European Journal of Computer Science and Information Technology, 12 (4),1-17, 2024
Print ISSN: 2054-0957 (Print),
Online ISSN: 2054-0965 (Online)
Website: https://www.eajournals.org/
Publication of the European Centre for Research Training and Development -UK

Exploring Agile Methodology in Developing a Web-Based Result Computation and Transcript System: A Case Study of Federal Polytechnic Ukana

Mfon Okpu Esang[1], Ekemini Anietie Johnson[1], Kingsley Attai[2], Jude Alphonsus Inyangetoh[3] Eduediuyai Ekerette Dan[4], Kitoye Ebire Okonny[5], Ifeanyi Bardi[6], Aniefiok Bassey [7]& Anietie John [2]

[1]Department of Computer Science, Federal Polytechnic Ukana, Nigeria
[2]Department of Mathematics and Computer Science, Ritman University, Ikot Ekpene, Nigeria
[3]Department of Statistics, Federal Polytechnic Ukana, Nigeria
[4]Department of Computer Engineering, Federal Polytechnic Ukana, Nigeria
[5]ICT Centre, Ignatius Ajuru University of Education Port Harcourt, Nigeria
[6]Department of Computer Science, Alvan Ikoku Federal University of Education, Nigeria
[7]Department of Computer Science, Ignatius Ajuru University of Education Port Harcourt, Nigeria

oi: https://doi.org/10.37745/ejcsit.2013/vol12n4117 Published June 05, 2024

Citation: Esang M.O., Johnson E.A., Attai K., Inyangetoh J.A., Dan E.E., Okonny E.E., Bardi I., Bassey A., and John A. (2024) Exploring Agile Methodology in Developing a Web-Based Result Computation and Transcript System: A Case Study of Federal Polytechnic Ukana, *European Journal of Computer Science and Information Technology*, 12 (4),1-17

ABSTRACT: *In today's rapidly changing technological landscape, there is a great demand for efficient, adaptable, and responsive software development methodologies. This article examines how agile methodology was used in practice to create a web-based transcript and result computation system for Polytechnic Ukana. Agile methodology, which is renowned for its incremental and iterative approach, was used to increase flexibility, foster better teamwork, and produce a workable product on schedule. The article describes the goals and requirements of the project, including how Scrum was chosen as the agile framework and what stages it went through in implementation. Along with the roles and responsibilities of the development team, the planning, execution, and review processes for sprints are highlighted. The team demonstrated flexibility in responding to evolving needs, effective problem-solving, and sustained stakeholder engagement through a sequence of sprint cycles. The system was deployed on schedule, fulfilling all requirements and substantially boosting the accuracy and efficiency of the institution's transcript generation and result computation. This is proof that the project was successful. This case study provides a useful manual for implementing agile approaches for comparable educational software development projects.*

KEYWORDS: result computation, transcript system, Federal Polytechnic, Ukana

INTRODUCTION

The demand for efficient, flexible, and responsive software development methodologies is highly sought after in today's evolving technological landscape. Therefore, agile methodology is becoming a shining example of innovation, providing businesses with a flexible approach to software development that puts flexibility, teamwork, and ongoing improvement first (Ouriques et al. 2023). Although several studies have utilized other methodologies in result computation and transcription(Akputu et al., 2020; Bello et al.2023), Agile methodology offers several advantages in software quality engineering (Pargaonkar, 2023).This paper unveils agile methodology through a detailed case study focused on the implementation of a web-based result computation and transcript system at Federal Polytechnic Ukana.The Federal Polytechnic Ukana, like many educational institutions, faces the challenge of managing and disseminating academic results and transcripts efficiently because they manage their day-to-day activities using the flat file system. This traditional flat files system often struggles to keep pace with the evolving requirements and expectations of stakeholders in a dynamic environment such as a tertiary institution. In response to these challenges, adopting agile methodology presents a promising solution, offering a framework that emphasizes iterative development, customer collaboration, and the ability to respond to change (Ogundipe et al., 2024).This case study aims to provide insights into the practical application of agile methodology within the context of developing a web-based result computation and transcript system. By examining the experiences, challenges, and successes encountered throughout the implementation process, this paper seeks to shed light on the effectiveness of agile practices in addressing the unique needs and complexities of academic institutions. Through a combination of qualitative analysis and empirical data, this study explores various aspects of agile methodology, including its principles, practices, and impact on project outcomes. By documenting the journey of transitioning from traditional development approaches to agile practices, valuable lessons and best practices can be gleaned, guiding organizations embarking on similar ventures.

Furthermore, this paper contributes to the growing literature on agile methodology by offering a real-world case study that illustrates its application in a specific domain. By examining the intricacies of software development within the academic sector, valuable insights can be gained into the challenges and opportunities inherent in leveraging agile methodologies in diverse contexts. This paper serves as a comprehensive exploration of agile methodology through the lens of a case study conducted at Federal Polytechnic Ukana. By delving into the intricacies of the implementation process and analyzing its implications, this study aims to enrich our understanding of agile practices and their potential to revolutionize software development in academic institutions and beyond.

RELATED WORKS

Agile methodology is a software development approach that prioritizes adaptability, teamwork, and change-responsiveness, by placing less emphasis on adhering to a strict plan and more emphasis on producing functional software in brief iterations.The core principles include customer collaboration, iterative development, and change response. It places a strong emphasis on stakeholders' or customers' ongoing active participation and collaboration during the development process which guarantees that the product satisfies the users' requirements and expectations. Agile projects are often broken up into manageable chunks, or iterations, where the goal is to produce a potentially useful product increment, enabling continuous feedback and improvement. Agile views change as a necessary component of the development process and the teams are always ready to adjust and modify their plans to accommodate new insights, as opposed to resisting changes to requirements or priorities (Ozkan et al., 2020; Al-Saqqa et al., 2022; Nazir et al., 2022).With the Agile methodology, project teams can react swiftly to shifting demands, market conditions, and customer feedback, making it a more responsive and flexible development process. It enables a quicker delivery of functional software by segmenting projects into smaller iterations and providing incremental updates. Agile approaches place a high value on ongoing customer feedback and collaboration, which results in products that better satisfy user requirements and expectations(Al-Saqqa et al., 2020; Sarangee et al., 2022; Grass et al., 2020).

What constitutes an agile software development process is the subject of research by Kuhrmann et al. (2021). To determine software development techniques and approaches that enhance or reduce agility, the researchers provide an empirical analysis based on a comprehensive worldwide survey. They examine how agile standard project disciplines are judged and how this relates to the development processes and methods that are employed, based on 556 data points. The results indicate that a relatively small percentage of participants less than 15% manage their projects in a strictly traditional or agile fashion. Nevertheless, a definite tendency toward greater levels of adaptability can be seen in the majority of project disciplines and practices. The choice of practices has a greater impact on a discipline's level of agility than the techniques utilized to produce software. Lastly, there are no procedures or approaches that either promise or prohibit agility. The researchers conclude that process definitions are not the only way to describe agility. Agility implementation or improvement in a software company requires consideration of other issues.

Al-Saqqa et al. (2020) carried out an extensive analysis that highlights the primary agile values and principles as well as the salient features that set agile approaches apart from traditional ones. The most widely used agile approaches are then covered, together with information on their responsibilities, life cycles, benefits, and drawbacks. The analysis reveals that big data and cloud computing have adopted agile development. Last but not least, this study emphasizes

how to select the most appropriate agile approach based on the task at hand, the product's sensitivity, and the organizational structure.

In a comprehensive analysis of the literature, Edison et al. (2021) compared the primary large-scale agile approaches, including SAFe, LeSS, Scrum-at-Scale, DAD, and the Spotify model. According to the researchers, this is the first study to evaluate and analyze all of the measurements, tools, practices, and principles of the method consistently. It includes all extensions and changes to each technique suggested by further empirical study, in addition to the original method specifications. This comparison encompasses not only large-scale commercial approaches but also custom-built ones from companies like Ericsson, Nokia, and others. Practitioners are better equipped to decide whether commercial technique, method component, or even custom-built method best suits their needs based on the study's findings. The analysis highlights several theoretical and practical problems with the literature now in circulation, including the emphasis placed on commercial framework practices at the expense of their fundamental ideas or even any unique methods. The application of large-scale agile methods is linked to several difficulties and success criteria. Several research gaps that need to be filled using various approaches are also identified by the study.

The goal of Perkusich et al. (2020) was to combine and examine the use of intelligent strategies in Agile Software Development (ASD). The report also identifies adoption concerns and evaluates its maturity. The study found 104 primary papers through a thorough literature review, yielding 93 distinct studies. The number of studies using intelligent techniques to address ASD is on the rise, according to the researchers, who also found that the most often used intelligent techniques in the context of ASD are machine learning, search-based solutions, and reasoning under uncertainty (primarily using Bayesian networks). The most common goals are requirements management, resource allocation, effort estimation, requirements prioritization, and requirements selection. It was also found that supporting decision-making is the main objective of using intelligent strategies. As a result, few adoption risks are associated with the present solutions' safety. Lastly, the report emphasizes the use of explainable intelligent approaches as a trend. The topic area is emerging, although it is still in its infancy for many areas of application, according to the researchers' conclusion. This indicates that there is a great demand for additional empirical study as well as a wealth of fresh prospects for academics.

To quantify the performance of ongoing agile software development projects in terms of cost, time, and customer satisfaction, Tam et al. (2020) created a model made up of five people factors. The results of a study conducted with 216 agile practitioners indicate that "customer involvement" and "team capability" are the primary elements influencing the success of ongoing agile software development projects. The findings were validated by triangulating

these results with a focus group using a mixed-methods technique. Teams and managers can set priorities and improve project outcomes by understanding which elements are critical to success.

The Agile Teamwork Effectiveness Model (ATEM) for Colocated Agile Development Teams is presented by Strode et al. (2022). The revised general team effectiveness model is based on data from multi-vocal literature, case studies, and focus groups. Redundancy, adaptability, shared leadership, team orientation, and peer feedback make up the paradigm of effective agile teamwork. It takes coordinating mechanisms to make these components possible. Mutual trust, communication, and shared mental models serve as the coordinating mechanisms. The researchers analyze the model thoroughly and talk about extensions for remote, multi-team, extremely tiny, and safety-critical development settings. The agile community's researchers, team members, coaches, and leaders are the target audience for this approach.

Mishra and Alzoubi (2023) did a comparative analysis of structured software development versus agile software development. The objective of this research is to design a decision tree that will assist in determining which waterfall and agile approaches are most suited for a given software development project. Several scenarios and examples are investigated with the hybrid development methodology. The research showed that even though agile approaches have numerous benefits, some projects or development phases may call for a mixed strategy because Waterfall is sometimes important. The viability of merging Agile and Waterfall approaches in software development management is investigated in this study.

Arcos-Medina and Mauricio (2019) conducted a systematic literature review on aspects of software quality applied to the process of agile software development. The objective of this work is to conduct a systematic evaluation of the literature on the following topics: agile practices, quality models, critical success factors, quality attributes, metrics to measure quality, and agile concepts taken into account by these models. 773 papers about quality in the agile development process were found; 71 of those were chosen for this article's preparation. This study lists 102 metrics, 70 quality traits, 137 agile practices, and 118 important success criteria that have an impact on agile software development. Additionally, to compile this article, 14 quality models were examined; nevertheless, they do not address all aspects of quality linked to agile development. The most researched feature of agile techniques is how they impact software quality, which is taken into account in 28.17% of the examined studies.

The use and effectiveness of software security engineering activities in the context of agile software development, as carried out by software developer experts, were empirically proven by Rindell et al. in 2021. Software practitioners in Finland were surveyed (N = 61) about their use of 16 agile software development items and activities, as well as their perceptions of the

security impact of using 40 standard security engineering methods. The selection of security engineering practices was quantified in response to the use of agile items and activities. The perception of the security practices' impact was found to be lower than the rate of use would suggest. This finding was interpreted as evidence of selection bias, which could be brought about by, for example, developers' limited awareness of specific security engineering practices or challenges integrating the practices into an iterative software development workflow. Proactive security procedures that were implemented early in the software development process were shown to have the greatest impact. Agile practices were observed to be systematically used in conjunction with security practices, and this compliance was noted. It was during the requirement and implementation phases when security operations were most prevalent. Generally speaking, the most influential activities were those that occurred early in the life cycle. There was a disparity found between the degree of use and the perceived security impact of certain security measures.

In software development, agile approaches have various benefits, such as increased adaptability and flexibility to shifting market conditions, stakeholder input, and requirements. Agile facilitates the timely delivery of functional software by giving priority to iterative development cycles and continuous improvement (Al-Saqqa,Sawalha, & AbdelNabi, 2020). Additionally, Agile actively involves stakeholders in the process. This method expedites early risk identification and mitigation while also cutting down on time-to-market (Mukherjee, 2020). Agile also helps to create a cooperative team atmosphere where trust and good communication are encouraged (Tyagi, Sibal, & Suri, 2022). All things considered, agile approaches enable teams to produce excellent software that more effectively satisfies the changing needs of consumers and the shifting demands of the market.

The implementation of Agile methodologies in software development poses several challenges, such as handling requirements volatility, allocating resources in the face of conflicting priorities, reconciling documentation and Agile principles, promoting productive team dynamics, and expanding Agile practices to larger projects or organizations(Iivari, 2021; Gupta, Poels, & Bera, 2022). Agile's dynamic nature can occasionally make it difficult to stabilize project scope, especially for larger projects or environments where priorities are constantly shifting (Layton et al., 2020). Continuous effort and coordination are also needed to maintain cohesive team dynamics and guarantee the availability of skilled resources. It can be difficult to strike a balance between Agile's emphasis on working software and the necessity for documentation, particularly in regulated industries where formalized documentation is crucial (Heeager& Nielsen, 2018). Finally, to preserve uniformity and alignment amongst various teams or departments, extending Agile practices to bigger teams or organizations necessitates meticulous preparation and coordination (Schtein, 2018).

Agile Software Development Architecture

Agile software development architecture refers to the high-level structure and organization of software components, designed to support agile methodologies. Agile methodologies emphasize iterative development, collaboration, and flexibility.

From the outset, the user/client, developers, and analysts are all involved in an agile architecture environment. And as the project develops, they continue to communicate with one another. One of the main advantages of this strategy is its capacity to prevent misinterpretations or misunderstandings. The capacity to adapt to changes is another, as an Agile methodology minimizes the need for change-driven coding because changes are likely to affect only tiny areas of the entire system.The agile architecture is depicted in Figure 1.

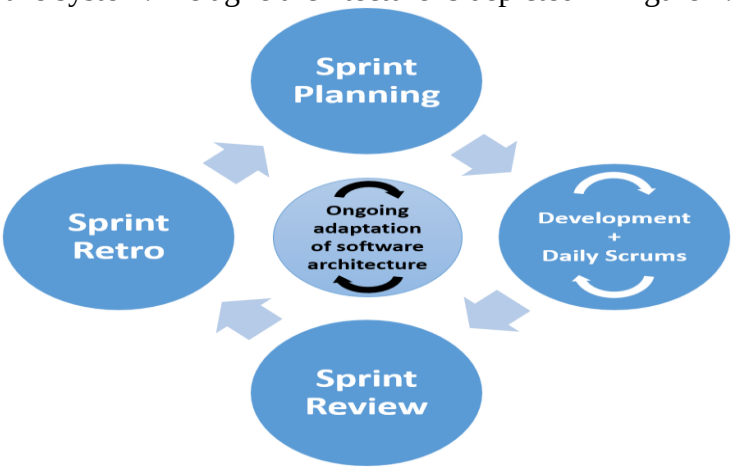

Figure 1: Agile software development Architecture

METHODOLOGY

This study used a case study approach with an emphasis on Federal Polytechnic Ukana's development of a web-based result computation and transcript processing system. The rationale behind selecting the case study approach is its capacity to offer a comprehensive, in-depth analysis of a particular phenomenon in the context of real-world situations, thereby facilitating a complete comprehension of the subject matter (Harrison et al., 2017).With this approach, the researchers can investigate how Agile methodology is applied in software development within the particular setting of the Polytechnic, taking into account variables like stakeholder dynamics, organizational culture, and project constraints. The case study attempts to provide insights into the application of Agile principles, the difficulties faced, and the results obtained in creating the result computation and transcript system through interviews, observations, and examination of project documentation.

Agile methodology was selected because it works well with changing project requirements, especially in the context of educational institutions, for the development of the web-based transcript and result computation system. Agile's iterative methodology facilitates regular stakeholder feedback and collaboration, which is essential when developing a system that must change to meet the needs of users and stakeholders. Furthermore, Agile's focus on producing

functional software in brief iterations is in line with the requirement for timely updates and enhancements in educational systems. The implementation of Agile methodology can enhance the project team's ability to handle ambiguities, adapt to modifications, and rank features according to their importance to end users, ultimately resulting in a system that is more user-friendly and responsive.

The implementation of Agile principles in the development of the web-based Secure Result Computation and Transcript Processing System was pivotal in ensuring flexibility, adaptability, and continuous improvement throughout the project lifecycle. The Agile product development methodology is a structured software development approach that prioritizes collaboration, adaptability, and iterative improvement. A simplified outline of the methodology is presented below.

Project Initiation: The goal and objectives of the project were defined and the necessary stakeholders which include students, Teaching staff, administrators, and IT personnel were identified and the scope of the project was also outlined.

Requirements Gathering: During this stage, collaborations with stakeholders took place to obtain and prioritize requirements for the proposed system. Techniques such as interviews and workshops were used to elicit requirements from these stakeholders.

Iterative Development: The project was organized into iterative development cycles known as sprints, typically lasting two to four weeks. Each sprint focused on delivering a set of prioritized features or functionalities, allowing for rapid iteration and feedback incorporation. Before the start of each sprint, the development team, along with stakeholders, conducted iteration planning meetings to prioritize the backlog of features and define the sprint goals. During these meetings, tasks were estimated, and commitments were made regarding what could be delivered within the sprint timeframe.

Continuous Feedback and Improvement: At the end of each sprint, a retrospective meeting was held to reflect on what went well, what could be improved, and what actions could be taken to enhance the development process. This continuous feedback loop enabled the team to identify bottlenecks, address challenges, and make necessary adjustments to improve efficiency and effectiveness in subsequent sprints.

Transparency and Communication: Stakeholder involvement was integral throughout the development process. Regular communication and collaboration ensured that stakeholder requirements were understood and prioritized effectively. Feedback from stakeholders was actively sought and incorporated into the development process, allowing for the alignment of the system with end-user needs and expectations.

Testing and Quality Assurance: Implement automated testing, conduct regular quality assurance activities, and address defects promptly.

Deployment and Release: Plan for deployment at the end of each sprint, provide user training and documentation and gather feedback post-deployment for further enhancements. The web-based result computation and transcript system is a digital platform designed to automate and streamline the process of computing results and generating transcripts for students.

System Description

This system serves as a centralized hub for managing academic records and facilitating efficient data entry, processing, and retrieval of student information. Key features of the system may include:

User Authentication and Access Control: Secure login functionality to authenticate users (such as students, faculty, and administrative staff) and assign appropriate access levels to ensure data confidentiality and integrity.

Result Computation Module: A module for inputting and computing student grades, taking into account various factors such as coursework, assignments, quizzes, and exams. The system may support automated grading calculations based on predefined criteria and grading scales.

Transcript Generation: Functionality to generate official transcripts for students, including comprehensive academic records, grades, credits earned, and other relevant information. The system may allow for the customization of transcript formats and templates to comply with institutional requirements.

Data Management and Reporting: Tools for managing student records, storing historical data, and generating reports on academic performance, course enrolment, graduation status, and other metrics. The system may include filters and search capabilities to retrieve specific information easily.

Communication and Notification: Integration with communication channels (e.g., email, SMS) to notify stakeholders about important updates, deadlines, or changes related to academic records and transcripts.

Security and Compliance: Implementing robust security measures to protect sensitive student data and ensure compliance with data privacy regulations. This may include encryption, data access controls, and regular security audits.

User Interface and Experience: An intuitive and user-friendly interface designed to enhance usability and accessibility for all stakeholders, including students, faculty, and administrative staff. The system features a responsive design to support access from various devices and screen sizes.

Data Collection Methods

The following techniques were used to gather information for the study of the creation and application of the web-based transcript and result computation system: focus groups, observations, document analysis, interviews, surveys, and user testing. While surveys offer quantitative input from a larger audience, interviews offer in-depth insights from important stakeholders. Through observations, our team observed how users interact with and use the existing system. Project documentation was examined through document analysis to provide context.Focus groups encouraged interactive discussions among stakeholders, while user testing assessed the usability and functionality of the system. Through the utilization of these diverse methodologies, investigators can amass extensive data to comprehend the project's progress, obstacles, and results from various angles.

Data Analysis Technique

In the context of the proposed system, "Iterative Review and Retrospectives" within Agile methodology involved regular reflection sessions conducted by the project team at the end of each iteration. These sessions, known as retrospectives, were aimed at assessing the progress made, identifying areas for improvement, and determining actionable steps to enhance future iterations of the development process. During retrospectives, team members from various roles, such as developers, project managers, academic staff, and administrative personnel, gathered to discuss their experiences, challenges encountered, and successes achieved during the iteration. The focus was on fostering open communication, collaboration, and continuous learning within the team, and the following aspects of the development process were discussed

System Functionality: Reviewing the implemented features and functionalities of the web-based transcript and result computation system to assess their alignment with stakeholder requirements and expectations. This included examining whether the system accurately computes results, generates transcripts, and meets regulatory requirements.

User Experience: Evaluating the usability and accessibility of the system from the perspective of end-users, including students, faculty, and administrative staff. This involved gathering feedback on the user interface, navigation, and overall satisfaction with the system's usability.

Development Practices: Assessing the effectiveness of Agile practices employed during the iteration, such as sprint planning, daily stand-up meetings, and backlog refinement. This involved identifying any bottlenecks, inefficiencies, or areas for improvement in the development process.

Team Dynamics: Reflecting on the collaboration, communication, and teamwork within the project team. This included discussing any challenges faced in coordinating tasks, resolving conflicts, or maintaining motivation among team members.

Process Improvement: Identifying actionable insights and improvement opportunities to enhance future iterations of the development process. This involved proposing changes to Agile practices, refining user stories, adjusting sprint goals, or addressing technical debt.

By conducting iterative reviews and retrospectives, the project team could continuously assess progress, adapt to changing requirements, and foster a culture of continuous improvement throughout the development of the web-based transcript and result computation system. This iterative feedback loop helped ensure that the system evolved iteratively, meeting the needs of stakeholders effectively and delivering value incrementally.

Case Study

Federal Polytechnic, Ukana, established in 2014, is an institution committed to providing practical and industry-relevant education in various technical and management fields. Recognizing the need for a more efficient and reliable system for result computation and transcript processing, the researchers embarked on a project to develop a web-based system for Federal Polytechnic Ukana using agile methodology. The primary goal was to replace the manual, time-consuming processes with a streamlined, automated solution that would enhance accuracy, reduce processing time, and improve accessibility for students and administrative staff.
The project was initiated in response to growing demands for a more efficient academic records management system. Key stakeholders included:

i. **Students:** The primary users of the system, requiring timely access to their academic results and transcripts.
ii. **Academic Staff:** Instructors responsible for inputting and verifying student grades.
iii. **Administrative Staff:** Personnel tasked with processing and issuing transcripts.
iv. **ICT Department**: The team responsible for developing, maintaining, and updating the system.
v. **Institutional Leadership**: The administrative body overseeing the project, providing strategic direction, and ensuring alignment with institutional goals.

Progression of the Agile Development Process in the case study was as follows:

i. **Initiation and Planning**: The project began with a series of meetings involving all stakeholders to gather requirements and set clear objectives. A product backlog was created, listing all features and functionalities needed for the system.

ii. **Sprint Planning**: The development process was divided into iterative sprints, each lasting two weeks. During sprint planning meetings, the team selected high-priority items from the product backlog to be addressed in the upcoming sprint.
iii. **Development and Testing**: Agile's iterative approach allowed the team to focus on developing small, functional parts of the system. Each sprint cycle included phases of coding, testing, and review. Continuous integration and testing ensured that new features were consistently evaluated and refined.
iv. **Review and Retrospective**: At the end of each sprint, a review meeting was held to demonstrate the progress to stakeholders and gather feedback. This was followed by a retrospective meeting where the team discussed what went well, what could be improved, and how to enhance the next sprint.
v. **Deployment and Feedback**: Once a minimal viable product (MVP) was ready, it was deployed in a controlled environment for real-world testing. Feedback from actual users was crucial in identifying further refinements and additional features.

Challenges were faced and resolved. They are as follows:

i. **Requirement Ambiguity:** Initially, some requirements were not clearly defined, leading to misaligned expectations. This was addressed by improving communication channels and involving stakeholders more closely in the requirement-gathering process.
ii. **Technical Debt**: As the project progressed, some technical debt accumulated due to the fast-paced nature of development. Regular refactoring sessions were introduced to address and manage this debt.
iii. **User Resistance:** Some users were resistant to change, preferring the old manual processes. To mitigate this, comprehensive training sessions were conducted, and user-friendly interfaces were developed to ease the transition.
iv. **Integration Issues:** Integrating the new system with existing institutional software posed challenges. A dedicated integration team was formed to focus on ensuring compatibility and seamless data exchange between systems.

The implementation of the web-based result computation and transcript processing system at Federal Polytechnic, Ukana, yielded significant positive outcomes:

i. **Improved Efficiency:** The new system drastically reduced the time required for result computation and transcript processing from weeks to days.
ii. **Enhanced Accuracy**: Automation minimized human errors associated with manual data entry and calculations.
iii. **Better Accessibility:** Students and staff could access the system remotely, providing greater flexibility and convenience.

iv. **User Satisfaction:** Training and iterative improvements based on user feedback led to high user satisfaction and acceptance of the new system.
v. **Scalability:** The system was designed with scalability in mind, allowing for easy adaptation to future growth and additional functionalities.

In general, the adoption of agile methodology in developing this system not only met the immediate needs of Federal Polytechnic, Ukana but also set a precedent for future digital transformation projects within the institution.

RESULTS

The development of the web-based result computation and transcript processing system at Federal Polytechnic, Ukana, provided several insightful findings. Through iterative cycles of development and continuous feedback from stakeholders, the following key results were observed:

i. Efficiency Gains: The time required for processing results and transcripts decreased significantly. Pre-implementation, the process could take several weeks, but post-implementation, the average processing time was reduced to a few days.
ii. User Satisfaction: Surveys conducted among students, academic staff, and administrative personnel indicated a high level of satisfaction with the new system. Over 85% of respondents reported the system as user-friendly and efficient.
iii. Error Reduction: The frequency of errors in grade computation and transcript generation was notably reduced, with the new system demonstrating a 95% accuracy rate compared to 80% in the manual process.
iv. Accessibility: The web-based nature of the system allowed users to access it remotely, which was particularly beneficial during periods when physical access to the campus was restricted.

The analysis focused on key performance indicators (KPIs) such as processing time, accuracy, user satisfaction, and system accessibility.

i. **Processing Time**: Before implementation, the average time for result computation and transcript processing was around 14-21 days. After implementation, this was reduced to an average of 3-5 days. This represents an improvement of approximately 75-85%.
ii. **Accuracy**: The reduction in errors was measured by comparing the number of incorrect transcripts issued before and after the implementation. The new system's error rate was reduced to less than 5%, indicating a significant improvement in accuracy.

iii. **User Satisfaction:** Based on survey data, user satisfaction improved from a baseline of 60% satisfaction with the old system to over 85% with the new system. Key areas of improvement noted were ease of use, speed, and reliability.

The interpretation of this data suggests that the implementation of the web-based system not only met but exceeded expectations in several areas, demonstrating the effectiveness of the agile methodology in delivering a high-quality solution tailored to the users' needs.

CONCLUSION

In conclusion, the development of a web-based transcript and result computation system was made possible by the successful application of agile methodology. Agile's iterative process allowed for constant feedback and adjustment, guaranteeing that the finished product closely matched institutional and user requirements. Improved teamwork, enhanced problem-solving skills, and the capacity to adapt to changes without causing major disruption were among the main advantages noted. The system's successful implementation increased operational effectiveness and proved the scalability and resilience of agile software development methodologies. The aforementioned project highlights the significance of implementing agile methodologies for projects that necessitate adaptability, continuous stakeholder engagement, and prompt response to changing requirements. This case study can provide important insights for future projects at Polytechnic Ukana and other similar institutions, demonstrating the viability and efficacy of agile development as a methodology for creating intricate, user-centered software solutions.

Declaration of Conflicting Interests
The author(s) declared no potential conflicts of interest concerning the research, authorship, and/or publication of this article.

Funding
The following financial assistance was revealed by the author(s) for the research, authoring, and/or publishing of this article: The Tertiary Education Trust Fund (TETFUND) provided funding for this study via the Institution Based Research Fund (IBRF).

REFERENCES

Abah, J. A., Honmane, O., Age, T. J., &Ogbule, S. O. (2022). Design of Single-User-Mode Computer-Based Examination System for Senior Secondary Schools in Onitsha North Local Government Area of Anambra State, Nigeria. VillageMath Educational Review (VER), 3(1).

Al-Saqqa, S., Sawalha, S., & AbdelNabi, H. (2020). Agile software development: Methodologies and trends. International Journal of Interactive Mobile Technologies, 14(11).

Akputu, O. K., Attai, K. F., Usoro, A., & Abiodun, A. O. (2020). Policy-Driven Academic Result Computation and Transcription: Ritman University Case. Policy, 9(1).

Amollo, B. O. (2021). Evaluation of School Management Information Systems (Smis) in Secondary Schools in Siaya County (Doctoral dissertation, JOOUST).

Arcos-Medina, G., & Mauricio, D. (2019). Aspects of software quality applied to the process of agile software development: a systematic literature review. International Journal of System Assurance Engineering and Management, 10(5), 867-897

Bello, U. A., Magaji, M. M., Jibril, A. Z., & Haruna, K.(2023). Design and Implementation of a Software Result Processing and Transcript Generation, 5(6).

Damasevicius, R., Maskeliunas, R., & Leon, M. (2019, October). Development of Online Clearance System for an Educational Institution. In In Applied Informatics: Second International Conference, ICAI 2019 (p. 327).

Edison, H., Wang, X., & Conboy, K. (2021). Comparing methods for large-scale agile software development: A systematic literature review. IEEE Transactions on Software Engineering, 48(8), 2709-2731.

Ekanem, A. J., Ozuomba, S., & Jimoh, A. J. (2017). Development of Students Result Management System: A case study of University of Uyo. Mathematical and Software Engineering, 3(1), 26-42.

Grass, A., Backmann, J., &Hoegl, M. (2020). From empowerment dynamics to team adaptability: Exploring and conceptualizing the continuous agile team innovation process. Journal of Product Innovation Management, 37(4), 324-351.

Gupta, A., Poels, G., & Bera, P. (2022). Using conceptual models in agile software development: a possible solution to requirements engineering challenges in agile projects. IEEE Access, 10, 119745-119766.

Harsha, H. K. (2016). Design and Development of School Management System with Database Encryption and Decryption.

Harrison, H., Birks, M., Franklin, R., & Mills, J. (2017, January). Case study research: Foundations and methodological orientations. In Forum qualitative Sozialforschung/Forum: qualitative social research (Vol. 18, No. 1).

Heeager, L. T., & Nielsen, P. A. (2018). A conceptual model of agile software development in a safety-critical context: A systematic literature review. Information and Software Technology, 103, 22-39.

Kuhrmann, M., Tell, P., Hebig, R., Klünder, J., Münch, J., Linssen, O., ... & Richardson, I. (2021). What makes agile software development agile?. IEEE transactions on software engineering, 48(9), 3523-3539.

Iivari, J. (2021). A paradox lens to systems development projects: The case of the agile software development.
Communications of the Association for Information Systems, 49(1), 4.

Layton, M. C., Ostermiller, S. J., & Kynaston, D. J. (2020). Agile project management for dummies. John Wiley
& Sons.

Mishra, A., & Alzoubi, Y. I. (2023). Structured software development versus agile software development: a
comparative analysis. International Journal of System Assurance Engineering and Management, 14(4), 1504-1522.

Mukherjee, S. K. (2020). The role of leadership in adopting agile practices/agile release management to reduce
product time-to-market: A case study research in the case company.

Nazir, S., Price, B., Surendra, N. C., & Kopp, K. (2022). Adapting agile development practices for hyper-agile environments: lessons learned from a COVID-19 emergency response research project. Information Technology and Management, 23(3), 193-211.

Ogundipe, D. O., Odejide, O. A., &Edunjobi, T. E. (2024). Agile methodologies in digital banking: Theoretical
underpinnings and implications for custom satisfaction. Open Access Research Journal of Science and Technology, 10(02), 021-030.

Onibere, E. I. (2013). Fuzzy logic modelling of a performance evaluation system for academicprogrammes in Nigeria higher education. Data Management and Security: Applications in Medicine, Sciences and Engineering, 45, 113.

Osunade, O., Ayinla, I. B., &Aduroja, O. O. (2019). Design and Implementation of aCentralized University Result Processing and Transcript System: Case study of University of Ibadan. International Journal of Computing Sciences Research, 2(3), 89-101.

Okikiola, M. A., & Samuel, F. (2016). Optimizing the processing of results and generation of transcripts in Nigerian universities through the implementation of a friendly and reliable web platform. Imperial Journal of Interdisciplinary Research, 2, 12.

Olamide, O. O., & Joshua, A. O. (2012). Design and simulation of an SMS driven microcontroller for home automation using proteus software. Journal Computer Science Department. University of Lagos.

Ouriques, R., Wnuk, K., Gorschek, T., & Svensson, R. B. (2023). The role of knowledge-based resources in
Agile Software Development contexts. Journal of Systems and Software, 197, 111572.

Ozkan, N., Gök, M. Ş., & Köse, B. Ö. (2020, September). Towards a better understanding of agile mindset by using principles of agile methods. In 2020 15th Conference on Computer Science and Information Systems (FedCSIS) (pp. 721-730). IEEE.

Pargaonkar, S. (2023). A Comprehensive Research Analysis of Software Development Life Cycle (SDLC)
Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering. International Journal of Scientific and Research Publications (IJSRP), 13(08).

Perkusich, M., e Silva, L. C., Costa, A., Ramos, F., Saraiva, R., Freire, A., ... &Perkusich, A. (2020). Intelligent
software engineering in the context of agile software development: A systematic literature review. Information and Software Technology, 119, 106241.

Rajegowda, P. M. (2017). Research on the Implementation of a Web Based Institutional Support System for Middle East College: Proof of Concept Study. Journal of Student Research

Rindell, K., Ruohonen, J., Holvitie, J., Hyrynsalmi, S., & Leppänen, V. (2021). Security in agile software development: A practitioner survey. Information and Software Technology, 131, 106488.

Sarangee, K., Schmidt, J. B., Srinath, P. B., & Wallace, A. (2022). Agile transformation in dynamic, high-technology markets: Drivers, inhibitors, and execution. Industrial Marketing Management, 102, 24-34.

Schtein, I. A. (2018). Management Strategies for Adopting Agile Methods of Software Development in
Distributed Teams. Walden University.

Strode, D., Dingsøyr, T., &Lindsjorn, Y. (2022). A teamwork effectiveness model for agile software
development. Empirical Software Engineering, 27(2), 56.

Tam, C., da Costa Moura, E. J., Oliveira, T., &Varajão, J. (2020). The factors influencing the success of on-
going agile software development projects. International Journal of Project Management, 38(3), 165-176.

Tyagi, S., Sibal, R., & Suri, B. (2022). Empirically developed framework for building trust in distributed agile
teams. Information and Software Technology, 145, 106828.

Yari, L. V., Cedeño, C. A., & Alarcón, J. T. (2017). Aplicación para la gestión de información del investigador MONOIL, proyecto de la Universidad de Guayaquil [Application for the information management of MONOIL researcher, project of Guayaquil's University]. International Journal of Innovation and Applied Studies, 20(4), 1160-1168.

Zeng, Y., &Boontasorn, N. (2022). Student management system based on intelligent technology of internet of things. In International Conference on Cognitive based Information Processing and Applications (CIPA 2021) Volume 1 (pp. 866-876). Springer Singapore.s
.