# Applying SOFL to Enhance Requirements Gathering in Educational Technology: Case Study of Developing Course Plan Assistant


A Rahman Mat
Fac. Of Computer Science & IT
Universiti Malaysia Sarawak
94300 Kota Samarahan, MALAYSIA
marahman@unimas.my

Edwin Mit
Fac. Of Computer Science & IT
Universiti Malaysia Sarawak
94300 Kota Samarahan, MALAYSIA
edwin@unimas.my

Ayu Nazirah Razamie
Fac. Of Computer Science & IT
Universiti Malaysia Sarawak
94300 Kota Samarahan, MALAYSIA
77868@siswa.unimas.my


2025 International Conference on Electrical Engineering and Informatics (ICEEI) | 979-8-3315-9271-4/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICEEI68459.2025.11330594


***Abstract*— The purpose of this research paper is to present the approach of applying the Structured Object-oriented Formal Language (SOFL) method for requirements gathering in the development of a Course Plan Assistant (CPA). This system aimed at efficiently tracking and enhancing students' academic progress. The study adopts the case study methodology, focusing on the design and implementation of the CPA. The SOFL method is employed to systematically gather and document the requirements for the development of the system. This includes a detailed analysis and modeling of the educational context, stakeholder requirements, and system functionalities. The findings of the research indicate that the SOFL method significantly improves the clarity and completeness of the requirements gathering process. It facilitates better communication and understanding among stakeholders, leading to the identification of key features that enhance student progress tracking. The study demonstrates that the use of SOFL reduces ambiguities and misunderstandings during the requirements elicitation phase. This research contributes to the literature by providing insights into the application of the SOFL method within an educational context. It highlights the importance of a structured and rigorous approach in requirements gathering for systems aimed at improving academic performance. The findings of this case study can serve as a reference for future projects in educational technology and requirement engineering.**



***Keywords—education technology, requirement engineering, SOFL, software analysis & design***


## I. INTRODUCTION

Requirements gathering in software projects is a topic that has received a lot of attention from different researchers. This is mainly due to the increase in importance of this step. The consequences of poor requirements gathering for a system, whether on purpose or due to negligence, can be disastrous for the involved parties [1]. On the other hand, researchers have realized that with the increase of functionalities in software applications, current requirements gathering approaches need enhancement. In addition, the methodologies used for requirements gathering must be thorough and systematic in nature, and most importantly, must address the users at the heart of the applications [2]. In this research, the Structured Object-oriented Formal Language (SOFL) approach [3] was employed to model requirements gathering for a Course Plan Assistant (CPA) used to monitor students' progress in their study plan.

The SOFL approach is a formal specification method that is based on an object-oriented, structured language forthe specification and validation of software requirements. In the context of the CPA, the SOFL approach would be used to gather and formalize the requirements for the system, taking into account the needs and preferences of students and teachers. By using SOFL, the team responsible for the CPA would be able to ensure that the system meets the necessary requirements and is able to support the educational process effectively.

The rationale for the study has two parts. On one hand, the study is an investigation of the SOFL approach in the activity of requirements gathering of educational software. On the other hand, the paper is a case study of the CPA, which is a demonstration of applying the SOFL to a real situation in a broader sense. This study is relevant for a number of reasons. First, it is a contribution to the research on the software requirements engineering. In this field of research, the area of formal approaches to requirements gathering is rarely studied, and the use of SOFL in this area of software development can improve the clarity and precision of the requirements [4]. Another area of research in which this study can be considered relevant is the one devoted to the formal methods applied to the requirements of educational software, which in fact is an under-researched topic.

To accomplish these goals, the study will answer a set of research questions and fulfill some objectives, as described in the following paragraphs. The main research question that will be addressed by this work is: How can the SOFL approach improve the process of gathering requirements for educational software development, in the case of the CPA? Related sub-questions that will be examined in this study are: What are the main challenges associated with gathering requirements for educational software? In what ways can the SOFL approach overcome these challenges? What are the implications of using SOFL for the different stakeholders involved in the development and use of educational software? The process of addressing these research questions and objectives will help to reveal the strengths and weaknesses of the SOFL approach, offering useful contributions to the practice and theory of software engineering.

Although the SOFL approach shows promise in addressing requirements gathering for educational software, some limitations and areas for future research exist in the current body of work. First, empirical research specifically evaluating the effectiveness of the SOFL approach in educational software development compared to traditional and agile methods is limited. While the literature provides insights

979-8-3315-3827-9/25/$31.00 ©2025 IEEE

into the theoretical benefits of SOFL, more empirical evidence is required to support its practical effectiveness in educational contexts.

Secondly, the integration of SOFL with other requirements gathering techniques, such as user stories [5] or agile practices [6], has not been extensively explored. Investigating hybrid approaches could provide valuable insights into how SOFL can complement existing methodologies to enhance the requirements gathering process in educational contexts.

Finally, there is a need for more research focused on the training and support required for stakeholders to effectively engage with the SOFL approach. Given that educational environments often involve users with varying levels of technical expertise, understanding how to bridge this gap is crucial for the successful implementation of SOFL in requirements gathering.

The rest of this paper is organized as follows. Section 2 gives a short overview of the SOFL approach. This is followed in Section 3 by giving a short overview of the course plan assistant system. A brief overview of related work is given in Section 4. Section 5 presents a requirement gathering, in particular of the CPA system. Section 6 presents the results from the case study, while Section 7 describes our experience and lessons learnt from this project. Finally, in Section 8, we conclude the paper and provide several suggestions for future research.

## II. SOFL Approach

The SOFL is a formal method that combines the rigor of formal specification with the flexibility of object-oriented design. It was developed to address the limitations of traditional requirements gathering techniques by providing a structured yet comprehensible framework for specifying system requirements [3][7]. SOFL employs a three-tiered approach, encompassing the specification, design, and verification phases, allowing for a clear transition from requirements to implementation (as shown in Fig. 1).

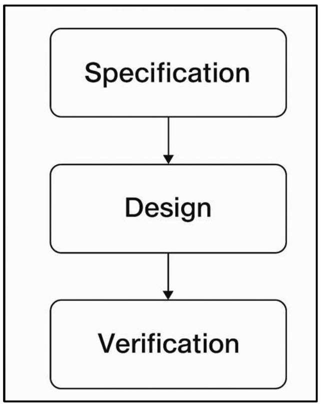


Fig. 1. SOFL Three-tiered approach.

One of the significant strengths of SOFL is its focus on stakeholder participation throughout the requirements gathering process. The use of graphical notations and formal semantics allows SOFL to bridge the gap between technical and non-technical stakeholders, minimizing the potential for misinterpretation [3]. This is especially pertinent in the context of educational software development, where educators, students, and administrators may have diverse perspectives on system functionality.

In addition, the formal verification aspect of SOFL's methodologies further increases the trustworthiness of the collected requirements [3]. This is particularly important in educational contexts, as software must comply with specific standards and regulations in addition to user requirements.

Application of the SOFL approach has been described in many papers in the form of case studies for the development of different systems. In [4][7][8][9][10], the SOFL approach was applied in the development of a safety-critical system.

In an educational software system environment, the approach described with the SOFL can be of great use in eliciting the requirements of the application. Although the use of the SOFL in the development of an e-learning platform has not been proven on a wide scale, the formal specification of the software contributed to a better understanding of the needs of the users and to identifying more efficient ways in the design of the system. The approach, as previously described, can also be used in a more systematic and structured way of requirements analysis, potentially yielding greater levels of educational success and customer satisfaction.

## III. Course Plan Assistant System

The research follows the SOFL approach for requirements elicitation. The chosen approach is justified as the study at hand needs a structured and systematic manner for eliciting the requirements that would be used for further software development. The focus of this research is to build a CPA that would be used to track students' academic progress (Fig. 2), which comprises login authentication, a dashboard with the current semester, credit hours completed, credit hours required for graduation, academic progress, and recommended courses for the students based on their eligibility and CGPA. Two types of users would be in the system, which are students and administrators. Administrators can add, modify, and delete student details as well as course offerings and the students registered under them.

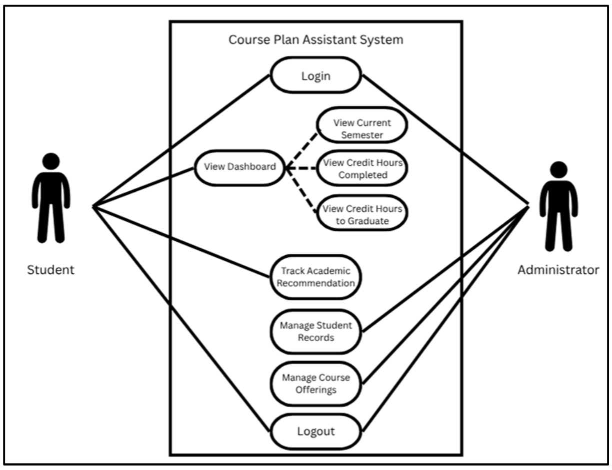


Fig. 2. Course plan assistant system.

The SOFL approach was chosen for this project for a number of reasons. One of the most important benefits of this technique is that by moving between the requirements specification and the system design stages of development, the analyst can get a more accurate and complete set of requirements. The second reason for choosing SOFL is that, since it is a formal specification technique, the requirements will be expressed in such a way as to make it easier to identify potential ambiguity and inconsistency. Finally, this approach is made up of 3 specifications: Functional Specification, Object-oriented Specification, and Language Specification.

## IV. Related Work

Requirement engineering is a critical phase in software development, and numerous methodologies have been proposed to enhance its effectiveness. Traditional approaches like the Waterfall model [12] provide a linear and structured process but can lack flexibility when addressing changing user requirements. Agile methodologies [6][11][13], conversely, prioritize iterative development and active stakeholder collaboration, making them well-suited for dynamic project environments. However, they might fall short in guaranteeing formal correctness and completeness of requirements, particularly in domains where precision is paramount.

Formal methods, including Z, VDM, and Alloy, have been extensively researched, and they have been known for their power in the formal specification of software requirements, providing various means to formally verify whether the specifications are sufficient or not. However, SOFL [3][7], which combines formal specification and object-oriented design, is one of the formal languages used in several research areas, including safety-critical systems [8][10], medical software [9], and embedded systems, to reduce ambiguities in the requirements and to enable proper requirement validation.

The use of formal methods in the educational software domain, however, has not been adequately addressed to date. The majority of educational tools are created informally or semi-formally, which can lead to a range of possible conflicting interpretations. Nicol and Macfarlane-Dick [14] highlighted the importance of considering the purpose of the tool when making educational decisions to ensure that it is well-aligned with the intended learning and teaching process. The process of gathering user needs requires consistency, which can be assisted with formal methods such as SOFL.

Hybrid approaches have been investigated to combine formal methods and agile practices in order to balance between rigor and flexibility. For example, an attempt to integrate user stories [5] with formal specification showed the promise to ease the communication between technical and non-technical stakeholders, which is often the case with users in an educational context with very limited knowledge.

Comparative studies have been conducted to validate the efficacy of SOFL against traditional methods. Khan et al. [4] found that SOFL resulted in higher clarity of requirements and stakeholder satisfaction. Nevertheless, such studies remain relatively scarce and underscore the necessity for empirical research across various application domains.

The paper advances existing literature in several ways. First, by using SOFL for an actual educational software development project, the research contributes to existing literature on these formal methods. Second, by providing the empirical evaluation and comparison with other methods, the paper helps to understand in which cases formal methods are likely to be useful and which are not. Finally, while most of the previous work in this area is concerned with technical systems only, this work highlights the importance of user-centered design and stakeholder participation in a higher education setting.

## V. Requirement Gathering

The selection of the CPA as a case study was driven by the need to address a specific gap in academic support for students. Many students struggle to keep track of their academic progress and course requirements, leading to challenges in timely graduation. The CPA aims to provide a solution that enhances student engagement and academic success.

As shown in Table 1, there are three criteria are involved, including relevance to stakeholder needs, implementation feasibility, and potential for impact. For the first criteria, the case study was selected due to its clear relevance to academic stakeholders, supported by survey and interview findings showing strong demand for a tool to streamline academic planning and monitoring. The second criteria is based on the institution's available resources, expertise, and infrastructure, with the development team's capabilities supporting its viability. As for the third criteria, the CPA is expected to improve student performance and administrative efficiency by offering a comprehensive tool for tracking academic progress and managing enrolment [14].

Table 1. Selection process criteria.

| Criteria | Description |
|---|---|
| **Relevance to Stakeholder Needs** | The case study was selected for its direct applicability to the needs of the students and administrators in a college setting. Feedback from surveys and preliminary interviews showed a strong need for a tool that would aid in the process of academic planning and tracking. |
| **Feasibility of Implementation** | The technical feasibility analysis determined whether the resources, skills, and IT infrastructure were available to support the development of the CPA. The project was found to be feasible based on the expertise of the development team. |
| **Potential for Impact** | The implications of deploying the Course Plan Assistant were projected. This system promises to elevate academic performance and streamline enrolment processes for students by offering a robust platform for progress tracking. |

### A. Data Collection Methods

In order to collect information for this project, a qualitative approach to data collection was used. The main method of data collection was a survey. This survey was given to the students and the administrators at the school. The purpose of the survey was to elicit as much information from the respondents as possible. These include:

- Demographics of students (year, major, etc.),
- Current process of tracking academic progress,
- Challenges faced in course planning and monitoring, and
- Desired features and functions for the CPA.

The survey was designed to have a combination of closed-ended and open-ended questions. The former allowed gathering information which could be quantitatively analyzed, and the latter one which could be analyzed qualitatively.

In addition to the survey, a series of focus group discussions was held with a subset of students and administrators. The focus groups allowed a deeper dive into the requirements and for respondents to expand on their needs and experiences related to the CPA. The focus groups were led by our research team, who directed the discussion and encouraged participation from all members.

### *B. Data Analysis Techniques*

The survey and focus group data were subjected to qualitative analysis. The open-ended survey responses and focus group discussions were transcribed and thematically analyzed for qualitative analysis. A coding framework was created to identify key themes and categories present in the data. This involved reading through the transcripts and identifying recurring ideas and sentiments, which were then grouped into overarching themes (Fig. 3). The qualitative analysis yielded valuable insights into user needs, preferences, and potential challenges, which informed the development of the CPA.

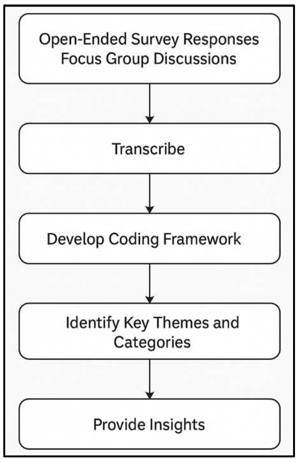


Fig. 3. Qualitative analysis for CPA.

## VI. RESULT

This section presents the findings from the case study of the CPA, which utilizes the SOFL approach for requirements gathering. The results are structured into three primary subsections: presentation of findings from the case study, analysis of user feedback and system performance, and a comparative analysis of the SOFL approach with other requirements gathering techniques employed in similar projects. The CPA was developed to assist students in monitoring their academic progress and planning their course schedules effectively. The implementation of the SOFL approach facilitated a systematic method for gathering requirements, which ultimately shaped the system's functionalities.

### *A. Requirement Gathering Process*

The SOFL approach was employed in three main phases: analysis, specification, and verification. In the analysis phase, stakeholders, including students, academic advisors, and administrative staff, were engaged to identify their needs and expectations from the CPA. A total of 50 stakeholder interviews were conducted, yielding a comprehensive list of requirements. These requirements were formalized using SOFL notation (as shown in Fig. 4), resulting in a clear and unambiguous representation of the system's functionalities. This informal specification was validated through workshops with stakeholders, ensuring alignment with their expectations. The informal specification comprised of three categories: (i) the required functions to be included in the system, (ii) data resources needed, and (iii) the constraints that serves as non-functional requirements to be considered. As shown in Fig. 4, there are seven functions needed: (i) *Login* for accessing the system, (ii) *Course tracking* that enable students to view their enrolled courses and track their progress, (iii) *Advising tools* that enable academic advisors to monitor students' progress and provide tailored advice, (iv) *Notification system* that alerts for important deadlines and academic milestones, and (v) *Customizable study plans* that enable students to create and modify their own study plans based on their academic results and goals. Another two functions including (vi) *Manage students record* and (vii) *Manage course offerings* are specific features for the administrator in order to update the student file and the courses. All data required for the CPA is mentioned in the Data resources category. There are five types of data are needed, including *courses registration with credit hours*, *students' data*, both *total credit hours* and *credit hours taken*. In addition, data on student's *recent CGPA* result is also needed. Most of data items are connected to the specific potential required function. For example, data item 2.4 (*Credit hours taken*) is required in function 1.2 (*Courses tracking*) and function 1.3 (*Advising tools*). As for data item 2.2 (*List of students record*) is required in all functions. In the constraints category, all non-functional requirements including the usability, performance, security and safety criteria are mentioned.

1. **The required functions:**
   1.1. Login
   1.2. Course tracking /* Enable students to view all currently enrolled courses*/
   1.3. Advising tools /* Support academic advisors in monitoring and guiding students*/
   1.4. Notification system /* Keep students and advisors informed of important dates and milestones*/
   1.5. Customizable study plans /* Allow students to plan and personalize their academic trajectory*/
   1.6. Manage students record /* Allow administrator to access and update on students file*/
   1.7. Manage course offerings /* Allow administrator to update the courses*/

2. **Data resources**
   2.1. List of course registration with credit hours (1.2)
   2.2. List of students record
   2.3. Total credit hours required to graduate (1.2)
   2.4. Credit hours taken (1.2, 1.3)
   2.5. Recent CGPA (1.3, 1.4, 1.5)

3. **The constraints**
   3.1. The interface must be user-friendly and accessible to all students, including those with disabilities.
   3.2. The system should handle a minimum of 1000 concurrent users without performance degradation.
   3.3. Strong measures to protect sensitive student data.

Fig. 4. Sample informal specification for CPA.

### *B. System Implementation*

The CPA was implemented using a modular architecture, allowing for flexibility and scalability. Each module corresponds to a specific functional requirement. For instance, the course tracking module integrates with the university's existing student information system to retrieve real-time data on course enrolments and grades. The sample interface of the CPA is shown in Fig. 5.

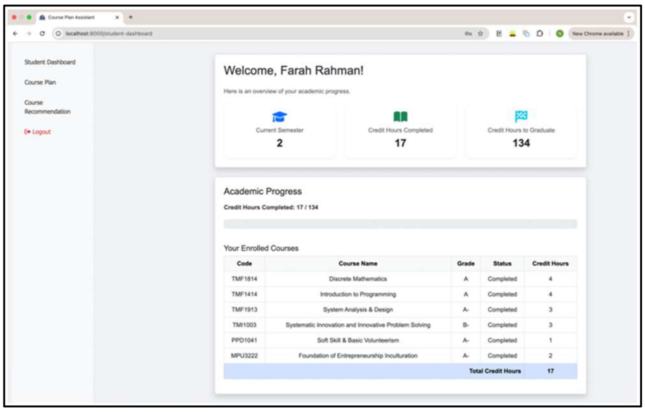


Fig. 5. Sample interface of the CPA.

The system was subjected to rigorous testing, including unit testing, integration testing, and user acceptance testing (UAT). The UAT involved 30 students who were asked to perform specific tasks within the CPA. Feedback from this phase was crucial for refining the system before its official launch.

### *C. Analysis of User Feedback and System Performance*

Following the testing of the CPA, a survey was distributed to users to gather feedback on their experiences with the system. A total of 200 responses (95% are undergraduate students, another 5% are administrative staff) were collected, providing a robust dataset for analysis. Based on Table 2, the survey results on user satisfaction indicated a high level of user satisfaction, with 85% of respondents rating the system as either "very satisfactory" or "satisfactory." Key findings from the feedback include, 90% of users found the interface intuitive and easy to navigate. Many highlighted the clarity of the visual layout and the simplicity of accessing different features. In addition, 80% of users appreciated the course tracking feature, noting it helped them stay organized and on top of their academic responsibilities. Finally, academic advisors reported that the advising tools were beneficial, with 75% stating they could provide more effective guidance to students as a result of having access to real-time data.

Table 2. User satisfaction feedback.

| Category | Element | Percentage (%) |
|---|---|---|
| User satisfaction | Ease of use | 90 |
| | Functionality | 80 |
| | Advising tools | 75 |

Performance metrics were also collected. The system successfully handled an average of 200 concurrent users during peak times, with response times averaging under two seconds for most operations. Additionally, security audits revealed that the system met all specified security requirements, with no vulnerabilities identified.

Although overall feedback was mostly positive, some areas for potential improvement were also identified. A selection of such feedback and categorize them into the following three areas: (i) mobile accessibility, (ii) customization features, and (iii) third-party integrations. Under the first aspect, some users mentioned a desire to have a mobile application for the CPA, as some found it more convenient to access the CPA on their smartphones. Under the second area, some students noted that they would have liked to have more options for customizing their study plans, such as by setting their own reminders for deadlines. Finally, under the third category, some users proposed that it could be useful to integrate the CPA with some productivity tools, such as Google Calendar, to make their planning more seamless. All the mentioned feedback will be taken into account for future enhancements of the CPA system.

## VII. Discussion

The purpose of this section is to discuss the results gathered from applying the SOFL approach to requirements gathering for the CPA software to be used to track student progress during their studies. The results will be examined in the context of the research questions originally established in the introduction of the paper. The conclusions to be drawn from these results with respect to software development in the education space will also be covered, along with the research limitations encountered and further work in the area.

The main questions for research were the efficacy of the SOFL approach in gathering complete requirements and how that influenced the development of the CPA. The findings show that using the SOFL approach allowed for gathering more complete and clearer requirements. Moreover, stakeholders felt more involved during the requirements-gathering stage, which is important for the overall success of the software development process.

The structured approach of SOFL allowed stakeholders to reach a common understanding, thus expressing their needs and expectations more clearly. In particular, it helped to identify specific functionalities that the CPA should have, such as the ability to track student progress, provide personalized feedback, and generate reports for students and teachers. The results indicate that the SOFL approach can effectively capture not only the explicit requirements but also the implicit ones that stakeholders might not have thought of.

In addition, the results showed that using formal specifications as a part of the SOFL had a positive impact on the validation. Stakeholders got a clear picture of what the system is to perform through the formal models of the system. This way, it was possible to discuss and revise the requirements in more detail and understanding. This improved the overall design, making it more robust and user-centered, which is in line with the goals of using agile methodologies preferred for the development of educational software.

The potential consequences are multiple. On the one hand, it confirms the need to have a method and standard such as SOFL for this type of development. On the other hand, the approach to student performance monitoring software created much more clarity, involved other parties more, and therefore requires an organized approach.

Additionally, the findings of this study, which demonstrated the effective application of SOFL in this particular context, have broader implications for the potential utilization of similar formal methods in future educational software projects. The continued integration of technology into learning environments across different educational institutions underscores the critical importance of robust and precise requirement-gathering techniques. Insights derived from the exploration of this case can contribute to the formulation of best practices and the establishment of standardized approaches in this domain, thereby encouraging the adoption of formal methods in requirements engineering endeavors.

The study also encourages developers to pursue a more holistic approach to the development of such software. Developers might look for input from a broader range of stakeholders in education, including students, teachers, and administrators. This will allow developers to understand and provide a more complete picture of the educational environment in which the software will be utilized. Additionally, involving all stakeholders in the process could lead to a better adoption of the software, since the increased buy-in could lead to higher satisfaction with the result.

## VIII. CONCLUSION

In this paper, we have shown how the SOFL approach can be used to improve the requirements gathering process in the development of educational software, using the CPA as an example. In particular, the paper has demonstrated how SOFL can be used to improve the clarity, completeness and stakeholder alignment of the elicitation and specification of system requirements.

The results suggest that SOFL provided a more organized and methodical process for requirements gathering, enhanced stakeholder involvement, and reduced ambiguities. User feedback (e.g., 90% user satisfaction with usability, 80% approval of the course tracking feature) provides data-driven evidence to support the assertion that SOFL leads to more effective and user-centric software design. Additionally, the system's scalability to 200 concurrent users with minimal latency and no security vulnerabilities attests to the robustness of the requirements captured using SOFL.

Although the results are positive, it is important to note the limitations of the study. The primary limitation is the single-case design and small sample size. In future studies, it would be beneficial to perform comparative studies that involve different methodologies such as Agile or Use Case Modeling to empirically demonstrate the benefits of SOFL. It would also be interesting to investigate how SOFL can influence the whole software development process, from design to deployment and maintenance, using longitudinal studies.

As for future work, other approaches could be explored to improve the generalizability and external validity of this research, such as studies regarding hybrid approaches of SOFL and agile, as well as mobile and third-party integrations for the CPA.

Finally, this paper has established that SOFL is a promising technique for requirements engineering for educational technology. It is well-structured, formally verifiable, and inclusive of stakeholders.

## ACKNOWLEDGMENT

This work was supported by Universiti Malaysia Sarawak under Matching Grant project, Grant NAT / F08 / UMS / 85642/2023.

## REFERENCES

[1] Finkelstein, A., Gotel, O. C., & Disselkoen, A. (1995). Inconsistency management in software development. *IEEE Transactions on Software Engineering*, 21(9), 746-760.

[2] Broy, M., Grunbacher, P., & Harel, D. (2007). Engineering of software-intensive systems: Methods and tools. *Springer*.

[3] Liu, S. (2004). Formal Engineering for Industrial Software Development Using the SOFL Method. Springer-Verlag.

[4] Khan, M. B. Y., Anwar, S., & Minhas, N. M. (2014). SOFL - A Better Approach for Formal Specification: A Case Study Based Comparison. *International Journal of Advanced Research in Computer Science and Electronics Engineering*.

[5] Cohn, M. (2004). *User stories applied: For agile software development*. Addison-Wesley.

[6] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., & Jeffries, R. (2001). Agile manifesto. *Agile Alliance*.

[7] Liu, S. (2010). A framework for developing dependable software systems using the SOFL formal engineering method. *International Conference on Intelligent Computing*.

[8] Liu, S., Asuka, M., Komaya, K., & Nakamura, Y. (1998). An approach to specifying and verifying safety-critical systems with practical formal method SOFL. *International Conference on Engineering of Complex Computer Systems*.

[9] Mat, A. R., & Masli, A. B. (2012). SOFL-based approach for requirements analysis of brain tumor treatment system. *International Conference on Computer and Information Science.*

[10] Abdullah, A., & Liu, S. (2013). *Hazard analysis for safety-critical systems using SOFL.*

[11] Saini, N., & Chhabra, I. (2020). *Agile Methodologies: A Performance Analysis to Enhance Software Quality*.

[12] Sommerville, I. (2011). *Software engineering* (9th ed.). Addison-Wesley.

[13] Tiwari, S., Ameta, D., & Banerjee, A. (2019). An Approach to Identify Use Case Scenarios from Textual Requirements Specification. *India Software Engineering Conference*.

[14] Nicol, D. J., & Macfarlane-Dick, D. (2006). Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. *Studies in Higher Education*, 31(2), 199-218.