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

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
aPS as long living systems

Development and Construction

Operation

Commissioning of System

Commissioning after Re-Engineering

20-50 Years

10-15 Years

1.5 Years

Automation hardware

Mechanics (Context)

Automation hardware and electrics/electronics

Software

1.5 Years

10-13 Years

Automation hardware

Sensors/Actuators

8-12 Years

Microcontroller

3-5 Years

HMI

1.5 Years

Chips

1.5 Years

Life-Cycle

Technical Constraints of software for aPS

- 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

- Requirements specification
- System specification
- System design
- Mechanical design
- Electrotechnical design
- Software design
- Component TEST
- Manufacturing
- Cabinet construction
- Software implementation
- Mechanical assembly
- Integrated TEST
- System integration
- Electrotechnical and Software integration
- System delivery
- Commissioning
- Startup
- System handover
- Project completion
- Operation / Maintenance
- System delivery
- Commissioning
- Startup
- System handover
- Project completion
- Operation / Maintenance

B. Vogel-Heuser
23.05.2015
Enlarged Types of Technical Debt (TD) in aPS according to the enlarged life cycle model

- Requirements TD
- Architectural TD
- Design TD
- Code TD
- Manufacturng 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, Mounting of devices in the cabinet
- Manufacturing (el.) = Circuit / Wiring diagram, Terminal connection table, Mounting of devices in the cabinet

Effect: sporadic faults

Source: systech-gmbh.de
Source: durau.de

© AIS
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 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.
  - Always heavy penalties for late delivery on-site, acceptance as well as plant availability
- **Time pressure**
- **Split of budget**
  - Lack of cross-discipline responsibilities, different teams for each discipline
  - Lack of budget for software maintenance

Sources:

23.05.2015  B. Vogel-Heuser
<table>
<thead>
<tr>
<th>Causes</th>
<th>Sub-causes</th>
<th>Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design and architecture documentation</td>
<td>• Only very rough textual specification given in contract</td>
<td>• Lack of clear specification, misunderstanding -&gt; inappropriate interfaces or functionality</td>
</tr>
<tr>
<td>Technology evolution</td>
<td>• Change of PLC-, drives or HMI-platform, heterogen. fieldbus</td>
<td>• Interoperability problems</td>
</tr>
<tr>
<td>Reuse of Legacy and third party code/components</td>
<td>• Outsourcing hardware and/or software development due to lack of development capacity/resources</td>
<td>• Poor quality (tools), violation of company standards, further development usage impossible</td>
</tr>
<tr>
<td>Effects uncertainty</td>
<td>• Changes have unforeseen effects on other discipline</td>
<td>• Change of motor, requires different frequency converter</td>
</tr>
<tr>
<td>Non-completed refactoring</td>
<td>• Need to maintain backward compatibility for a decade</td>
<td>• Hinders new architectures and structures</td>
</tr>
<tr>
<td>Human factors</td>
<td>• Not invented here phenomenon</td>
<td>• New designed solutions on-site (costly)</td>
</tr>
<tr>
<td>Parallel development</td>
<td>• Gap of knowledge between commissioning and start-up staff and design staff</td>
<td>• Violation of internal standards</td>
</tr>
</tbody>
</table>
ATD parallel development
Hardware change and software change: additional sensor during start-up onsite parallel development

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

Functionality

- **Extract Pusher**
  - B1.1 AND B1.2
  - NOT B1.1

- **Retract Pusher**
  - B1.1 AND B1.4
  - NOT B1.1

```
Functionality Error Code Calculation

<table>
<thead>
<tr>
<th>FB Interface</th>
<th>ErrCode</th>
</tr>
</thead>
<tbody>
<tr>
<td>retract</td>
<td>extract</td>
</tr>
<tr>
<td>B1.1</td>
<td>Error</td>
</tr>
<tr>
<td>B1.2</td>
<td>Calcul</td>
</tr>
</tbody>
</table>

ErrCode

<table>
<thead>
<tr>
<th>extract</th>
<th>retract</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1.2</td>
<td>NOT</td>
</tr>
</tbody>
</table>

AND

<table>
<thead>
<tr>
<th>S</th>
<th>M1</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>retract</th>
<th>extract</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>M1</td>
</tr>
</tbody>
</table>

AND

<table>
<thead>
<tr>
<th>B1.4</th>
<th>1,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;=</td>
<td>&lt;=</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B1.4</th>
<th>30,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;=</td>
<td>&lt;=</td>
</tr>
</tbody>
</table>
- **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

