

Article

Not peer-reviewed version

---

# Hybrid Intelligence in Requirements Education: Preserving Student Agency in Refining User Stories with Generative AI

---

[Leon Sterling](#)\* and [Eduardo Oliveira](#)

Posted Date: 19 December 2025

doi: 10.20944/preprints202512.1658.v1

Keywords: generative artificial intelligence; LLM; user stories; motivational modelling; software engineering



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# Hybrid Intelligence in Requirements Education: Preserving Student Agency in Refining User Stories with Generative AI

Leon Sterling \*  and Eduardo Oliveira 

School of Computing and Information Systems, University of Melbourne

\* Correspondence: leonss@unimelb.edu.au

## Abstract

We discuss the use of Generative Artificial Intelligence (Gen AI) to support the teaching of Requirements Engineering. While Gen AI systems cannot independently specify requirements, they can play a useful helping role. We introduce motivational modelling which we have used extensively for teaching. We describe a novel method for using Gen AI to assist in the development of compelling user stories derived from the modelling activities. Gen AI helps in three ways: suggesting new possible requirements, encouraging careful consideration of how requirements are expressed by offering alternative language, and improving the readability of requirements.

**Keywords:** generative artificial intelligence; LLM; user stories; motivational modelling; software requirements engineering

## 1. Introduction

Software Requirements Engineering (RE) is a critical component of the software development lifecycle, laying the groundwork for delivering products that meet the needs and expectations of stakeholders [1]. The RE phase traditionally involves eliciting, analysing, specifying, and validating requirements. It is labour-intensive and prone to errors due to its reliance on human judgement, communication, and interpretation [2]. Persistent challenges in RE may jeopardise project outcomes, particularly in complex or agile environments [1,3,4]. Recent literature continues to identify requirements-related issues as a major source of project failure, rework, and cost overruns [5,6]. These challenges are often intensified when stakeholders differ significantly in technical expertise, domain knowledge, or availability, creating communication gaps that impede shared understanding [7–10].

User stories have emerged as a lightweight method to support shared understanding and reduce the need for frequent updates to requirements documents [11]. However, despite their value, user stories are not always well integrated with broader RE processes. Prior research has highlighted the lack of methodological support for ensuring consistency, traceability, and validation between user stories and other RE artefacts [12–14]. This gap is particularly salient in educational and industry-based learning settings, where novice developers must coordinate requirements with clients under time constraints. Teaching effective RE practices is therefore challenging, especially when students must balance conceptual understanding, communication with stakeholders, and the production of coherent artefacts such as user stories.

To support requirement elicitation and complement user story development, The University of Melbourne has continuously developed and maintained Motivational Modelling (MM) since 2017. This initiative centers on a dedicated software tool that has been refined across five different subjects within the Undergraduate, Master of Information Technology and Master of Software Engineering programs. To date, over 2400 students have utilised the various versions of our tool, providing a rich testbed for educational RE practices.

MM provides a high-level view of the requirements by representing stakeholder roles, system goals, and underlying rationales in forms accessible to both technical and non-technical participants [15]. Originally developed as an extension of agent-oriented RE, MM emphasises the motivations that underlie system behaviours rather than focusing solely on functional requirements. Motivational models can facilitate early validation of requirements, uncover implicit assumptions, and improve communication between clients and development teams [16]. These models also provide a structured foundation for subsequent design and planning.

Despite these benefits, translating motivational models into well-structured user stories remains a time-consuming and interpretive process. Student-generated user stories often vary in quality, lack consistency, or sound awkward, especially when multiple teams engage with the same client. These issues limit the scalability of MM-based practices and challenge their integration into agile workflows. Although motivational models can be translated into user stories [17], doing so reliably continues to present difficulties.

Recent advances in large language models (LLMs) offer opportunities to enhance this translation process. LLMs such as ChatGPT and Claude demonstrate strong capabilities in understanding and generating natural language text [18]. In this paper, we use the term Generative Artificial Intelligence (Gen AI) to refer to tools powered by such models. A rapidly growing body of work has begun exploring the role of Gen AI in RE, including elicitation, analysis, specification, and validation [5,6,19]. These studies identify potential benefits such as increased efficiency, improved natural language processing, and greater support for iterative refinement. However, challenges related to trust, hallucinations, reproducibility, and interpretability continue to limit adoption in both educational and industrial contexts [5,10]. These concerns are particularly significant when LLMs generate or reformulate requirements, where inaccuracies or biased phrasing may affect the reliability of the resulting artefacts. Ethical considerations relating to privacy, transparency, and responsible use add an additional layer of complexity.

This paper investigates the role of Gen AI in assisting with the translation of motivational models into user stories, addressing longstanding challenges in RE through a structured and scalable approach. We adopt a semi-automatic, human-in-the-loop process so that Gen AI supports, rather than replaces, the reasoning and judgement of students and software developers. Earlier attempts at a fully automated pipeline resulted in inconsistent or incorrect outputs, reinforcing the need for guided human oversight. Building on these insights, we propose a structured workflow for LLM-assisted translation. Our findings indicate that the workflow improves efficiency and enhances interpretive accuracy compared with manual practices.

The contribution of this work describes a brand-new initiative within this established project. We present a practical workflow that integrates Gen AI into the Requirements Engineering tool, utilising a design that strictly preserves student agency and guards against metacognitive offloading. While the tool has successfully supported manual modelling for years, this new design addresses the specific cognitive challenges of the Gen AI era. Rather than outsourcing the cognitive work involved in analysing and translating requirements, our semi-automatic approach is designed to support reflection, interpretation, and verification, which are essential in educational contexts. This aligns with emerging perspectives on hybrid human–AI intelligence, which emphasise AI as an extension of human cognition rather than a replacement for it [20–22]. It also resonates with recent empirical findings showing that uncritical reliance on Gen AI can lead to metacognitive “laziness”, reduced self-regulation, and diminished learning processes if students are not actively engaged [23]. The workflow presented in this paper balances efficiency with pedagogical integrity by using Gen AI to enhance, rather than diminish, the learning process. Here, we investigate the following research question: *How can large language models be effectively integrated into requirements engineering teaching to improve the translation of motivational models into user stories while strengthening scalability, consistency, reliability, and student agency?*

The structure of the paper is as follows. After this introduction, we give our literature review and then outline the challenges associated with relying on Gen AI to directly elicit requirements. Section 4 describes our proposed workflow. We then introduce motivational modelling, an evolving RE process used for over 15 years in both research and teaching contexts. We describe a promising novel application of Gen AI, following an explanation of the context of translating motivational models into user stories. Finally, we present conclusions and directions for future work.

## 2. Background and Literature Review

### 2.1. Requirements Engineering, User Stories, and Educational Challenges

Decades of empirical research have shown that many software project failures can be traced back to problems in Requirements Engineering, including ambiguous statements, missing or conflicting requirements, and weak traceability between artefacts [1]. These issues are particularly acute in contemporary development contexts that involve heterogeneous stakeholders, complex system behaviour, and rapid delivery cycles. Recent systematic work confirms that requirements related defects remain a major driver of rework and cost, despite advances in methods and tooling [5,6]. Habiba et al. [10], for example, report that RE for AI based systems is still characterised by immature practices, insufficient attention to non-functional concerns, and limited methodological guidance for aligning stakeholder expectations with the behaviour of data driven components.

User stories have become a central artefact in agile RE, intended to provide a lightweight and human readable representation of stakeholder needs. Their narrative structure is designed to facilitate communication across technical and non-technical participants and to support incremental refinement of scope [11]. However, several studies have criticised the informality of user stories and the difficulty of maintaining consistency and traceability when large numbers of stories are created [12–14]. In practice, user stories are often written at inconsistent levels of abstraction, omit quality concerns, or drift away from higher level project goals. This makes it difficult to reason systematically about coverage, dependencies, and validation.

These challenges are amplified in educational contexts. Students who are new to RE frequently struggle to elicit and articulate stakeholder goals, to distinguish between functional and quality requirements, and to maintain coherence across different artefacts. Mahnic and Hovelja [24] report that specific training in user story writing can improve students' ability to connect stakeholder needs to implementable features, but also note that many students still produce vague or incomplete stories. Péraire and Lopez [25] similarly emphasise the need for structured teaching approaches that help students move between high level visions and detailed user stories while maintaining traceability and justification.

In summary, the literature highlights three interrelated problems. First, RE remains a high risk phase in practice, with persistent issues of ambiguity, misalignment, and weak validation [1,10]. Second, while user stories help communication, they can introduce fragmentation and inconsistency if not grounded in a broader modelling framework [12,14]. Third, teaching RE to novice developers is challenging because students must simultaneously learn conceptual foundations, engage in authentic communication with clients, and produce coherent artefacts under time pressure [24]. These limitations motivate the use of modelling approaches that make stakeholder motivations explicit and that can provide a stable reference point for subsequent artefacts such as user stories.

### 2.2. Motivational Modelling in RE and Education

Motivational Modelling (MM) builds on a lineage of goal oriented and agent oriented approaches, including Gaia [26], ROADMAP [27], and Tropos [28]. These methods represent systems in terms of actors, goals, and dependencies, and seek to capture why a system is needed rather than only what it should do. However, diagrammatic complexity and steep learning curves have limited their adoption in both industry practice and teaching. MM addresses this by providing a simpler visual vocabulary

that foregrounds stakeholder roles (who), functional goals (do), quality goals (be), and emotional goals (feel) in a single hierarchical diagram [15].

In MM, requirements elicitation typically begins with a do/be/feel brainstorming process involving technical and non technical stakeholders. Participants collaboratively identify stakeholders (who list), functional expectations (do list), qualities (be list), and desired emotional outcomes (feel list). These lists are then transformed, through a structured sequence of steps, into a motivational model where functional goals are organised hierarchically, and qualities and emotions are attached to appropriate nodes. The resulting model is intended to fit on a single page, making it easier to discuss with clients and to identify gaps or inconsistencies [15]. The visual simplicity of the notation and the explicit separation of functional, quality, and emotional concerns help mitigate some of the comprehension issues observed in more complex goal modelling notations.

MM has been adopted as a central RE technique in several software engineering and IT project subjects. Students work with external clients to develop motivational models during an inception phase, validate them with stakeholders, and then use them as a hub for subsequent artefacts such as personas and user stories [16]. To support internal coherence across artefacts in this process, nine Consistency Principles have been devised to systematically verify alignment between motivational models, personas, and user stories [29]. These principles require, for instance, that every role in the motivational model has at least one corresponding persona, that leaf nodes and specific quality and emotional goals are supported by user stories, and that any quality or emotional goal that appears in a user story also appears in the model. When applied in teaching, the Consistency Principles provide students with concrete checks for completeness and traceability, and they reduce the risk that user stories drift away from the original stakeholder motivations.

Despite these advantages, the translation from motivational models to user stories remains largely manual. Students must interpret abstract, high level goals and decide how they should be operationalised as user stories. This interpretive step is cognitively demanding and can lead to variability in style, level of detail, and fidelity to the underlying model. In large cohorts, the process is also time consuming for both students and instructors, who must provide feedback on many sets of user stories. These limitations restrict the scalability of MM based teaching and create an opportunity for technology enhanced support that can preserve the pedagogical benefits of MM while reducing repetitive manual work.

### *2.3. Generative Artificial Intelligence in Requirements Engineering and Human–AI Collaboration*

Recent work on Generative Artificial Intelligence and large language models has opened new possibilities for automating and augmenting different stages of the RE lifecycle. Marques et al. [6] present a comprehensive review of the use of ChatGPT across RE activities and categorise existing studies into four broad application areas: requirements elicitation, analysis, documentation, and refinement. Their survey highlights that ChatGPT is frequently used to generate candidate requirements from unstructured descriptions, rewrite existing requirements to improve clarity, and support exploratory analysis by identifying potential inconsistencies or missing information. They also note that most studies focus on proof-of-concept experiments rather than large-scale deployments, and that there is a lack of standardised evaluation frameworks for assessing the quality of Gen AI-generated requirements.

Cheng et al. [5] complement this work through a systematic literature review of Gen AI and RE more broadly. Their findings show that LLMs have been used in tasks such as generating requirements from user scenarios, classifying and clustering large requirement sets, extracting requirement-relevant information from documentation, and assisting with traceability and defect detection. Importantly, their review identifies significant variation in prompting strategies, model selection, and evaluation metrics, which complicates direct comparison across studies. Cheng et al. also report that while LLMs can accelerate the handling of textual artefacts, their outputs often require human verification due to inconsistencies and hallucinated content.

Arora et al. [19] provide a practitioner-oriented perspective by examining LLMs as RE assistants. They argue that LLMs can support analysts in repetitive or routine natural language processing tasks, such as paraphrasing requirements, proposing alternative formulations, or flagging unclear statements. Their evaluation emphasises the potential of LLMs to facilitate sense-making and early ideation, but they also caution that LLM outputs may obscure underlying reasoning processes, thereby complicating justification and traceability.

Across these studies, several common themes emerge. First, Gen AI shows strong potential for reducing manual effort and accelerating the processing of large volumes of textual artefacts [5,6]. Second, LLMs can assist analysts by surfacing ambiguities, identifying missing detail, or proposing clarifying questions, which can support early validation. Third, however, there are persistent concerns regarding hallucinations, variability in outputs across prompts, and opaque chains of inference [5]. These issues undermine trust and accountability, particularly in safety-critical or regulated domains. Furthermore, Habiba et al. [10] highlight that RE practices for AI-based systems are already immature, with limited methodological rigour; introducing Gen AI tools without robust frameworks or safeguards risks amplifying existing weaknesses rather than mitigating them.

Taken together, this body of work suggests that Gen AI can enhance—but not replace—the work of human requirements analysts. The literature strongly indicates a need for human-in-the-loop designs that preserve stakeholder agency, maintain verification processes, and ensure that generated artefacts remain grounded in well-defined modelling practices.

### 3. Potential and Pitfalls of Gen AI in Requirements Engineering

Systematic reviews have identified a growing number of use cases for large language models across elicitation, analysis, and documentation, but have also pointed to issues of misalignment, and weak methodological grounding [5,6,10,19]. From an educational perspective, research on hybrid intelligence and metacognitive laziness further underlines the need to embed Gen AI in ways that extend rather than replace learners' cognitive work [20–23]. In this section, we illustrate how these concerns manifest in practice when Gen AI is used naively for elicitation, and how this motivates the design choices behind our human-in-the-loop workflow.

Gen AI systems such as ChatGPT, Claude, and SparkAI are trained on very large corpora that include software requirements, design documents, and domain descriptions. As a result, they readily respond to prompts such as “Give me the requirements for System X”, often producing long, well structured lists that appear plausible and comprehensive. For novice analysts and students, this can create the impression that Gen AI is a suitable substitute for early elicitation, particularly when direct engagement with stakeholders feels challenging or time consuming.

A teaching example from a final year IT project subject illustrates the limitations of this approach. In class, SparkAI (a secure environment at The University of Melbourne, powered by Claude, where staff can experiment with Gen AI using private or confidential information) was prompted with “Can you give me software requirements for an app to manage recipes online?”. The system immediately returned a detailed set of potential requirements, grouped into categories such as user management, recipe management, user interaction, meal planning, nutritional information, data storage, accessibility, localisation, security, and monetisation. The response concluded by describing the list as a comprehensive foundation for development and suggested that prioritisation might be required depending on the target audience.

Viewed in isolation, this output exemplifies many of the strengths reported in the Gen AI for RE literature: fluent text, coverage of multiple functional areas, and a structured breakdown of features [6]. However, when situated in the context of an actual client, the limitations become clear. The example was closely aligned with a real student project previously undertaken at the University of Melbourne. In that project, the client was primarily interested in organising personal recipes collected from websites and emails and sharing them with family members. Most of the requirements proposed by SparkAI were either irrelevant or premature for this scope. In particular, nutritional analysis, complex meal

planning, and monetisation were not part of the client's goals. By contrast, a requirement that was central to the client—the ability to extract recipes from photos—was not mentioned at all.

Gen AI produces requirements that are generic rather than grounded in the specific context, intentions, and constraints of actual stakeholders. It has no access to the latent goals and trade offs that shape real projects, and it can only extrapolate from patterns in its training data. As a result, it may foreground features that are common in similar systems while omitting needs that are distinctive or locally important. For experienced analysts, these outputs might be treated as a checklist or a source of ideas. For students, however, the polished presentation risks being mistaken for an authoritative specification.

The educational implications are significant. As discussed in the Background section, many students already struggle with engaging stakeholders and with articulating requirements that faithfully reflect user needs. There is a documented tendency for novices to invent or assume requirements rather than to negotiate them, especially when they lack confidence in communication or when stakeholder access is limited. In an earlier honours examination context at the University of Mauritius, the first author observed that students often produced complete sets of requirements without ever consulting prospective users or system owners. The resulting systems were largely misaligned with genuine needs and had little practical value.

Introducing Gen AI without careful framing can amplify this tendency. If students learn that a single prompt can produce an apparently complete list of requirements, they may feel less need to conduct interviews, workshops, or modelling sessions. This aligns with Fan et al.'s findings on metacognitive laziness, where reliance on Gen AI can lead learners to offload planning, monitoring, and evaluation processes [23]. Rather than using Gen AI outputs as material for critical analysis, students may accept them at face value, bypassing the very practices that RE education seeks to cultivate. In the language of hybrid intelligence, this would represent a shift from AI supporting human cognition to AI substituting for it [20].

These observations do not imply that Gen AI has no place in RE. Rather, they suggest that its role should be carefully delimited and embedded within robust modelling and validation practices. The Background section described how Motivational Modelling provides a structured, stakeholder facing representation of roles, functional goals, qualities, and emotional outcomes, and how Consistency Principles help maintain alignment between models, personas, and user stories. If Gen AI is introduced only after such artefacts have been developed through human-centred elicitation, its function can shift from generating requirements from scratch to supporting refinement, translation, and consistency checking.

In this paper we adopt precisely this stance. Gen AI is deliberately excluded from the initial elicitation of goals and requirements. Instead, it is employed only once motivational models have been constructed and validated with stakeholders. Within that boundary, Gen AI is used to assist students in translating elements of the motivational model into candidate user stories and in polishing the language of those stories. Students remain responsible for judging whether the generated stories accurately reflect stakeholder motivations, comply with consistency principles, and fit within the broader project context. This design is intended to preserve student agency, promote critical engagement with Gen AI outputs, and support the development of RE competences rather than replacing them.

Having outlined these practical and pedagogical pitfalls and the rationale for constraining the role of Gen AI, the next section presents our LLM-assisted workflow for translating motivational models into user stories and describes the research design used to evaluate its effectiveness in a teaching context.

## 4. Materials and Methods

### 4.1. Research Design

This study adopts a case-based methodological approach to investigate how large language models can be integrated into the translation of motivational models into user stories. Rather than

collecting participant data, we analyse a set of instructional artefacts—motivational models and manually derived user stories—originally created for teaching software requirements engineering. These artefacts are routinely used in our classroom practice to demonstrate motivational modelling, user story construction, and persona-based reasoning. In the present study, we extend these same examples to illustrate and evaluate a semi-automatic, LLM-assisted workflow. Although no student data, personal information, or behavioural measures were collected for this study, some students had previously contributed suggestions and refinements to the motivational models as part of our teaching and research activities, which were conducted under ethics approval #24272.

Our research method involves (1) establishing a baseline manual translation process, (2) introducing a structured semi-automated workflow incorporating Gen AI, (3) applying both methods to the same set of motivational models, and (4) conducting an interpretive comparison of outputs to assess alignment, additions, omissions, and shifts in intent. This method is consistent with design science and technical case study methodologies commonly employed in software engineering research to examine process innovations and methodological feasibility.

#### 4.2. Artefacts and Baseline Process

A motivational model was selected as the basis for this case study. This model was originally developed for classroom use and has been refined over several years of teaching motivational modelling across software engineering and IT project subjects. Over this period, students have contributed to the evolution of this example through coursework, capstone projects, and internal research activities centred on improving motivational modelling practices and tooling. As a result, the model is representative of the kinds of systems students work with in industry-partnered projects, containing multiple functional, emotional, and quality goals. Its structure makes it suitable for demonstrating both the strengths and limitations of the translation workflow.

To establish a baseline, we apply the previously published algorithmic procedure for translating motivational models into user stories [17]. The process involves identifying subtrees, creating corresponding epics, and generating user stories for each branch based on associated roles, functional goals, and quality or emotional constraints. Although systematic, this manual translation process requires careful reading of the model and is sensitive to individual interpretation. Prior work has shown that user stories generated manually from motivational models often lack consistency or completeness and may be phrased awkwardly. These limitations provide a strong motivation for investigating how LLMs might support refinement while preserving alignment with stakeholder intent.

#### 4.3. LLM-Assisted Workflow

The proposed workflow extends the manual baseline by incorporating LLM-based refinement as a semi-automated step. The workflow consists of six stages:

1. **Motivational Model Creation:** Motivational models are created using the AMMBER tool [30], which provides a visual environment for representing stakeholder roles, functional goals, quality attributes, and emotional goals.
2. **Manual Extraction:** Key elements of the model (leaf functional goals, associated quality and emotional goals, and stakeholder roles) are manually extracted. This ensures that any LLM input is grounded in an accurate, human-curated representation of stakeholder motivations rather than inferences made by the model.
3. **Prompt Formulation:** Extracted elements are converted into structured prompts. Prompts specify the epic, role, and associated goals and request improved phrasing or clearer user story articulation. Prompt design follows recommended best practices for minimising hallucination and guiding LLM attention toward narrative consistency.
4. **LLM Processing:** Prompts are submitted to SparkAI, the University of Melbourne's LLM interface powered by GPT-4o. All model runs were executed with a temperature of 1.0, allowing the LLM

to propose alternative formulations while retaining general coherence. The model generates refined user stories that expand, rephrase, or reorganise the original content.

5. **Human Review and Error Correction:** Generated user stories are analysed for alignment with the original motivational model. Additions, deletions, and modifications are examined to determine whether they reflect stakeholder intent or introduce discrepancies. This step maintains a human-in-the-loop process, consistent with pedagogical goals of preserving student agency and avoiding overreliance on automation.
6. **Finalisation:** A final set of user stories is produced, incorporating beneficial refinements while discarding incorrect or speculative additions. This final set represents the output of the LLM-assisted workflow.

This workflow is designed to enhance traceability, consistency, and clarity while ensuring that human judgement remains central to the requirements process.

#### 4.4. Analysis Procedure

Our analysis focuses on comparing differences between (1) manually generated user stories (baseline) and (2) LLM-refined user stories (workflow output). To enhance the trustworthiness of the analysis, two researchers independently examined the outputs produced by SparkAI. Each researcher generated annotations and reflections on structural changes, intent drift, additions, omissions, and alignment with the motivational model. Following this independent review, a series of consensus meetings were held to discuss discrepancies, reconcile interpretations, and validate the final analytical judgements. This process provided analyst triangulation, allowing us to identify systematic patterns in the LLM's behaviour while minimising individual bias. Our analysis was qualitative and interpretive in nature.

The comparison examines:

- **Structural changes:** whether the LLM reorganised or restructured stories into clearer hierarchies.
- **Lexical and semantic modifications:** including changes in verb choice, tone, or granularity.
- **Intent drift:** instances where the LLM adds purpose statements ("so that" clauses) not present in the motivational model.
- **Additions and expansions:** such as new constraints, quality attributes, or assumptions inserted by the LLM.
- **Omissions or simplifications:** cases where model elements were ignored or condensed.
- **Alignment with functional, quality, and emotional goals:** assessing whether LLM output remains faithful to the structure and logic of the original model.

This interpretive analysis allows us to characterise the types of refinements LLMs can beneficially contribute, the risks they introduce, and the degree to which the workflow supports consistency and correctness.

## 5. Motivational Modelling in Practice: Teaching Context and Example Artefacts

To demonstrate how large language models can support the transition from motivational models to user stories, this study draws on a set of instructional artefacts routinely used in our software engineering teaching. These artefacts are created using AMMBER [30], a tool that represents a longitudinal educational project in itself. Since 2017, AMMBER has been iteratively developed and maintained by successive cohorts of students, allowing us to rapidly adapt the platform to new technologies. In this latest iteration, we utilise this flexible infrastructure to investigate a new workflow design involving Gen AI.

The artefacts originate from an established pedagogical approach that introduces motivational modelling through a two-stage process. The first stage is a controlled brainstorming activity using the do/be/feel method [31], in which students and clients collaboratively identify stakeholders (who), functional goals (do), quality goals (be), and emotional goals (feel). The second stage involves converting the four lists into a single motivational model that captures the project on one page. This

model is then revised iteratively as client understanding deepens. In our teaching practice, these models are validated with clients and assessed by project mentors, who provide feedback on clarity and alignment with stakeholder intent.

### 5.1. Running Example

To contextualise the workflow, we adopt a simplified motivational model used frequently in class.

The scenario involves an undergraduate project team building a simple real estate web application for a client. The account here is a simplification and a more complete version can be found in [32]. During the do/be/feel exercise, the team produced the following lists:

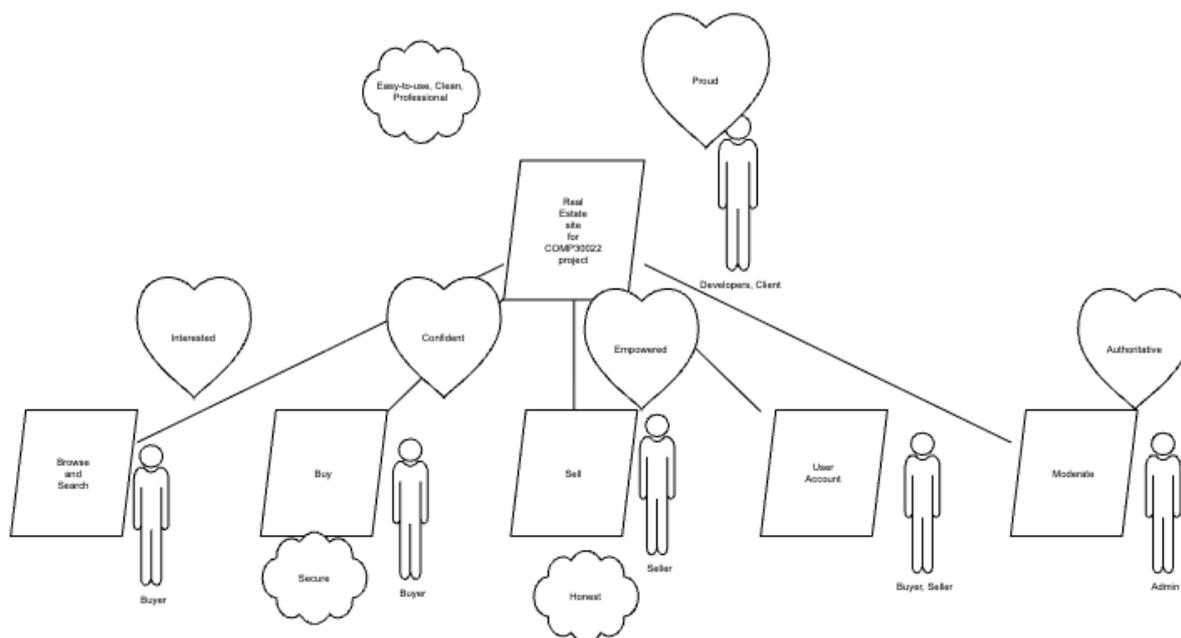
- Who list: Buyer, Seller, Developers, Client, Admin
- Do list: Real Estate for COMP30022 project, Browse and Search, Buy, Sell, User Account, Moderate
- Be list: Easy-to-use, Clean, Professional, Secure, Honest
- Feel list: Proud, Interested, Confident, Empowered, Authoritative

These terms are intentionally broad. As noted by [33], ambiguity in desired qualities or emotions can be productive because it prompts clarification and negotiation with clients. After completing the brainstorming, students apply a seven step procedure taught in lectures to construct the motivational model. The steps involve clustering functional goals, building the functional hierarchy, and annotating it with stakeholder roles, quality attributes, and emotional goals. The first and final stages of the process require validation with the client to ensure that the diagram accurately captures their expectations.

Figure 1 summarises the notation used in motivational models. A simplified version of a motivational model for a real estate app developed by students is shown in Figure 2. This model illustrates typical structural elements, including functional decomposition, qualities attached to specific goals, and emotional goals linked to stakeholder experiences. The larger model has multiple levels of functional goals, rather than the two levels in the figure.



Figure 1. Symbols used in motivational models



**Figure 2.** A simple motivational model developed by students

In this example, five functional goals appear under the central application node: View Reviews, Browse and Search, Buy, Sell, User Account, and Moderate. Stakeholder roles are placed in close proximity to the functional goals with which they are associated. Qualities and emotional goals appear as clouds and hearts respectively. Although the model could be improved, it provides a realistic illustration of the artefacts students produce. The model also serves as the central hub for downstream requirements engineering. Students use it to subsequently derive user stories and ensure traceability between high level motivations and detailed requirements.

This teaching context is important for the present study because the same artefacts are used to evaluate our semi automated workflow. The real estate app model and additional models used in class allow us to explore how an LLM refines manually produced user stories, where intent is preserved, and where new interpretations are introduced. The AMMBER tool [30] is used to construct and revise these models, ensuring consistency with contemporary practice in motivational modelling.

## 6. Transforming Motivational Models into User Stories

This section presents the application of the baseline algorithm and the LLM assisted workflow to the example motivational models described earlier. It serves as both an illustration of the translation process and an examination of the types of refinements, additions, and discrepancies introduced by the LLM. While prior work established an algorithmic mapping from motivational models to epics and user stories [17], our focus here is on how the integration of GenAI modifies the resulting artefacts and what this reveals about the feasibility and risks of semi automated requirements refinement.

### 6.1. Baseline Algorithm

We begin with the established procedure for translating a motivational model into user stories. The algorithm, originally presented in [17] and refined in subsequent internal work, proceeds as follows:

- Create an epic for each subtree of the motivational model.
- Name each epic using the label of its corresponding subtree.

- For each branch, create a user story in which the role is taken from the stakeholder associated with the subtree and the action is the functional goal, qualified by associated quality or emotional goals where relevant.

This algorithm is systematic, yet sensitive to interpretation. Differences in phrasing, granularity, and narrative coherence frequently arise when students execute the method manually. These limitations motivate the integration of an LLM based polishing step.

### 6.2. Example 1: Real Estate Platform

We first demonstrate the simplified motivational model for an electronic marketplace supporting real estate transactions. The simplified version of the model is given in Figure 2. A version with more functional goals is given in [32]. We perform an analysis of the result of polishing part of the model below. The following prompt was submitted to SparkAI:

Can you improve the expression of the following user stories:

Epic: Buy, sell, search real estate properties, proudly and in an easy-to-use, clean, professional manner.

2. As a buyer, I want to be able to browse and search properties, and remain interested through the process.

As a buyer, I want to also be able to:

- (a) Look up items using keywords
- (b) Browse listings
- (c) Look up a seller account
- (d) Contact a seller

3. As a Buyer, I want to be able to buy properties confidently and feel secure.

As a buyer, I also want to be able to:

- (a) Place an order
- (b) Modify an order
- (c) Add listings to cart

SparkAI returned an expanded and reorganised output:

Real Estate Platform User Stories

Epic: Buy, Sell, and Search Real Estate Properties

Goal: Provide a user-friendly, professional, and engaging platform.

As a Buyer

- Search using keywords and filters
- View listings with images, virtual tours, and neighbourhood details
- Contact sellers and schedule viewings
- Save favourites, compare listings, and receive notifications
- Place, modify, or withdraw offers using secure payment options

As a Seller

- Create professional listings with images and pricing
- Respond to buyer enquiries and manage viewings
- Review and process offers securely

Analysis

The LLM introduced several substantial changes:

- A new seller role was added, despite not appearing in the model fragment. As depicted in Figure 2, there is a seller role, and the addition is sensible.
- The LLM introduced numerous capabilities (virtual tours, notifications, neighborhood details).
- Purpose clauses and hierarchical structuring were added throughout.
- Several assumptions were inserted, including secure payments and buyer–seller messaging.

These changes result in a more complete looking specification. The polished user stories also introduce new requirements not grounded in the underlying motivational model. This example demonstrates two phenomena. One is the potential of LLMs to enhance clarity. The other is the risk of unintentional scope expansion. Both phenomena reinforce the importance of human-in-the-loop validation.

### 6.3. Example 2: Motivational Modelling Tool

Our second example concerns the development and promotion of a modelling tool. The complete model is described in [34]. The leftmost subtree of the model is shown in Figure 3, redrawn using AMMBER.

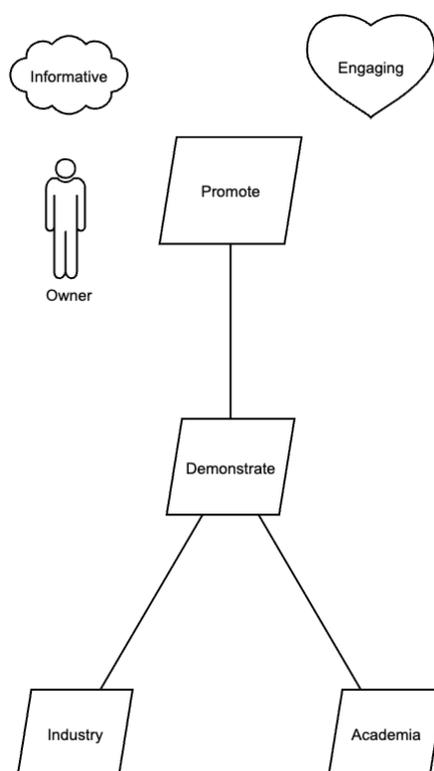


Figure 3. Fragment of a simple motivational model

Applying the baseline algorithm to the complete model yields the following set of user stories. The first epic corresponds to the model fragment in Figure 3.

- Epic: Promote
  - As Owner, I want to be able to demonstrate to industry.
  - As Owner, I want to be able to demonstrate to academia.
- Epic: Research
  - As a Researcher, I want to be able to collect data by survey.
  - As a Researcher, I want to be able to collect data by interview.
  - As a Researcher, I want to be able to analyse data.

- Epic: Improve MMTool
  - As a Student, Intern or Software Engineer, I want to be able to extend the MM tool.
  - As a Student, Intern or Software Engineer, I want to be able to test by system testing.
  - As a Student, Intern or Software Engineer, I want to be able to test by regression testing.
  - As a Student, Intern or Software Engineer, I want to be able to maintain by corrective maintenance.
  - As a Student, Intern or Software Engineer, I want to be able to maintain by perfective maintenance.

To refine the phrasing, these user stories were provided to SparkAI using the following prompt:

Can you improve the expression of the following user stories?

We deliberately employed simplified prompts to both minimise the influence of prompt engineering and to simulate a minimum effort scenario typical of novice use. This enabled us to focus analytically on the LLM's default refinement behaviours, including additions, omissions, and shifts in intent, rather than on performance optimised through highly structured instructions. SparkAI produced the following refined set of user stories:

#### User Stories for Project Management Tool

##### Epic: Promotion and Outreach

As an Owner

1. I want to showcase our tool's capabilities to industry professionals so that we can attract potential clients and partnerships.
2. I want to present our tool's research applications to academia so that we can foster collaborations and gain credibility.

##### Epic: Research Facilitation

As a Researcher

1. I want to design and conduct surveys through the tool so that I can efficiently collect quantitative data.
2. I want to schedule and manage interviews so that I can gather qualitative data in a structured manner.
3. I want to use built-in data analysis features so that I can process and interpret collected data effectively.

##### Epic: MM Tool Enhancement

As a Student, Intern, or Software Engineer

1. I want to extend the tool's functionality so that I can add new features or integrate with other systems.
2. I want to perform comprehensive system testing so that I can ensure the tool's functionality and reliability.
3. I want to conduct regression testing after updates so that I can confirm changes have not affected existing features.
4. I want to carry out corrective maintenance so that I can fix bugs promptly.
5. I want to implement perfective maintenance so that I can improve performance and user experience.

## Analysis

Several types of changes are evident. The most immediately visible is the renaming of the system to a *Project Management Tool*. This is an incorrect inference and a clear example of an error that must be corrected.

More subtle changes involve the insertion of intent clauses. For instance, the clause *so that we can attract potential clients and partnerships* does not appear in the original motivational model. While plausible, this addition reflects speculation and requires validation.

Similarly, the refinement of *test by system testing* into *perform comprehensive system testing* introduces a new quality attribute (*comprehensive*) that was not present in the model. Such additions may be beneficial, but only when confirmed with stakeholders. The replacement of verbs with near synonyms (e.g., *demonstrate* → *showcase*) also requires inspection for unintended shifts in meaning. These kinds of semantic expansions echo findings from prior work using LLMs to refine student writing [35].

## 7. Discussion

Developing software requirements is an iterative and socially negotiated process. Requirements rarely originate as complete and stable statements; instead, they emerge gradually through cycles of elicitation, clarification, and refinement among software engineers, system owners, and end users. Motivational modelling has proven effective in our teaching practice as a high-level elicitation technique because it foregrounds functional, emotional, and quality goals in a single shared artefact. Prior work has demonstrated that motivational models can serve as a generative structure for deriving rudimentary user stories [17,29,31]. However these first-pass stories often remain awkwardly phrased, incomplete, and inconsistently mapped to the model. These shortcomings motivate our exploration of how large language models might assist in refining user stories while preserving the strengths of the modelling process.

We also reflect on how the findings of our case-based investigation address the research question: *How can large language models be effectively integrated into requirements engineering teaching to improve the translation of motivational models into user stories while strengthening scalability, consistency, reliability, and student agency?* Our discussion is organised around four dimensions central to this question: (1) the effective integration of LLMs into RE teaching, (2) improvements to the translation process, (3) implications for scalability, consistency, and reliability, and (4) the preservation of student agency and the avoidance of metacognitive offloading.

Reflecting on our experience with over 2400 students using previous versions of the tool, we place particular emphasis on how this new Gen AI workflow impacts the cognitive process (unlike the purely manual modelling of the past).

### 7.1. Integrating LLMs into Requirements Engineering Teaching

Our analysis indicates that LLMs can be productively integrated into RE teaching when embedded within a structured, human-in-the-loop workflow. Motivational modelling provides a high-level representation of stakeholder roles, functional goals, and emotional and quality attributes that can be used as a stable anchor for LLM-assisted refinement. The LLM is not asked to generate requirements from scratch; instead, it builds on an existing, validated model, making its role complementary rather than generative. This approach mitigates the risk of LLMs fabricating system goals or introducing domain-incongruent interpretations. As a result, the workflow fits naturally into RE teaching practices, in which students already apply structured modelling techniques and engage in iterative refinement cycles.

### 7.2. Improving the Translation of Motivational Models into User Stories

The examples illustrate that LLMs can meaningfully enhance the clarity, structure, and completeness of user stories derived from motivational models. The LLM routinely reorganises fragmented stories into coherent hierarchies, refines awkward formulations, and introduces narrative conventions

common in user story practice such as “so that” clauses. These refinements can support novice requirements engineers who may struggle with articulating intent, naming actions, or producing consistent phrasing. At the same time, the analysis shows that LLMs introduce additions (e.g., new actions, quality attributes, purposes) that exceed the motivational model. While these expansions may be reasonable from a domain-knowledge perspective, they still require human validation to ensure they reflect stakeholder intent. This confirms that LLMs offer value primarily as *refinement assistants* rather than autonomous translators.

### 7.3. Scalability, Consistency, and Reliability

A key finding is that the workflow offers pedagogical and practical scalability. With large cohorts, the manual translation of motivational models into user stories often produces wide variation in quality, phrasing, and completeness. The LLM-refined outputs demonstrate more standardised narrative structure and wording, supporting consistency across student teams. In teaching contexts where multiple groups work with the same client, this consistency is especially beneficial.

Reliability, however, remains a qualified outcome. The LLM sometimes introduces inaccurate assumptions, as demonstrated by the transformation of *demonstrate* into *showcase*, the insertion of emotional or quality attributes (*comprehensive system testing*), or the incorrect inference of an entire subsystem (e.g., transforming *Notice* into a full notification system). These examples show that while LLMs can elevate linguistic quality, they cannot infer authentic stakeholder motivations. This reinforces the need for a workflow that embeds human judgement at multiple points, ensuring that scalability gains do not compromise the fidelity of requirements.

### 7.4. Preserving Student Agency and Avoiding Metacognitive Offloading

An important dimension of the research question concerns student agency. Recent work warns that GenAI tools can induce metacognitive laziness [23], leading learners to outsource judgment, exploration, and problem-framing to automated systems. Our workflow counters this risk by requiring students to extract model elements manually, formulate prompts, critically review LLM outputs, and validate refinements against the motivational model. These steps encourage reflective comparison, deepen understanding of functional and emotional goals, and maintain conceptual ownership over the requirements. In this respect, the workflow aligns with the principles of hybrid human–AI intelligence [20], where AI augments but does not replace human reasoning. The iterative cycle—modelling, LLM refinement, human validation, and model adjustment—strengthens students’ modelling competence and fosters a more active engagement with requirements engineering tasks.

### 7.5. Lessons Learned

From this investigation, we identify several takeaway lessons that can inform both software engineering education and future RE research:

1. **LLMs are most effective when refining, not generating, requirements.** Starting from a validated motivational model reduces hallucinations and anchors LLM output in authentic stakeholder goals.
2. **Human-in-the-loop validation is essential for reliability.** LLMs introduce plausible but unverified additions; without structured review, these can lead to intent drift or requirements misalignment.
3. **The workflow enhances scalability and consistency in teaching.** Standardised, LLM-assisted refinements help large cohorts produce clearer and more uniform user stories without suppressing individual reasoning.
4. **Comparative review of LLM outputs supports deeper learning.** Students learn from observing how the LLM rephrases, elevates, or extends their initial stories, producing opportunities for reflection on ambiguity, scope, and stakeholder intent.

5. **The approach preserves student agency and mitigates metacognitive offloading.** By requiring manual extraction, careful prompt design, and critical evaluation of outputs, the workflow avoids overreliance on the LLM and strengthens students' analytical and modelling skills.

Taken together, these findings demonstrate that LLMs can support, but not replace, the cognitive and interpretive work required in requirements engineering. When integrated within a principled workflow grounded in motivational modelling, LLMs offer a promising avenue for improving RE teaching while maintaining the centrality of human judgment.

## 8. Conclusions

We have described a role for Gen AI in polishing user stories generated from the motivational model. Gen AI assistance provides three advantages. First, it makes suggestions as to the intent of the user story, which the requirements engineer is free to adopt or reject. Second, by changing the language, it is potentially suggesting new requirements, which the engineer is similarly free to adopt and reject. Third, it improves the readability of the requirements, and makes the process of developing polished user stories more efficient.

The method has been shown to be effective in improving student work in a requirements engineering subject and also a software engineering projects. Future work will be to test it in more ongoing projects. Another piece of future work is to augment the AMMBER tool to generate user stories directly from AMMBER, and develop a workflow that integrates an LLM to refine the user stories.

Our conclusion came after attempting unsuccessfully to have students generate user stories directly usually a carefully structured prompt. In one case, students involved in a masters software engineering project spent several weeks trying to develop coherent stories. It would have been much easier to stick to concepts directly related to the motivational model. We also supervised a student research project [36] to generate user stories automatically. The student tried to validate his process without reference to the underlying model and no insights were gained.

**Author Contributions:** The authors contributed equally to the process, building on their previous collaboration.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** While this study did not specifically require ethics, some of the example models collected were covered under University of Melbourne Ethics #24272.

**Informed Consent Statement:** Consent is not required for this work.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Inayat, I.; Salim, S.S.; Marczak, S.; Daneva, M.; Shamshirband, S. A systematic literature review on agile requirements engineering practices and challenges. *Computers in human behavior* **2015**, *51*, 915–929.
2. Beck, K.; Beedle, M.; Van Bennekum, A.; Cockburn, A.; Cunningham, W.; Fowler, M.; Grenning, J.; Highsmith, J.; Hunt, A.; Jeffries, R.; et al. Manifesto for agile software development **2001**.
3. Schon, E.M.; Thomaschewski, J.; Escalona, M.J. Agile Requirements Engineering: A systematic literature review. *Computer Standards & Interfaces* **2017**, *49*, 79–91.
4. Bhat, J.M.; Gupta, M.; Murthy, S.N. Overcoming requirements engineering challenges: Lessons from offshore outsourcing. *IEEE software* **2006**, *23*, 38–44.
5. Cheng, H.; Husen, J.H.; Lu, Y.; Racharak, T.; Yoshioka, N.; Ubayashi, N.; Washizaki, H. Generative ai for requirements engineering: A systematic literature review. *Software: Practice and Experience* **2025**.
6. Marques, N.; Silva, R.R.; Bernardino, J. Using chatgpt in software requirements engineering: A comprehensive review. *Future Internet* **2024**, *16*, 180.
7. Daneva, M.; Van Der Veen, E.; Amrit, C.; Ghaisas, S.; Sikkil, K.; Kumar, R.; Ajmeri, N.; Ramteerthkar, U.; Wieringa, R. Agile requirements prioritization in large-scale outsourced system projects: An empirical study. *Journal of systems and software* **2013**, *86*, 1333–1353.

8. Blomkvist, J.K.; Persson, J.; Åberg, J. Communication through boundary objects in distributed agile teams. In Proceedings of the Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, 2015, pp. 1875–1884.
9. Tenso, T.; Norta, A.H.; Rootsi, H.; Taveter, K.; Vorontsova, I. Enhancing requirements engineering in agile methodologies by agent-oriented goal models: Two empirical case studies. In Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW). IEEE, 2017, pp. 268–275.
10. Habiba, U.e.; Haug, M.; Bogner, J.; Wagner, S. How mature is requirements engineering for AI-based systems? A systematic mapping study on practices, challenges, and future research directions. *Requirements Engineering* **2024**, *29*, 567–600.
11. Bjarnason, E.; Wnuk, K.; Regnell, B. A case study on benefits and side-effects of agile practices in large-scale requirements engineering. In Proceedings of the proceedings of the 1st workshop on agile requirements engineering, 2011, pp. 1–5.
12. Castro, J.W.; Acuña, S.T.; Juristo, N. Integrating the personas technique into the requirements analysis activity. In Proceedings of the 2008 Mexican International Conference on Computer Science. IEEE, 2008, pp. 104–112.
13. Kamthan, P. Using Personas to Support the Goals in User Stories. In Proceedings of the 2015 12th International Conference on Information Technology-New Generations. IEEE, 2015, pp. 770–770.
14. Faily, S.; Lyle, J. Guidelines for integrating personas into software engineering tools. In Proceedings of the Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems, 2013, pp. 69–74.
15. Sterling, L.; Taveter, K. *The art of agent-oriented modeling*; MIT Press, 2009.
16. Sterling, L.; Oliveira, E. Using Motivational Models to Promote Emotional Goals Among Software Engineering Students. In Proceedings of the IEEE 31st International Requirements Engineering Conference Workshops (REW), Hannover, Germany, 2023, pp. 30–35.
17. Oliveira, E.A.; Maram, V.; Sterling, L. Transitioning from motivational goal models to user stories within user-centred software design. In Proceedings of the RESOSY@ APSEC, 2021.
18. Oliveira, E.; Song, Y.; Saqr, M.; López-Pernas, S. An introduction to large language models in education. In *Advanced Learning Analytics Methods: AI, Precision and Complexity*; Springer, 2025; pp. 191–205.
19. Arora, C.; Grundy, J.; Abdelrazek, M. Advancing requirements engineering through generative ai: Assessing the role of llms. In *Generative AI for Effective Software Development*; Springer, 2024; pp. 129–148.
20. Cukurova, M. The interplay of learning, analytics and artificial intelligence in education: A vision for hybrid intelligence. *British Journal of Educational Technology* **2025**, *56*, 469–488.
21. López-Pernas, S.; Misiejuk, K.; Oliveira, E.; Saqr, M. The dynamics of the self-regulation process in student-AI interactions: The case of problem-solving in programming education. In Proceedings of the Proceedings of the 25th Koli Calling International Conference on Computing Education Research, 2025, pp. 1–12.
22. Oliveira, E.; Mohoni, M.; López-Pernas, S.; Saqr, M. AI collaboration or cheating? Using explainable authorship verification to measure AI assistance in academic writing. *Educational Technology & Society* **2026**, *29*, 100–117. [https://doi.org/10.30191/ETS.202604\\_29\(2\).RP06](https://doi.org/10.30191/ETS.202604_29(2).RP06).
23. Fan, Y.; Tang, L.; Le, H.; Shen, K.; Tan, S.; Zhao, Y.; Shen, Y.; Li, X.; Gašević, D. Beware of metacognitive laziness: Effects of generative artificial intelligence on learning motivation, processes, and performance. *British Journal of Educational Technology* **2025**, *56*, 489–530.
24. Mahnič, V.; Hovelja, T. Teaching user stories within the scope of a software engineering capstone course: analysis of students' opinions. *The International journal of engineering education* **2014**, *30*, 901–915.
25. Péraire, C. Learning to write user stories with the 4c model: Context, card, conversation, and confirmation. In Proceedings of the 2023 IEEE/ACM 5th International Workshop on Software Engineering Education for the Next Generation (SEENG). IEEE, 2023, pp. 33–36.
26. Wooldridge, M.; Jennings, N.R.; Kinny, D. The Gaia methodology for agent-oriented analysis and design. *Autonomous Agents and multi-agent systems* **2000**, *3*, 285–312.
27. Juan, T.; Pearce, A.; Sterling, L. ROADMAP: extending the Gaia methodology for complex open systems. In Proceedings of the Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1, 2002, pp. 3–10.
28. Bresciani, P.; Perini, A.; Giorgini, P.; Giunchiglia, F.; Mylopoulos, J. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems* **2004**, *8*, 203–236.
29. Oliveira, E.A.; Sterling, L. Motivational models for validating agile requirements in software engineering subjects. *arXiv preprint arXiv:2306.06834* **2023**.

30. AMMBER Tool: Motivational Modelling Editor. <https://motivationalmodelling.github.io/mm-local-editor/>, 2025.
31. Lorca, A.L.; Burrows, R.; Sterling, L. Teaching motivational models in agile requirements engineering. In Proceedings of the 2018 IEEE 8th international workshop on requirements engineering education and training (REET). IEEE, 2018, pp. 30–39.
32. Sterling, L.; Hirve, A.; Ahmed, H.; Vo, Q. Understandability of Requirements Artefacts - A Small Survey: Queue Solutions Report Series 23-01, 2023.
33. Paay, J.; Sterling, L.; Pedell, S.; Vetere, F.; Howard, S. Interdisciplinary Design Teams Translating Ethnographic Field Data Into Design Models: Communicating Ambiguous Concepts Using Quality Goals. In *Research Anthology on Recent Trends, Tools, and Implications of Computer Programming*; IGI Global Scientific Publishing, 2021; pp. 173–201.
34. Oliveira, E.; Maram, V.; Sterling, L. Transitioning from Motivational Goal Models to User Stories Within User-Centred Software Design. In Proceedings of the Proceedings of the First International Interdisciplinary Workshop on Requirements Engineering for Sociotechnical Systems (RESOSY 2021) co-located 28th Asia Pacific Software Engineering Conference (APSEC 2021), Virtual event, Taipei, Taiwan, December 8-9, 2021 / Taveter, K., Mooses, K., Gambo, I., and Iqbal, T. (eds.), Vol. 3107, 2021.
35. Sterling, L.; Ye, C.; Ying, H.; Chen, Z. Finding Your Voice: Using Generative AI to Help International Students Improve Their Writing. *Information* **2025**, *16*.
36. Lyu, J. Exploring the Feasibility of Generating User Stories from Motivational Model using Artificial Intelligence Language Model: Student report, Advanced computing studies unit, University of Melbourne, 2023.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.