

Session: AI and Virtual Reality in Education

Enhancing Quality of Computer Science Bachelor Theses by Preparing Students via a Prior Research Methodology Course Including AI Tools

Authors:

Charlotte Sennersten & Kamilla Klonowska
Kristianstad University, Sweden

Brief Index of Contents

Main Objectives

Where we are from [Sweden, Skåne, Kristianstad]

Main Challenge and Goal for Computer Science Theses

Staff Method for Enhancing Quality in Education

Research Methodology Course DA311A

Bachelor Thesis Course DA399E and DT399F

AI Tools Case Study

Results

Student Feedback

References

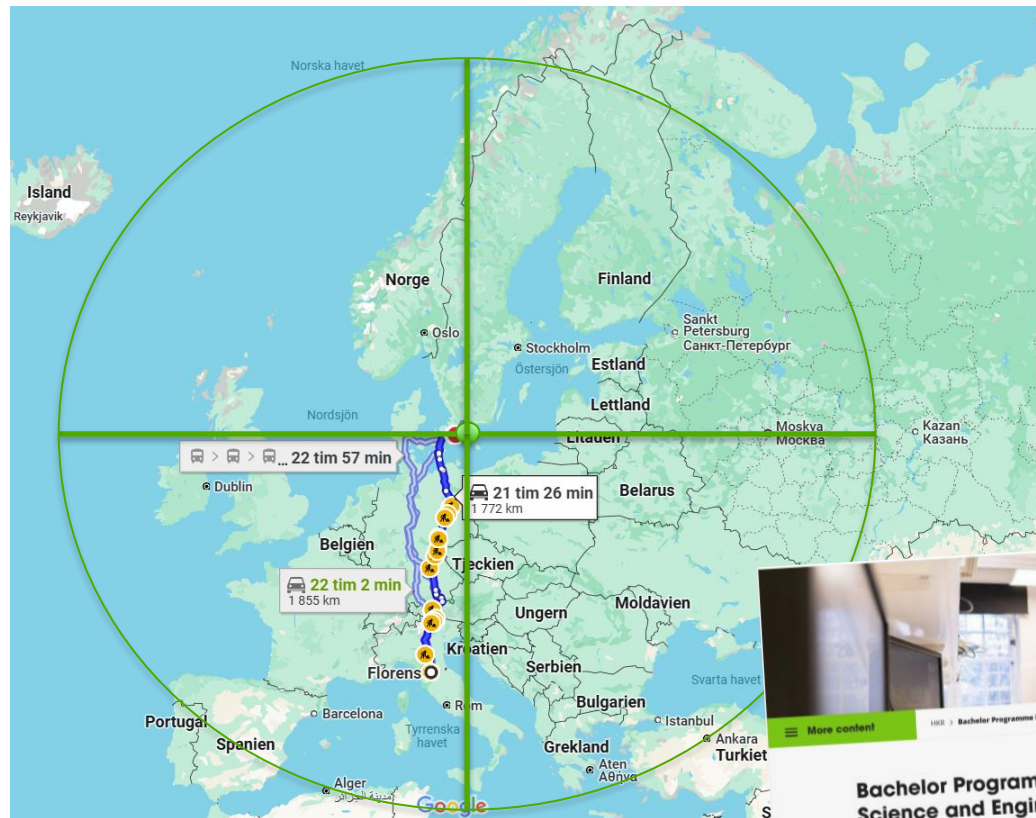
Main Objectives:

Ensuring ACADEMIC_EMPIRICAL quality via a prior Research Methodology Course incorporating AI tools to meet later requirements in a Thesis Course.

How to ensure student comprehension and quality using AI Tooling.

“...growing challenge due to increased complexity in both scientific approaches, engineering skills and the AI tools palette entering the stage. Problem solving skills are fundamental for students to be able to navigate through these information flows to be able to express and formulate their own set up criteria following a logical order to be able to reflect and state outcomes from this process.”

Where are we from? Location.



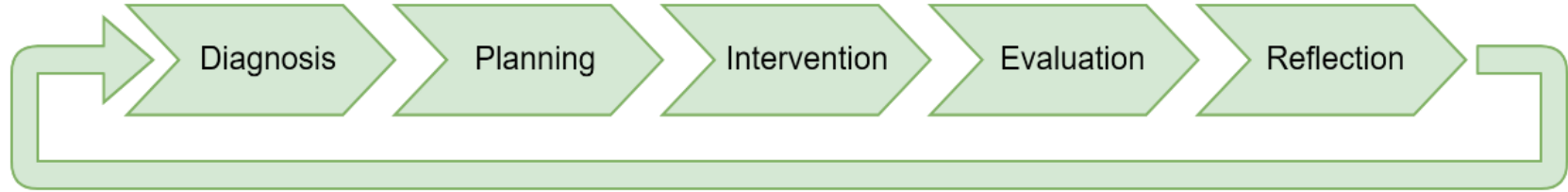
Main Challenge and Goal

To reverse-engineer the thesis process, the end goal must be clearly defined - specifying the requirements and expectations from both scientific and engineering perspectives for computer science students. This end goal can be understood through three key dimensions:

- (1) Achieving the learning outcomes specified in the course syllabus,
- (2) Producing a high-quality thesis, and
- (3) Applying appropriate assessment and grading criteria to the thesis work.

“Integrating AI tools into education [7], along with our evolving roles as educators in response to these changes [8], is crucial in fostering curiosity, supporting independent inquiry, and addressing knowledge gaps as they arise.”

Staff Method for Enhancing Quality in Education



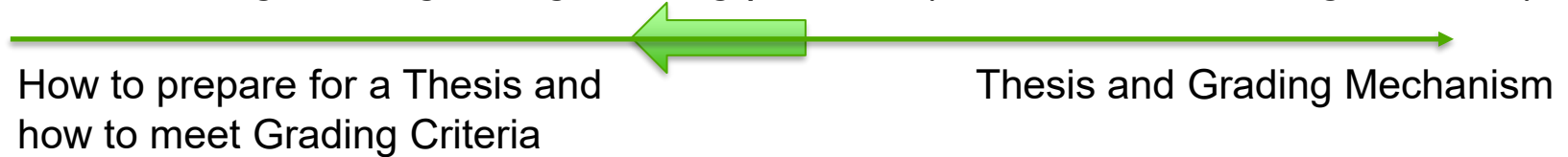
“The educators in the Computer Science Courses at the Kristianstad University Computer Science Department employ an iterative reflection process, grounded in action research (as illustrated in Figure 1 [9]). This ongoing approach involves regular meetings where staff collaboratively discuss and refine the courses to support continuous improvement. Fig 1. The cycle of five stages in action research.”



How do we flip the sequential order and keep to it at the same time?

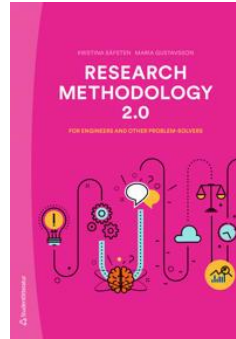
Research Methodology Course, DA311A, 7.5 HEC	Bachelor Thesis Course, DA399E and DT339F, 15 HEC

Reverse Engineering = engineering process (Result and how to get there?)



For teaching staff to be clear!
For students to have clear requirements.

Research Methodology Course, DA311A, 7.5 HEC



*Stronger choice for implementation
purpose*

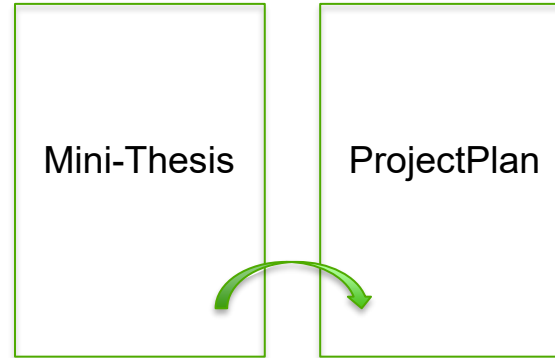
“Wohlin et al. [12] the authors present essential principles for conducting empirical studies in software and hardware engineering, effectively linking research questions and their variables through implementation to achieve meaningful results.”

Research Methodology Course, DA311A, 7.5 HEC

Previous
Course Emphasis



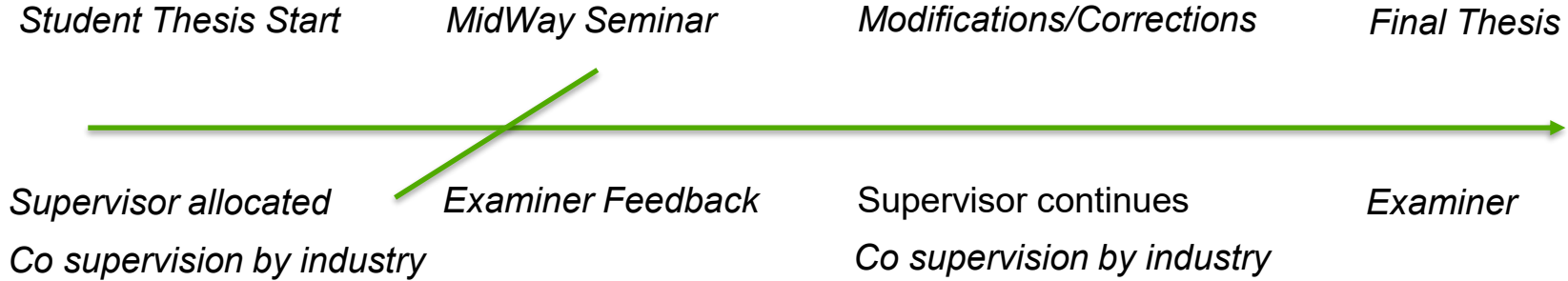
New
Course Emphasis



*A smaller task at the end and
can be lifted into the Thesis Course later.*

“As a result, asking them to draft a project plan for something they have never created or written before can lead to disconnects in purpose and execution.”

Bachelor Thesis Course, DA399E and DT339F



“...to deepen students’ knowledge and skills within a defined area of Computer Science. It is offered with a focus on Software Development (DA399E, 2024) [14] and on Computer Science and Engineering with a specialization in the Internet of Things (DT339F, 2024) [15]. Through the completion of a scientific study, students produce a bachelor thesis. The chosen topic must be relevant to the student’s educational program.”

Bachelor Thesis Course, DA399E and DT339F

“The overall aim of the course is to enable students to independently plan, conduct, and present a scientific study, both in written and oral form. This includes developing the ability to:

Apply scientific methods within the field of computer science or computer engineering.

Demonstrate a deep understanding of relevant theoretical foundations and current research.

Clearly and effectively communicate technical and scientific knowledge at an advanced level.”

AI Tool Case Study: Support via Large Language Model Tools

“As a thought-provoking example, perhaps instead of asking students to produce a flawless, LLM-generated thesis draft, we could ask them to generate ten mini-theses and critically evaluate what is lacking or incorrect in each. While slightly tongue-in-cheek, this suggestion highlights the need to reconsider how we approach content creation and how we evaluate the academic integrity and educational value of student work.”

AI Tool Integration

- Use of LLMs (e.g., ChatGPT) in support—not substitution
- Example: AI-generated grading rubrics
- Prompt use for thesis improvement
- Emphasis: avoid “copy-paste,” foster critical thinking

AI Tool Case Study: Support via Large Language Model Tools

Prompting requesting a grading scheme categorized into Pass, Good, and Very Good, displayed in a clear table format. This type of AI-generated material can serve as a helpful reference for students, offering guidance on what criteria are expected and how their work will be assessed from the outset.

Prompt example, using ChatGPT:

“Show a grading scheme with grade 3, 4 or 5 on a 5-grade scale for an empirical computer science thesis in Computer Science with a focus on software development (DA399E) and Computer Science and Engineering with a focus of Internet of Things (DT339F) at Kristianstad University. What should be included for the different grades?”



Criterion	Grade 3 (Pass)	Grade 4 (Good)	Grade 5 (Very Good)
Problem Formulation	Clear but basic problem; aligned with course scope	Clearly defined and justified; some depth or originality	Well-defined, clearly motivated, and shows originality or innovation
Literature Review	Sufficient to support understanding; sources are relevant	Comprehensive with relevant comparisons and some analysis	Critical, analytical, well-structured; integrates sources to form arguments
Methodology	Appropriate method chosen; basic understanding shown	Method well-chosen, explained, and adapted to the context	Strong methodological rigor, justified choices, and critical understanding demonstrated
Implementa-tion / Experimenta-tion	Functionally correct; limited complexity	Effective implementation, shows problem-solving and software engineering skills	High-quality, innovative, or technically complex solution; strong engineering practices
Evaluation / Results	Basic evaluation; limited but valid conclusions	Clear evaluation using relevant metrics; thoughtful interpretation	In-depth, critical evaluation; data well-analyzed; limitations discussed
Academic Writing	Mostly correct language and structure; citations mostly used correctly	Well-structured, good language use, formal tone; references follow academic standards	Excellent academic writing; coherent, well-structured, precise; citations and formatting consistently correct
Reflection & Contribution	Basic reflection on work and learning	Good self-awareness and understanding of results' significance	Deep reflection; clear articulation of contributions and broader impact
Independence & Project Management	Completed with some supervision; basic planning shown	Shows initiative; manages time and tasks with limited supervision	Highly independent; excellent planning, problem-solving, and communication with supervisor
Relevance to Field (SW Dev / IoT)	Relevant to the field; meets minimal expectations	Adds some value or insight to the chosen area	Strong relevance; demonstrates contribution or innovation within Software Dev or IoT

AI Tool Case Study: Support via Large Language Model Tools

Research Question Improvement With Grade Scale



	Stage	Research Question	Clarity	Specificity	Feasibility	Novelty	Grade Scale
1	Low Quality	How is software implemented using computer science theory?	Too broad and lacks focus	Covers a vast area with many interpretations	Difficult to answer due to wide scope	General topic with extensive prior research	Fail or Needs Major Improvement
2	Medium Quality	How can algorithmic principles improve software performance?	More specific but still general	Narrowed down to algorithmic principles	More feasible but still needs refinement	Somewhat original but could be explored further	Pass
3	High Quality	How can graph theory-based algorithms optimize large-scale software system performance?	Clearly defined with a precise scope	Focused on a well-defined theoretical and practical aspect	Achievable with clear methodologies and data	Addresses a specific research gap with innovation	Pass with Distinction

AI Tool Case Study: Support via Large Language Model Tools

“Students can use LLMs such as ChatGPT, Gemini, or similar tools by crafting targeted prompts to support their own knowledge development in parallel with meeting formal academic requirements.”

“These models can swiftly generate relevant content, aiding students in refining their research questions and developing well-structured project plans. By leveraging the capabilities of AI, students can cultivate a deeper understanding of their subject area and advance their academic work with increased clarity and precision.”

“However, to fully benefit from these tools, it is essential to ensure, nurture and maintain logical consistency and coherence throughout the research and writing process. Given the vast amount of content LLMs can generate, adopting a structured approach guided by a clear and focused line of reasoning is crucial.”

Student Feedback Highlights

“Solid Foundation in Subject Matter”

“Appreciated Structured and Supportive Environment”

“Course Improvements are Evident and Impactful”

**Courses improved based on
student feedback
and
staff consolidations!**

References

1. ACM/IEEE Computer Science Curricula, <https://ieeecs-media.computer.org/media/education/reports/CS2023.pdf>, 2023, Visited 06/05/2025.
2. The Swedish Higher Education Act (1992:1434) - Swedish Council for Higher Education, accessed 06/05/2025.
3. IEEE Computer Society, IEEE Computer Society and Industry Partners Release Guidance in the Form of Undergraduate Computer Science Curricula, New CS2023 Curriculum Guide Released by IEEE CS and Partners, accessed 06/05/2025.
4. Higher Education Ordinance (1993:100), Ministry of Education and Research (Sweden), in Swedish: Regeringskansliets rättsdatabaser, accessed 06/05/2025.
5. Veletanlic, B., Klonowska, K. "Generic Competencies and Research Integration in Education – An Academic Track for Computer Science Students" (published in Swedish: "Generella kompetenser och forskningsanknytning i utbildningen - en akademisk slinga för dataloger"), Lärarlärdom Conference 2015, Växjö, Sweden, 163-181, <https://doi.org/10.15626/lld.201508>, 2016.
6. Klonowska, K., Frisk, F., Einarson, D., "The win-win of synchronizing last semester's Computer Engineering courses", 17th International CDIO Conference, Bangkok, Thailand, Online, June 2021.
7. Bista, K., Bista, R. "Leveraging AI tools in academic writing: Insights from doctoral students on benefits and challenges", American Journal of STEM Education, 6, 32-47. <https://doi.org/10.32674/9m8dq081>, 2025.
8. Zawacki-Richter, O., Marín, V.I., Bond, M. et al. "Systematic review of research on artificial intelligence applications in higher education – where are the educators?". Int J Educ Technol High Educ 16, 39, 2019, <https://doi.org/10.1186/s41239-019-0171-0>.
9. Dickens, L., & Watkins, K., "Action Research: Rethinking", Lewin. Management Learning, 30(2), 127-140, <https://doi.org/10.1177/1350507699302002>, 1999.
10. DA311A, Course syllabus –Research Methodology, 7.5 HEC, English, HKR.se, Course syllabus - Research Methodology in Computer Science - da311a , English | HKR.se, 2022.
11. Säftsen, K., Gustavsson, M., "Research Methodology 2.0 -for engineers and other problem-solvers", Studentlitteratur, 2nd edition, 2023.
12. Wohlin, C., Runeson, P., Höst, M., Ohlsson, C., M., Regnell, B., Wessle'n, A., "Experimentation in Software Engineering", Second edition, Springer, 2024.
13. Carless, D. "Feedback loops and the longer-term: towards feedback spirals", Assessment & Evaluation in Higher Education, 44(5), 705–714. <https://doi.org/10.1080/02602938.2018.1531108>, 2018.
14. DA399E, Course syllabus – Bachelor Thesis in Computer Science, 15 HEC, English, HKR.se, Course syllabus - Bachelor Thesis in Computer Science - da399e , English | HKR.se, 2024.
15. DT339F, Course syllabus – Bachelor Thesis in Computer Engineering, 15 HEC, English, HKR.se, Course syllabus - Bachelor Thesis in Computer Engineering - dt339f , English | HKR.se, 2024.
16. Hong, Yining, et al. "3d-llm: Injecting the 3d world into large language models", Advances in Neural Information Processing Systems, 36, 20482-20494, 2023.