Karlsruhe Institute of Technology

Teaching Research Methodologies with a Robot in a CS Lab Course

Mathias Landhäußer, Sebastian Weigelt, Martin Blersch

KIT – Department of Informatics – Institute for Program Structures and Data Organization (IPD).

The Past

Educating Practitioners

Focus on Employability

[BCKM1997] [WR1999] [DSTWP2014]

The Present

Focus on Methodologies

Full-blown Approach: Conducting Research to Teach Research

[B2005] [R2007] [KB2009] [BGBKBBR2016]

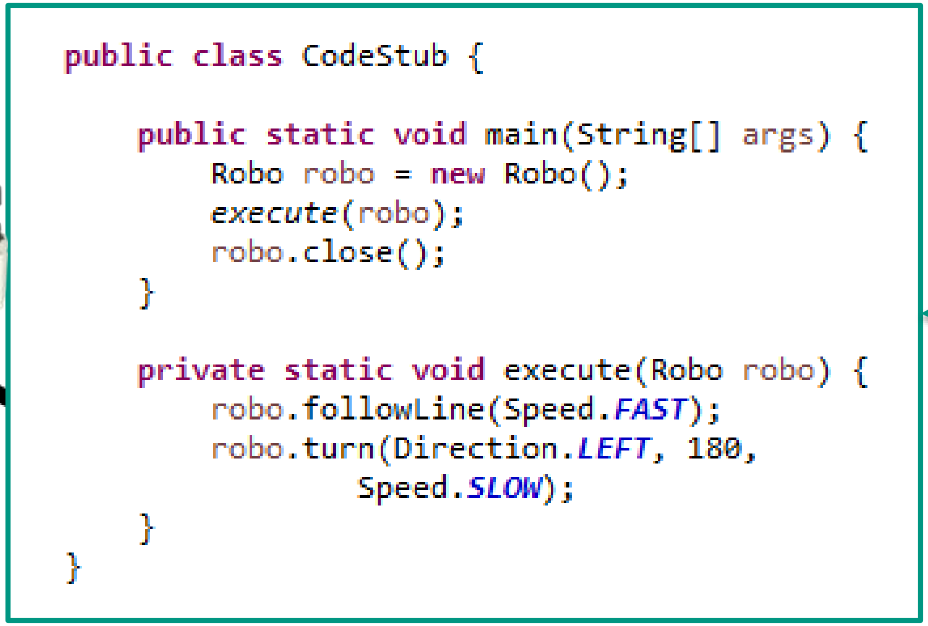
(Goals for) The Future

Educating Practitioners and Researchers Alike

Expose Students to Research as Early as Possible

Lower the Hurdles for Getting into Research

Lab Course: Programming in Natural Language



```
public class CodeStub {

    public static void main(String[] args) {
        Robo robo = new Robo();
        execute(robo);
        robo.close();
    }

    private static void execute(Robo robo) {
        robo.followLine(Speed.FAST);
        robo.turn(Direction.LEFT, 180,
                Speed.SLOW);
    }
}
```

This is what we want the students to build.

Hey Gizmo, follow the black line, quickly. Then turn around slowly.

Translation Engine

Lego Mindstorms Robot „GIZMO“

- GIZMO – Grammar Identification Zombie with a Monstrous Ontology
- Technical Data:
  - Size: 244mm x 183mm x 321mm
  - Powered by a Lego Mindstorms EV3 Brick
    - ARM 9 processor
    - 64 MB RAM, 16 MB flash disk
    - 1x USB 2.0 port
    - 178x128 pixel display (monochrome)
    - 4 push buttons
    - 4x ports for actuators and sensors (each)
  - Battery: 2050 mAh
  - 2x (independent) caterpillar tracks
  - 1x Grappler
  - Adjustable head (motor-operated)
  - Sensors: Color, infra red, ultra sonic
- Operating System: LeJOS → Java VM w/ pre-defined API (basic functionalities w/ hw abstraction)

Lab Course:
Natural Language Processing in Software Engineering

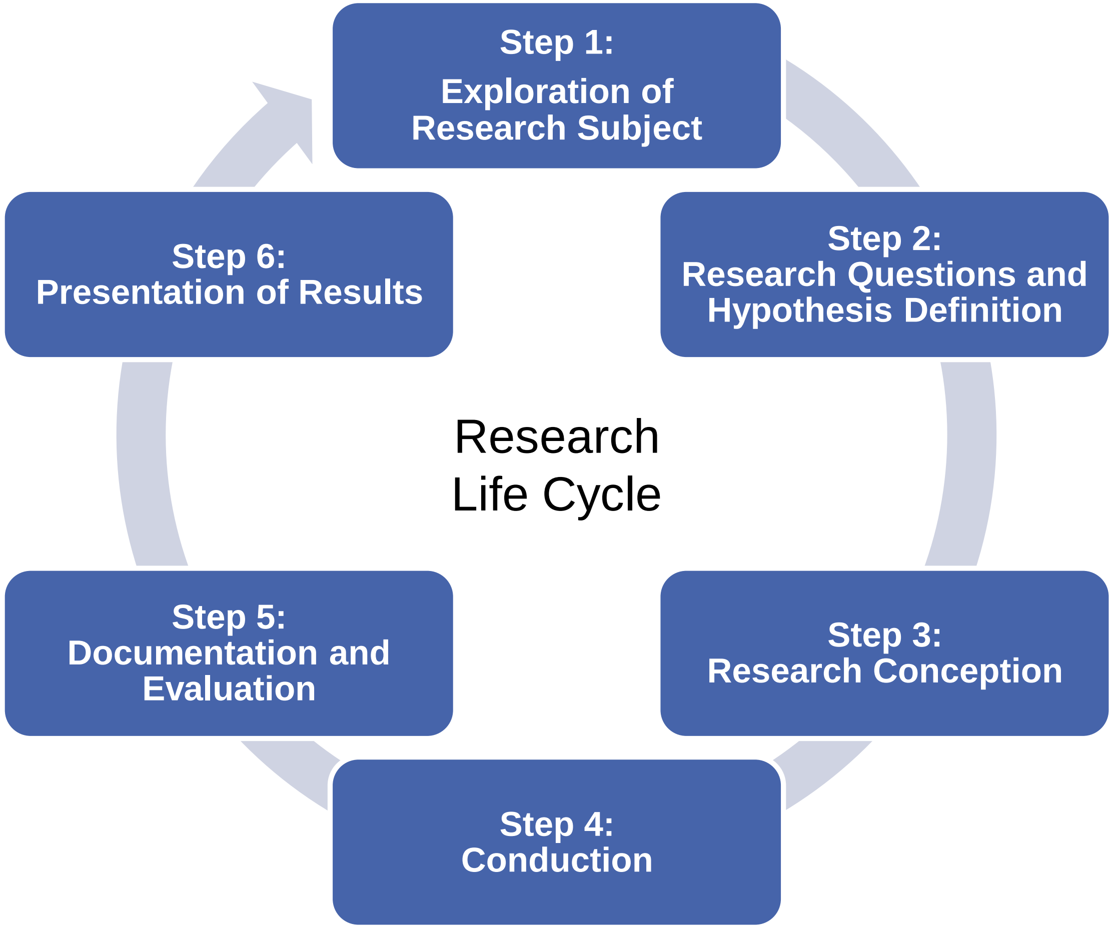
- Topic: Development of a system for programming in natural language.
- Example/Toy System: Lego Mindstorms Robot
- Learning Objectives
  - Apply knowledge from lecture “Natural Language Processing in Software Engineering” to a live project
    - Develop a NLP pipeline
    - Benchmark and use NLP tools
    - Build a text corpus
    - Build an ontology and build an ontology generator
    - Development of a program detecting semantics in NL text
  - Apply a research methodology
  - Work in a team, use SCRUM
- Why “Natural Language Processing in Software Engineering”?
  - Programming (in NL) is a subset of the lecture
  - We adopt and adapt many techniques and concepts that are being taught in the lecture

Ingredients: Process Model and Learning Objectives

- Process Model: Research Life Cycle (RLC)
- Learning Objectives (excerpt)

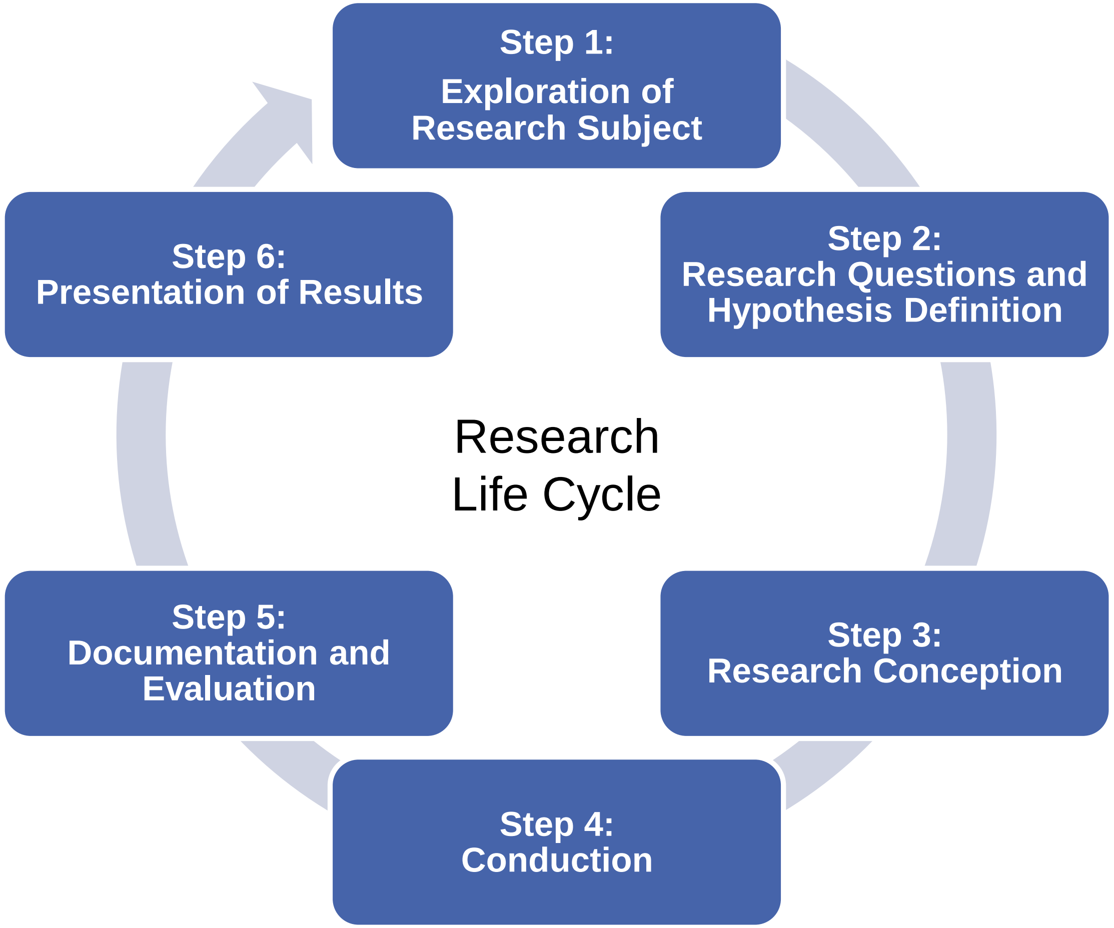
No	Objective (Students are able to:)	Level*
O_1	Perform a literature review for a given topic	L_2 - L_4
O_2	Build up a benchmark for a given problem	L_4 - L_6
O_8	Implement and benchmark a prototype for a given problem at hand	L_3 - L_6
O_{10}	Present their results and insights gained in a concise and precise manner	L_1 - L_3

*according to Bloom's taxonomy [AKB2001]



Ingredients: Process Model and Learning Objectives

- Process Model: Research Life Cycle (RLC)
- Learning Objectives
- Programming in natural language (build a NLP pipeline that generates code from text)
  - Tokenize
  - Parse
  - Detect actions
  - …
  - Generate code
- Final task: Parcours



Step 1:
Exploration of Research Subject

Step 2:
Research Questions and Hypothesis Definition

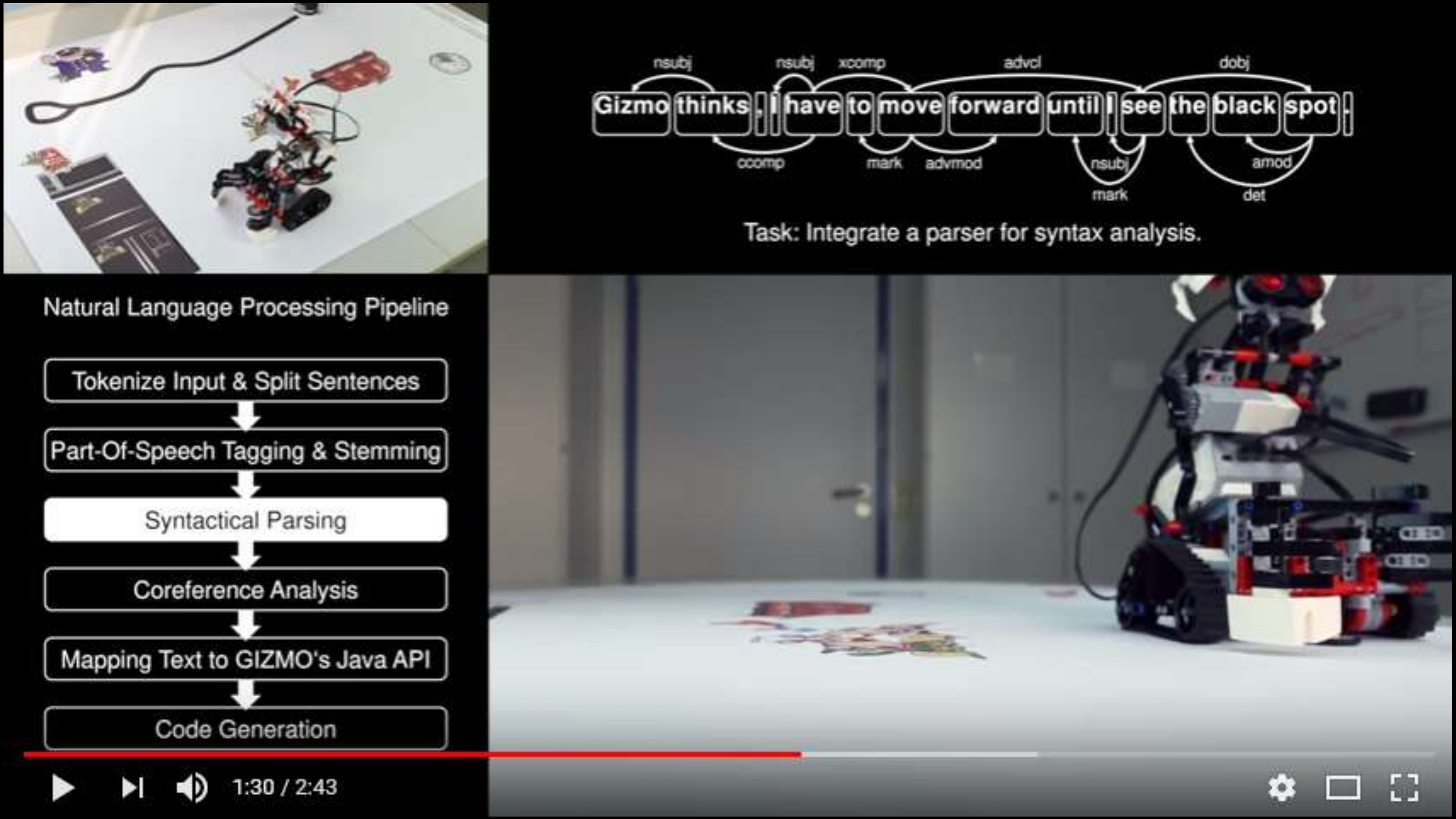
Step 3:
Research Conception

Step 4:
Conduction

Step 5:
Documentation and Evaluation

Step 6:
Presentation of Results

Research Life Cycle


nsubj
nsubj
xcomp
advcl
dobj
Gizmo thinks , I have to move forward until I see the black spot .
ccomp
mark
advmod
nsubj
mark
amod
det
1:30 / 2:43

Task: Integrate a parser for syntax analysis.

Natural Language Processing Pipeline

Tokenize Input & Split Sentences

Part-Of-Speech Tagging & Stemming

Syntactical Parsing

Coreference Analysis

Mapping Text to GIZMO's Java API

Code Generation

Review Gizmo‘s technical details and watch it master our parcours

https://youtu.be/Z_vt1-imBUE (165 seconds ☺)

Didactic Approach

- Assignment = one (or more) pipeline stages
  - Continuous improvement
  - Transparent assessment

- Final presentation = entire pipeline
  - Evaluates fitness of the approach
  - Rewards continuous improvement

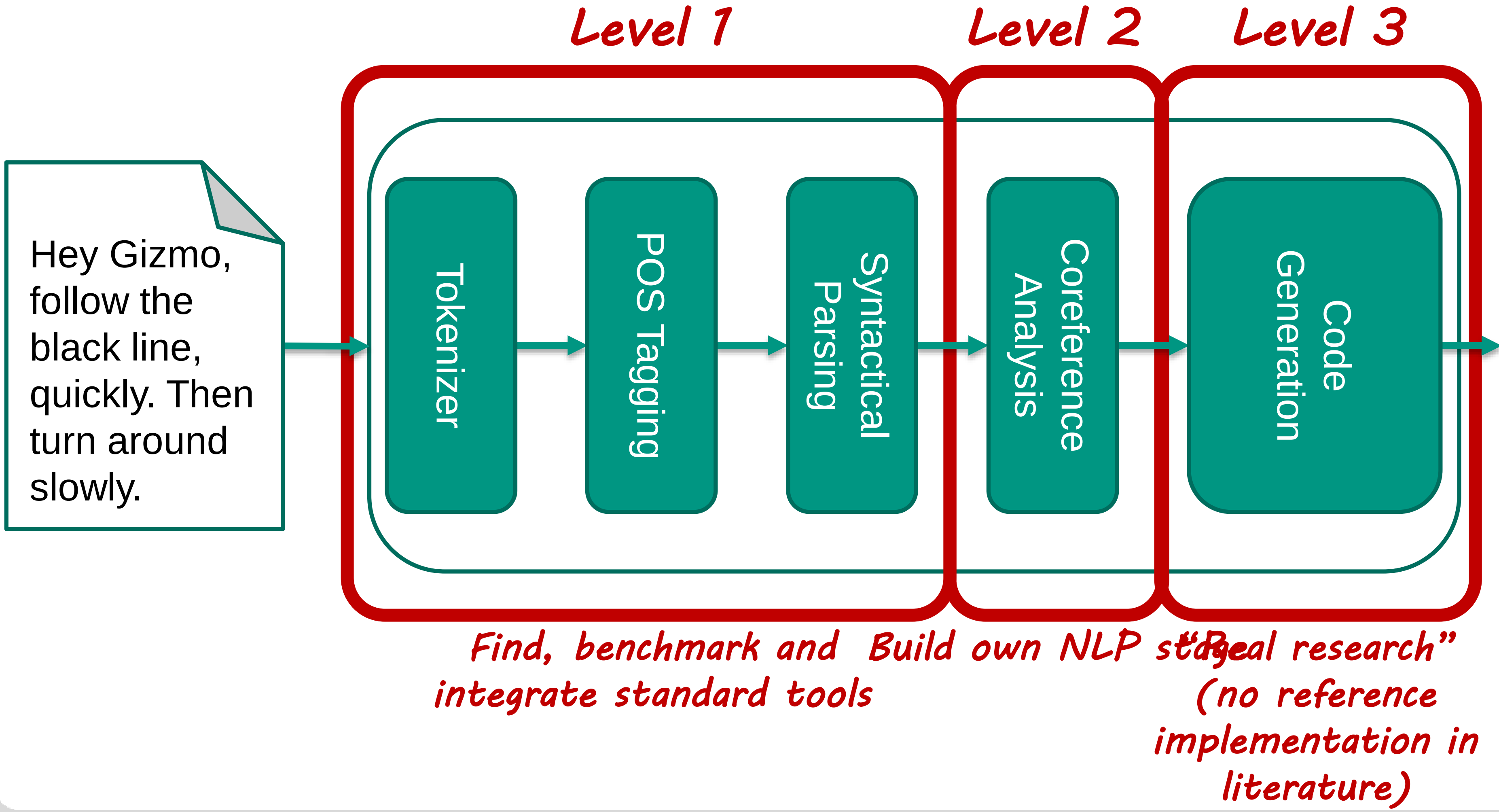
- For every stage: “Run” Research Live Cycle (at least) once
  - Repetition intensifies learning effect
  - Shows applicability of life cycle in different contexts, for different types of problems, …

- Every stage teaches a subset of our learning objectives
  - Technical and methodical skills
  - Some skills are needed at every stage

IT’S ALL ABOUT

MOTIVATION

Course Design & Assignments: NLP Pipeline



Level 1

Level 2

Level 3

Hey Gizmo, follow the black line, quickly. Then turn around slowly.

Tokenizer

POS Tagging

Syntactical Parsing

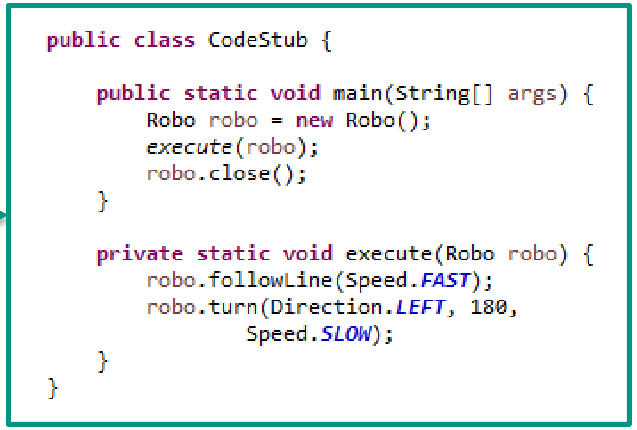
Coreference Analysis

Code Generation

Find, benchmark and integrate standard tools

Build own NLP stage

“Real research” (no reference implementation in literature)



```
public class CodeStub {

    public static void main(String[] args) {
        Robo robo = new Robo();
        execute(robo);
        robo.close();
    }

    private static void execute(Robo robo) {
        robo.followLine(Speed.FAST);
        robo.turn(Direction.LEFT, 180,
                Speed.SLOW);
    }
}
```

15/16 Course

- 2 Teams, 4 students each
  - MSc. students (first year)
  - Self-assigned teams
  - Teams worked self-organized

- 7 Assignments
  - 1 – 6 weeks
  - 1 – 3 pipeline stages each
  - 1 – 2 learning objectives
  - Increasing complexity

- Final presentation
  - 1 parcours per team + 1 parcours by TAs = 3 parcours/stories to solve
  - Team A: fully successful, Team B: struggled with unknown courses

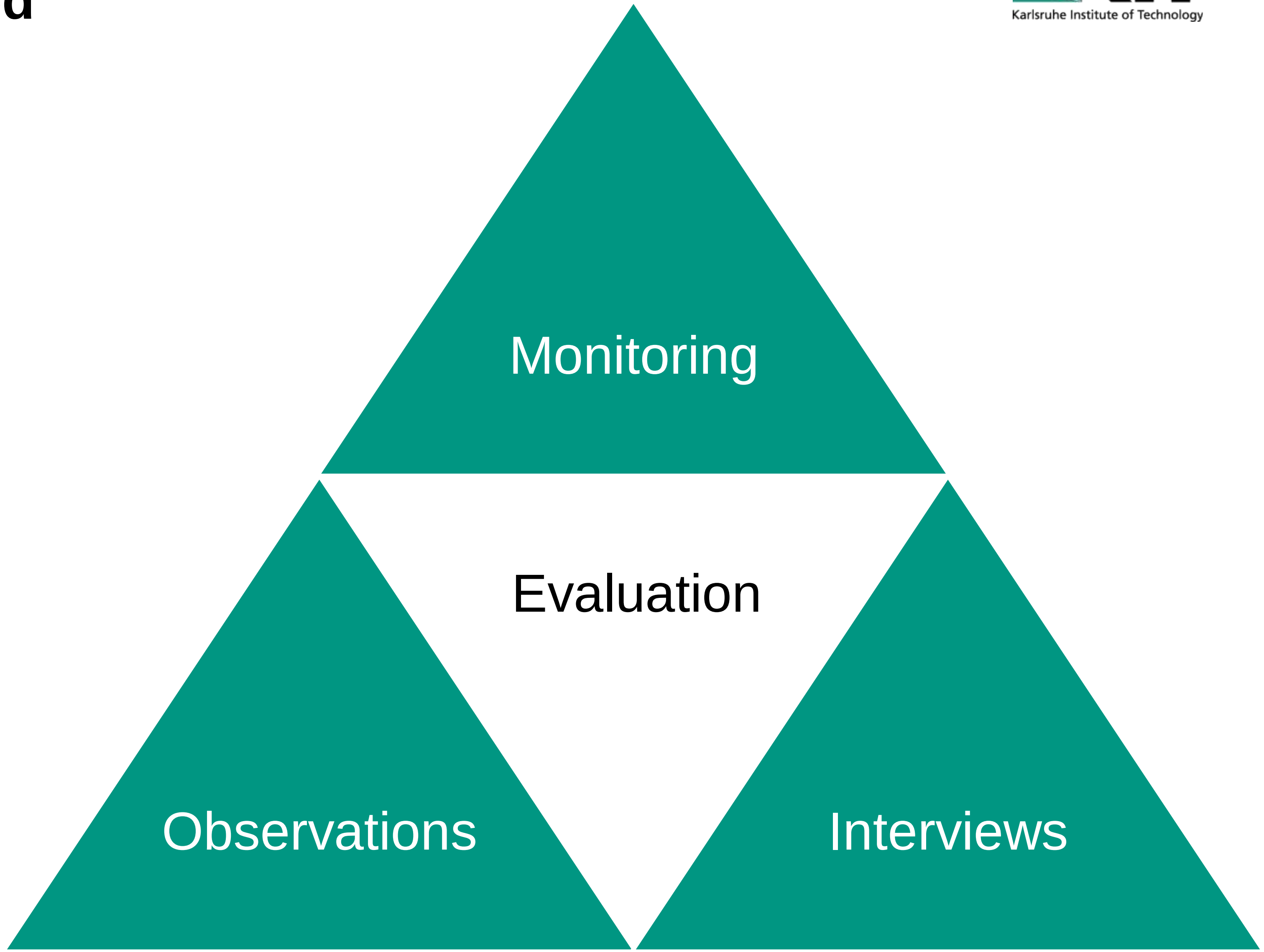
Findings and Lessons Learned

We have had eight participant only!

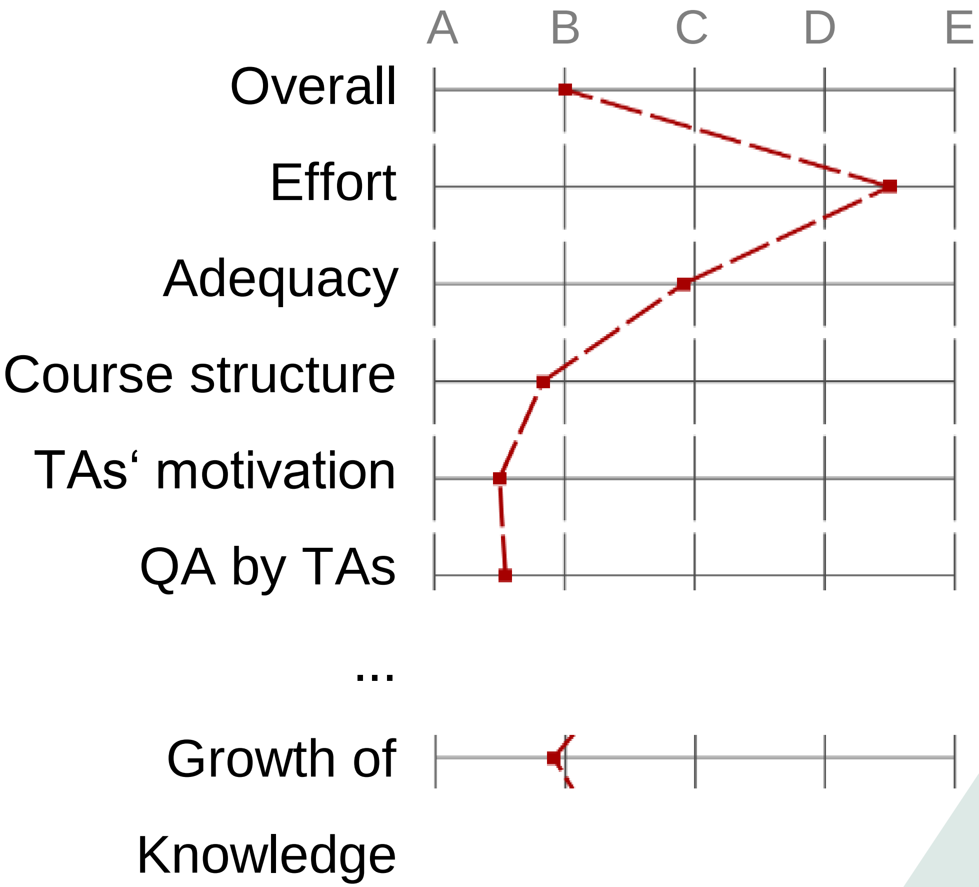
How to conduct a *proper* Evaluation?

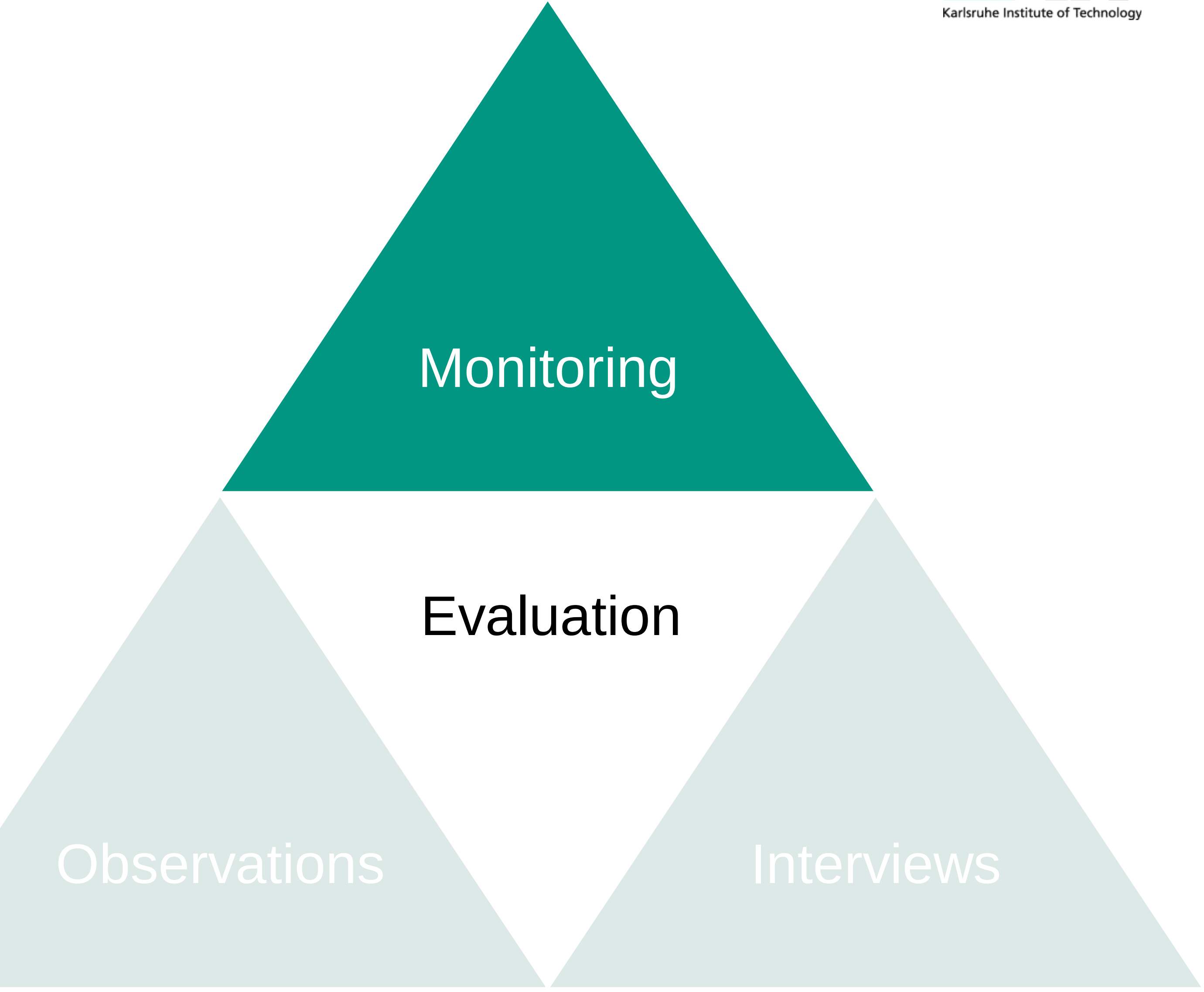
Note: This is a case study…

Findings and Lessons Learned



Findings and Lessons Learned



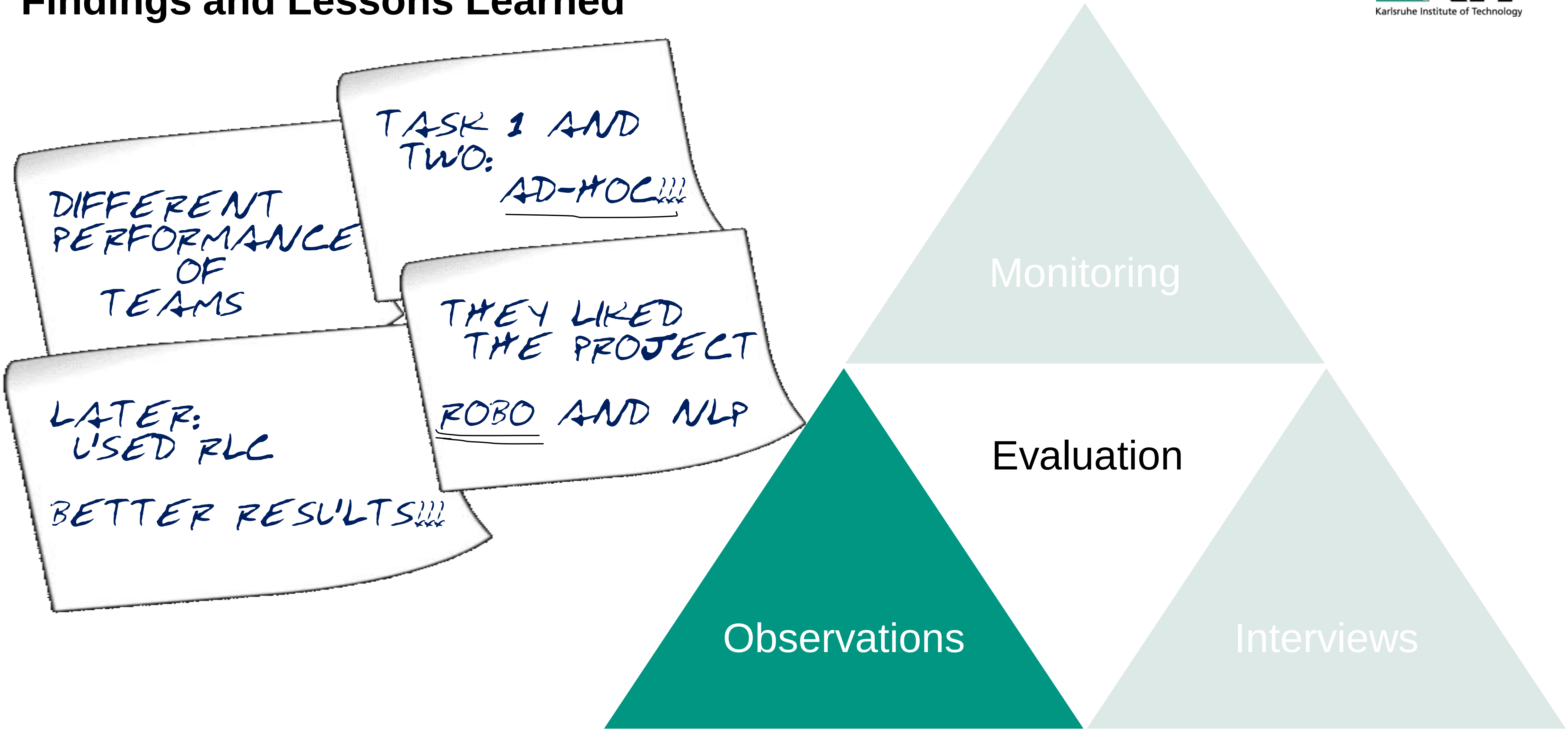

Monitoring

Evaluation

Observations

Interviews

Findings and Lessons Learned



DIFFERENT PERFORMANCE OF TEAMS

TASK 1 AND TWO: AD-HOC!!!

THEY LIKED THE PROJECT

ROBO AND NLP

LATER: USED RLC

BETTER RESULTS!!!

Monitoring

Evaluation

Observations

Interviews

Findings and Lessons Learned

“The **robot** was the most interesting thing!”

“The **structure** – the research life cycle and the pipeline – was very helpful!”

“Doing **research** in the future? Why not?!”

“I could definitely imagine […] to further engage in **research**.”

“Working […] with the **robot** was fun!”

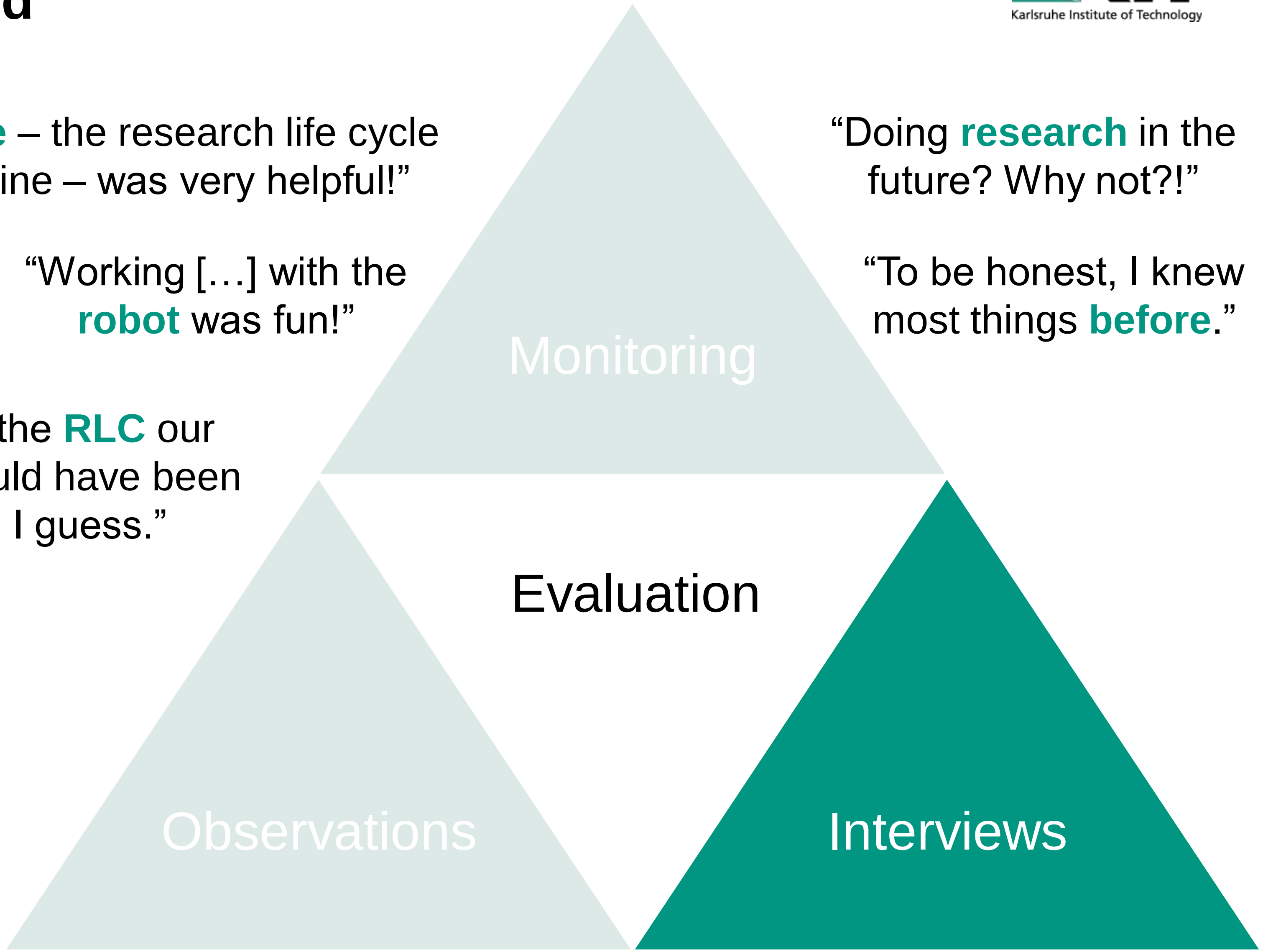
“To be honest, I knew most things **before**.”

“I’m really into NLP. […] building a full **NLP pipeline** to control a robot was really cool.”

“Without the **RLC** our results would have been worse, I guess.”

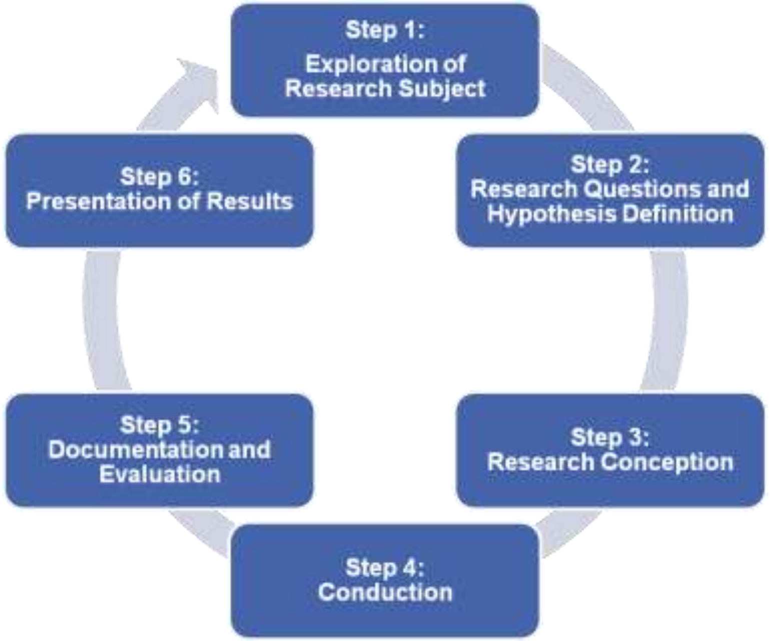
“Developing a **complete system** was very motivating.”

“No, […] we **did not use the RLC** consistently.”



Summary

Research Life Cycle


Step 1:
Exploration of Research Subject

Step 2:
Research Questions and Hypothesis Definition

Step 3:
Research Conception

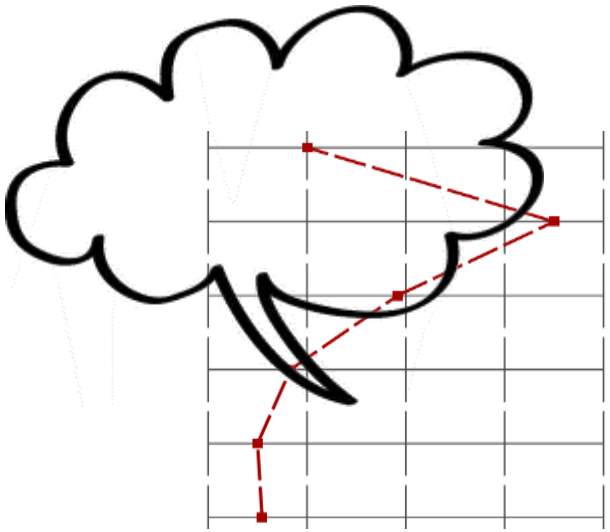
Step 4:
Conduction

Step 5:
Documentation and Evaluation

Step 6:
Presentation of Results

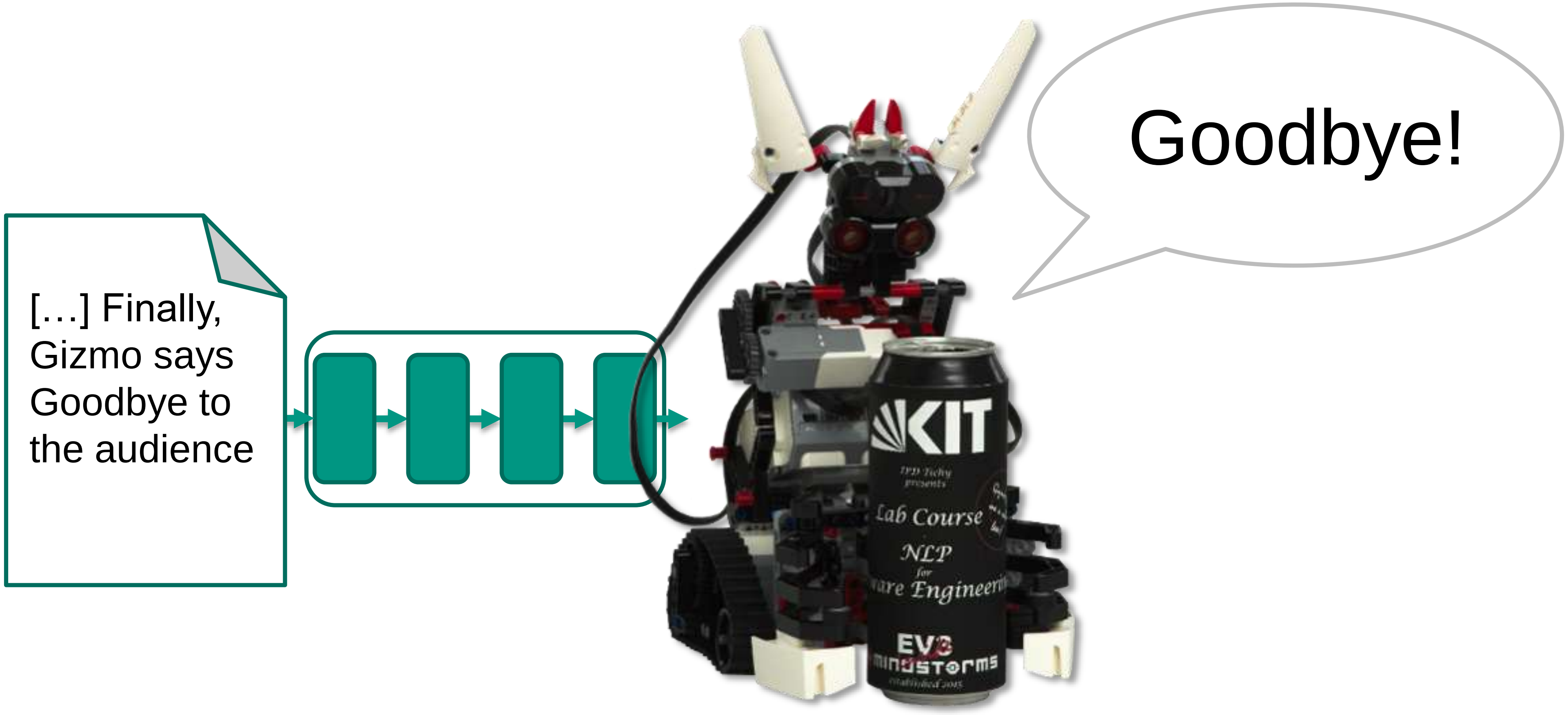
Program a robot with English prose

Interviews, monitoring and observations



Topic | Project | Evaluation

Thank you



[…] Finally, Gizmo says Goodbye to the audience

Goodbye!

References

- [BCKM1997] K. Beckman, N. Coulter, S. Khajenoori, and N. R. Mead, “Collaborations: Closing the industry-academia gap”, *IEEE Software*, vol. 14, no. 6, pp. 49{57, Nov. 1997.
- [WR1999] C. Wohlin and B. Regnell, “Achieving industrial relevance in software engineering education”, in *12th Conf. on Software Engineering Education and Training, 1999. Proceedings*. DUZ Verlags- und Medienhaus GmbH, Mar. 1999, pp. 16 –25.
- [DSTWP2014] M. Daun, A. Salmon, B. Tenbergen, T. Weyer, and K. Pohl, “Industrial case studies in graduate requirements engineering courses: The impact on student motivation”, in *2014 IEEE 27th Conf. on Software Engineering Education and Training*, Apr. 2014, pp. 3–12.
- [B2005] R. Barnett, “*Reshaping the university: new relationships between research, scholarship and teaching”*. McGraw-Hill Education (UK), 2005.
- [R2007] I. Richardson, “Preparing Students for Software Engineering Research”, in *20th Conf. on Software Engineering Education Training*, Jul. 2007, pp. 367–367.
- [KB2009] S. Koolmanojwong and B. Boehm, “Using Software Project Courses to Integrate Education and Research: An Experience Report”, in *2009 22nd Conf. on Software Engineering Education and Training*, Feb. 2009, pp. 26–33.
- [BGBKBBR2016] M. Budde, S. Grebing, E. Burger, M. Kramer, B. Beckert, M. Beigl, and R. Reussner, “Praxis der Forschung: Eine Lehrveranstaltung des forschungsnahen Lehrens und Lernens in der Informatik am KIT”, in *Neues Handbuch Hochschullehre (NHHL)*, Feb. 2016, no. A 3.19
- [AKB2001] L. W. Anderson, D. R. Krathwohl, and B. S. Bloom, A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom’s Taxonomy of Educational Objectives. Longman, 2001.