MOOSE Seminar Keynote

“State of the practice in European embedded software engineering”

Rini van Solingen

Moose Project Manager
Principal Consultant in SW Management, LogicaCMG
Professor in Quality Management & Engineering, Drenthe University

Rini.van.Solingen@LogicaCMG.com

http://www.mooseproject.org/
Agenda

- Introduction
- MOOSE web-repository
- Analysis of state-of-the-practice
- Strategies for increasing “deployability”
- Conclusions

Introduction
Embedded SW Market

- Product is sold, not software
- Dominant hardware restrictions (memory, timing)
- Strongly based on previous products
- Increasing amount of software
- Application of software engineering technology

Embedded Software Market

Number of embedded systems per household

http://www.mooseproject.org/
Challenges embedded sw market

- Faster development
- More functionality with increasing complexity
- Increasing quality and performance demands
- Financial pressure on product cost, mostly hardware only

- Increasing demands on business drivers
- Technological innovation as a solution
  - Hardware technology
  - Software engineering technology
  - SE technology: methods, techniques, tools, and processes

Innovation = New Technology + Usage

- Innovation includes usage
  - Technology development without usage is NOT innovation
  - Many research initiatives focus on technology development only
  - Technology adoption by industry often lasts long
- Two types of innovation
  - Initial innovation
    - First time development and industrial application
  - Evolutionary innovation
    - Continuous improvement from application experiences
- Moose facilitation through experience exchange
  - Sharing what worked and did not work in which situation and why
MOOSE web-repository

http://www.mooseproject.org/
MOOSE web-repository

- Project centred storage and exchange of experiences
- Project characterisation
  - Effort, persons, lead-time, business driver, etc.
- Product characterisation
  - Software, Hardware, Real-time criticality, Market
- Technology characterisation
  - SE technology used
  - Satisfaction with the technology for that project (scale)
  - Short textual reason for this satisfaction rating
- Project evaluation report
  - Optional attachment with detailed findings and experiences

Content of the web-repository

- Currently: 89 projects included (analysis based on 78)
- Projects originate from voluntary submission
  - Registered users can enter project experiences
  - Evaluation board evaluates submission on completeness and reliability
  - Registered users can contact project owners
  - Unregistered users can only browse anonymous projects and technologies
- Analysis of the web-repository
  - Provides insight in state-of-the-practice
  - First idea on trends
  - Limited validity of findings and conclusions

http://www.mooseproject.org/
Analysis of the web-repository

What is the state-of-the-practice?

Project characterisation (1)
[Business driver and Spread]

- **Main business driver**
  - 80% of the projects are driven by schedule or functionality demands
  - 70% of the projects are undertaken on 1 site with max. 2 teams

- Number of sites
- Number of teams

http://www.mooseproject.org/
**Project characterisation (2)**

**[Lead-time and Effort]**

- **Lead-time (months):**
  - <3
  - 4-6
  - 7-12
  - 13-24
  - >24
  - 60% of the projects have a duration of <1 year
  - 80% of the projects have a duration of <2 years

- **Effort (person years):**
  - <1
  - 1-5
  - 6-10
  - 10-50
  - >50
  - 50% of the projects cost less than 5 person years of effort
  - 5% more than 50 py

---

**Product characterisation (1)**

**[Type and Real-time criticality]**

- **Product Type:**
  - Professional
  - Consumer
  - OEM
  - 50% have soft or not

- **Real-time criticality:**
  - Hard real-time
  - Soft real-time
  - Not real-time
  - 50% of the products have hard real-time constraints
  - 50% have soft or not
Product characterisation (2)  
[SW Size and HW Size]

- 60% of the products have more than 100,000 lines of software code

- 35% of the products have memory boundaries below 1 Mbyte
  - 30% have over 16 Mbyte

Product characterisation (3)  
[SW and HW]

- 30% of the products have MsWindows operating system

- 30% of the products are Intel/PC-based
Technology Characterisation

What is the state-of-the-practice in SE technology application?

Which Requirements Engineering Methods are used?

- None used
- Proprietary method
- RequisitePro
- Problem Frame approach
- Interviews
- Use cases
- Other

55% of the products are developed without use of a RE method.

http://www.mooseproject.org/
Which Requirements Engineering Tools are used?

- MsWord: 20%
- No tool used: 25%
- RequisitePro: 5%
- MsExcel: 10%
- Other: 5%

50% of the projects use MsWord/Excel for RE
20% use dedicated RE tools

Which Design methods are used?

- None used: 30%
- UML: 25%
- Proprietary method: 15%
- SA/SD: 10%
- Object Oriented Design: 5%
- UML-RT: 5%
- Other: 5%

30% of the products is designed without use of a method
25% by use of UML
Which Design tools are used?

- No tool used: 25%
- MsVisio/Visual studio: 10%
- Rational Rose: 15%
- MsWord: 5%
- Rational Rose RT: 5%
- Prosa: 5%
- Other: 5%

25% of the products is designed without use of a tool
20% by a generic drawing tool

Which coding languages are used?

- C: 30%
- C++: 25%
- Assembler: 20%
- Java: 15%
- Other: 5%

55% of the products is made in C/C++
20% in Assembler
15% in Java
Which Test Tools are used?

**Test Tooling**

- No tool used/manual testing
- Proprietary
- QAC/C++
- Hitex Debugger
- PurifyPlus
- GNU tooling
- TLCS Debugger
- Hardware test benches
- ConTest/TestFrame
- Other

30% of the products is manually tested.

Which Configuration Management tools are used?

**Configuration Management Tooling**

- CMSynergy
- ClearCase
- PVCS
- No tool used/manual CM
- VisualSourceSafe
- CVS
- Proprietary tool
- Other

90% of the products is developed using a CM tool.
Which PR/CR handling tools are used?

- No tool used/manual: 40% of the projects use a PR/CR tool
- Other: 20% have no tool
- MsExcel, ClearDDTS, Lotus Notes, Proprietary tools: 20% use a generic tool

Which Management processes are installed

- ISO900x: 35% of the projects use CMM for process management
- CMM level 2
- Proprietary
- CMM level 3
- None used
- Other
Which Engineering process is used?

![Engineering process diagram](image)

- **Incremental**: 50% of the products is developed by an incremental engineering process.
- **Waterfall**
- **Proprietary**
- **Time-boxing**
- **Parallel development**
- **No engineering process**
- **Other**

Overview over technologies and SE

<table>
<thead>
<tr>
<th>SE Technology</th>
<th>Value</th>
<th>Most Used</th>
<th>Most Satisfied</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management process</td>
<td>★★★★☆</td>
<td>ISO900x</td>
<td>CMM level 2</td>
</tr>
<tr>
<td>Engineering process</td>
<td>★★★★☆</td>
<td>Incremental</td>
<td>Incremental</td>
</tr>
<tr>
<td>RE method</td>
<td>★★★★☆</td>
<td>No method</td>
<td>Interviews / Use-cases</td>
</tr>
<tr>
<td>Design method</td>
<td>★★★★☆</td>
<td>No method</td>
<td>UML-RT</td>
</tr>
<tr>
<td>Programming language</td>
<td>★★★★☆</td>
<td>C/C++</td>
<td>C/C++</td>
</tr>
<tr>
<td>RE tooling</td>
<td>★★★★☆</td>
<td>MsWord</td>
<td>RequisitePro</td>
</tr>
<tr>
<td>Design tooling</td>
<td>★★★★☆</td>
<td>No tool</td>
<td>Rose RT</td>
</tr>
<tr>
<td>Test tooling</td>
<td>★★★★☆</td>
<td>No tool</td>
<td>HW tools / ConTest</td>
</tr>
<tr>
<td>CM tooling</td>
<td>★★★★☆</td>
<td>CMSynergy</td>
<td>CMSynergy</td>
</tr>
<tr>
<td>PR/CR tooling</td>
<td>★★★★☆</td>
<td>No tool</td>
<td>ChangeSynergy</td>
</tr>
</tbody>
</table>
Summary state-of-the-practice

- **Product characterisation**
  - Most products contain more than 100,000 lines of software code
  - 70% of products have memory boundaries above 1 Mbyte
  - Half of the products have limited or no real-time constraints
  - 30% of the products have MsWindows OS and are Intel/PC-based

- **Project characterisation**
  - Most projects are driven by schedule or functionality, have a duration of <2 years, and are undertaken on 1 geographic location
  - Half of the projects cost less than 5 person years of effort

- **Technology characterisation**
  - Half of the products are built without a RE method using MsWord/Excel
  - 30% of the products is designed without a method, 25% by use of UML.
  - Half of the products are programmed in C/C++
  - Almost all products are developed using a configuration management tool

Opportunities for innovation

- Real-time specific tools score well, but are hardly used. Maturisation of RT specific tools might be interesting
- No method or tool support common for Requirements Engineering. RE seems most promising improvement area for embedded systems
- CMM most actual used and appreciated, so adoption of CMM-I seems opportunity
- Introduction of embedded test tooling, integrated with HW test tooling potential improvement area
- Integration of technologies continues to be large opportunity, however, highly context dependent. Default integrated tool set based on most used technologies could be interesting for industry
**Reasons for not using innovative technologies**

- Legacy in technologies is leading
- Time-pressure in project does not leave time for new things
- Immaturity of new technologies and complexity (learning investment) is too high
- Risks for new technology introduction are too high
- Benefits of technologies are not clear upfront, guarantees are not given
- Experiences or measurements are hardly available
- Sentiment
- Deployment is major challenge in industry

**Increasing “Deployability”**

- Maturity assessment of technologies
- Impact specification of technologies
- Interfacing for technology chains
- Measurement and exchange of experiences
- Variations of technologies to application domains
- Increased collaboration between technology providers and users
- Paradigm shift in SW engineering research:
  - from revolution to evolution
  - from introduction to maturisation
  - from development to evaluation
Conclusions

State-of-the-practice

- Large gap between available and industry used technologies
- Industry acts conservative towards SE technologies
  - Often no methods used
  - Mainly generic tools used
  - Proven technology is used at low risk
  - Pragmatic approaches
- Industry is often not able to make rigorous changes
- So: **Minimal changes but with maximal results**
  - Not a revolution strategy towards innovation
  - Evolution strategy towards innovation
  - However, this is not supported by most technologies

http://www.mooseproject.org/
MOOSE web-repository

- Sharing experiences among practitioners
- Finding projects that are similar to own situation
- Finding proposals for new/innovative technologies
- Getting in touch with other projects directly
- Support for minimal change maximum effect strategy
- Web-repository in public domain and maintained on open-source concepts
- Feasibility of the web-repository depends on continuous addition of new project experiences
- Joint benefit from joint effort

http://www.mooseproject.org/

Rini.van.Solingen@LogicaCMG.com

http://www.mooseproject.org/
Clustering Web-repository projects