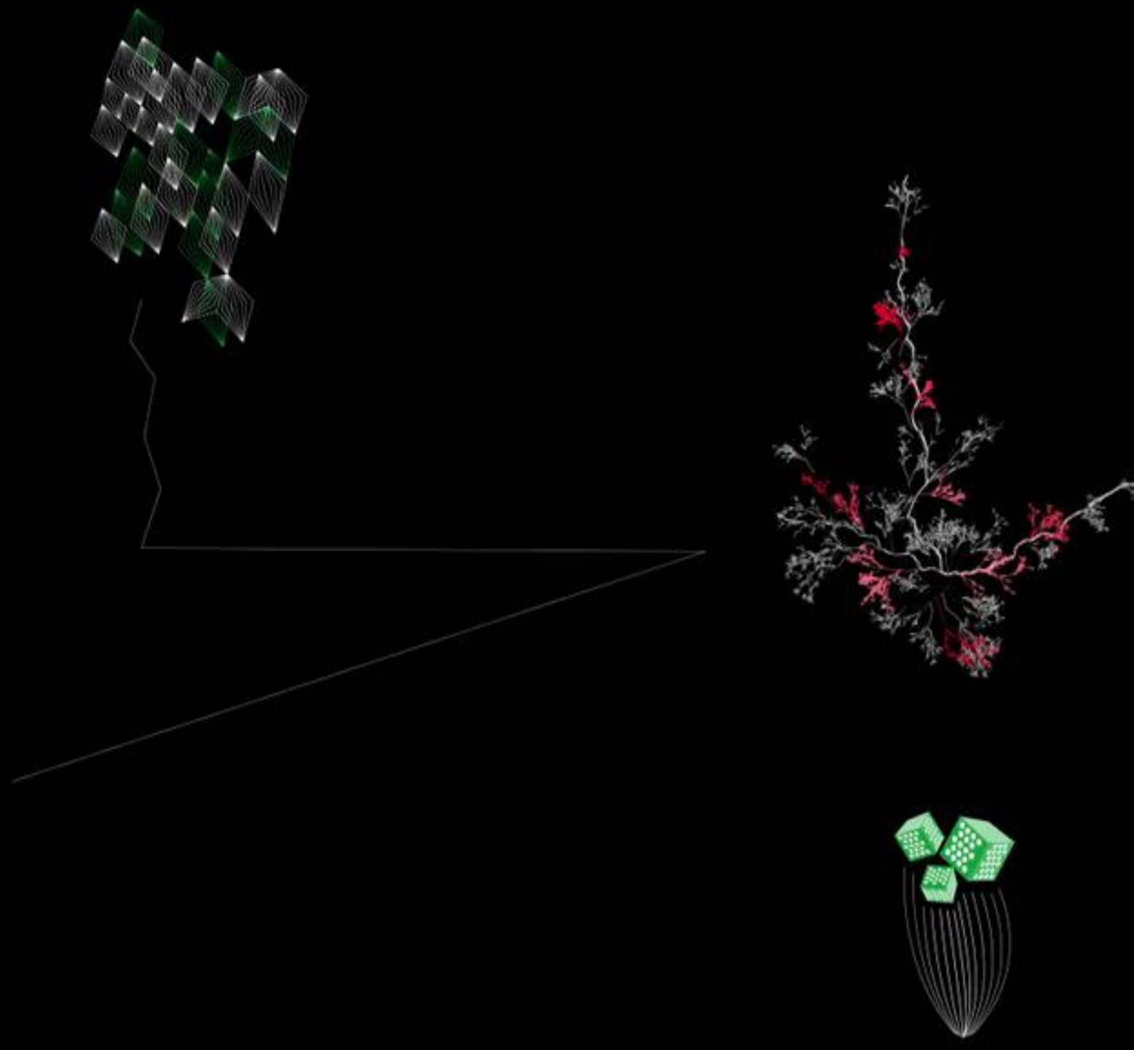
Formal Methods and ~~Tools~~Teaching
University of Twente

GEN-AI TOOLS IN PROJECT-BASED LEARNING

LETTING STUDENTS PLAY WITH COPILOT AND CHATGPT IN THEIR FIRST SEMESTER PROGRAMMING PROJECT



UNIVERSITY
OF TWENTE.

INVESTIGATORS

- Tom van Dijk
  - Coordinator of Q2 Software Systems (module) (15 EC)
  - Teacher of Q2 Object-oriented Programming (8 EC)

- Vadim Zaytsev
  - Programme Director of Technical Computer Science
  - Teacher of Q2 Software Design (4 EC)

- Internally funded by the University of Twente

UNIVERSITY
OF TWENTE.

OVERVIEW OF THE STUDY

- AI is upon us! Doom! [citation needed]
- Lack of understanding what implications are on the learning process
- Do students even need to learn programming? Or report writing?
- Should we force students to (learn to) use AI for programming?
- Is using ChatGPT academic misconduct? Always or sometimes?
- Can everyone use AI to successfully do the programming project?
- Are any playing fields being leveled? Or is everything more unequal?

BUT WAIT, WHAT IS THE CONTEXT?

- First-semester 8 EC programming course in **Java**
  - Object-oriented programming, Java collections, JML, basics of concurrency, a little bit of networking with Java sockets
- Students start with (near) zero programming experience
- ± 7-8 weeks of programming lectures and practicals
- ± 2 weeks of programming project
  - Implement a **simple** board game
  - Write a server and a client for this board game
  - Implement a very basic AI and play against each other

UNIVERSITY
OF TWENTE.

RESEARCH QUESTIONS

1. Impact on quality of code and quality of report
2. Influence on time to completion, amount of effort, efficiency
3. Which rubric criteria / learning objectives are affected and how?
4. How is understanding of code and problem-solving affected?
5. How does it affect the learning process and student engagement?

METHODOLOGY

- Hire student assistants (~40) to repeat the project over the summer

- Four groups
  1. Control group
  2. Use Github Copilot ("advanced autocomplete")
  3. Use ChatGPT
  4. Use both

- **Group assignment** such that each group has the same mixture of low/medium/high programming skill, low/high Copilot skill, low/high ChatGPT skill (obtained via self-report)

UNIVERSITY
OF TWENTE.

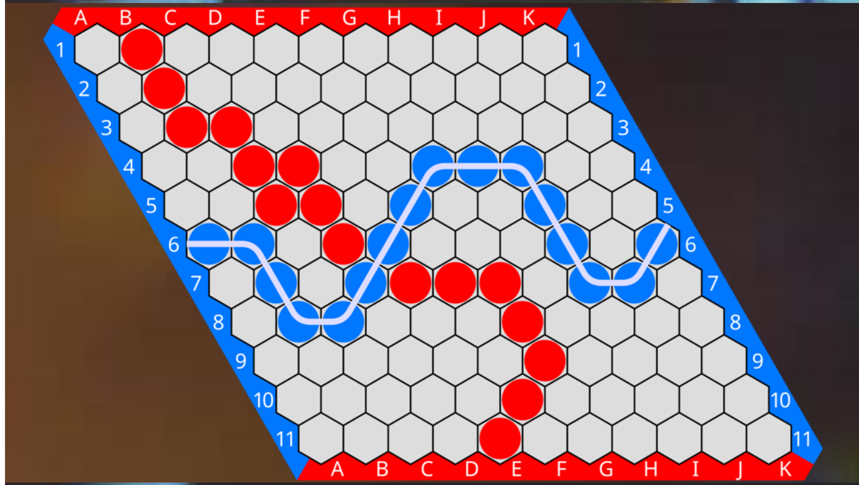
METHODOLOGY

- 36 hours of programming project, code + report, aiming for a high grade
- Maintain **reflective journal** and **monitoring spreadsheet**
- Afterwards: **self-assessment** and **peer review** to establish grade
- Followed by: **focus group meetings**

PROGRAMMING PROJECT

(FOR THE PILOT STUDY)

- Each project is done **solo**
- Each project is done in **full**: server, client, AI
- The game is **Hex**, slightly more difficult than normal projects
- Consider the **best strategy** to get a high grade within 36 hours



UNIVERSITY
OF TWENTE.

REFLECTIVE JOURNAL

- Every 1-2 hours, write an entry in the reflective journal
  - Should take 5-10 minutes
  - Also for the control group!

- Additional entries whenever there are observations or experiences they'd like to note down

- Instructed to be specific and detailed, screenshots, etc.

- Manually tagged afterwards

FOCUS GROUP MEETINGS

- Three focus group meetings with 5-10 humans

REFLECTION JOURNALS

(READ AND TAGGED MANUALLY)

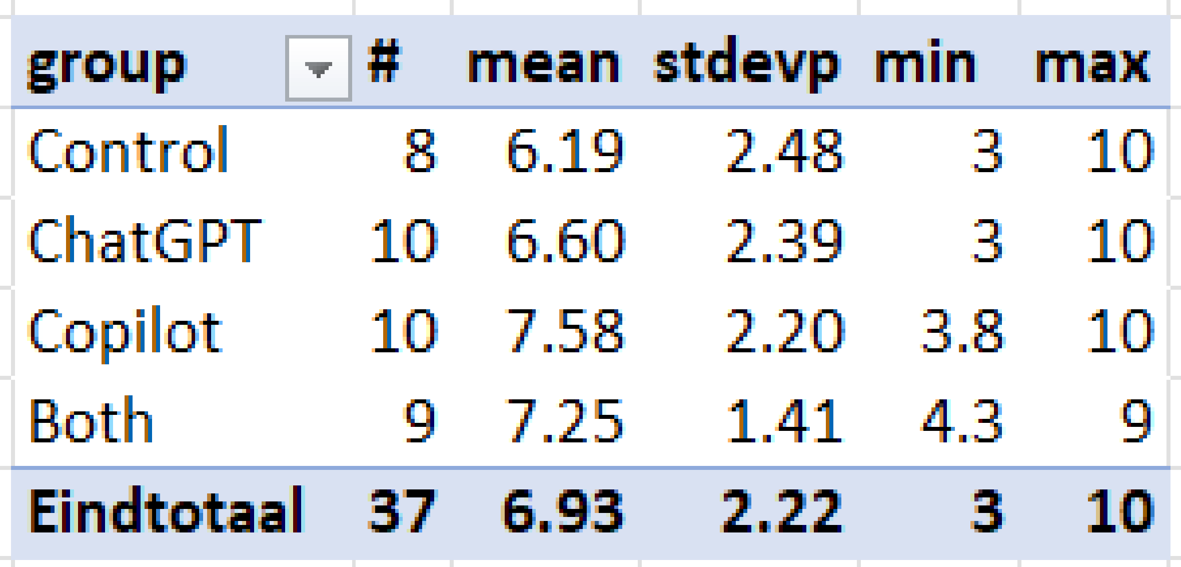
- Process
  - **joy** — positive reactions/emotions
  - **pain** — negative reactions/emotions
  - **boost** — saving time or energy
  - **reason** — getting (or not getting) explanations from AI

- Activities
  - **refactoring** — intentional changes in the code
  - **debugging** — dealing with errors and defects
  - **testing** — probing, writing or running tests
  - **planning** — project outlining together with AI
  - …

- Concrete
  - **concurrency** — threads, race conditions, etc
  - **class diagram** — apparently more popular than sequence diagrams
  - ...

QUANTITATIVE RESULTS (GRADES)

- Functionality (20%)

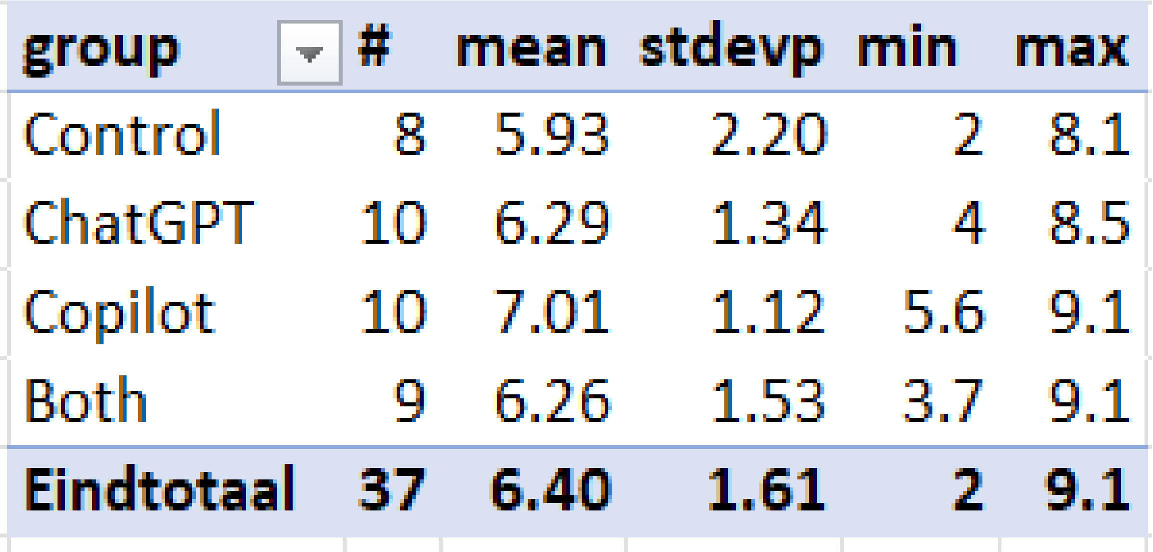


group ▼	#	mean	stdevp	min	max
Control	8	6.19	2.48	3	10
ChatGPT	10	6.60	2.39	3	10
Copilot	10	7.58	2.20	3.8	10
Both	9	7.25	1.41	4.3	9
Eindtotaal	**37**	**6.93**	**2.22**	**3**	**10**

UNIVERSITY
OF TWENTE.

QUANTITATIVE RESULTS (GRADES)

- Software (40%)

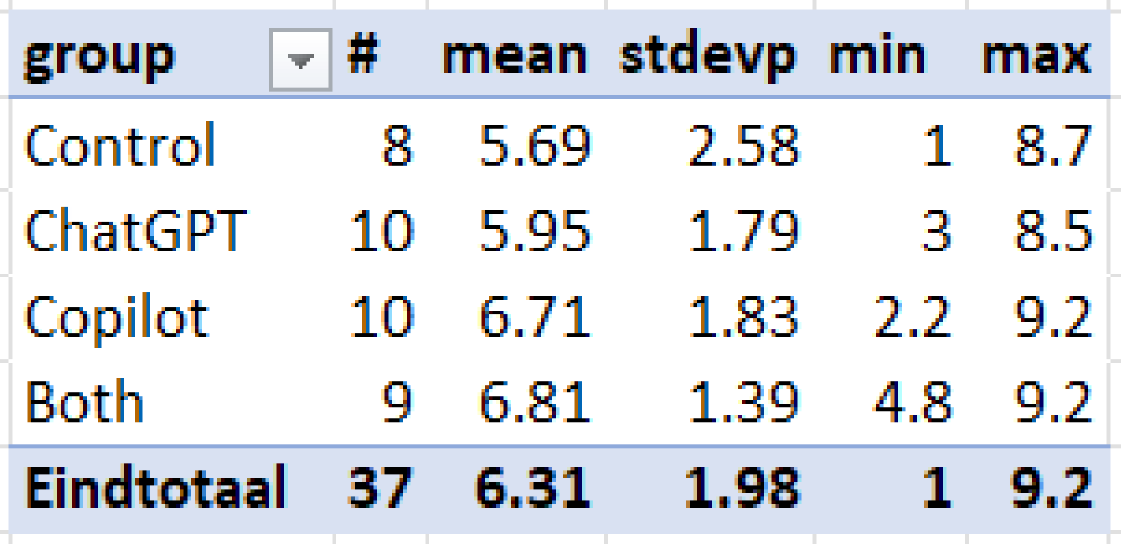


group	#	mean	stdevp	min	max
Control	8	5.93	2.20	2	8.1
ChatGPT	10	6.29	1.34	4	8.5
Copilot	10	7.01	1.12	5.6	9.1
Both	9	6.26	1.53	3.7	9.1
Eindtotaal	**37**	**6.40**	**1.61**	**2**	**9.1**

UNIVERSITY
OF TWENTE.

QUANTITATIVE RESULTS (GRADES)

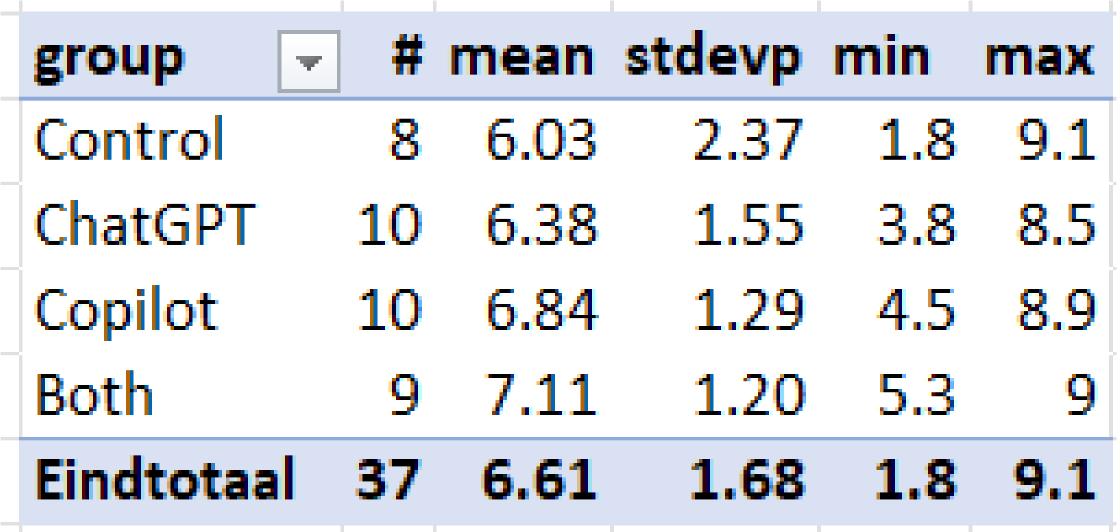
- Report (40%)



group	#	mean	stdevp	min	max
Control	8	5.69	2.58	1	8.7
ChatGPT	10	5.95	1.79	3	8.5
Copilot	10	6.71	1.83	2.2	9.2
Both	9	6.81	1.39	4.8	9.2
Eindtotaal	**37**	**6.31**	**1.98**	**1**	**9.2**

QUANTITATIVE RESULTS (GRADES)

- Final grade



group	#	mean	stdevp	min	max
Control	8	6.03	2.37	1.8	9.1
ChatGPT	10	6.38	1.55	3.8	8.5
Copilot	10	6.84	1.29	4.5	8.9
Both	9	7.11	1.20	5.3	9
Eindtotaal	**37**	**6.61**	**1.68**	**1.8**	**9.1**

CODE ANALYSIS

Files: 14..63 [25.7] Classes: 12..32 [18.6] LOC: 2671..25095 [11473] CC: [2.01] Fields: 32..56 [47.3] Methods: 49..158 [97.0]	Files: 10..73 [25.4] Classes: 9..53 [20.7] LOC: 7018..23064 [14282] CC: [2.35] Fields: 23..64 [48.9] Methods: 57..316 [128.1]
Files: 13..47 [26.1] Classes: 14..35 [22.6] LOC: 6758..38918 [19676] CC: [1.83] Fields: 48..136 [71.3] Methods: 72..247 [158.0]	Files: 10..37 [23.1] Classes: 8..36 [19.5] LOC: 3423..18721 [12849] CC: [2.28] Fields: 20..148 [65.1] Methods: 33..172 [114.5]

UNIVERSITY
OF TWENTE.

HIGHLIGHTS

(BASED ON REAL QUOTES FROM JOURNALS)

- *I started to believe that the more questions I ask, the dumber it gets.*
  - (technical problems with token limit, other hiccups and shortcomings)
  - (we can expect this to be fixed in next versions)

- *I didn’t want to go over the details, which is why I sent what is inside the file. It solved the issue!*
  - (YOLOing through the project works; if you want to avoid learning, you will)

- *The quality of that I didn’t check.*
  - (testing and documentation suffered most)

MORE HIGHLIGHTS

(BASED ON REAL QUOTES FROM JOURNALS)

- *And then I spotted another assumption that was made by the AI and it was wrong.*
  - (limited context leads to many nontrivial assumptions forming technical debt)

- *I overspend time on trying to figure out the problem using ChatGPT.*
  - (straightforward debugging/coding could have been easier)

- *I found this extremely unhelpful and tried resolving it myself and I did succeed.*
  - (taking over in complex situations works best)
  - (just like with more junior colleagues!)

UNIVERSITY
OF TWENTE.

EVEN MORE HIGHLIGHTS

- Constant anthropomorphising of technology
  - "I ask *him* to fix..."
  - "At first … but then it *understood* when I explained..."
- Context is often lacking
  - token limit or negligence of the "developer"
- Requires skill
  - should we teach it?
- If it works, it speeds up code writing
  - It's not necessarily correct code!
  - development not necessarily sped up
  - no feeling of complexity

UNIVERSITY
OF TWENTE.

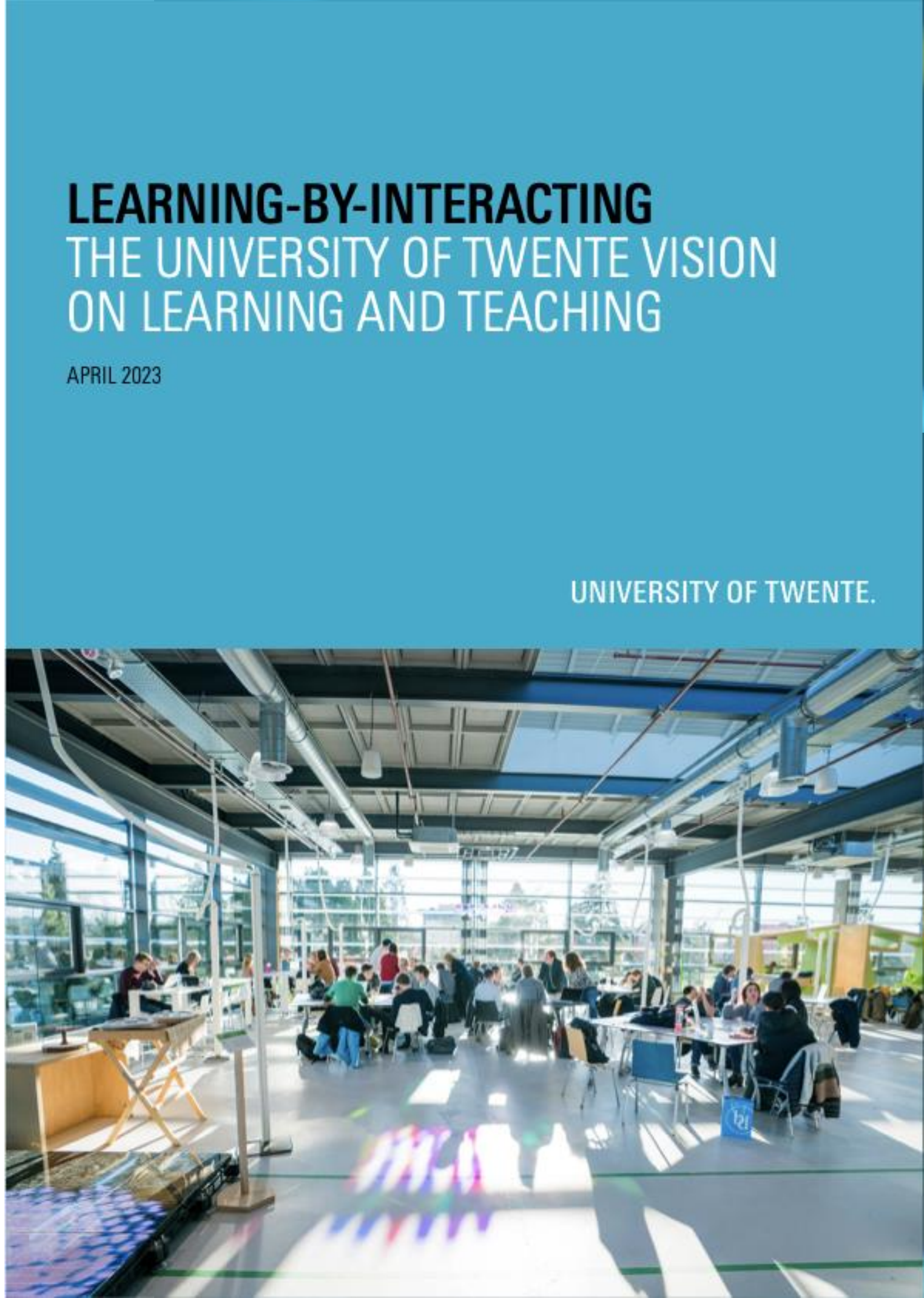
IMPACT ON QUALITY OF CODE AND REPORT

- **Code quality**
  - Mixed impact in general
  - Good for writing documentation, Javadoc, comments, (some JML)
  - Great for simple methods
  - ChatGPT has tendency to hallucinate methods
  - ChatGPT tends to write a lot of redundant code
  - Helpful for boilerplate, less helpful for multiple classes

- **Report**
  - ChatGPT can create content but lacks critical thinking, detail, human touch

LEARNING BY DOING

- Currently
  - large "integrating" projects
  - you learn because you **do** (yourself)
  - testing equates quality of artefact with quality of learning
- With AI
  - no doing necessary => no learning guaranteed
  - quality of artefact is even more removed from learning
- Possible solution
  - project is pass/fail (signed off)
  - opportunities to demonstrate what was learnt



LEARNING-BY-INTERACTING
THE UNIVERSITY OF TWENTE VISION
ON LEARNING AND TEACHING

APRIL 2023

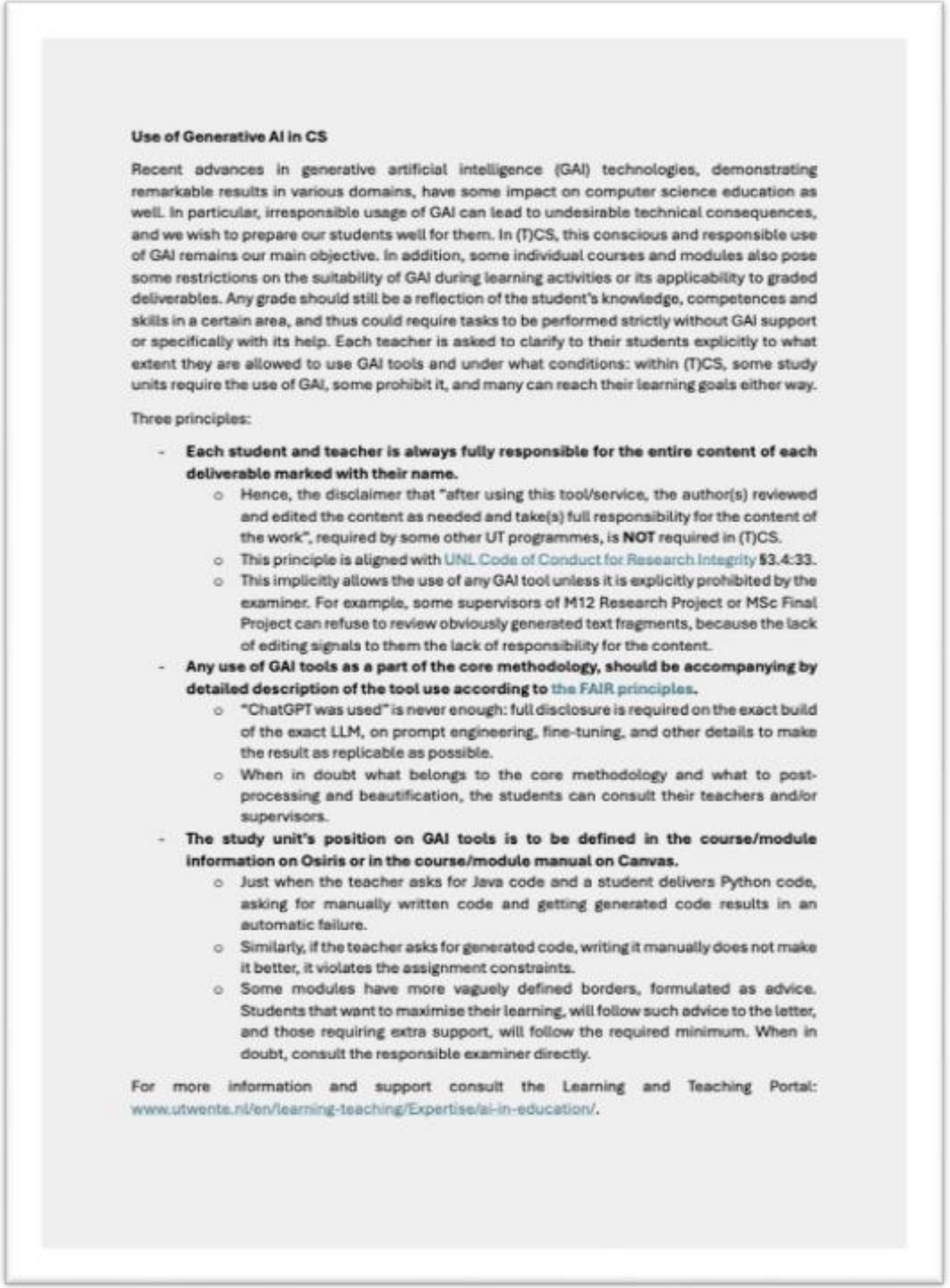
UNIVERSITY OF TWENTE.

AI IN CS POLICY

(UNIVERSITY OF TWENTE EDITION)

- **First proposal (another faculty)**
  - ~~let's ban AI on this campus!~~
  - let's copy from Elsevier
  - prohibited unless permitted
  - clear statement if used
  - clear statement if not used
  - focused on fraud and liability

- **CS-specific policy**
  - permitted unless prohibited
  - be responsible for what you submit
  - if AI is core, tell all the details
  - if not core, nobody cares
  - may vary per study unit
  - focused on not interfering with learning practices



Use of Generative AI in CS

Recent advances in generative artificial intelligence (GAI) technologies, demonstrating remarkable results in various domains, have some impact on computer science education as well. In particular, irresponsible usage of GAI can lead to undesirable technical consequences, and we wish to prepare our students well for them. In (T)CS, this conscious and responsible use of GAI remains our main objective. In addition, some individual courses and modules also pose some restrictions on the suitability of GAI during learning activities or its applicability to graded deliverables. Any grade should still be a reflection of the student's knowledge, competences and skills in a certain area, and thus could require tasks to be performed strictly without GAI support or specifically with its help. Each teacher is asked to clarify to their students explicitly to what extent they are allowed to use GAI tools and under what conditions: within (T)CS, some study units require the use of GAI, some prohibit it, and many can reach their learning goals either way.

Three principles:

- **Each student and teacher is always fully responsible for the entire content of each deliverable marked with their name.**
  - Hence, the disclaimer that "after using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the work", required by some other UT programmes, is **NOT** required in (T)CS.
  - This principle is aligned with UNL Code of Conduct for Research Integrity §3.4:33.
  - This implicitly allows the use of any GAI tool unless it is explicitly prohibited by the examiner. For example, some supervisors of M12 Research Project or MSc Final Project can refuse to review obviously generated text fragments, because the lack of editing signals to them the lack of responsibility for the content.
- **Any use of GAI tools as a part of the core methodology, should be accompanying by detailed description of the tool use according to the FAIR principles.**
  - "ChatGPT was used" is never enough: full disclosure is required on the exact build of the exact LLM, on prompt engineering, fine-tuning, and other details to make the result as replicable as possible.
  - When in doubt what belongs to the core methodology and what to post-processing and beautification, the students can consult their teachers and/or supervisors.
- **The study unit's position on GAI tools is to be defined in the course/module information on Osiris or in the course/module manual on Canvas.**
  - Just when the teacher asks for Java code and a student delivers Python code, asking for manually written code and getting generated code results in an automatic failure.
  - Similarly, if the teacher asks for generated code, writing it manually does not make it better, it violates the assignment constraints.
  - Some modules have more vaguely defined borders, formulated as advice. Students that want to maximise their learning, will follow such advice to the letter, and those requiring extra support, will follow the required minimum. When in doubt, consult the responsible examiner directly.

For more information and support consult the Learning and Teaching Portal: www.utwente.nl/en/learning-teaching/Expertise/ai-in-education/.

FINAL THOUGHTS

- Students ***will*** use generative AI for take-home work
  - It is ***almost the same*** as having a tutor / senior student / mentor / parent that is available all day and rarely complains

- Consider role of take-home work in the course: assessment or learning?

- Consider an assessment strategy that is generative-AI-aware
  - Oral exams / presentations after submission
  - Written “project exam” after submission
  - Just changing the rubric may ***not*** be sufficient!

- Guidelines on using generative AI
  - Clarification of difficult concepts
  - Generate feedback rather than primary output
  - Do ***not*** use as a crutch when debugging or learning, and use it to give additional feedback after completing an exercise