

Innovative Software Development and Project Management Framework for Technology Startups

Song-Kyoo Kim

Faculty of Applied Sciences, Macao Polytechnic University, Macau, SAR, China
Email: amang@mpu.edu.mo

Abstract—This paper introduces a new process that integrates inventive problem-solving methods into modern software development. The central research question addresses how tech startups can enhance their software development processes with minimal project management expertise. The Systematic Innovation Mounted Software Development Process, which blends Agile and Systematic Innovation, offers an alternative framework to facilitate idea generation in software products. This intuitive project management framework empowers technology-driven companies to manage their projects more effectively. The study aims to provide guidelines for entrepreneurs to manage projects successfully. By collaborating with the existing Agile model, the Systematic Innovation model fosters creativity and innovative problem-solving. Ultimately, this new software development process and its techniques have the potential to significantly impact the software industry, particularly for startups, as they alleviate managerial burdens and allow companies to focus on their core technologies.

Keywords—software development process, systematic innovation, agile process, software process improvement, Theory of Inventive Problem Solving (TRIZ), technology startup, project management, lean software development

I. INTRODUCTION

In the modern era, software complexity is essential for fostering innovation. The development processes have evolved to encompass a vast array of software applications. This paper presents an enhanced software development framework tailored for the recent software industry, recognized as one of the fastest-growing sectors globally. Furthermore, software development startup companies within this domain serve as critical economic foundations for many countries worldwide [1]. Consequently, efforts to improve the software development process, while maintaining minimal quality within a short product lifecycle and low costs, have become a primary focus. A software process framework is essential for producing high-quality software, despite the well-established understanding that no universal software development method exists that is suitable for all scenarios [2, 3]. Distinct cultures, characteristics, and markets are maintained by each software development

company for its products. Consequently, the effective application of a single software development process to all predefined development processes is not feasible. Customization of development processes is necessitated by specific organizational protocols, needs, and contexts. Furthermore, inherent conflicts among delivery time, cost, and quality exert a significant influence on the software development process [2].

The contemporary software process continues to prioritize maintaining a certain level of quality over mere efficiency. Research suggests that addressing issues after a software product launch can incur costs up to one hundred times greater than rectifying the problems during the initial phase. Consequently, continuous refinement and improvement of software development processes are essential. Software Process Improvement (SPI) serves as a practical solution for software enhancement, representing a systematic approach to improving development process performance through modifications to existing processes [4]. SPI also involves implementing changes to the development process to achieve specific goals, such as increasing development speed, enhancing product quality, or reducing costs. A leader in SPI must understand the relevant methodologies and tools suited to the current context and be aware of the state of practice and ongoing process improvement initiatives. SPI has been applied across various areas within the software development sector. Software Process Improvement (SPI) can be applied to highly integrative projects, including electronic commerce system development [5] and machine learning application development [6], to mitigate the risk of project failure. Implementation of SPI may involve incorporating new activities into the development process while simultaneously removing certain existing ones. Some researchers offer guidelines for engaging in software process improvement for small and medium-sized companies by analyzing critical SPI requirements. Although research exists on SPI models aimed at small and medium-sized companies, dedicated SPI models for technology startups are lacking. Furthermore, existing SPI models primarily cater to large companies and are unsuitable for smaller organizations (fewer than ten employees) due to their expense and complexity [7]. Recently, Crowne [3] proposed modified SPI models

applicable to both small and large companies through a competitive advancement strategy. Lean principles from the original Lean manufacturing process have also been integrated into software development, making it suitable for technology startups developing software products [8]. Sulayman *et al.* [7] indicates that SPI implementation is an effective approach for enhancing the software development process. In the current environment of rapid change and development, the software industry increasingly demands a more flexible development process to accommodate changes during projects. While software development often favors the Lean process, recent trends indicate that products must be more innovative and appealing to consumers. The uniqueness of the software sector continuously shifts. Consequently, the development process must evolve to enhance flexibility and foster innovation [9]. Tech startups can enhance their software development processes with minimal project management expertise through the adoption of the Systematic Innovation Mounted Software Development Process (SIM-Process). This framework integrates Agile and Systematic Innovation, providing an intuitive structure that facilitates idea generation and effective project management for technology-driven companies.

This paper introduces newly integrated frameworks that utilize systematic innovation for software development and project management, which have not been previously presented. The lean and the systematic innovation are combined to integrate with existing agile development processes, imposes no side effects or overload. A combination of systematic innovation with a software development process suitable for dynamic development environments is presented in this research. The Systematic Innovation Mounted software development process (SIM-Process) constitutes an innovative framework designed to enhance creativity and problem-solving in software development. This newly proposed development process is conceptually similar with the Agile and the Lean process but it is designed for the innovative activities including the story producing and the user experience design. The primary pillars are the integration of systematic innovation methodologies with a suitable software development process, such as Agile. This foundational combination underpins the SIM-Process, designed to foster creativity and problem-solving within dynamic development contexts, particularly for startups.

II. LITERATURE REVIEW

The software development process is often considered as a subset of a Systems Development Lifecycle (SDLC) for developing software products. There are several models for processes that describe approaches to a variety of tasks and activities that take place during the procedure. Generally, the system development lifecycle is a broader term and the software development process is a more specific term. The software lifecycle typically includes the following steps: requirements, analysis, design,

construction (or coding), testing (validation), installation, operation, and maintenance [10]. The international standard for software lifecycle (ISO, 2017) has mentioned that many software development processes fit the spiral lifecycle model from the system development lifecycle model. Agile software development is also adapting the spiral cycles (recursive, iterative) for enhancing the development process. Agile software development, based on iterative and incremental development, is practical for startup companies. Agile development processes were introduced in the 1990s, to minimize a process bureaucracy by removing unnecessary milestones because of the administrative workloads [11]. Agile software development is targeted to deliver a software product quickly to consumers, who could also propose new business requirements into products.

This philosophy behind agile methods is reflected in the Agile manifesto which values individuals and interactions, working software, customer collaboration and responding to change [11]. Central to contemporary market realities is the philosophy of Agile Software Development. The emergence of agile software processes addresses challenges posed by rapidly changing and unpredictable markets [12]. Exploration of each value facilitates understanding of the agile process philosophy and activities for applying the philosophy to enhance software development, aligning with the latest volatile markets for startups [12]. Feature Driven Development [13], Scrum [14], Extreme Programming, Crystal, Dynamic System Development Method [15] and Adaptive Software Development [16] are common software development methodologies that are aligned with the values of Agile Software Development [17]. From an organizational perspective, agile development activities are considered suitable for teams that are small, physically co-located, highly dedicated, and characterized by strong collaborative interactions [18, 19].

In general, agile development is regarded as the extreme opposite of Waterfall development. In the Agile process, a series of these processes are repeated, known as reputation development [20]. There are some variances of the Agile development processes, which usually starts with Planning phase and defines Requirements, Design, Implementation, Testing (and Integration) and Evaluation (see Fig. 1) phases. It has several recursive cycles (iterations) between Design and Testing and Integration phases. Once all requirements are determined as actual implementation during the Evaluation phase, the project is completed (Product phase). The recursive cycle (Design; Implementation; Testing and Integration) is a minimum set of the development cycle, executed daily. A Daily-based Scrum framework is commonly applied for this recursive cycle [14]. Scrum functions as a flexible and holistic development strategy, representing an iterative and incremental agile software development framework wherein a development team operates cohesively to achieve a shared objective [14].

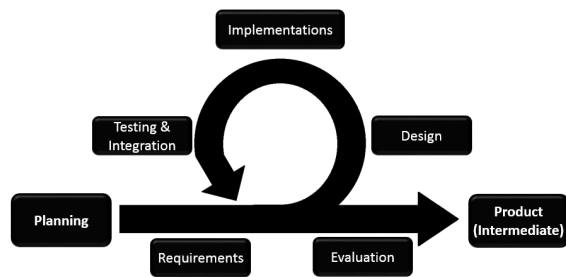


Fig. 1. Agile software development process.

Systematic Innovations (SI) is a structured process and the set of practical tools for new idea generations and application to technical problems, including software implementation issues [21]. The tools of Systematic Innovation have been widely used for technical breakthroughs and system improvements [22]. In general, problem-solving and innovation processes which include 6-sigma, Lean Thinking, IDEO process, ARIZ (Algorithm of Inventive Problem-solving) and SI usually contain one or several task blocks (also called phases) to generate and implement new ideas and solutions. This phase approach of innovation method provides check points to use inventive problem-solving tools more effectively. An 8-step phase approach is widely used in the systematic innovation process [23], however, a 3-step phase process is used for mounting with the existing Agile processes in this research. This 3-step process for Systematic Innovation has recently been introduced for those who use complicated problem-solving tools with ease. Systematic Innovation is originally targeted to solve engineering problems but the method has expanded to various areas included in new software development [24–27]. This 3-step process (see Fig. 2) is simple and easy to use, even for beginners [28].

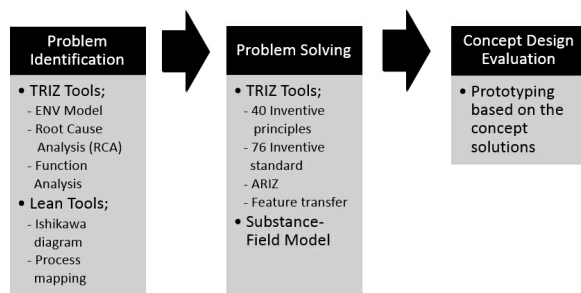


Fig. 2. Systematic innovation process [28].

The Agile process introduces the idea of simplicity. The more effort needed to find required information, the more effort is needed to keep the information up to date [29]. Agile software development fits well in terms of flexibility to reflect the requirements from astute customers. The innovative properties of software products have become a mandatory factor for success in business under uncertain circumstances. Software companies need to think differently to generate new ideas that appeal to customers. Adapting distinct innovation processes that can be widely used by world-leading companies could be one way to move for delivering innovation into products. DeepDive

(IDEO), Lead User Research (3M) and Design Thinking (Apple) are well known innovation processes that have been adapted and are being used by various companies. Integrating an innovation process into an existing software development process is not an easy task. The innovation process must be lean enough to avoid associated side effects, confusion, and overload during the integration. Consequently, the innovation processes mentioned above might be not applicable for software startup companies due to characteristics of the companies, which are small and lean.

The project management role is vital and project management for software development is also a critical factor for success. Project management is the process and activity of planning, organizing, motivating, and controlling resources, procedures, and protocols to achieve specific goals in scientific or daily problems [30]. A project is a temporary endeavor designed to produce a unique product, service or result with a defined beginning and end (usually time-constrained, and often constrained by funding or deliverables) [29], undertaken to meet unique goals and objectives, typically to bring about beneficial change or added value (see Fig. 3) [31].

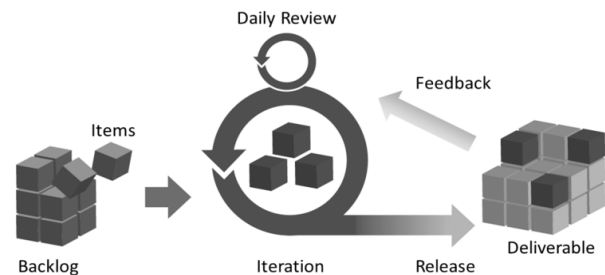


Fig. 3. Agile project management diagram [32].

III. PROPOSED METHODS FOR SOFTWARE DEVELOPMENT

A. SIM Software Development Process

Systematic Innovation can be adapted to the agile development process to implement the software innovatively. SIM-Process is a framework to help developers to generate more innovative ideas systematically. Some steps in the software development are required to create something new and it requires to use the innovative problem-solving skills. The SIM-Process could be adopted for these types of steps including the (story) idea generation and the graphic design especially for mobile game developments.

The general procedure of a SIM-Process is like an Agile software development processes except for some phases in the agile process. Requirements, Design, and Implementation phases in the agile process are replaced with Problem Identification, Problem-solving, and Concept Design phases in the SI method (see Fig. 4). Sometimes, software development requires to be generated new idea innovatively. The SIM-Process is suitable for developing the new idea in the different perspective and it is really handy to be attached with atypical software development process like the Agile

process. Beside of the Systematic Innovation, other components of the software development are the same as an existing software process. The SIM-Process could be adopted under the exact same setups including the setups of human resources, time managements for daily scrum, the release and sprint plans, and development roadmap.

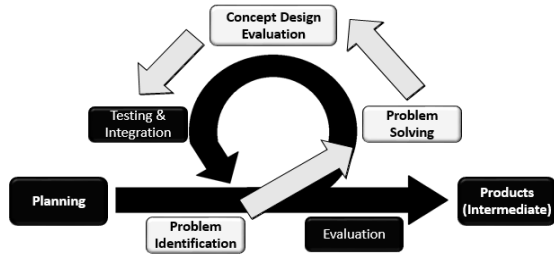


Fig. 4. Systematic Innovation Mounted software development process.

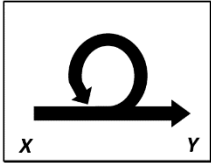
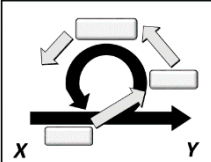
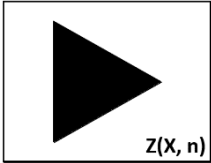
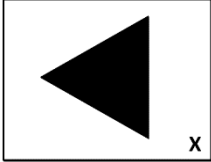
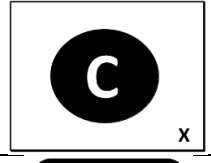
B. Intuitive Project Management Framework

The new framework in this paper suggests modules for project management, and the whole project consists of building blocks as unit modules.

Unlike existing project management, the resources of

each task such as duration of the task, number of human resources and costs are defined into the unit module. So, the project manager can determine workloads by counting number of the unit modules in the project diagram. Single unit module is either an Agile process which contains a whole cycle of the development process or a SIM-Process which contains a whole process for idea generations. The resource usage (duration, human resource, and cost) of one SIM-Process module is assumed to be the same as what an applied agile process module is (see Fig. 4). For instance, the project manager determines one agile process module of five daily iterations as one unit, which means that single unit module is completed within one week (assume one week as five working days; $\alpha = 5$). Sometimes the project manager needs to split the project into several sub-projects (splitter) and vice versa (integrator). The project manager may need to reconsider the whole project even if in the middle of the development (checker) phase because of changing requirements. This set of new project management planning requirements is built, based on various circumstances of the software development process. Each module is one single block and a project manager can build up the software development process by adding these modules (see Table I).

TABLE I. MODULE SET OF SOFTWARE DEVELOPMENT PROJECT

Module	Symbol	Text Symbol	Variable	Descriptions
Agile Process		#	X: one recursive (iteration) cycle Y: process duration	Module of Agile software development process
SI Mounted Process		*	X: one recursive (iteration) cycle Y: process duration	Module of Systematic Innovation mounted software development process
Integrator		}	Z: integration duration depend on nodes	Module for Integration of the development modules
Splitter		{	X: split duration	Module for splitting a project into parallel sub-projects
Checker		C	X: check durations	Checksum and reconsidering the whole project (the project could be terminated).
Start/End		@	N/A	Start and end of the project

IV. COM2US: ACTION PUZZLE FAMILY

Com2uS has been a leader of the mobile game industry since its inception in 1998 and the company built its reputation as the number one mobile games provider in South Korea [26]. Com2uS was a successful developer of many premium titles. The name of Com2uS remains as the leader in the mobile game industry even though the company merged with Gamevil in 2013.

Action Puzzle Family (APF) is in the form of delightful classical easy puzzle games. This game title has been known as one of the popular Freemium casual games, which has more than six million users of eight puzzle games [26]. The goal of the game is collecting all puzzle pieces to move the family into a house with wacky features and each one of the eight eccentric puzzle games has a different theme. The APF game project started in June, 2006 with 10 members, including a producer and software developers that lasted until the end of the project, August 2007. The APF was developed by a small group (less than

ten people in total) and the process was flexible than the general Com2uS development process. According to Com2uS case research [26], idea generation and graphic design phases are required for generating the new and fresh ideas; the SIM-Process module might be suitable for this process. In the general Com2uS development process, bug-fix tasks before starting alpha and beta testing phases are mandatory. The APF project could be described by using the intuitive project management framework more effectively and two steps are required for creative problem-solving skills. So, the SIM-Process is applied into these two steps which are the Idea Generation and the Graphic Design (see Fig. 5).

The APF producer could determine the unit of modules and four parameters (X , Y , Z , and α) during the project initiation. The values of the parameters (X , Y , Z , and α) are as follows:

- $X = 1$ [week]
- $Y = 1$ [month]
- $\alpha = 4$ [weeks/month].

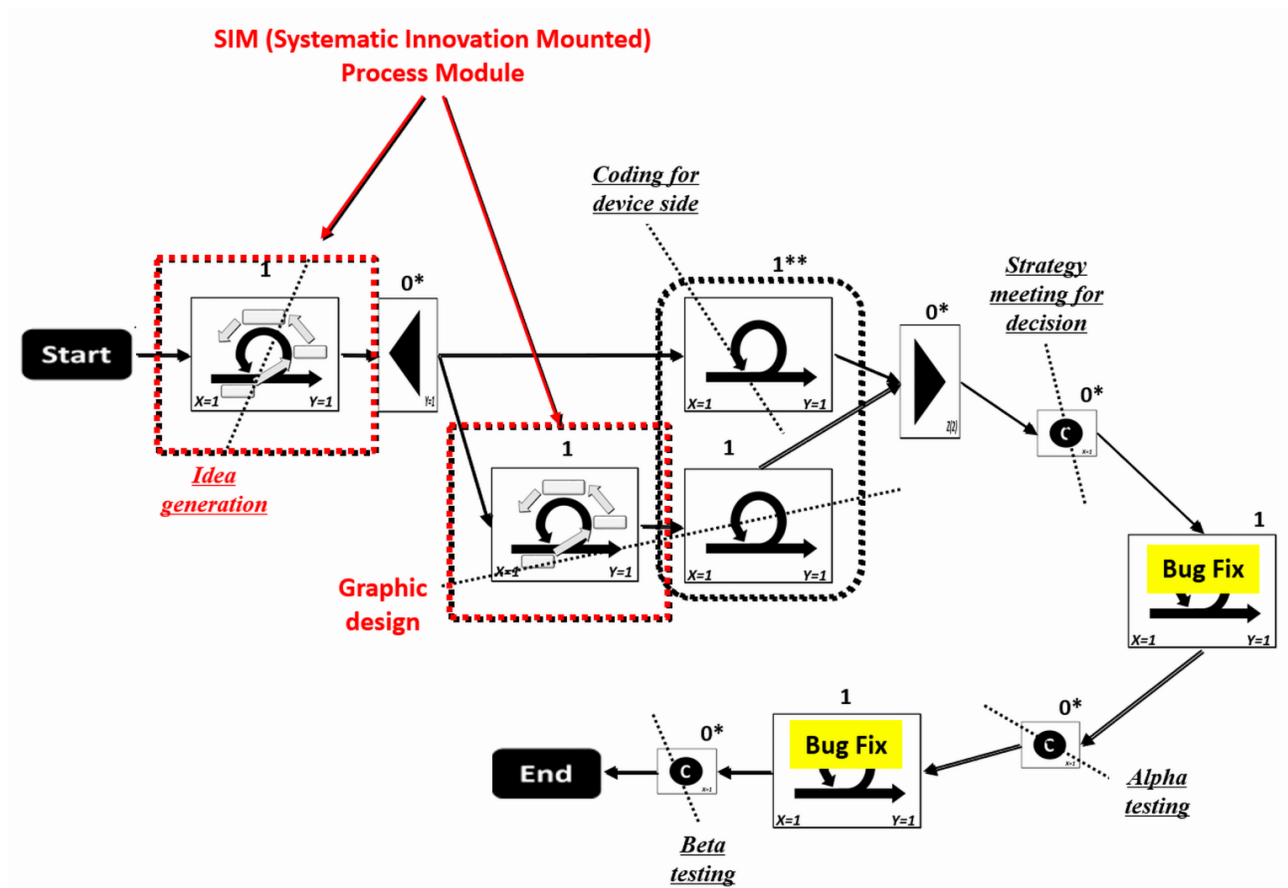


Fig. 5. Intuitive APF project planning diagram. *) Based on the pre-defined functions (Table II); **) Assume the human resources are not limited.

The duration of the *integrator* before the *strategic decision meeting* is calculated as follows:

$$Z(X, \alpha, n) = Z(1, 4, 2) = \text{Round} \left(1 \cdot \frac{2 \cdot (2 - 1)}{2 \cdot 4} \right) = 0$$

and duration of the integration completion is less than one month (0^* ; 1 week). Three *checkers* of the project include, the moment to check if the APF project is worthy enough to be completed based on the current plan (the project might be dropped at this moment). The APF project planning was also assumed that each member in the project was a highly-skilled programmer or engineer and each task may have different members. The APF project diagram in

Fig. 4 instantly shows that the project would takes 6 months and 1 week with 14 developers (2 members 7 exclusive tasks) by simply counting the number of 1 s and 0*s (5 months and 5 weeks = $1+0*+1+1+0*+0*+1+0*+1+0*$). The duration of all modules is calculated based on the predefined functions in Table II.

TABLE II. PARAMETERS AND PRE-DEFINED FUNCTIONS FOR THE MODULE SET

Variable	Meaning	Unit	Default Function
X	Duration of recursive period (Unit period)	[day], [week]	N/A, constant
Y	Duration for completion of one module process	[week], [month], [year]	$Y(X, \alpha) = \text{Round}(\frac{X}{\alpha})$
Z	Duration of integrating the module	[week], [month], [year]	$Z(X, \alpha, n) = \text{Round}(X \cdot \frac{n(n-1)}{2\alpha})$
n	Number of nodes for integration	N/A	N/A
α	Duration unit between the cycle and module (i.e., number of cycles for one module process).	N/A	$\alpha = \frac{5[\text{day}]}{1[\text{week}]} = 5 \text{ or } \alpha = \frac{4[\text{week}]}{1[\text{month}]} = 4$

V. CONCLUSION

The research highlights the significance of the Systematic Innovation Mounted Process (SIM-Process) in software development, particularly for technology-driven startups. The SIM-Process provides a structured framework that enhances creativity and fosters innovative solutions within the fast-paced environment of software development. By focusing on problem identification, problem-solving, and concept design, this process aligns well with agile methodologies, ensuring flexibility and responsiveness to market demands. The project planning diagram drawn by the software development process building blocks in this framework shows the project managing information intuitively. However, the research also identifies a critical limitation: the necessity for high-skilled team members who can navigate both the SIM-Process and Agile methodologies effectively. As small startups may lack extensive resources, the essential for team members to possess adequate knowledge and skills to leverage these processes successfully. Thus, holistic training and skill development in systematic innovation are vital for the continuous growth and sustainability of technology startups in today's competitive landscape. Even regarding the limitations, this practical method could impact the current software development industry and provide the powerful weapons in the development and management capabilities especially for startup companies.

CONFLICT OF INTEREST

The author declares no conflict of interest.

FUNDING

This work was supported in part by the Macao Polytechnic University (MPU), under Grant RP/FCA-05/2024.

The APF project case has been used to demonstrate how the SIM-Process could be applied into the mobile game software. It is noted that the SIM-Process could be more practical when it is applied into simpler software development. The next case is a SIM software development application for another software development process which is simpler than the APF project.

ACKNOWLEDGMENT

This paper was revised using AI/ML-assisted tools. Special thanks to the reviewers who provide valuable advice for improving this paper.

REFERENCES

- [1] I. Sommerville, *Software Engineering*, 10th ed. Essex, UK: Pearson Education, 2016, pp. 1–808.
- [2] R. S. Pressman and B. R. Maxim, *Software Engineering a Practitioner's Approach*, 9th ed. New York, USA: McGraw Hill, 2020.
- [3] M. Crowne, "Why software product startups fail and what to do about it. Evolution of software product development in startup companies," in *Proc. International Engineering Management Conference*, 2002, vol. 1, pp. 338–343.
- [4] S. Alonso, M. Kalinowski *et al.*, "A systematic mapping study and practitioner insights on the use of software engineering practices to develop MVPs," *Information and Software Technology*, vol. 156, 107144, 2023.
- [5] V. Gruhn and L. Schöpe, "Software processes for the development of electronic commerce systems," *Information and Software Technology*, vol. 44, no. 14, pp. 891–901, 2002.
- [6] A. Zorto, S. Lansdell *et al.*, "Machine learning-based techniques for assessing critical factors for European tick abundance," *International Journal of Computer Theory and Engineering*, vol. 17, no. 1, pp. 13–20, 2025.
- [7] M. Sulayman *et al.*, "Towards a theoretical framework of SPI success factors for small and medium web companies," *Information and Software Technology*, vol. 56, no. 7, pp. 807–820, 2014.
- [8] M. Poppendieck and T. Poppendieck, *Lean Software Development: An Agile Toolkit*, Boston, USA: Addison-Wesley, 2003.
- [9] S. K. Kim, "Enhanced user experience design based on user behavior data by using theory of inventive problem-solving," in *Proc. 2010 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2010, pp. 2076–2079.
- [10] S. Cohen *et al.*, "A software system development life cycle model for improved stakeholders' communication and collaboration," *Int. Journal of Computers, Communications & Control*, vol. 5, no. 1, pp. 20–41, 2010.
- [11] S. K. Kim, "Strategic decision spectrum for software engineering," in *Proc. 2023 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2023, pp. 1708–1712.

- [12] G. C. L. Leal, R. Prikladnicki *et al.*, "Practices and tools for software start-ups," *IEEE Software*, vol. 37, no. 1, pp. 72–77, 2020.
- [13] J. M. Felsing and S. R. Palmer, "A practical guide to feature-driven development," *IEEE Software*, vol. 7, pp. 67–72, 2002.
- [14] M. Cohn, *Succeeding with Agile: Software Development Using Scrum*, New York, NY: Addison-Wesley Professional, 2010.
- [15] J. Stapleton, *DSDM: Business Focused Development*, New York, NY: Addison-Wesley Professional, 2003.
- [16] J. A. Highsmith, *Adaptive Software Development: A Collaborative Approach to Managing Complex System*, New York, NY: Dorset House, 2000.
- [17] R. H. AL-Taani and R. Razali, "Prioritizing requirements in agile development: A conceptual framework," *Procedia Technology*, vol. 11, pp. 733–739, 2013.
- [18] B. Boehm, "Get ready for agile methods, with care," *IEEE Computer*, vol. 35, no. 1, pp. 64–69, 2002.
- [19] P. D. O. Santos and M. M. de Carvalho, "Exploring the challenges and benefits for scaling agile project management to large projects: A review," *Requirements Engineering*, vol. 27, no. 1, pp. 117–134, 2022.
- [20] K. Furugaki *et al.*, "Innovation in software development process by introducing toyota production system," *Fujitsu Science Technology Journal*, vol. 43, no. 1, pp. 143–150, 2007.
- [21] J. Terninko *et al.*, *Systematic Innovation: An Introduction to Theory of Inventing Problem-Solving*, Boca Raton, FL: CRC Press, 1998.
- [22] D. Petkovic *et al.*, "Application of the TRIZ creativity enhancement approach to the design of a passively adaptive compliant robotic gripper," *Assembly Automation*, vol. 33, no. 3, pp. 231–239, 2013.
- [23] C2C Solution. (2016). Systematic innovation intro. [Online]. Available: <http://c2c-solutions.com/>
- [24] S. -K. Kim, "mobile terminal and message transmitting/receiving method for adaptive converged IP messaging," U.S. Patent 8 050 269 B2, 2011.
- [25] S. K. Kim, "Effective Wi-Fi setting user experience design by using systematic innovation method," in *Proc. 2012 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2012, pp. 643–646.
- [26] S. K. Kim, "Com2uS mobile game development," *Journal of Information Technology Case and Application Research*, vol. 16, no. 3–4, pp. 155–167, 2014.
- [27] S. K. Kim, "Systematic innovation practice for enhanced mobile advertisement protocol design," *International Journal of Social Science and Humanity*, vol. 6, no. 3, pp. 235–238, 2016.
- [28] S. K. Kim, "Innovative design guidebook for game changers," *Three Step Innovation Process for New Business Developments*, 3, 2018.
- [29] S. Nokes *et al.*, *The Definitive Guide to Project Management*, 2nd ed. London, U.K.: Prentice Hall, 2007.
- [30] J. Meredith *et al.*, *Project Management: A Managerial Approach*, New York, NY: John Wiley and Sons, Inc., 2011.
- [31] P. C. Dinsmore *et al.*, *The Right Projects Done Right*, New York, NY: John Wiley and Sons, 2005.
- [32] Planbox. (2012). Agile Project Management. [Online]. Available: https://commons.wikimedia.org/wiki/File:Agile_Project_Management_by_Planbox.png

Copyright © 2025 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).