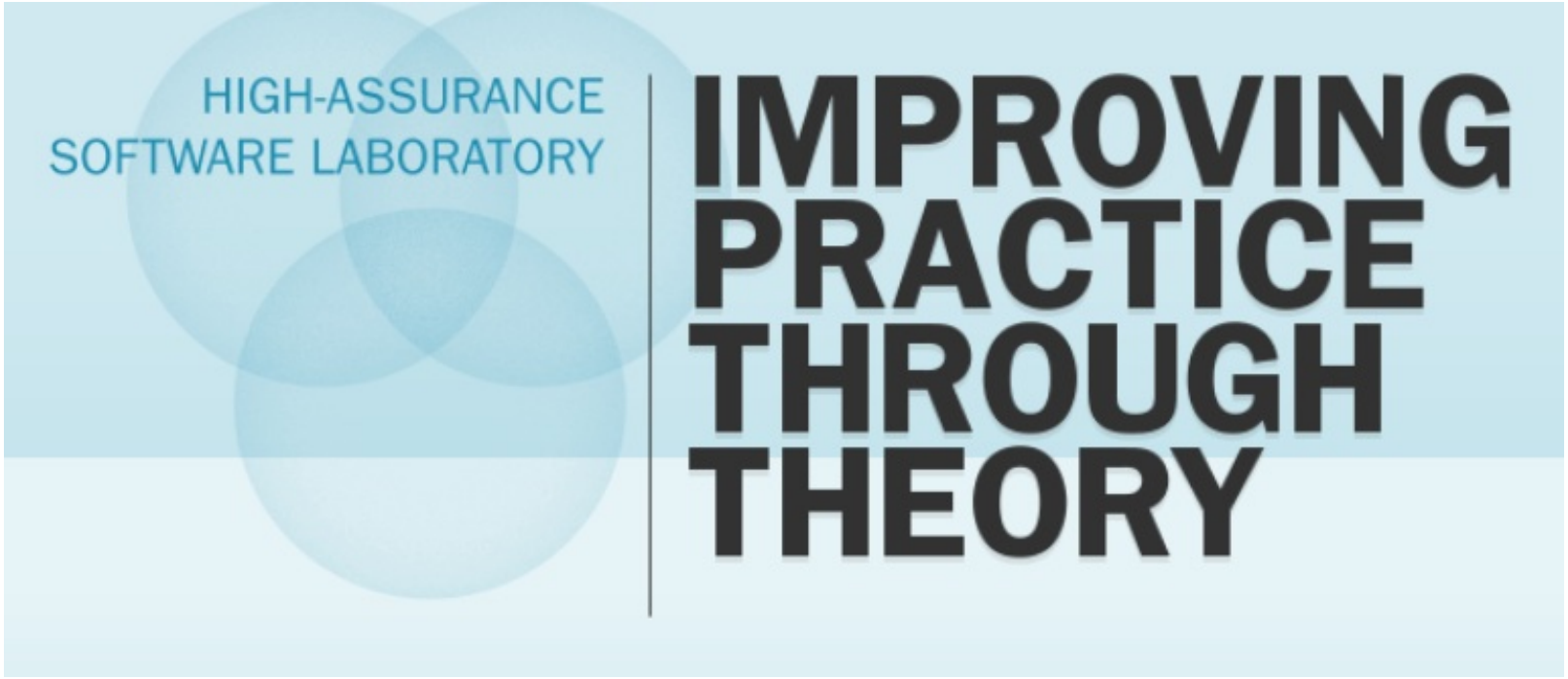
Algebraic and Coalgebraic Methods in Software Development

Alexandre Madeira, UM (coord.)
Dirk Hoffman, UA
Manuel A. Martins, UA
Luís S. Barbosa, UM
José Nuno Oliveira, UM

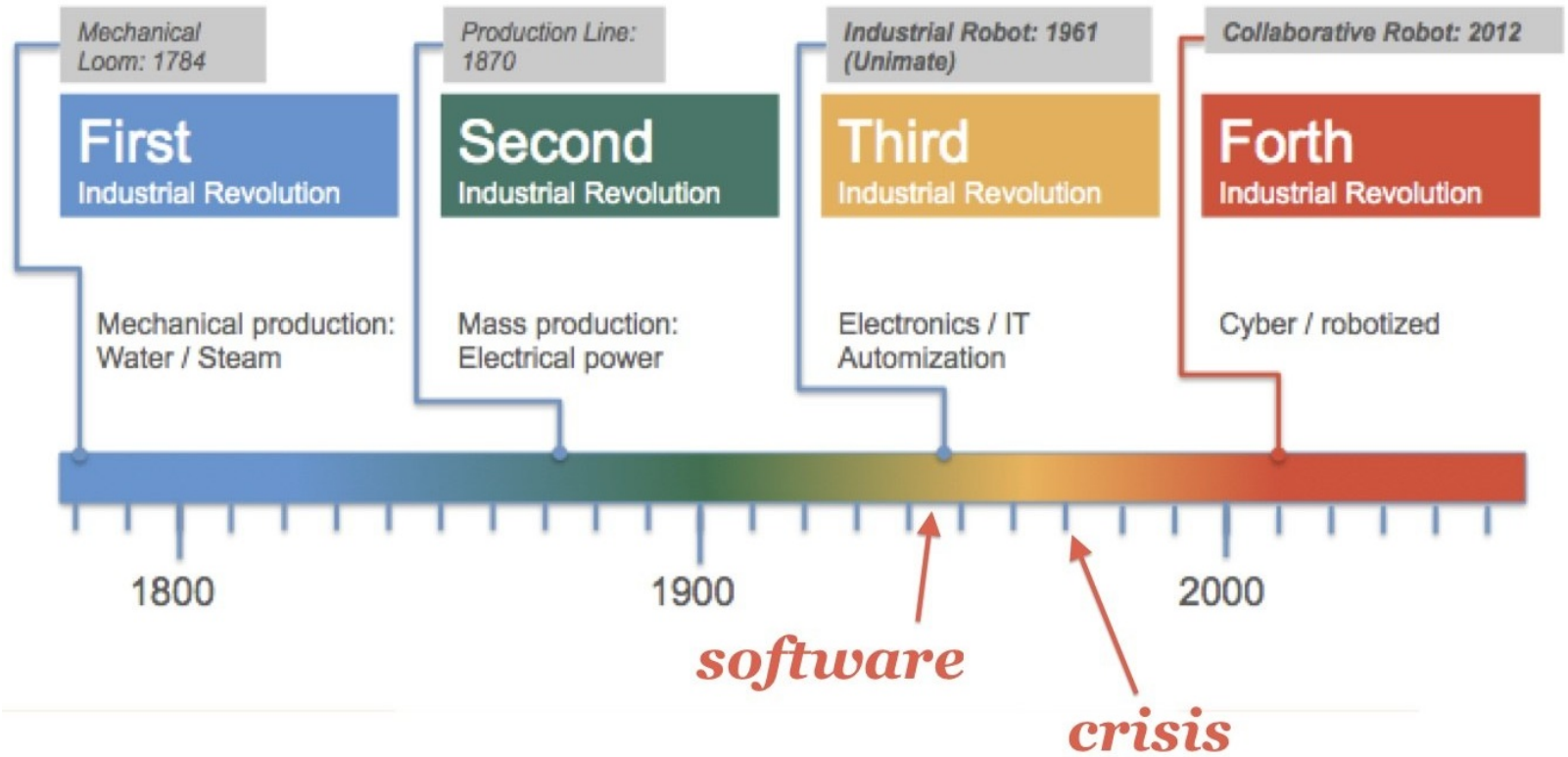
16 September, 2016

Our motto


HIGH-ASSURANCE
SOFTWARE LABORATORY
IMPROVING
PRACTICE
THROUGH
THEORY

Big picture

Industrial revolutions:



1st NATO Conference on **Software Engineering**, Darmstadt, October **1968**

Big picture

Industrial revolutions made possible by advances in

- Physics
- Software

Industry 4.0 will rely on **software** on an unprecedented scale (CPS, data-mining, robotics...)

However

- Physics is a **truly scientific** discipline
- Software is a **pre-scientific** discipline

Uupps!

> *As happened in physics, the software sciences need to find a* **unified theory**.

This course

This course is about such **unifying theory**.

This will provide you with **better** knowledge of what **computing** is all about.

Abstraction essential to better knowledge,

> *"The purpose of* **abstraction** *is not to be* **vague**, *but to create a new semantic level in which one can be* **absolutely precise**." *(E. Dijkstra)*

---

even where you would expect the opposite (next slide).

Why abstraction matters

From a Robot Programming Tutorial on the web:

> *"The fundamental challenge of all* **robotics** *is this: It is impossible to ever know the true state of the* **environment**. *A* **robot** *can* **only guess** *the state of the real world based on measurements returned by its* **sensors**."

There is a sound theory for such (safe) **guessing** called abstract interpretation — and tools supporting it.

About the course title

Intimidated by

> *Algebraic and Coalgebraic Methods in Software Development ?*

Don't worry — you've seen this before without noticing:

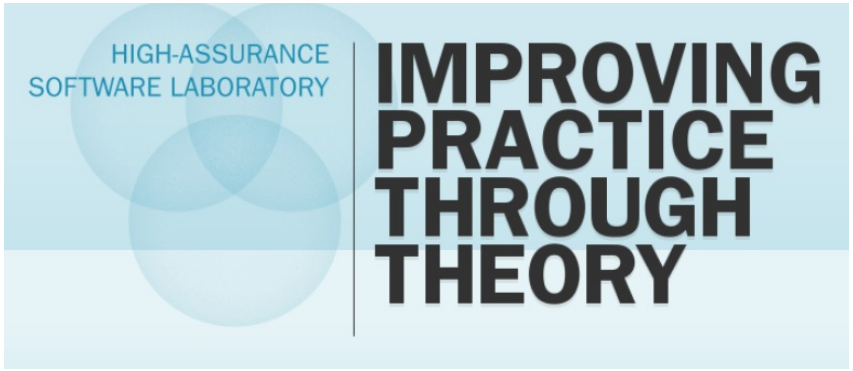
Programs = Algebras + Coalgebras

Examples:

- a **programming language** forms an **algebra**
- an **automaton** forms a **coalgebra**

Want to know why? — come to the course :-)

What for?



Which **practice**?

- data mining
- cyber-physical systems
- software architecture
- risk analysis
- probabilistic programming
- ....

Course structure

Course plan:

1. **Category theory for computer science** (JNO)
2. **Advanced Category theory** (DH)
3. **Coalgebras and coalgebraic modelling** (LSB)
4. **Algebras and algebraic specification** (MAM)
5. **Modal logics – a logics on-demand approach** (AM)

where

- AM — Alexandre Madeira, UMinho
- DH — Dirk Hoffman, UAveiro
- MAM — Manuel Antonio Martins, UAveiro
- LSB — Luís Soares Barbosa, UMinho
- JNO — José Nuno Oliveira, UMinho

Course structure

Operating mode

- Organized in 5 modules (one per lecturer)
- One lecture per afternoon (with breaks!)

Bibliography

- Papers / books suggested by lecturers

Grading

- Paper recitation (individual assessment).