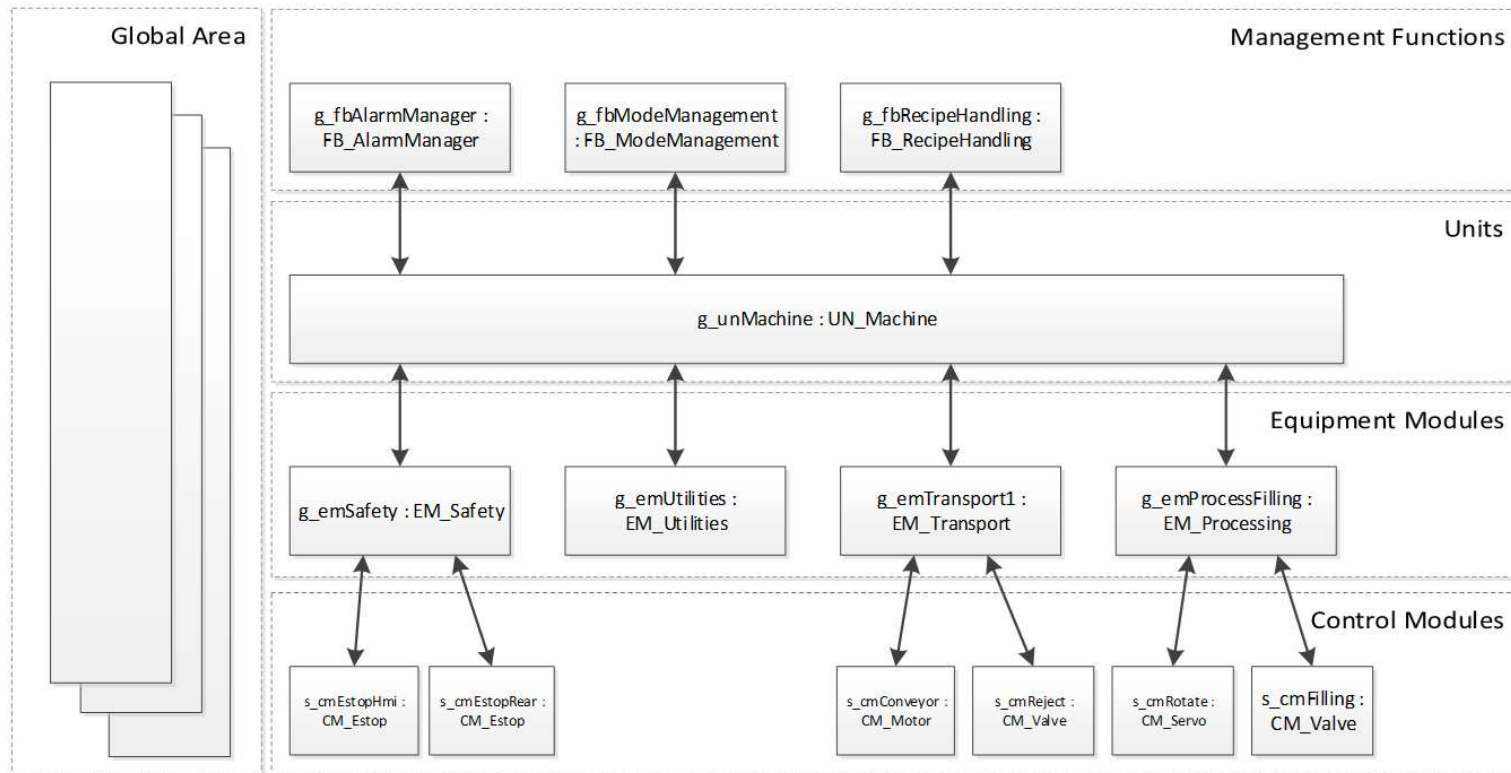
Automation
Technology

Architectuur

PackML programma stuctuur PackML state machine

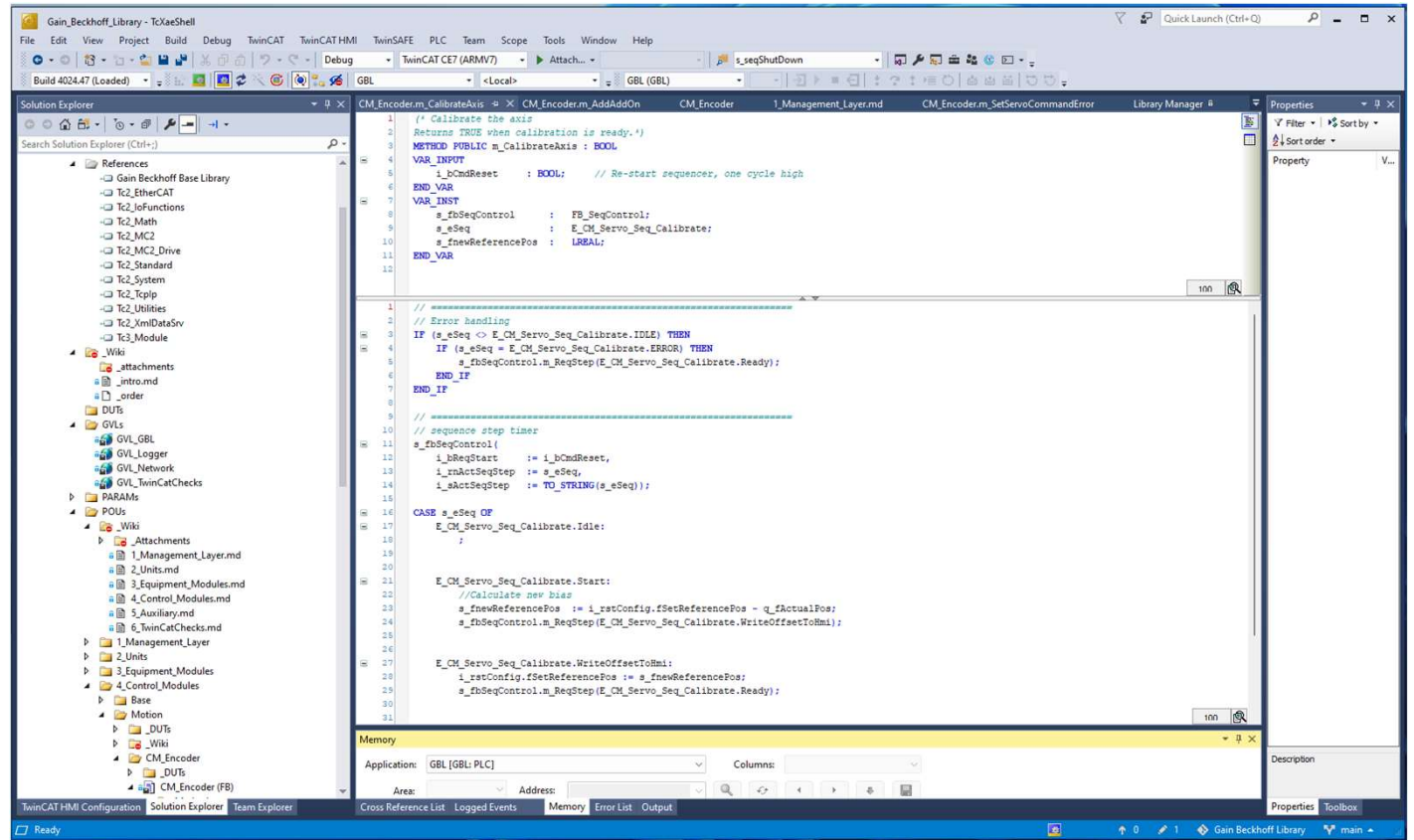


Architectuur



Coderen

- Libraries
- Framework
- Tooling
- Inrichten GIT



Machinebouw

12 december 2023, Congressentrum 1931



Functies

Framework	Control modules	Motion	Auxiliary
Modecontrol	Kleppen	Encoder	Logging
PackML statemachine	Discrete motor	Virtuele servo	Filters
Sequencers	Freq. Motor	AX5000 servo	Stacklights
Base class Unit		AX8000 servo	Motion calculations
Base class Equipment module		XTS servo	Matrix calculations
Base class Control module		Camming	EtherCAT diagnostics
Alarm management		Flying saw	HMI interfacing



Machinebouw

12 december 2023, Congrescentrum 1931

GAIN

Automation
Technology

Coding guidelines

- Naming Conventions
- Coding patterns
- Software structure

Object name	Type	Example usage
PRG_Main	Program	Main PLC program.
GVL_Constants	Global variable list	Global area containing common constants
DB_Constants	Data block	Data block containing common constants
F_fLimit	Function	Function to limit floating point variables
FB_SeqTimer	Function block	Function block to measure step times.
EM_Conveyor	Equipment module	
F_EM_Conveyor_SeqEvents	Equipment module	Events occurrence steps for the equipment

No.	Region	Description
1	Init	Unit initialization, set initial data.
2	Inputs	Process unit inputs.
3	State handling	Determine PackML states and modes for all sub-modules
4	Modules	Call sub-modules (equipment and/or control modules)
5	Alarms	Collect alarm conditions from sub-modules.
6	Outputs	Process unit outputs.

```
s_nValue1 := 50 + 20; // Correct code
s_fbFunction1(i_bReqStart := TRUE); // Correct code and preferred.
s_fbFunction2(i_bReqStart:=TRUE); // Correct code but not preferred.
```

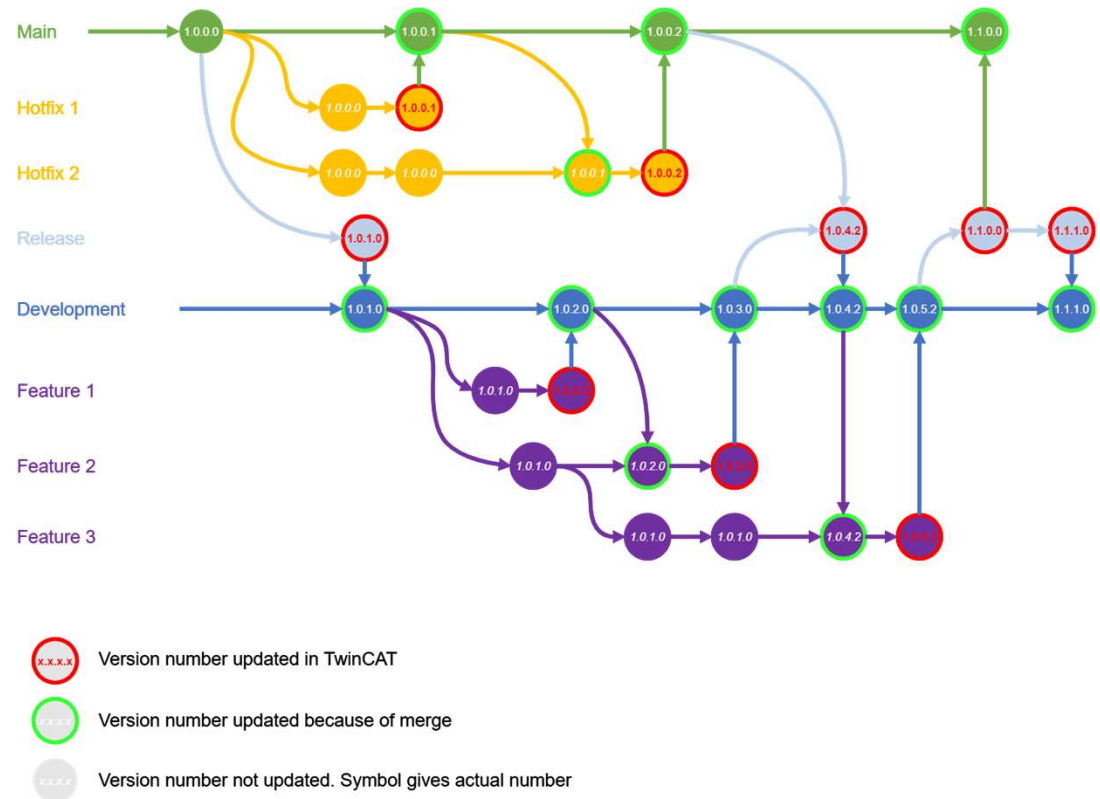
wards HMI.



Versiebeheer

- Main branch
 - Hotfixes
- Development branch
 - Features
- Versienummering

Team kan gelijktijdig ontwikkelen



Project beheer

DevOps

- Backlog
 - Requirements
 - Taken
 - Bugs
- Sprint planning

The screenshot shows a Jira backlog and Kanban board for a project named 'PLC Libraries'. The backlog is filtered by 'Control modules general' and shows various work items with their states and business areas. The Kanban board below shows the workflow of these items, categorized into Proposed, Active, Resolved, and Closed.

Order	Work Item Type	Title	State	Effort	Busin...	Value Area	Tags
1	Category	PLC Libraries	Active				
	Epic	Gain Beckhoff Library (GBL)	Active			Business	
	Feature	Gain Beckhoff General Library	Active			Business	
	Feature	Software Structure	Proposed			Business	
	Feature	Control Modules	Proposed			Business	
	Requirement	Control modules general	Proposed			Business	
	Task	Control modules: Hoe goed te simuleren?	Proposed				
	Task	Control modules: Emulatie methode at all CM's	Proposed				
	Task	Convert Private methods to Protected methods	Proposed				Discussion
	Requirement	CM_MotorDOL	Proposed			Business	
	Requirement	CM_MotorVFD_HW	Proposed			Business	
	Requirement	CM_MotorVFD_Lenze_i5xx	Proposed			Business	
	Requirement	CM_Servo_Group	Proposed			Business	
	Requirement	CM_Encoder/Servo	Proposed			Business	
	Feature	Alarm Management	Proposed			Business	

Proposed	Active	Resolved	Closed
<ul style="list-style-type: none"> 4614 CM_Servo (Marco Conrads, Active) 4624 Upgrade Alarm management to Beckhoff eventhandler structure (Marco Conrads, Active) 4632 FB_SeqControl (Marco Conrads, Active) 4640 Bugs Gain 2023 (Marco Conrads, Proposed) 	<ul style="list-style-type: none"> 4617 Update according PLC-open guidelines (Gert van Lagen) 4619 Servo Addons, some improvements (Marco Conrads) 4625 Create new alarm handling (John de Jaeger) 4638 Replace obsolete FB_SeqTimer with FB_SeqControl (Marco Conrads) 4641 All links in wiki class diagram are broken (Marco Conrads, Completed) 		<ul style="list-style-type: none"> 4637 FB_SeqControl Create status methods (Marco Conrads)



Machinebouw
12 december 2023, Congressentrum 1931

Documentatie

- Genereren vanuit code
- Wiki

Public-Gain-Beckhoff-Librarywiki

Filter pages by title

- POUs
 - 1_Management_Layer
 - 2_Units
 - 3_Equipment_Modules
 - 4_Control_Modules
 - Base
 - Motion
 - _DUTs
 - CM_Encoder
 - CM_Encoder_L...
 - _DUTs
 - CM_Servo_Base
 - CM_Servo_Virtual
 - CM_ServoAXS000
 - CM_ServoAXS000
 - CM_ServoEL72xx
 - CM_ServoStepper
 - Motor
 - Valve
 - 5_Auxiliary
 - 6_TwinCatChecks
 - FB_ProgramBase (FB)
 - GBL_Camming library
 - GBL_Flyingsaw library
 - GBL_HMI library
- How to
- New page

```

// Alarm
s_AlarmFailure1 : FB_AlarmHandler;
s_AlarmFailure2 : FB_AlarmHandler;
    
```

DUTs

ITF_EM_XXX

Add properties to the EM-interface needed at the HMI_EM_XXX_Gain block, so it can be shown at the HMI. This usually includes the interfaces of all CMs.
Example:
image.png

ST_EM_XXX_Config

Add configuration variables and structs which:

- must be editable during runtime by operator.
- structs from lower level functions (like CM's, IO functions and alarms) which may have the incorrect default values.

Allow manual control

Write 's_bAllowManualControl', so manual control from HMI is allowed. Preferred location is at the method _m_Logic.
For example:

```

// Allow manual control from HMI when the actual mode is 'Manual' and actual state is 'Execute'
s_bAllowManualControl := s_FBPackML_p_ActState = E_PackML_Mode_Manual AND s_FBPackML_p_ActState = E_PackML_State_Execute;
    
```

Enable of the EM

An EM can be enabled/disabled by s_bStoEnable, which is set by L_bReqEnable. This input does nothing by default, but can be used to:

- Disable the alarm collector, so alarms are not changing PackML state and alarms can not be activated.
- Disable the PackML, so the PackML methods like _m_Execute and _m_Starting are never called even when those states are active.

For example at _m_Logic:

```

// Allow alarming when module is enabled
s_bModuleAlarmCollector_Enable := s_bStoEnable;
// Allow PackML when module is enabled;
s_FBPackML_p_ActEnabled := s_bStoEnable;
    
```

Disable all alarms at once

Write 's_bReqAlarmCollector_Enable', so the alarm collector can disable all alarm handlers when this variable is inactive. Preferred location is at the method _m_Logic.
TRUE.
For example:

```

// Disable all alarm handlers when the EM is disabled.
s_bModuleAlarmCollector_Enable := s_bStoEnable;
    
```

Simulation

To implement simulation use the following variables:

Public-Gain-Beckhoff-Library / Overview / Wiki / PLC Library documentation / GBL library / POU's / 4_Control_Modules / Motion / CM_Encoder / CM_Encoder (FB)

Public-Gain-Beckhoff-Librarywiki

Filter pages by title

- Introduction
- Gain PLC Standard
 - Best practices
- PLC Library documentation
 - GBL library
 - GVLS
 - PARAMS
 - POUs
 - 1_Management_Layer
 - 2_Units
 - 3_Equipment_Modul...
 - 4_Control_Modules
 - Base
 - Motion

```

+ q_nActualTurnModulo: DINT
+ q_nDAxis: POINTER TO AXIS_REF
+ q_nEncoderState: ST_AxisStatus
+ q_nParameterSet: ST_AxisParameterSet
+ FB_CM_Base
+ l_bAllowManualCtrl: BOOL = FALSE
+ l_bReqEnable: BOOL = FALSE
+ l_bReqReset: BOOL
+ l_bReqSimulationOn: BOOL = FALSE
+ l_bAlarmCollector_Parent: ITF_AlarmCollector_Connector
+ q_bStoEnabled: BOOL
+ q_nStoAlarms_CollectedAction: ST_AlarmCollector_ActioDates
+Property
+ p_bAllowManualCtrl: BOOL
+ p_nActualTorque: LREAL
+ p_nDriveErrorCode: UDINT
+ p_nNCEErrorCode: UDINT
+ p_nStoAlarmsParameters: REFERENCE TO ST_AxisParameterSet
+ p_nStoAlarmsRef: REFERENCE TO AXIS_REF
+ p_nStoConfig: REFERENCE TO ST_CM_Servo_Config
+ p_nStoPctOn: REFERENCE TO PCTONC_AXIS_REF
+ p_nStoState: REFERENCE TO ST_AxisStatus
+ p_nStoTpc: NCTOPLC_AXIS_REF
+ FB_CM_Base
+ p_nAlarmCollector: ITF_AlarmCollector
+ FB_Base
+ p_nStancePath: T_MaxString
+ m_AggAddOn()
+ m_CalibrateAxis(): BOOL
+ m_FastTask(): BOOL
+ m_Main()
+ m_NCReadParam(): BOOL
+ m_NCWriteParam(): BOOL
+ m_NCWriteParam(): BOOL
+ m_ResetReference(): BOOL
+ m_SetActualPosition(): BOOL
+ m_SetReference(): BOOL
+ m_SetServoCommandError()
+ m_UpdateAxisState()
+ m_WriteEncoderSettings(): BOOL
+ FB_CM_Base
+ m_AddHmi(): BOOL
+ m_Main()
+ FB_Base
+ m_Main()
    
```

Class Diagram

Inheritance/Extension

Inheritance Diagram:

```

graph LR
    CM_Encoder --> FB_CM_Base
    FB_CM_Base --> FB_Base
    
```

CM_Encoder --> FB_CM_Base --> FB_Base

Interface/Implementation

Implementation Diagram:

```

graph LR
    CM_Encoder --> ITF_CM_Encoder
    
```

CM_Encoder --> ITF_CM_Encoder

Inputs/Outputs

VAR_INPUT

```

VAR_INPUT
    l_nStoConfig : REFERENCE TO ST_CM_Servo_Config ; (* Reference to configuration/settings of CM_Encoder *)
    s_PctOn : STRING(30) ; (* Postfix added to alarm value *)
END_VAR
    
```

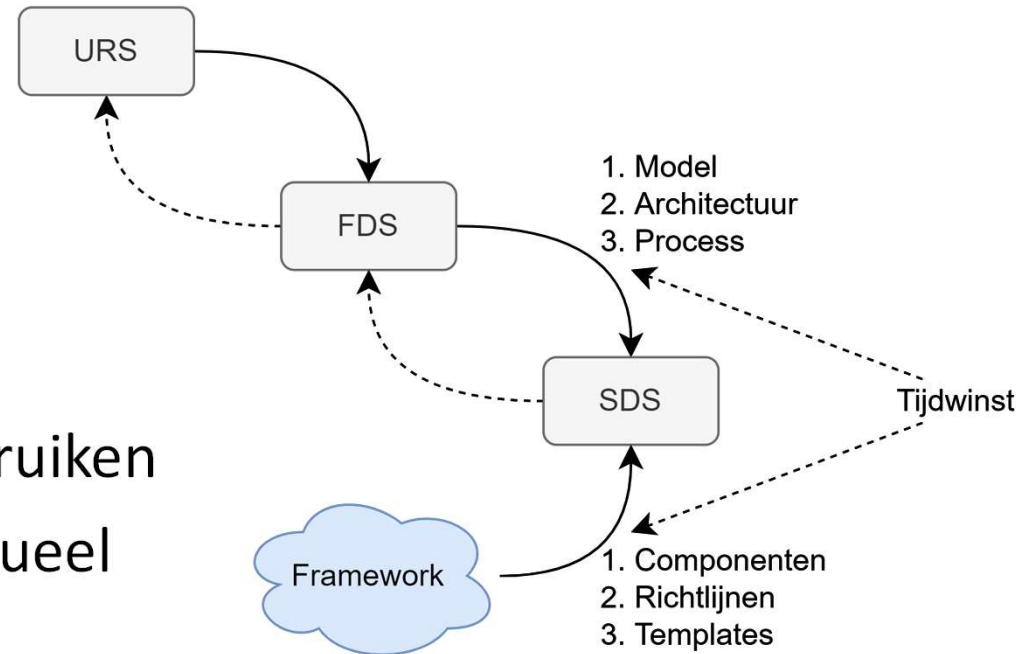


Hoe implementeren wij dit?

Specificatie documentatie

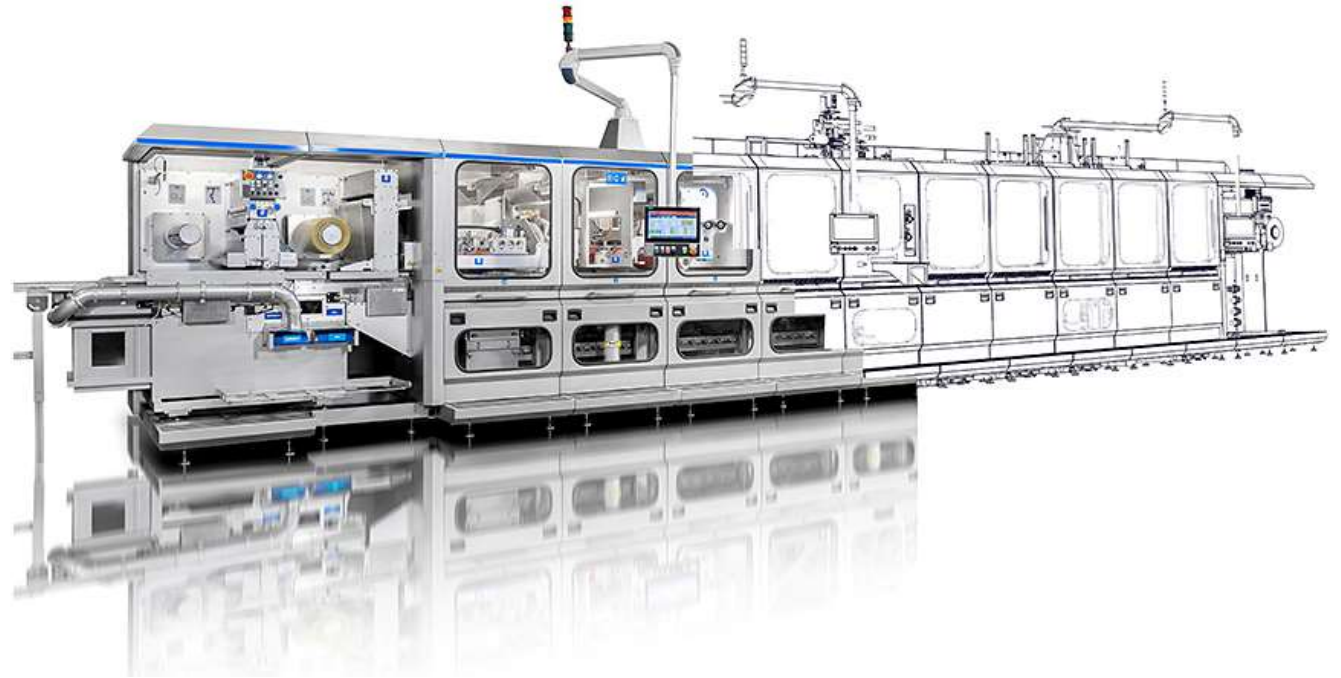
→ sluit aan op framework

- URS
- FDS: PackML filosofie aanhouden
- SDS: Framework componenten gebruiken
- Code: Framework invullen en eventueel uitbreiden



Voorbeeld

 **Tembo**



Machinebouw

12 december 2023, Congrescentrum 1931

GAIN

Automation
Technology

Voorbeeld



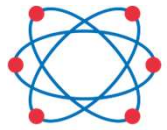
Machinebouw

12 december 2023, Congrescentrum 1931

GAIN

Automation
Technology

Voorbeeld



CREMER

UHLMANN GROUP



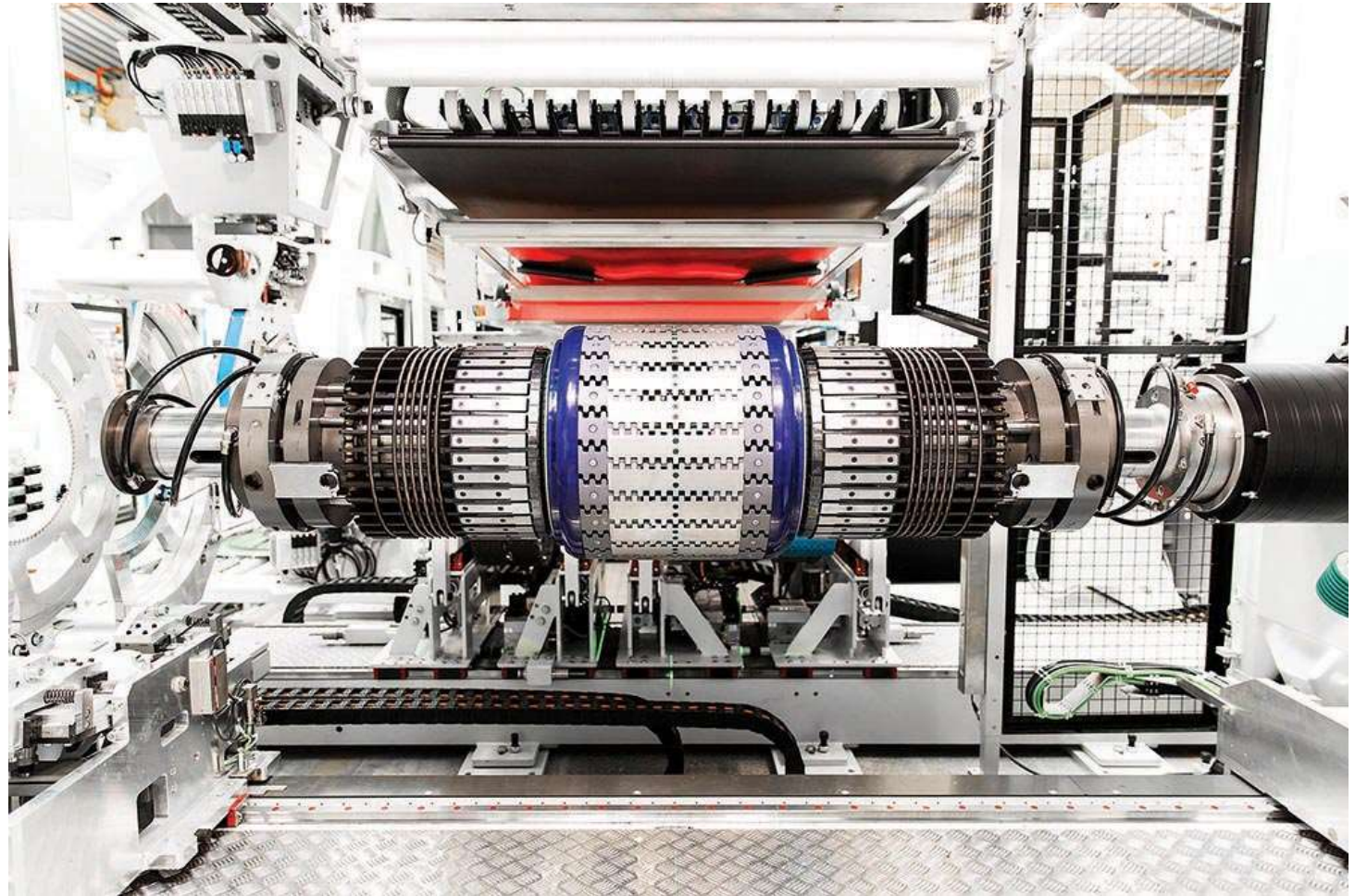
Machinebouw

12 december 2023, Congrescentrum 1931

GAIN

Automation
Technology

Voorbeeld



Machinebouw

12 december 2023, Congrescentrum 1931

GAIN

Automation
Technology

Bedankt voor uw aandacht

Zijn er vragen?



Machinebouw

12 december 2023, Congrescentrum 1931

GAIN

Automation
Technology

We zien jullie graag bij de borrel!



Machinebouw

12 december 2023, Congrescentrum 1931

GAIN

Automation
Technology

Alle tips op een rij

- Informatie delen van werk gedaan voor meerdere projecten/klanten
- Zoek de componenten die je wil hergebruiken
 - Motion, kleppen, aansturing, communicatie
- Zoek de functionaliteit die je altijd hebt
 - HMI
 - Alarmen
 - Modes, state machine en algemene logica
- Blijf flexibel! Custom zaken blijven komen. Het gaat om je engineers helder laten communiceren
- Blijf toepassingsgericht.
- Framework blijft een basis die je iedere keer weer opnieuw erbij pakt.
 - NIET het vorige project.
- Versiebeheer toepassen vanaf het eerste moment



Machinebouw

12 december 2023, Congrescentrum 1931

GAIN

Automation
Technology