Panel on SOFTENG / ALLDATA
On-Device Software: Challenges of Embedded Software

Mira Kajko-Mattsson, KTH, Sweden
Roberto Meli, DPO Srl, Italy
Bernard Stepien, University of Ottawa, Canada
Yoshihisa Udagawa, Tokyo Polytechnic University, Japan

NexComm 2017, Venice, April 23-27, 2017

Topic # 1:
How can academia and industry cooperate in the IoT days?

- How do we do the knowledge transfer best?
- How can we make academic research relevant for industry?
- Are there special challenges for embedded software?

Topic # 2:
Managing complexity in embedded software

- How can the complexity in the development of embedded software be managed?
- Lack of developer system knowledge:
  can it be related to Topic 1?
- Can you measure complexity?

Summary

- Lack of system knowledge is a major contributor to software glitches and errors in the development process.
- Giving developers the opportunity and time to update and get better system knowledge is essential.
- Cooperation between academia and industry could be important in this process.
- The complexity of a software project is a determining factor for the probability of a project being successful in terms of time, budget and functionality.
- Complexity can be measured along many axes, but there are no established methodologies for measuring overall complexity of av software project.
- Embedded software production process is typically non-standardized and difficult to fit to established software production methodologies.

On-Device Software: Challenges of Embedded Software

Mira Kajko-Mattsson
KTH Royal Institute of Technology
SWEDEN
mekm2@kth.se

Developing embedded device must be precise and succinct to meet

- Quality
  - memory constraints
  - performance constraints
  - maintainability
  - …
- Specialized needs of the product that the embedded system resides in, whether it be medical, automotive or consumer oriented.

The challenge is to manage **COMPLEXITY** from the following perspectives:

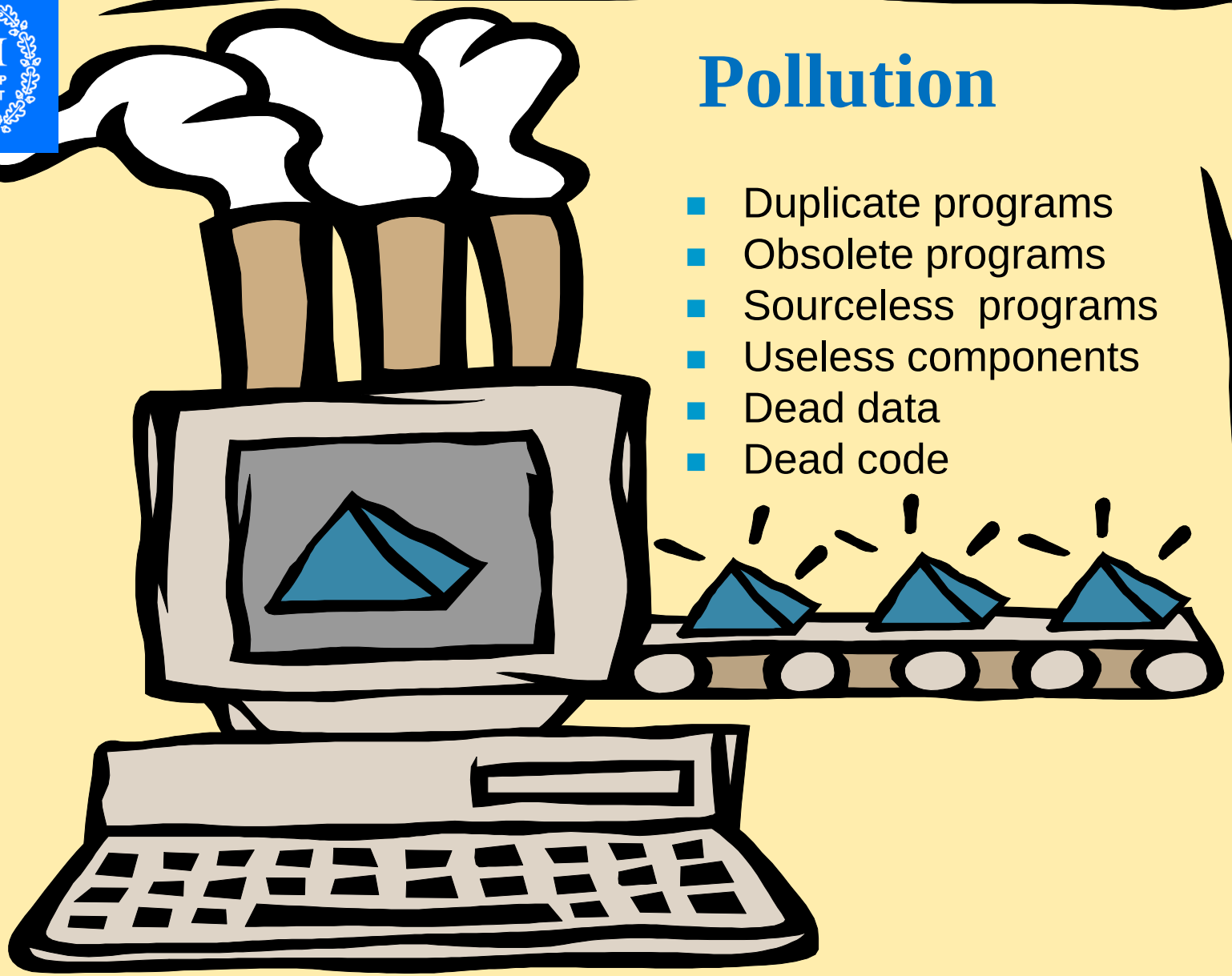
- Organizational
- Technological
- Domain
- Methodological
- System Quality
- Educational

System complexity

- System knowledge complexity
  - More and more functionality onto a single system
  - As networking capabilities are becoming persuasive in embedded systems, a design becomes a system of systems, adding yet another layer of complexity
  - **One of the main root causes of software problems is lack of developer system knowledge**
- Software polution

Pollution

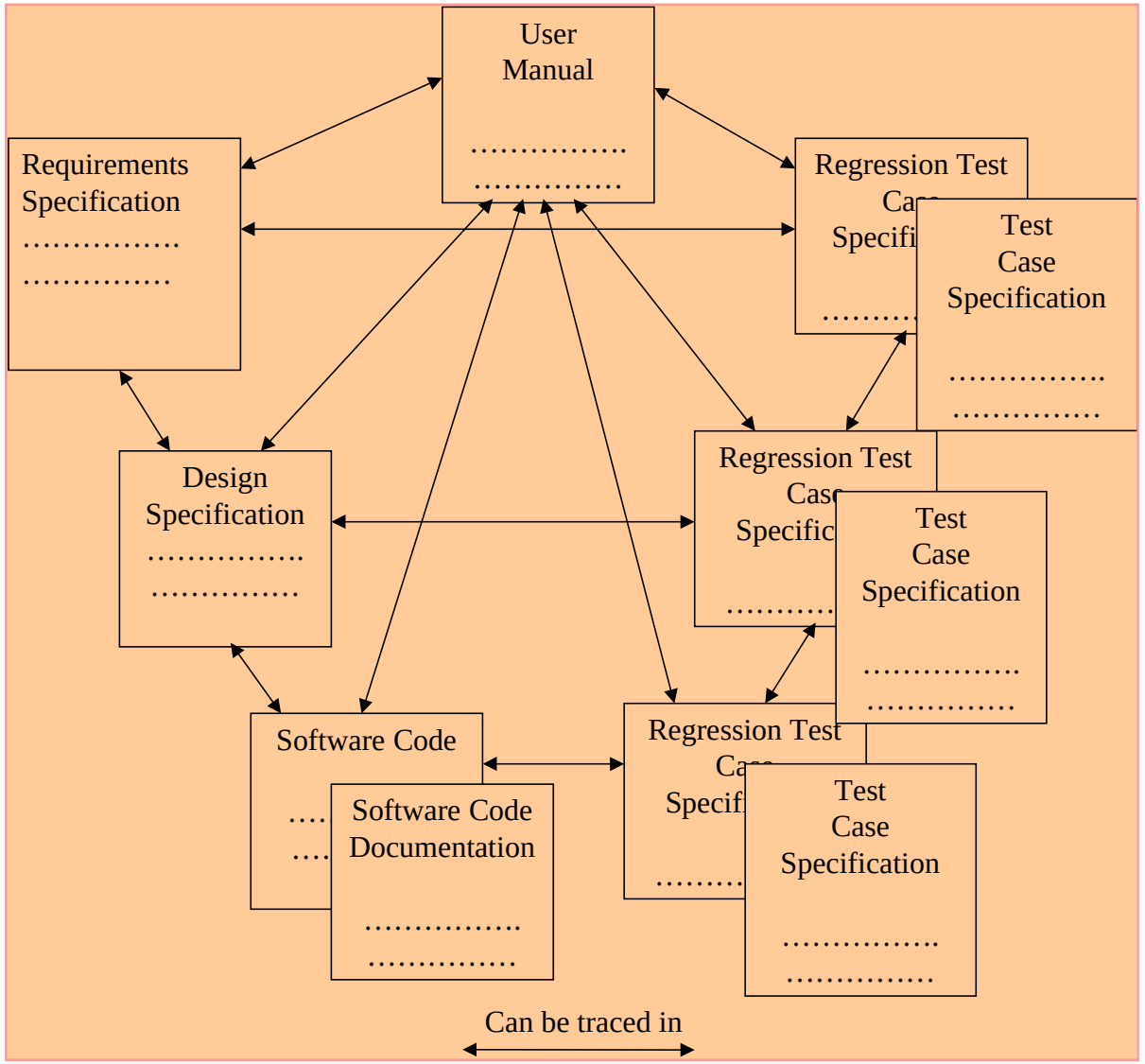
- Duplicate programs
- Obsolete programs
- Sourceless programs
- Useless components
- Dead data
- Dead code



System complexity

- System knowledge complexity
  - More and more functionality onto a single system
  - As networking capabilities are becoming persuasive in embedded systems, a design becomes a system of systems, adding yet another layer of complexity
  - **One of the main root causes of software problems is lack of developer system knowledge**
- Poor traceability

Traceability is an important complexity mngt factor



System complexity

- More and more functionality onto a single system
- As networking capabilities are becoming persuasive in embedded systems, a design becomes a system of systems, adding yet another layer of complexity
- Software polution
- Poor traceability
- Embedded knowledge
- Poor lexicon

Methodological complexity

- Developers lack a holistic perspective of the overall lifecycle
- **They should not be blamed for this.**
- They develop complex systems within complex environments using very complex methods.
- The methods should be elastic enough to allow some developer freedom. However, they should not jeopardize system and process quality.

Panel
On-Device Software: Challenges of Embedded Software

the measurement problem

Roberto Meli (CEO)
roberto.meli@dpo.it - www.dpo.it

April 2017

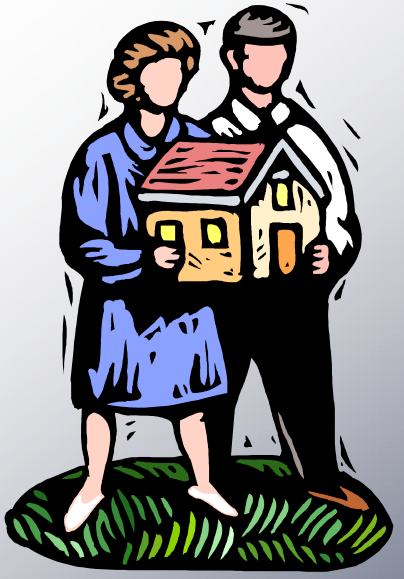
SOFTENG 2017

Why do we need measures ?

- To estimate (sw quality, resource consumption, work effort, work duration, costs…)
- To monitor work progress
- To represent and communicate progress and results to the various stakeholder
- To manage risks
- To establish baselines and hystorical data bases
- To benchmark against the market
- … even more

What kind of measures are available today ?

Technical & Logical



User Requirements - ISO/IEC 14143

- Functional User Requirements
  - Subset of the user requirements regarding what the software will do to support business or operational processes.
- Non Functional User Requirements
  - Subset of the user requirements regarding how the software will perform to support business or operational processes.
    - Quality Requirements according to ISO/IEC 25010:2011 (ex ISO/IEC 9126).
    - Technical Requirements.

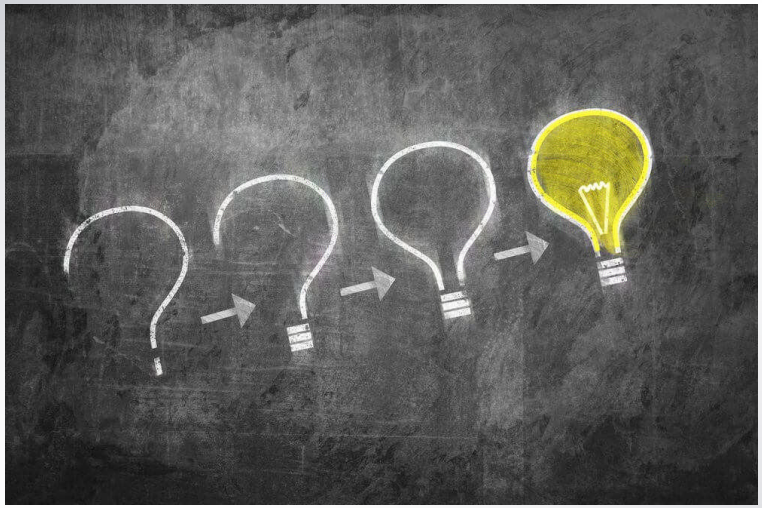
Embedded SW peculiarities

- Hardware, time, space constraints
- Very different languages
- Direct control of devices drivers
- Very different levels of abstraction
- Ad hoc programming (vs standardization)
- Cross compiling and testing with emulators
- Communication protocols are relevant
- High technical competence
- Inter-system complexity

Threats on measurement

- It’s difficult to learn from experience
- Estimation is highly “human dependent”
- Productivity is unknown
- Comparison among projects is difficult
- Technical measurement may lead to inefficiency
- “Doing”is the principle focus of teams: measurement may become a “luxury” item

Solutions



Have you got any ?

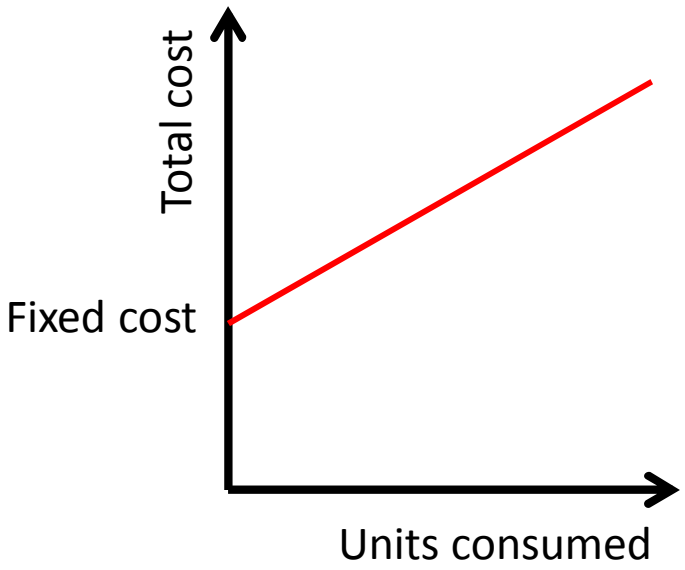
About the cost of testing and more precisely where the costs really are especially in the context of open source mania

By Bernard Stepien

University of Ottawa

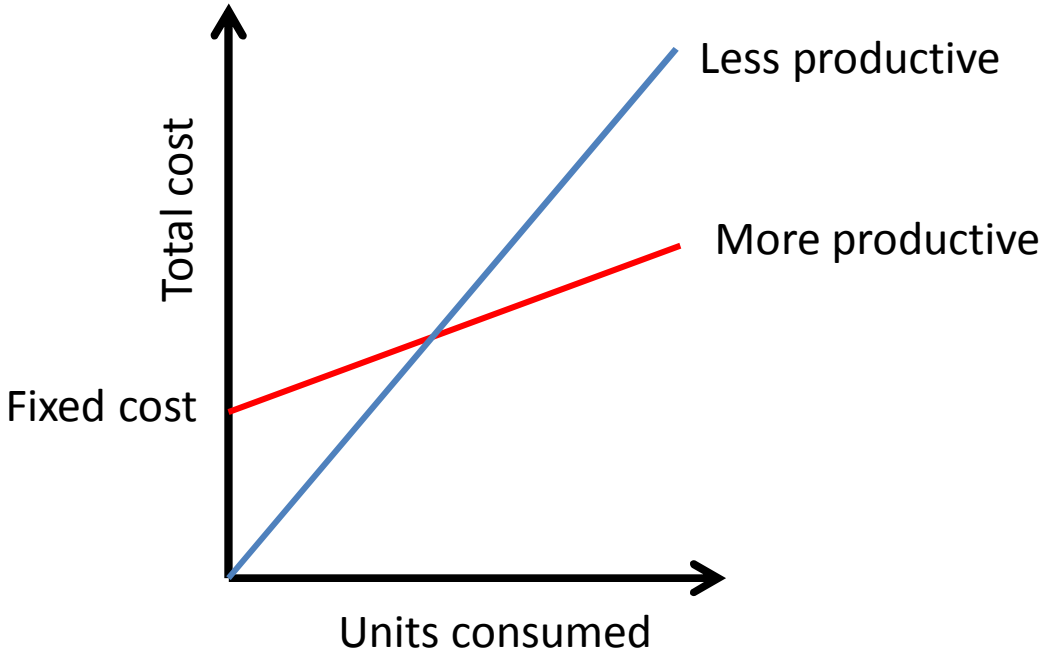
Basic principle of costing

- Fixed cost
- Variable cost
- Productivity (total cost divided by number of units consumed)



Productivity

- Productivity can be improved by using design and testing tools
- The problem is that tools cost money, thus a fixed cost
- Open source vs commercial software does not always resolve the problem because of learning curves and the absence of support in open source tools (A hidden fixed cost).



How is cost really perceived?

- Commercial tools cost money (a tangible cost)
- An employee is not seen directly as a cost (an intangible cost). He already sits there and his budget has already been approved a long time ago.
- Also, formal methods and their tools that improve productivity, require training costs (a fixed cost that repeats itself every time an employee moves on)
- A cost is really perceived as something outside of the budget

Experiences lived in Industry

- Managers have major troubles about how to do costing.
- They perceive fixed costs as something that can be avoided.
- They will avoid training costs as much as possible because they are repeatable
- Productivity is not always tangible, thus less efficient solutions are chosen to avoid asking for more budget

The Nortel Networks syndrom

- Back in the 1980s, Nortel Networks that had an incredible pool of scientists decided to cut completely the hundreds of internally developped programming languages and related tools to General Purpose Languages (C++ and Java).
- This example has been largely followed by Industry in North America.
- Ironically, the result is a loss of productivity

Recommendations

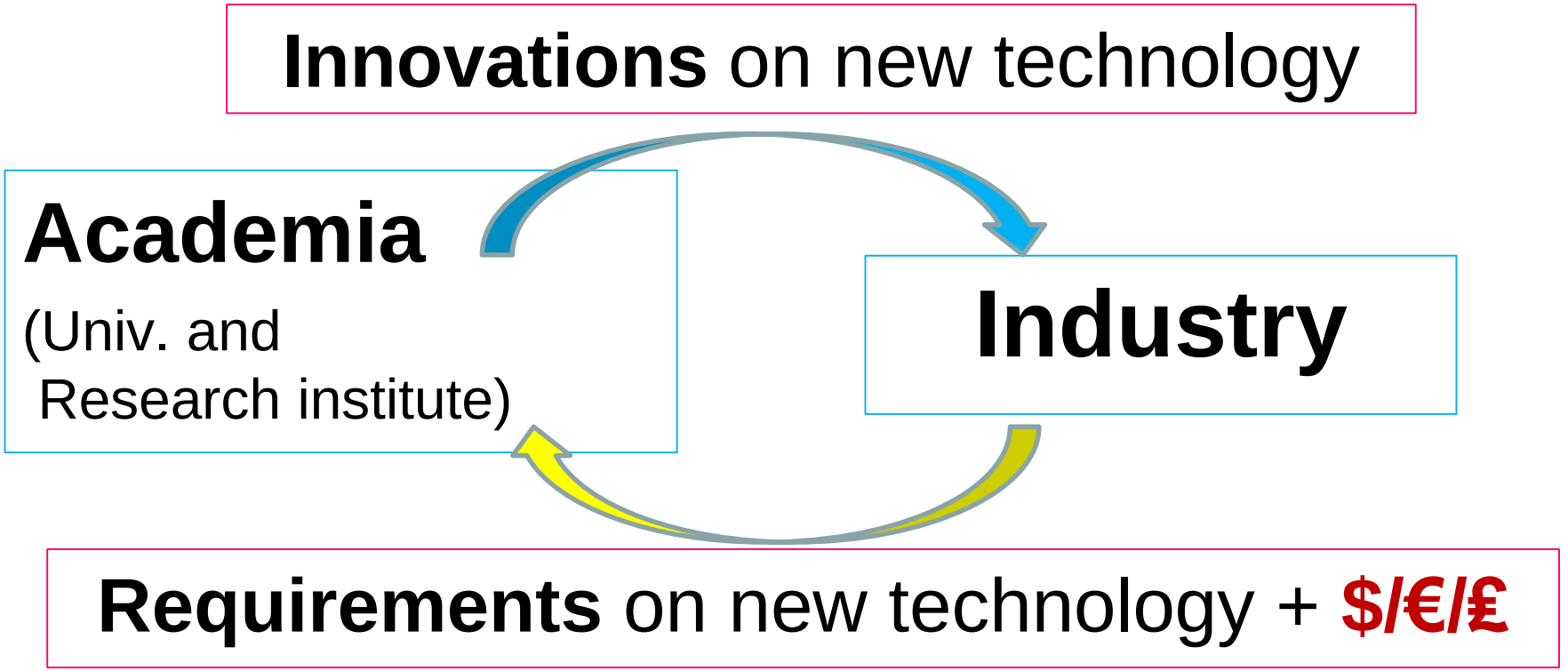
- Select a limited amount of novel languages
- New comers like Python are a good example
- Put in place knowledge conservation measures using competence centers where the turn around is less frequent
- Write ample documentations

How can academia and industry cooperate in IoT (Internet of Things) days?

Yoshihisa Udagawa (PTU, Japan)

- In the coming IoT days, systems for industry become increasingly sophisticated.
- This means engineers are expected to have wide range of knowledges and skills.
- How can academia cooperate with engineers in industry? --- a never ending problem
- I'd like to stress the importance of conference like IARIA.

Fig. 1 Concept of ideal cycle



The ideal cycle doesn't work because **Innovation** does not necessarily meet **Requirements**.

Fig. 2 A view of reality

- People in academia seek for **Originality** and may not catch up with advances in industry.
- Industry tends to save **budgets**.
- The only connection is **flow of human resources** !

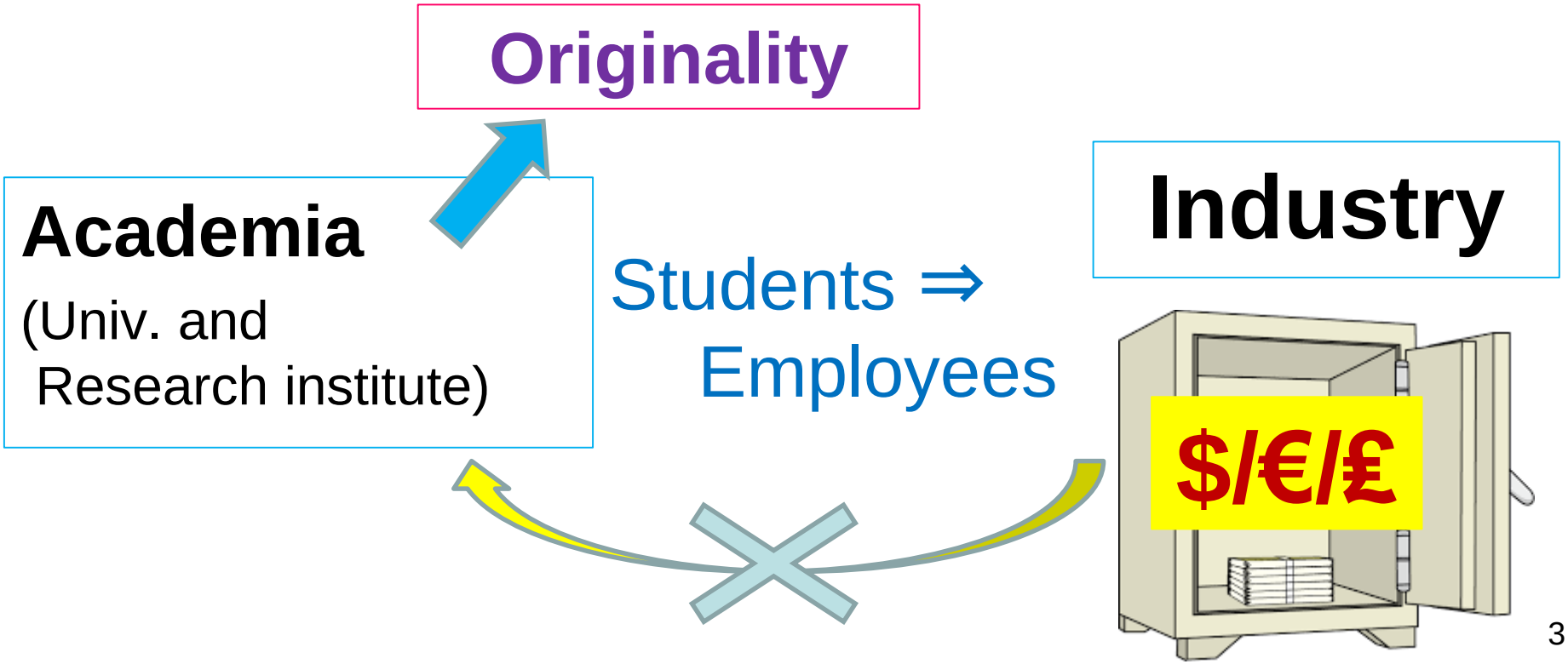
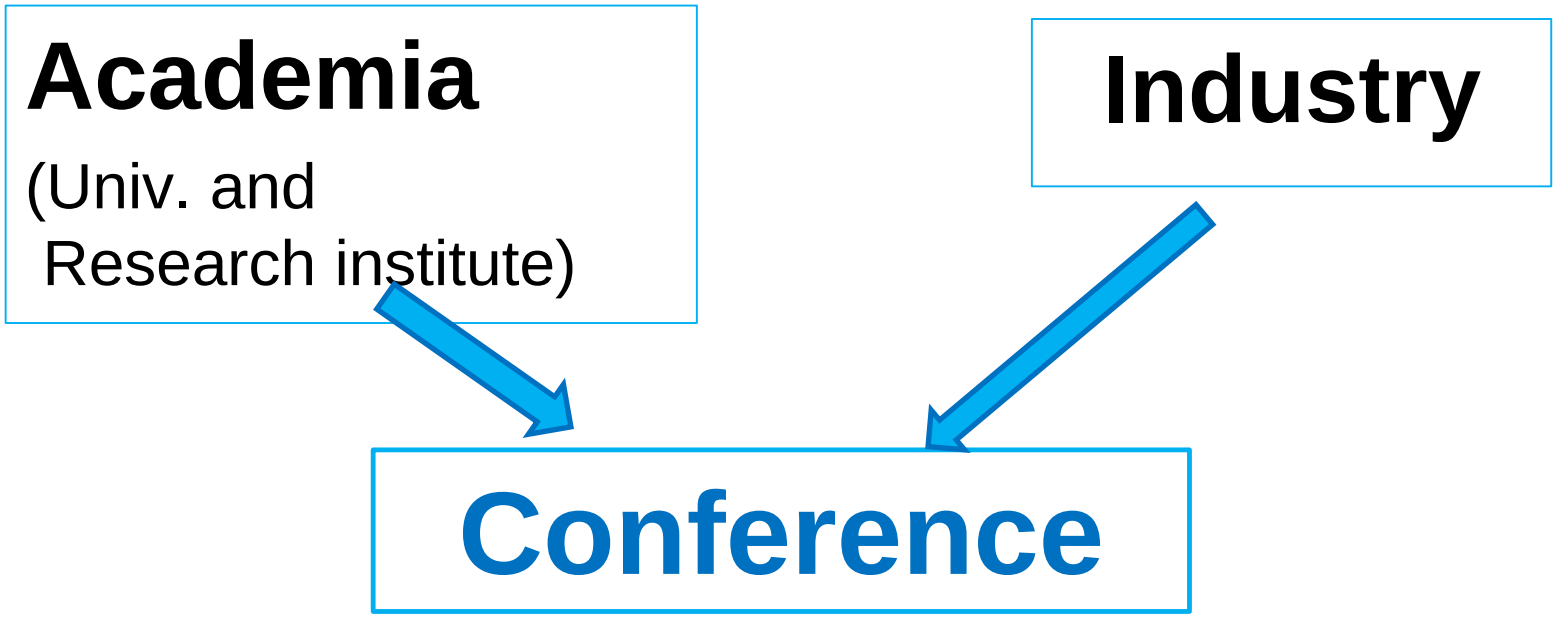


Fig. 3 A proposal



- Conferences like IARIA provides valuable time and space to meet people in both A & I.
- Innovation of conference leads to better cooperation between A & I.

Fig. 4 A proposal (continue)

- **Typical Conf.**
  - Paper (Reg./Short)
  - Keynotes
  - Panels

It is nice, but can be improved.

- **Proposed Conf.**
  - Paper (Reg./Short +Industry)
  - Keynotes
  - Panels
  - Hands-on using internet pages
  - **???**

Glossary

- Paper on Industry: A paper accepted by criterion on interest to industry, soundness and/or impact of solution, etc. not much stress on originality. Strictly, it may not be an academic paper.
- Hands-on using internet pages: events that uses pages on internet to solve a specific subject such as programming. In other words, best practice of using internet pages to solve problems to implement some functions hopefully on IoT industry.