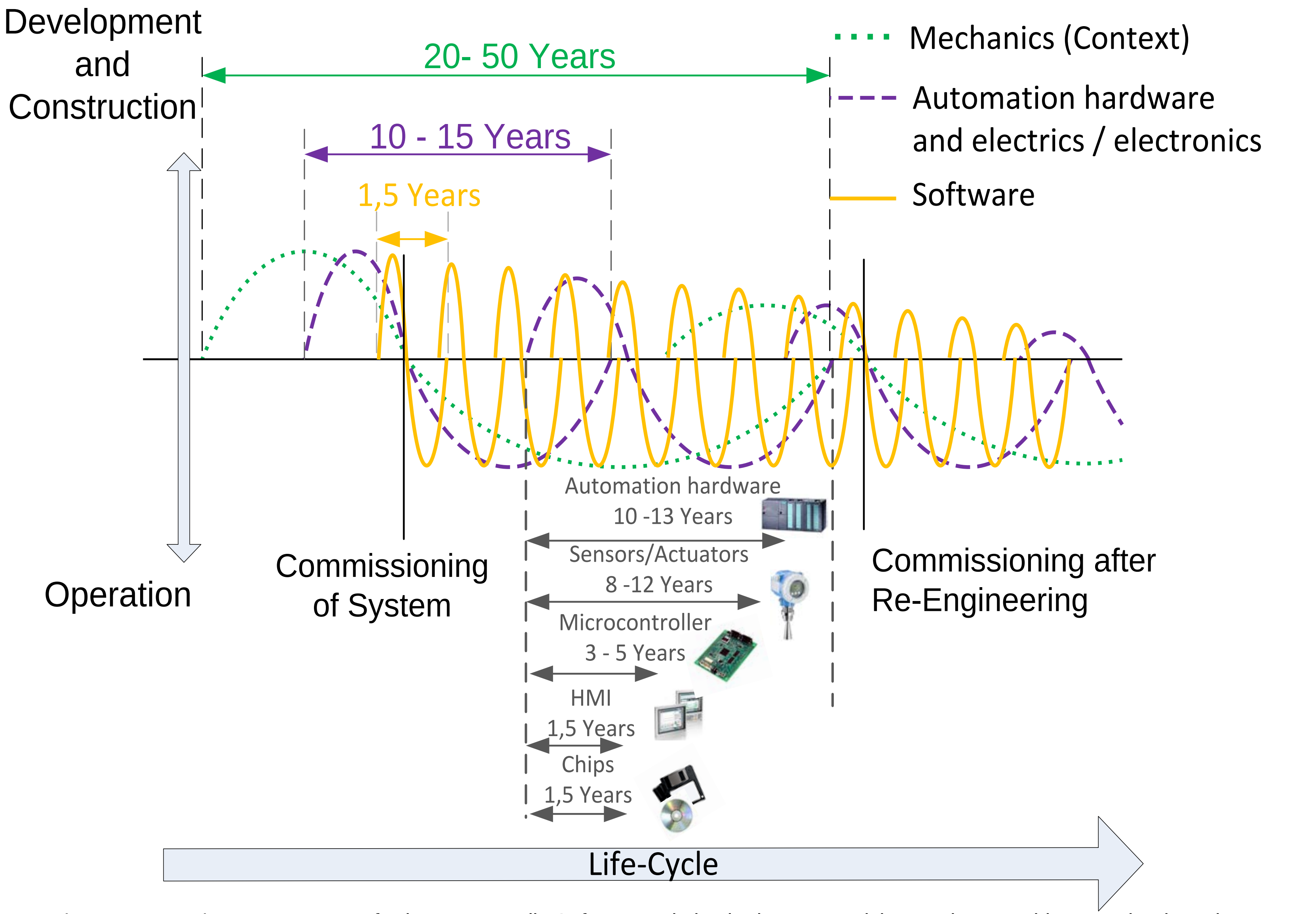
Transferring Technical debt to automated Production Systems (aPS)

1. **Domain specific constraints**
2. Types of Technical Debt
3. Causes of Architectural Technical Debt in aPS
4. ATD- Parallel development – Pick&Place Unit
5. Accumulation and Recovery Models
6. Conclusion and Outlook

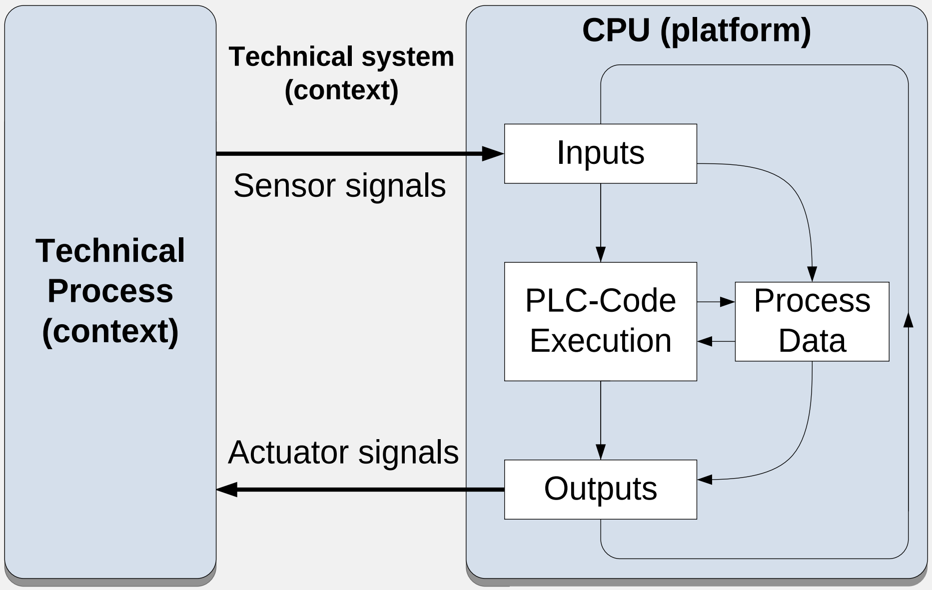
Source: Bayer AG, Leverkusen

Source: Siemens AG

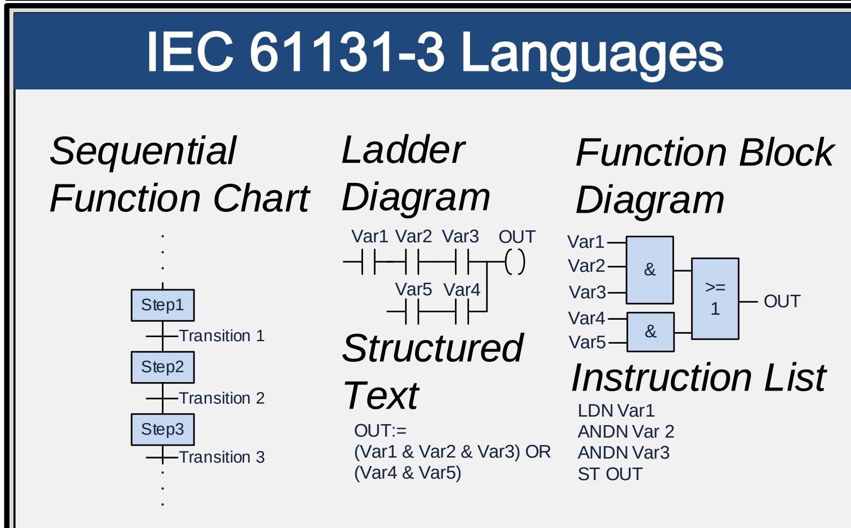
Univ.-Prof. Dr.-Ing. Birgit Vogel-Heuser
vogel-heuser@tum.de
Head of Chair and Director of Institute, Automation and Information systems (AIS)
Faculty of Mechanical Engineering, Technische Universität München



Source: B. Vogel-Heuser, J. Folmer, C. Legat: *Anforderungen an die Softwareevolution in der Automatisierung des Maschinen- und Anlagenbaus*. In: at – Automatisierungstechnik, 62(3), 3/2014



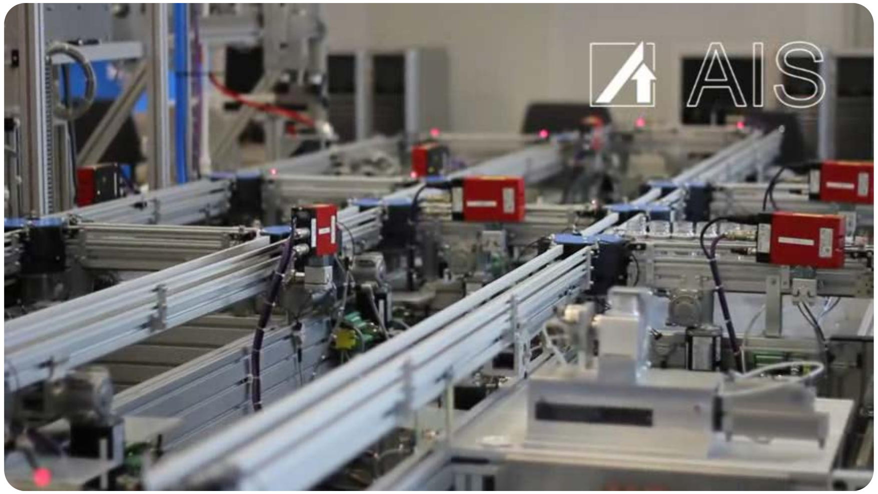
- Real-time requirements of aPS
  → hard real-time for the used platform PLC
- Cyclic behavior of the platform (1µs – 1s)
- Classical PLC as well as Soft-PLC (PC-based) programmed in IEC 61131-3 Languages
- Increasing amount of IPC and C, C-derivatives
- *Online change is mandatory*

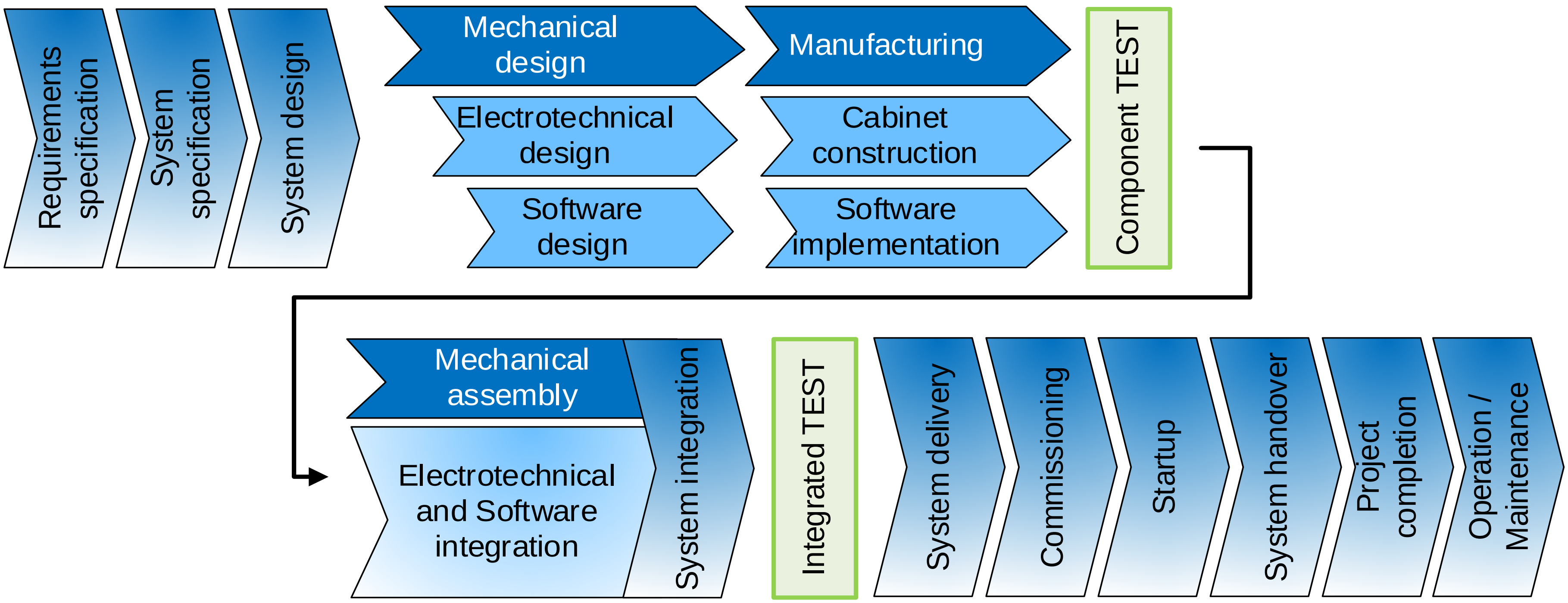


IEC 61131-3 Programming Languages

- Proprietary programming languages: Structured Text (ST), Ladder Diagram (LD), Instruction List (IL), Sequential Function Chart (SFC), Function Block Diagram (FBD)
- **Upcoming: C**

Life cycle phases for aPS





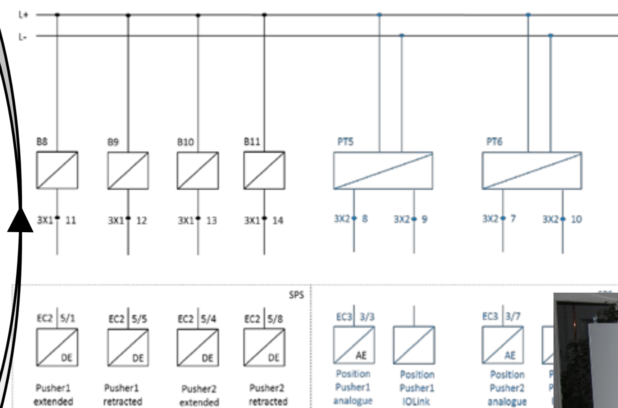
Enlarged Types of Technical Debt (TD) in aPS according to the enlarged life cycle model

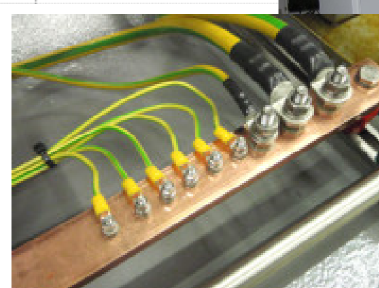
Enlarged Types of Technical Debt for aPS based on (Li et al. 2015)

- Requirements TD = textual description in contract plus layout
- Architectural TD = different architectural approaches and artifacts in disciplines
- Design TD
- Code TD / Manufacturing TD
- Test TD
- Build TD
- Assembly TD
- Documentation TD
- Infrastructure TD
- Versioning TD
- Defect TD
- Commissioning TD / Start Up TD
- Operational & Maintenance TD

Design (el.) = → Circuit / Wiring diagram
→ Terminal connection table

Manufacturing (el.) = → Mounting of devices in the cabinet




Source: systech-gmbh.de


Source: durau.de

© AIS

Effect: sporadic faults

Types of Technical Debt (Li et al. 2015)

- Requirements TD
- Architectural TD
- Design TD
- Code TD
- Test TD
- Build TD
- Documentation TD
- Infrastructure TD
- Versioning TD
- Defect TD

Causes of Architecture Technical Debt (ATD) (Martini et al. 2014)

- Business factors
- Design and architecture documentation
- Reuse of Legacy and third party code
- Parallel development
- Effects uncertainty
- Non-completed refactoring
- Technology evolution
- Human factors

Business factors:

Business evolution

- New application domain, e.g. food & beverage
- Varying standards in different countries e.g. CSA, UL, VDI/VDE
- Different environmental conditions e.g. humidity, temperature, etc.

Time pressure

- ✓ Always heavy penalties for late delivery on-site, acceptance as well as plant availability

Split of budget

- Lack of cross-discipline responsibilities, different teams for each discipline
- Lack of budget for software maintenance

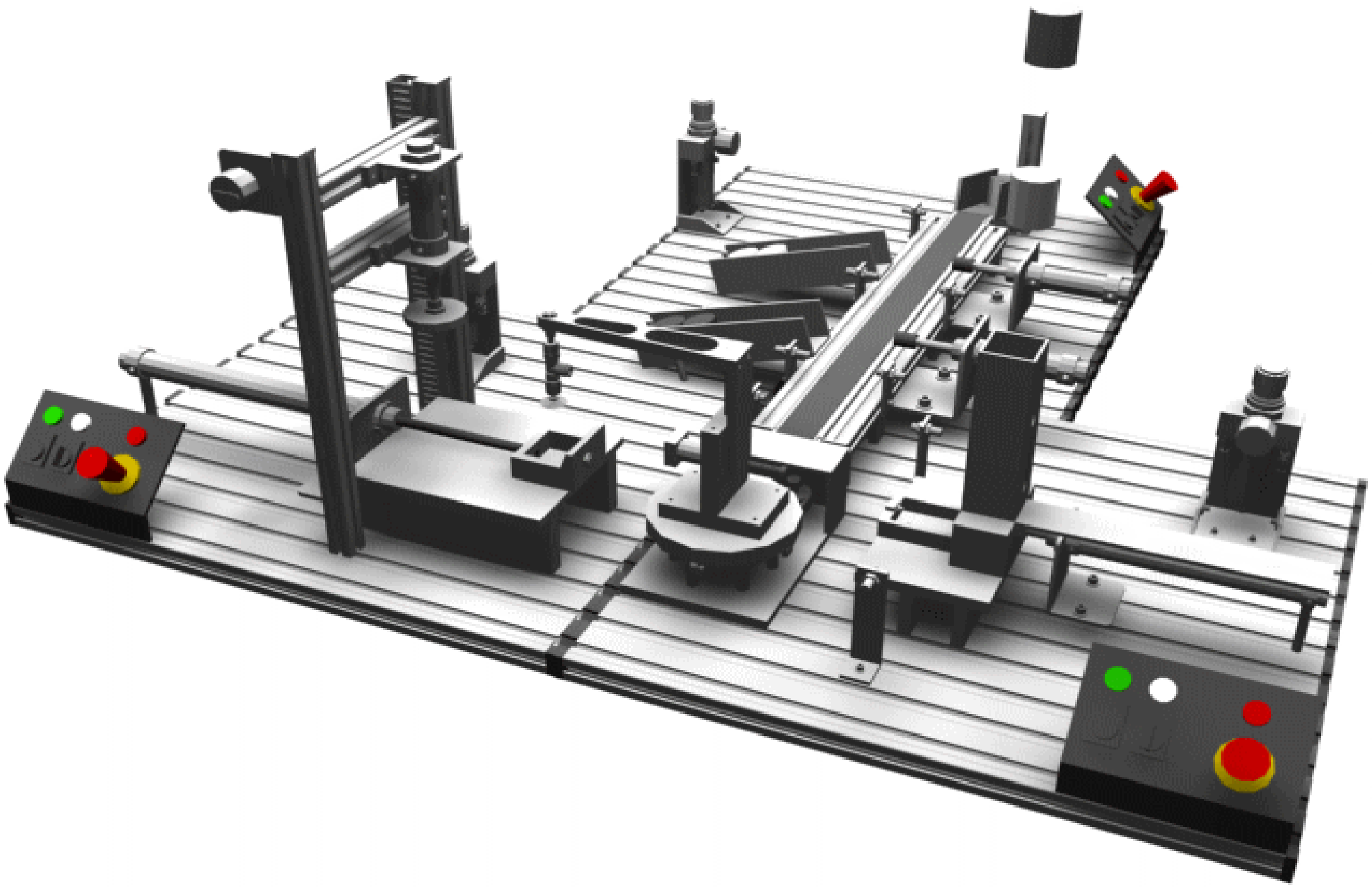
Sources: Z. Li, P. Avgeriou, P. Lang: *A systematic mapping study on technical debt and its management.* In: The Journal of Systems and Software, pp. 293-220, 2015.
A. Martini, J. Bosch, M. Chaudron: *Architecture Technical Debt: Understanding causes and a qualitative model.* In: Conference on Software Engineering and Advanced Applications, pp. 85-92, 2014

Causes and effects of ATD in aPS – selected examples

Causes of Architecture Technical Debt (ATD) (Martini et al. 2014)

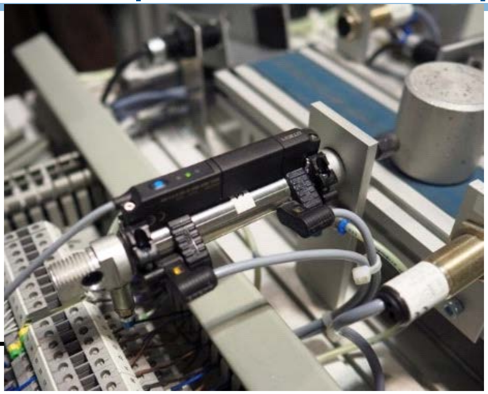
causes	Sub-causes	effects
Design and architecture documentation	• Only very rough textual specification given in contract	• Lack of clear specification, misunderstanding -> inappropriate interfaces or functionality
Technology evolution	• Change of PLC-, drives or HMI-platform, heterogen. fieldbus	• Interoperability problems
Reuse of Legacy and third party code/components	• Outsourcing hardware and/or software development due to lack of development capacity/resources	• Poor quality (tools), violation of company standards, further development usage impossible
Effects uncertainty	• Changes have unforeseen effects on other discipline	• Change of motor, requires different frequency converter
Non-completed refactoring	• Need to maintain backward compatibility for a decade	• Hinders new architectures and structures
Human factors	• Not invented here phenomenon • Gap of knowledge between commissioning and start-up staff and design staff	• New designed solutions on-site (costly) • Violation of internal standards • inappropriate solutions from design department
Parallel development	⇨ See application example	

ATD parallel development



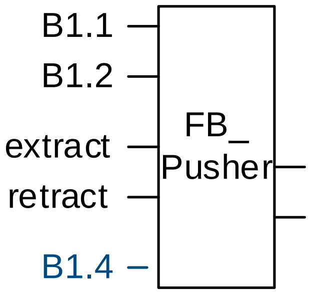
© AIS

Hardware change and software change: additional sensor during start-up onsite parallel development

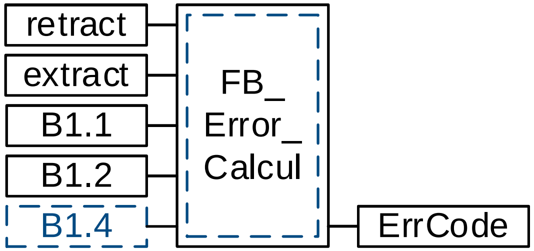


Machine group	Device number	Device	Function	Location	Device/Signal type	Power supply [V]	Remarks
310	M1	sorting line	sort/push WP in slide 1	Pusher	pneum. valve DO	24V	mand.
310	B1.1	sorting line	pusher is extended	Pusher	reed switch DI	24V	mand.
310	B1.2	sorting line	pusher is retracted	Pusher	reed switch DI	24V	mand.
320	M1	sorting line	sort/push WP in slide 2	Pusher	pneum. valve DO	24V	mand.
…	…	…	…	…	…	…	…

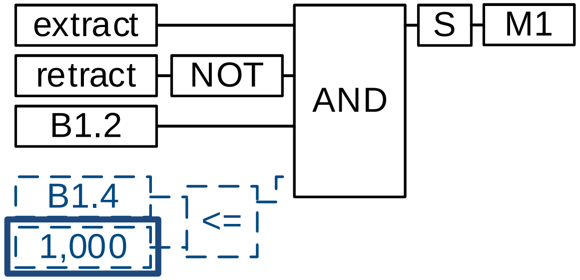
FB Interface



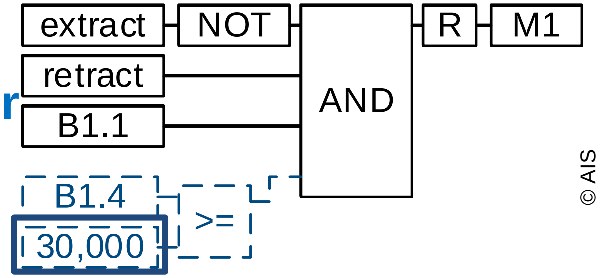
Functionality Error Code Calculation



Functionality Extract Pusher



Functionality Retractf Pusher



- ***Crisis-based ATD management***
  - software tends to be unreadable and unmaintainable or lead to unpredicted behavior
  - refactoring is unavoidable but not planned – leading to a weak solution

- ***ATD accumulation and recovery during feature development***
  - ATD grows with every modification of software or electrics not compliant with valid explicit or implicit rules
    - during optimization of existing plants
    - during interdisciplinary development of new plants

- ***Events initiating ATD recovery***
  - *development of a new machine or machine generation (continuous product change)*
  - *based on a new technology*
  - *different market requirements (products), e.g. thinner or thicker particle boards*
  - *different tools, e.g.,* the introduction of a new engineering tool in electrical engineering or software engineering
  - the change of a team leader
  - the limitations of a numbering system e.g. for MCL implicitly representing the variants

- ✓ Concepts of TD and ATD are in principle applicable to aPS
- ✓ Dimensions, some causes and some effects were introduced
- ✓ Challenges of interdisciplinary relations

- ➢ Should lead to a deeper understanding of obstacles for a systematic evolution of aPS

- ➢ Management strategies to deal with TD and ATD can be developed in future work focusing on the plant manufacturing industry
- ➢ Need for industrial case studies to gain more data and classify different ATD recovery models

- Z. Li, P. Avgeriou, P. Lang: *A systematic mapping study on technical debt and its management.* In: The Journal of Systems and Software, pp. 293-220, 2015.
- B. Vogel-Heuser, J. Folmer, C. Legat: *Anforderungen an die Softwareevolution in der Automatisierung des Maschinen- und Anlagenbaus*. In: at – Automatisierungstechnik, 62(3), 3/2014.
- E. Tom, A. Aurum, R. Vidgen: *An exploration of technical debt.* In: The Journal of Systems and Software, pp. 1498-1516, 2013.
- A. Martini, J. Bosch, M. Chaudron: *Architecture Technical Debt: Understanding causes and a qualitative model*. In: Conference on Software Engineering and Advanced Applications, pp. 85-92, 2014