Integrating Large Language Models into Software Engineering Education

Current Trends and Future Directions

Cigdem Sengul, **Rumyana Neykova**, and Giuseppe Destefanis

Computer Science Department
Brunel University London

Overview

- **State of the Art:** AI in SE Industry & Education
- **Our Project:** LLM Toolkit for SE Education
- **Hands-on Practice:** Gamifying Requirements Engineering
- **LLM Platform Comparison:** Insights from Experiments
- **Student Experience:** Requirements Engineering Detective Game

Part 1:

State of the art
AI in SE Industry and Education

frontiers | Frontiers in Artificial Intelligence

TYPE Review
PUBLISHED 29 August 2024
DOI 10.3389/frai.2024.1436350

OPEN ACCESS

EDITED BY
Deborah Richards,
Macquarie University, Australia

REVIEWED BY
Toby Dragon,
Ithaca College, United States
Ansgar Fehnker,
Macquarie University, Australia

*CORRESPONDENCE
Cigdem Sengul
cigdem.sengul@brunel.ac.uk

RECEIVED 21 May 2024
ACCEPTED 10 July 2024
PUBLISHED 29 August 2024

CITATION
Sengul C, Neykova R and Destefanis G (2024) Software engineering education in the era of conversational AI: current trends and future directions. *Front. Artif. Intell.* 7:1436350. doi: 10.3389/frai.2024.1436350

Software engineering education in the era of conversational AI: current trends and future directions

Cigdem Sengul*, Rumyana Neykova and Giuseppe Destefanis

Computer Science Department, Brunel University London, Uxbridge, United Kingdom

The developments in conversational AI raised urgent questions about the future direction of many aspects of society, including computing education. The first reactions to the fast-paced evolution of conversational agents were varied: Some announced "the end of programming," while others considered this "premature obituary of programming." Some adopted a defensive approach to detecting the use of conversational AI and avoiding an increase in plagiarism, while others questioned, "So what if ChatGPT wrote it?" Nevertheless, questions arise about whether computing education in its current form will still be relevant and fit for purpose in the era of conversational AI. Recognizing these diverse reactions to the advent of conversational AI, this paper aims to contribute to

Survey Overview

Methodology & Scope

- PRISMA-guided rapid review: 183 peer-reviewed articles (2018-2024)
- Three distinct categories analyzed:
  - Software Engineering Industry (SE I): 115 papers
  - Computing Science Higher Education (CS HE): 52 papers
  - Software Engineering Higher Education (SE HE): only 16 papers

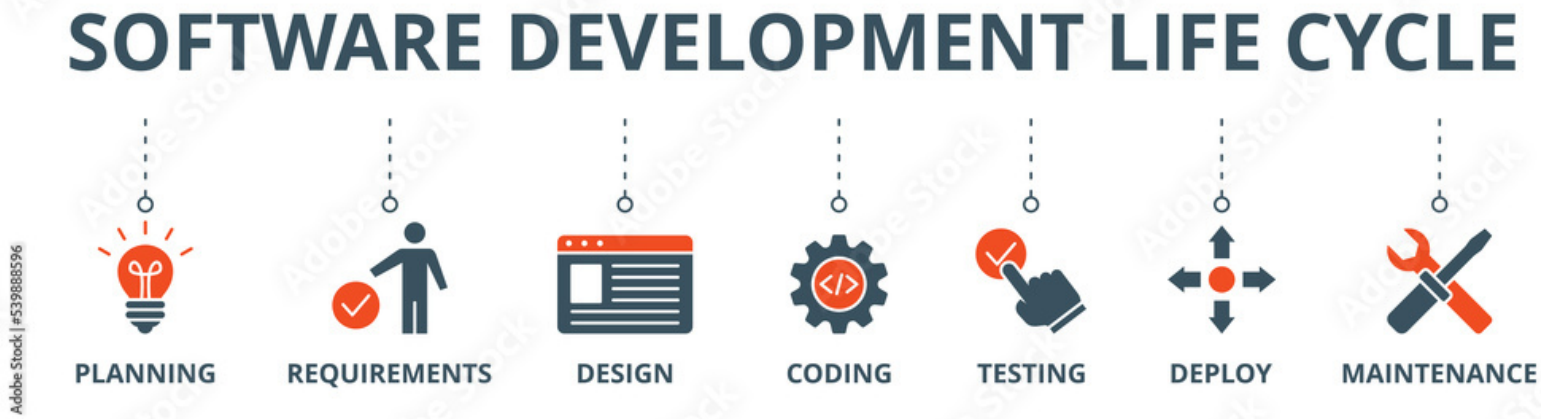
Citation

Sengul C, Neykova R and Destefanis G (2024) *Software engineering education in the era of conversational AI: current trends and future directions*. Front. Artif. Intell. 7:1436350. doi: 10.3389/frai.2024.1436350

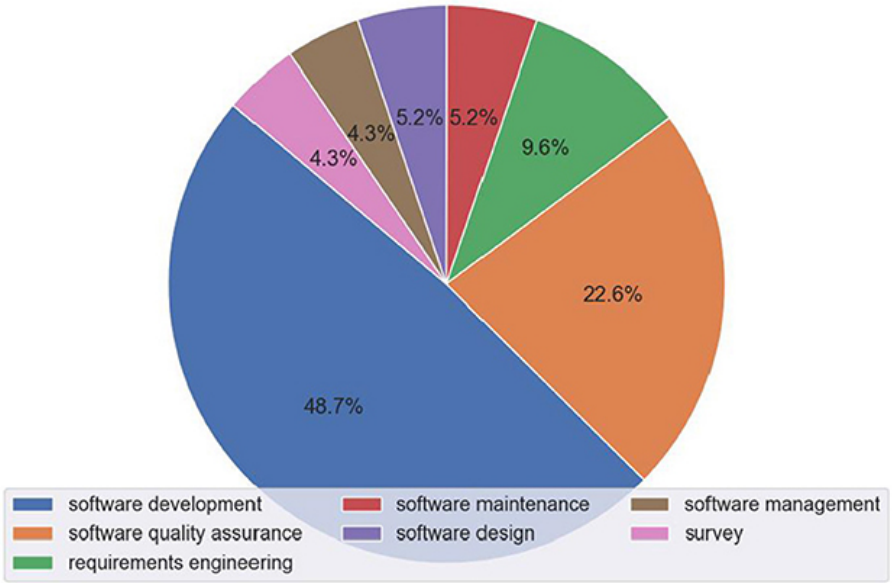
Background: The main phases of software development

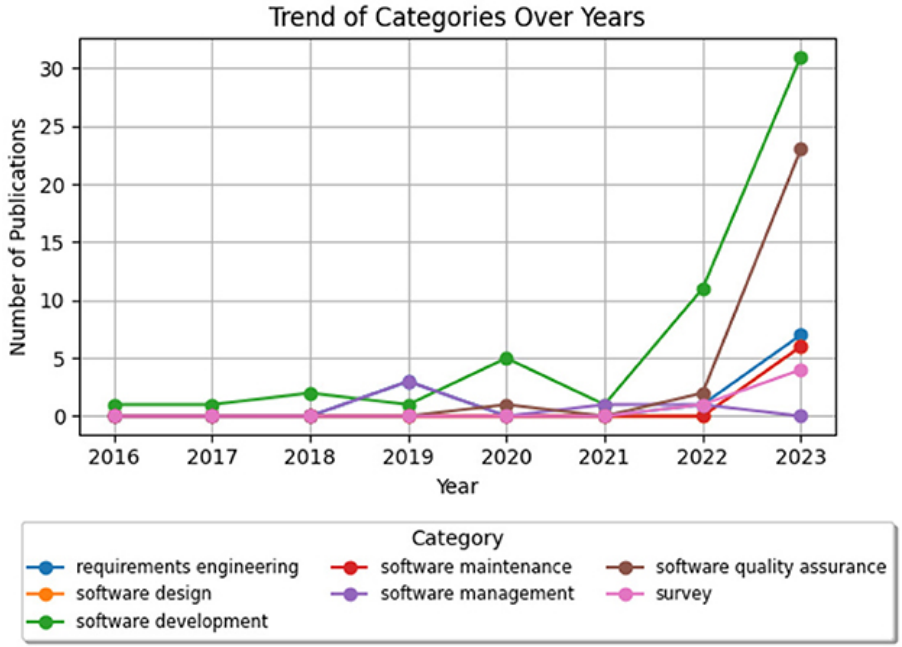
LLM applications in

- Requirements Engineering
- Design
- Development
- Testing
- Maintenance



LLM papers in SDLC





Current State of LLMs in SE

- Development phase most mature (48.7% of studies)
- Quality Assurance emerging (22.06%)
- Requirements Engineering and Design promising (14.8%)

Key Findings

Publication Trends

- Sharp increase from 2018-2023, especially post-ChatGPT release
- US leads (followed by Canada, Germany, China)
- Strong international collaborations network

Industry vs Education Adoption

- Industry adoption far outpaces educational integration
- Software development (48.7%) and quality assurance (22.6%) lead industry applications
- CS education papers primarily focus on coding tasks (70%)
- Non-coding aspects students struggle with are neglected:
  - Requirements specification
  - Architectural design
  - Testing beyond unit tests
  - Collaborative processes

Promising Educational Applications

Innovative Applications in SE Education

Requirements Engineering Using LLMs to generate realistic stakeholder interviews with *intentional mistakes* for training (Gorer & Aydemir, 2023)

Software Design SOCIO chatbot for UML modeling outperformed traditional tools in student satisfaction and efficiency (Ren et al., 2023)

Secure Development ChatGPT-4 identified 20/28 vulnerabilities in web application code **plus** found 4 previously *undetected* issues (Li et al., 2023)

Underexplored Areas

Requirements Engineering

System Design

Complex Testing

Key Observations

The Chicken-Egg Problem

- Students need SE knowledge to effectively prompt AI tools
- But AI tools are changing how SE knowledge should be taught

Findings

- Students struggled to express requirements to AI and wished "AI could read their minds"
- Self-described "introverts" preferred AI pair programming over human partners
- Professionals were more comfortable with AI collaboration than students
- Students expressed: "basics of software engineering must be taught without AI and before AI-assisted programming"

Moving Forward

Our Take on the Findings

- SE education shouldn't just react to AI tools—it should proactively shape them
- Learning approach shift: from "coding first" to "design and analysis first"
- Need for tools that showcase AI's limitations as well as capabilities

Practical Next Steps

- Develop AI-enhanced teaching modules for underrepresented SE areas:
  - Requirements engineering activities
  - Software architecture visualization
  - Advanced testing strategies
  - Collaborative development workflows
- Create frameworks for ethical AI integration in SE curriculum
- Design tailored tools for different student personality types

Part 2:

CPHC-funded project

LLM Toolkit for Software Engineering Education

LLM Toolkit for Software Engineering Education (1/2)

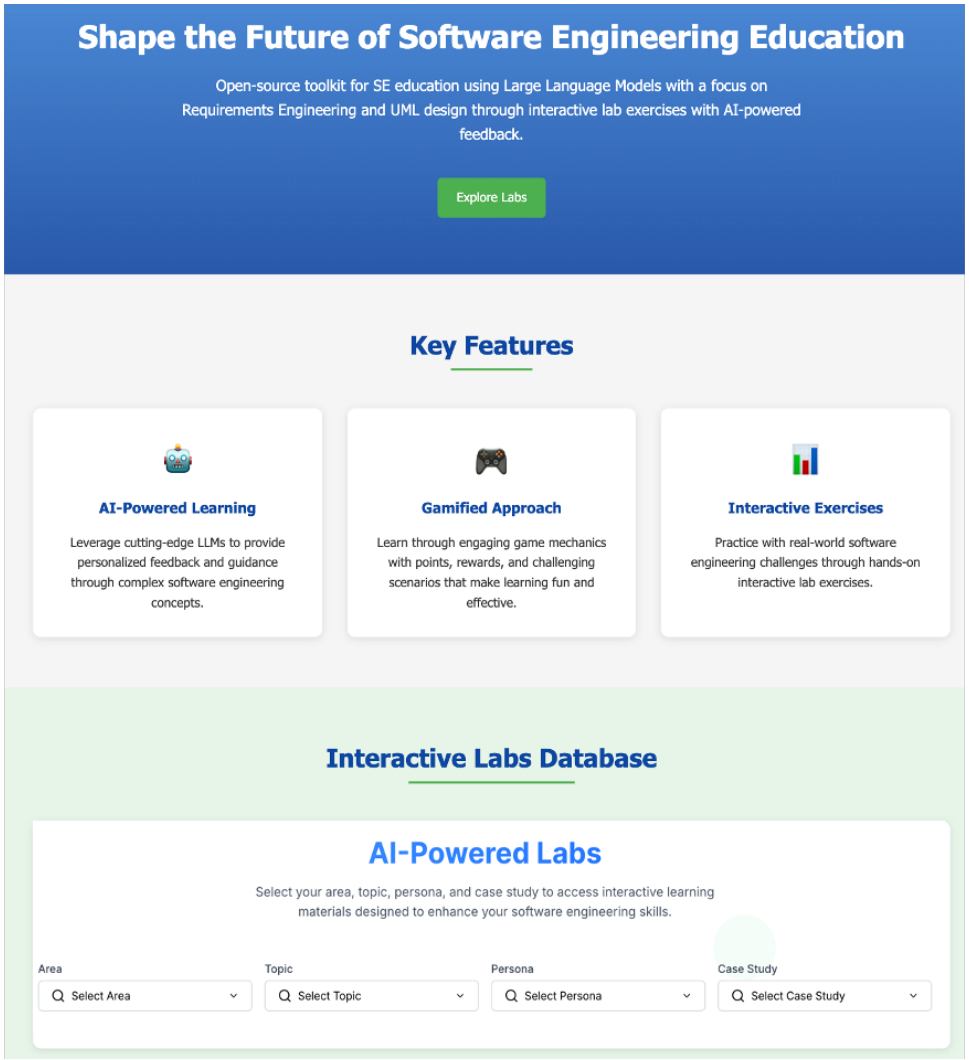
Project Overview </>

- Open-source toolkit for SE education using LLMs
- Focus: Requirements engineering and UML design
- LLM-guided interactive Lab exercises
- Structured dataset of prompt templates and case studies

Explores the Personas Prompt Engineering Pattern

- **Tutor**: Explains concepts and guides through exercises
- **Expert (Senior/Principal Engineer)**: Provides detailed technical feedback on designs
- **Peer (Cybernatic Teammate)**: Collaborative problem-solving and design reviews

LLM Toolkit for Software Engineering Education (2/2)



Access our website at: `https://se-education-toolkit.vercel.app/`

LLM Toolkit: Design and Methodology

Requirements Engineering

Learn how to identify and document software requirements using industry-standard techniques with personalized AI feedback.

Use Case Detective Game

Identify and fix issues in use case main flows while earning points and receiving tailored AI guidance on your solutions.

UML Diagram Mastery

Design and validate UML diagrams with the help of AI guidance and expert feedback to develop better modeling skills.

1 Experiment in Class

- ✓ Integration into Year 3 Software Engineering module
- ✓ Testing with LLM of choice
- ✓ Collecting student feedback and performance metrics
- ✓ Refining toolkit based on classroom experience
- ✓ We will report on findings today

2 Controlled Group Experiment

- ✓ ~100 Brunel CS students in four comparative groups
- ✓ Control group (no LLM) vs. different LLM platforms
- ✓ Measuring task completion time and output quality
- ✓ Evaluating student engagement and concept comprehension
- ✓ Comparing effectiveness of different LLMs for SE tasks

Next Steps

What's coming up...

- **Your Experience**: Try one of our interactive games
- **Our Experience**: Designing the toolkit
- **Student Feedback**: What we've learned so far

Part 3:

Practice

Gamifying Requirements Engineering: The User Story Wizard Game

Background: What Are Agile User Stories?

- **User stories** are short, informal descriptions of what the user wants.
- They are used in **Agile development** to capture software requirements.
- **Format:**

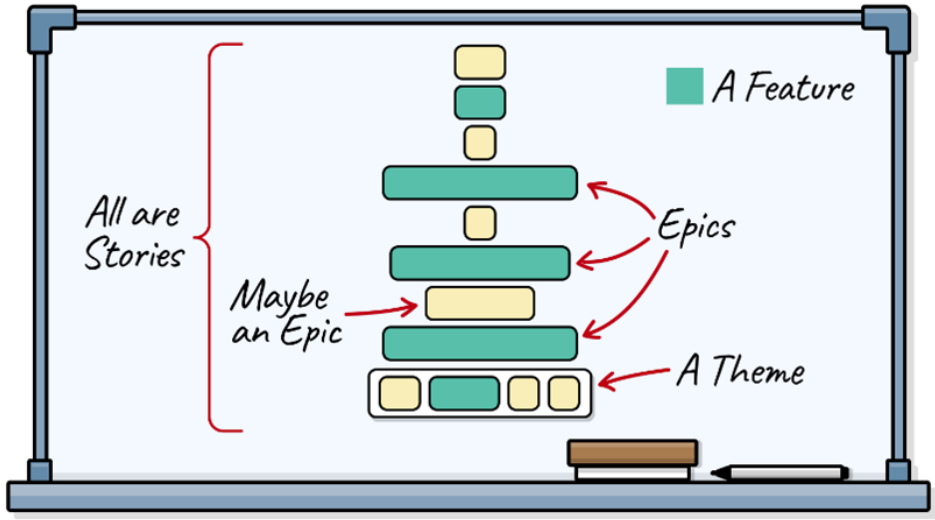
  As a [user role], I want to [goal], so that [reason/value].

Example

As a teacher, I want to give personalised feedback on each student's work, so that I can help students better.

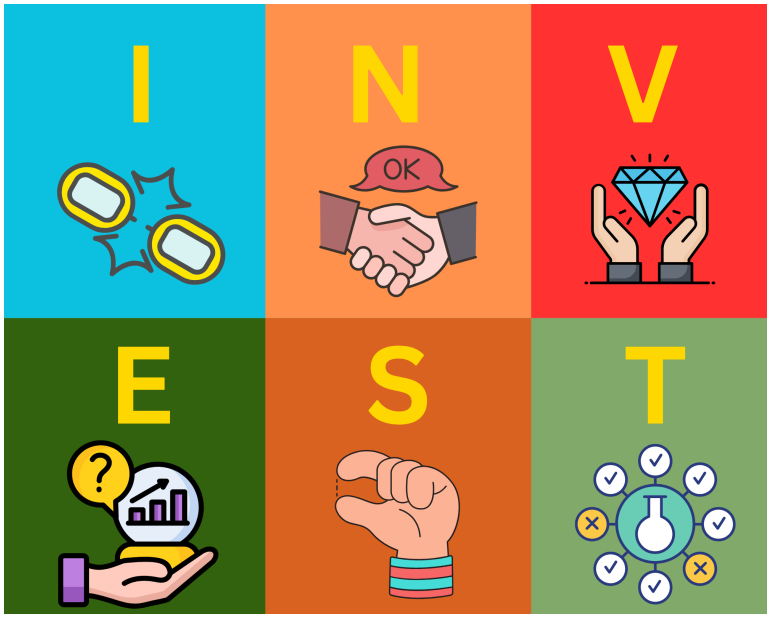
Background: Epics vs. User Stories

- **User Story:** Shorter, simpler, specific
- **Epic:** A larger, less-detailed user story
- **Theme:** A collection of epics and user stories linked by a "theme"
- **Feature:** A feature is a user story (or collection) that is big enough to be released



User Stories, epics, themes. The green ones can be released independently, i.e., a feature. Source: Mike Cohn

Background: What Makes a Good User Story?



A good user story should be:

- **I** **Independent** – can be developed separately
- **N** **Negotiable** – open to discussion and iteration
- **V** **Valuable** – delivers value to the user
- **E** **Estimable** – effort can be estimated
- **S** **Small** – fits within one sprint
- **T** **Testable** – can be validated with tests

Background: Acceptance Criteria

- Describe how to confirm the user story is complete.
- There is typically a list of criteria
- **Format:**
  - **Given** [initial context]
  - **When** [event occurs]
  - **Then** [expected outcome]

Example

- **Given** the teacher is viewing a student submission
- **When** they click on the feedback button
- **Then** a text modal opens to enter feedback

Why A Game on Agile User Stories?

Writing agile user stories is challenging

- Students struggle with writing requirements as user stories:
  - Do not use the right format: **As a [persona], I [want to], [so that]**.
  - Write vague or overly broad (epic) user stories
  - Forget to include acceptance criteria
  - Fail to test or validate requirements
- Traditional instruction doesn't always provide enough practice or feedback.

The LLM-driven game-based approach

- Provides **guided, scaffolded practice** to write high quality, testable user stories
- Helps identifying/fixing issues in an **interactive and fun** way
- Enables **learning concepts at own pace**

Introducing the User Story Wizard Game

Game Initialisation:

- Set the LLM context, the Tutor persona, RE background and the game rules
- After the set-up prompt, enter:
  - Problem Statement + Persona(s)
  - At least one user story with its acceptance criteria

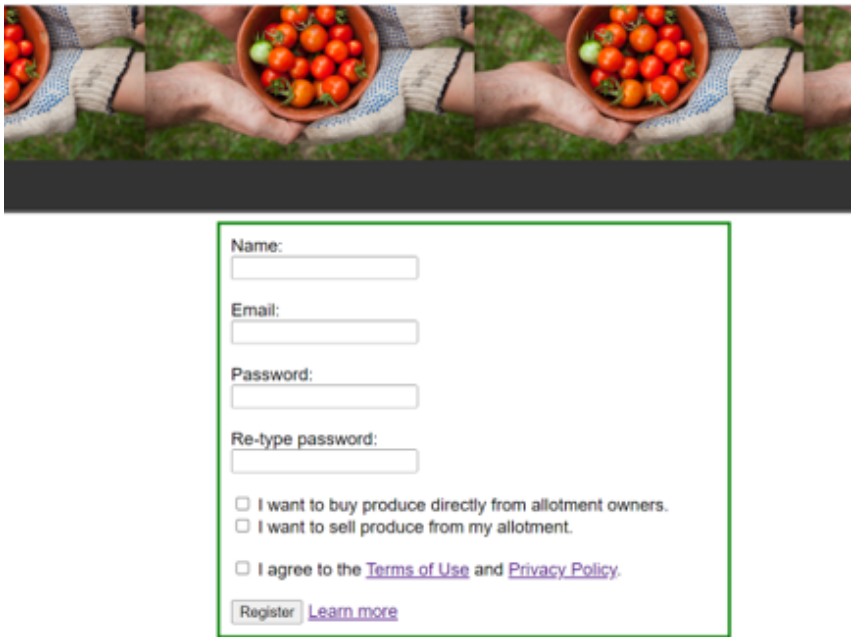
Feedback Loop:

- Tutor helps iterate toward complete, testable user stories

Game scoring at each iteration:

- +10: New missing user story
- +5: Finding/fixing issues, suggesting/fixing acceptance criteria
- -0 to -10: Optional hint system

Game Guidelines



Use-case in the game: Enable people in food poverty to access healthy food easily, and without putting their dignity at stake by locating fresh produce grown by allotment owners.

Follow these steps to play the game:

1. Open your favourite LLM tool (e.g., ChatGPT, Claude, etc.)
2. Type the link provided or scan the QR Code
3. Navigate to the lab sheet
4. Follow the instructions on the page and start playing!

Time to Play!

Visit:

`https://se-education-toolkit.vercel.app/`

To find the lab:

- Go to the **Labs** section
- Select:
  - **Area:** Requirements Engineering
  - **Topic:** User Stories and Acceptance Criteria
  - **Persona:** Tutor
  - **Case Study:** Food Sharing App

Feedback and Discussion

"How might this format be adapted to your teaching context?"

Opportunities and barriers for:

- Courses where students write requirements or design systems
- Peer feedback or formative assessment
- Integration into workshops, labs, or flipped classrooms

Part 4:

Our experience

Insights from Experiments Across Different Platforms

Testing the Game Across LLM Platforms

User Story Wizard Game was tested with four large language models:

- **ChatGPT-4o and GPT-4**
- **Claude 3.7 Sonnet**
- **Gemini Flash 2.0**

ChatGPT

4o

- Most playful and conversational
- Encouraged creativity and supported multiple iterations
- Helped revise acceptance criteria gently

GPT-4

- Accepted input with minimal critique - "Issues Found: None explicitly yet; potential for refinement."
- Less iterative — limited interactivity

GPT4o Steps	Issues	Score	Feedback
1 – Input	2	0	Divide US; expand AC
2 – Revised US	1	+5	Expand AC
3 – Add to AC	2	+5	Add missing AC
4 – Add to AC	1	+5	Refine AC
5 – Add to AC	0	+5	All covered!

Claude

- More pedantic: focused on precision
- Feedback on format and complexity
- Prompted revisions emphasizing value to user, e.g., dignity, privacy and anonymity

Claude Steps	Issues	Score	Feedback
1 – Input	0	0	Fix format/US, review AC
2 – Revised US	2	0	Same as above
3 – Revised US	3	0	Same as above
4 – Revised US	3	5	Same as above
5 – Add to AC	3	10	Same as above
6 – Add to AC	3	15	Same as above
7 – Add to AC	3	20	Same as above
8 – Reject format suggestion	3	20	revise AC
I stopped interacting!			

Gemini - 2.0 Flash

- Focuses on clarity and emphasises INVEST
- Subtler hints with more generic questions
- Does not present specific next steps

Gemini Steps	Issues	Score	Feedback
1 – Input	4	0	US Clarity Review AC
2 – Revise US	3	0	Revise AC
3 – Add to AC	2	5	Revise AC
4 – Add to AC	2	10	Revise AC
5 – Add to AC	2	15	Revise AC
6 - Revise AC	1	20	Revise AC
7 - Revise AC	0	25	Congrats!

Final User Stories

Original: As an allotment user, I want to create an app account so that I can save my preferences.

ChatGPT: Guides towards a smaller user story: registering and getting started

As an allotment user, I want to create an app account by registering my email and password **so that I can start using the app**.

Claude: Adds ethical/social context

As an Allotment User, I want to create an app account **so that I can start using the app to get access to fresh food without having to register to a food bank**.

Gemini: Prioritises UE and data-driven onboarding

As an Allotment User, I want to create an app account **so that I can save my email, password, location, and fresh food preferences**.

Acceptance Criteria

Acceptance Criteria	Claude	ChatGPT-4o	Gemini
Submit form	✓	✓+(valid email, registration success)	✓+(data saved, success shown)
Empty fields	✓	✓+(prevent submission)	✓
Password validity check	✓+(criteria specified)	✓	✓+(criteria specified)
Invalid email error	✓	✓	✓
Email already registered	✓	✓	✓
Server error		✓	

Comparison Across LLMs

Each model responds differently to the same prompt, offering distinct tone, focus, and feedback style.

Model	Tone/Style	Focus	Final Solution
GPT-4	Reserved/Concise	Basics	Underdeveloped
GPT-4o	Playful/Conversational	Gentle guidance	Fit-for-purpose
Claude	Pedantic/Formal	Clarity & Ethics	Comprehensive
Gemini	Measured/Subtle	INVEST	Good user experience

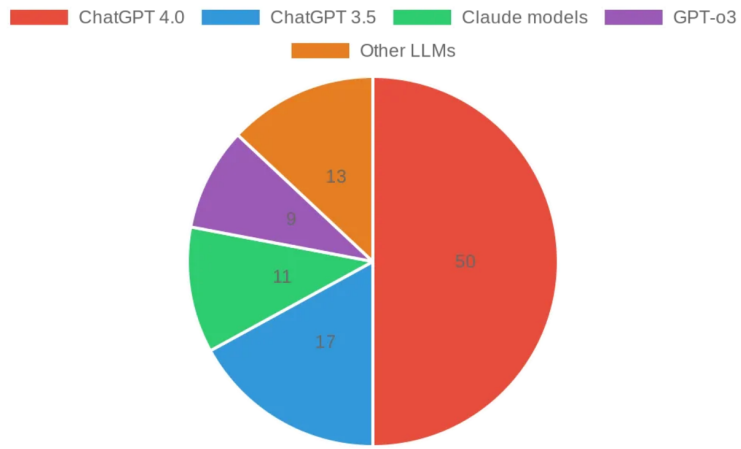
Part 5:

Students' experience

Requirements engineering Detective Game

Exercise Overview

- **Participants:** 46 students
- **Format:** Gamified prompt
- **Task:** Generate requirements for a reset password feature
- **Time limit:** ∼20 minutes (average)
- **Approach:** Students used LLM of their choice



LLM Distribution

Exercise Process

1. Students selected an LLM of their choice
2. They received a gamified prompt for creating requirements for user stories
3. The task focused on password reset functionality
4. Points were awarded for identifying issues and solutions

Key Insights

Learning Experience (out of 5)

Improved understanding	3.77/5
Enjoyed the exercise	3.85/5

Lab Design (out of 5)

Motivated to improve	3.98/5
Clear AI guidance	3.81/5
Helpful points system	3.89/5

Key Findings: Lab Format

- **77% of students** want more labs in this format:
  - 24% would prefer all labs in this format
  - 54% would prefer a mix with traditional labs
- **Only 13%** preferred traditional labs exclusively

What Students Liked 👍

- **Interactivity & Engagement** (10 mentions)
  - "Significantly more interactive, paired with an interesting topic makes it fun to do"
- **LLM/AI Integration** (9 mentions)
  - Appreciated learning to use AI tools effectively
- **Learning Experience** (7 mentions)
  - Practical approach helped solidify concepts
- **Guidance & Support** (4 mentions)
  - "It's quick and easy to get help"

Key words students used:

- **Easy** (7 mentions)
- **Fun** (5 mentions)
- **Interactive**
- **Quick**
- **Points system**

What Could be Improved

- **Clarity of Instructions** (5 mentions)
  - "Slightly more description on the exercise sheet"
- **Guidance Refinement** (3 mentions)
  - "Sometimes it says you found one issue if you found multiple"
- **Feedback Mechanisms** (2 mentions)
  - "Better error messages when something is wrong. It was hard to understand what needed to be corrected"

Lessons Learned:

- Detailed task instructions
- LLM training for more efficient prompting
- Multi-modal variations

Student Feedback

What Students Liked

"Significantly more interactive, paired with an interesting topic makes it fun to do"

"It was fun and engaging"

"It's quick and easy to get help"

"I like the step-by-step guidance and immediate feedback"

Positive Themes: Interactivity (10), LLM Integration (9), Learning Experience (7), Support (4)

What Could Be Improved

"More gamified UI/Maybe more elements other than it being just text."

"Sometimes it says you found one issue if you found multiple"

"Better error messages when something is wrong"

Improvement Areas: Clarity of Instructions (5), Complexity (3), Feedback (2)

Thank You! Let's Keep the Conversation Going

We would love your feedback!

Please scan the QR code to complete our short survey.

Interested in collaborating or adapting our toolkit for your course?

Reach out to us — we are happy to connect!

Cigdem Sengul cigdem.sengul@brunel.ac.uk

Rumyana Neykova rumyana.neykova@brunel.ac.uk