

A sustainable procedural method of software design process improvements

Khalid T. Al-Sarayreh¹, Kenza Meridji², Mamdouh Alenezi³, Mohammed Zarour⁴, Mohammed D. Al-Majali⁵

¹Department of Software Engineering, The Hashemite University, Zarqa, Jordan

²Department Software Engineering, University of Petra, Amman, Jordan

^{3,4}Department of Computer Science, Prince Sultan University, Riyadh, Saudi Arabia

⁵Department of Computer Education, Mu'tah University, Karak, Jordan

Article Info

Article history:

Received Apr 24, 2020

Revised Jun 27, 2020

Accepted Jul 22, 2020

Keywords:

A sustainable procedural method

Engineering design process

Generic design process

Scientific design process

Software design process

Software process improvements

ABSTRACT

In practice, the software process is an intermediate phase for enhancement and improvements the design for different types of software products and help developers to converts the specified requirements into prototypes that implement the design into reality. The objective of this paper is to provide software developers, designers and software engineers who work in small companies with a standards-based process improvement using a procedural method technique including detailed steps for designing the small software systems into their companies. The method used in this paper includes 1) analysis four different types of commonly design processes used by industry such as CMMI, conventional or software process in ISO 19759, generic and engineering design processes. 2) mapping between those four design processes. 3) collect the dispersed design concepts proposed by those four processes. 4) proposed a sustainable procedural method of software design process improvements 5) Illustration of the applicability of the proposed approach using A template-based implementation. The primary result of this study is a guideline procedure with detailed steps for software design process improvements to help and guide developers in small companies to analyze and design a small software scales with limited cost and duration. In conclusion, this paper proposed a method to improve the design process for different kinds of the software systems using a template-based implementation to reduce the cost, effort and time needed in the implementation phase in small companies. The scientific implication behind a template-based implementation helps the system and software engineering to use this template easily in their small companies; because most of the time those engineering developers are responsible for analyzing, designing, implementing and testing their software systems during the whole software life cycle.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Khalid T. Al-Sarayreh

Department of Software Engineering

Hashemite University

Zarqa 13133, Jordan

Email:khalidt@hu.edu.jo

1. INTRODUCTION

Currently, design Phase is a logical and rational process for producing and specifying design for new systems [1, 2]. During this phase, the detailed plan should provide a systematic and step at a time process. The design process in engineering projects has the maturity of the work efforts, time and cost of the product

[3-5], even though the engineering and scientific design life cycle is not the same as the software life cycle. However, the scientific design process [6] is a set of the design steps used to explore and predict solutions for most of the time for innovative software, engineering products or a combination of both.

The software design phase is used to describe the software structure and behaviour of the software product [3] (i.e. software components, collaborations and constraints). Consequently, the software design process defines the set of design activities, tasks and relationships between all of the product aspects [7]. Software design in software engineering domain should be build using detailed knowledge of the practice [8] (i.e. software application and interactions design) and a strategic realization vision considerable for their design [8] (i.e. implementation and the architecture design).

Recently, [9] proposed an approach for the implementation of the process area for small and medium-sized software development organizations in Saudi Arabia. They guide software development organizations for implementing the process areas of CMMI Level 2 in their environments. Followed by [10] defined an agile software development framework supported by some of the categories of CMMI. CMMI presented as a process improvement model in conventional methods, and many organizations depend on process maturity models to improve their particular processes and their product quality. However, [11] discussed the process maturity model for the software industry to guide managers of information-development teams in assessing and improving the work of their organizations. More specifically, [12-17] proposed different design frameworks to integrate user experience design into the software development cycle. In [18] RISDM ("Requirements Inspection Systems Design Methodology") is proposed to design the RIS ("Requirements Inspection System") to be conducted by a third party inspection team. Furthermore, [19] proposed a methodology for improving agile processes based on CMMI, this achieves by defining improvement on the GQM approach, that allow precise and intentional selection of CMMI practices. Therefore, [20] presents a performance management method that covers all process areas of CMMI level 5 and [21] proposed a method to improve the software development process to achieve maturity level 2 to conduct the assessment of the five process areas of CMMI level 2. In [22], they believed that poor design is implemented; it may affect the later stages of the software development life cycle. Research of the software design process done by [23] showed that bad software design could lead to productivity and quality problems.

In this paper, the research problem appears when the system and software engineers in small companies start working on the development process within the product life cycle. They have difficulty during the design process. More specifically, system and software engineers or developers involve a broad set of activities and tasks within the design process that are less understood and ignore some essential design aspects for executing a successful design phase due to the lack particular analysis, design and implementations departments for producing their products.

The endless customers' demands for new products features and functionality as well as looking for high product quality [24-28]; This creates a set of particular design challenges System, and software engineers encouraged to work on improving the design process improvements in their work product environment such as using well-known design principles, processes and the (Engineering and software) design in practice. However, each small company has adapted its suitable design process depending on their work products, and typically, this design process is frequently changed. At the same time, it is dependent on the product type due to the overlap between the scientific, engineering and software design processes [29-31] for different types of products.

In this paper, the solution to this problem, we proposed the procedural method of the software design process improvement. This proposed method is bridging the gap between the specified requirements and product implementation phase on the software life cycle. The procedural method of the software design process improvement proposed in this paper is the mapping results between the software design process with other three types of conventional design processes used by the industry such as a generic, engineering and scientific design processes. The procedural method is the implementation of this design process improvement using the dispersed design process concepts proposed by CMMI standards models.

2. RESEARCH METHOD

The research methodology is implemented in two phases, the first phase includes the procedural research method, and the second phase includes the sustainable indicators for the procedural research method. The Research procedural method used in this paper includes the following steps – See Figure 1:

- 1) Survey and analysis four different types of commonly design processes used by industry such as CMMI, conventional or (software design process in ISO 19759), generic and engineering design processes.
- 2) Mapping between those four design processes and collect the dispersed design concepts proposed by those four processes.
- 3) Proposed a sustainable procedural method of software design process improvements
- 4) Illustration of the applicability of the proposed approach using A template-based implementation.

The sustainable Indicators for the software design process improvements includes the following steps can see Figure 2.

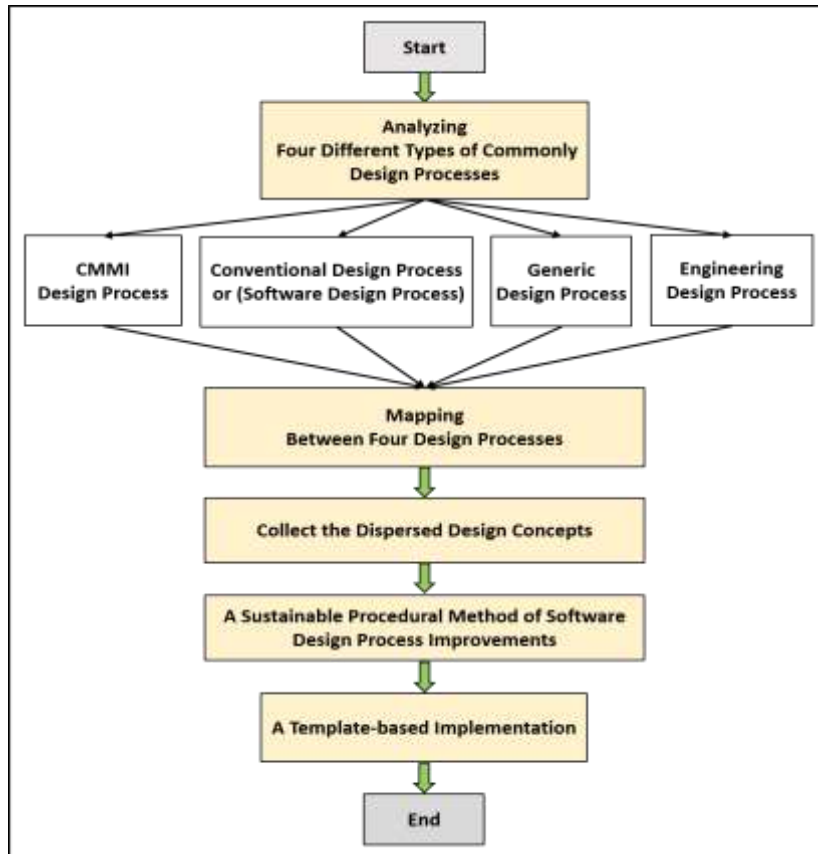


Figure 1. The research methodology

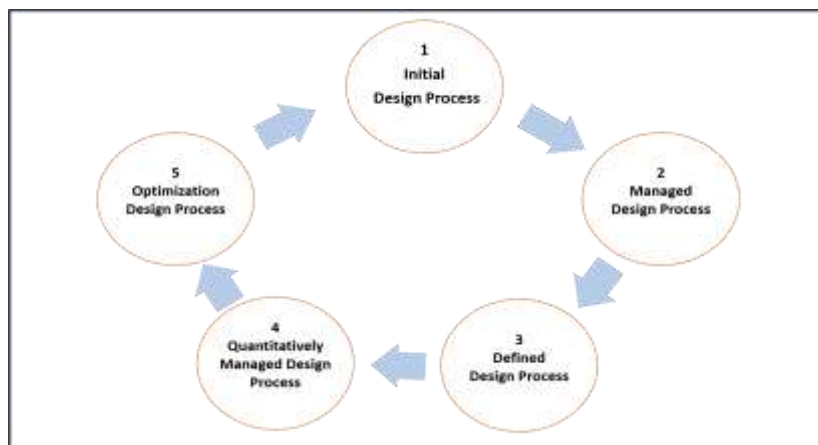


Figure 2. The sustainable Indicators for the software design process

3. RESULTS AND DISCUSSION

This section illustrates the results of our survey and analysis for different types of commonly design processes used by industry. The mapping results and the proposed procedural method for software process improvements as well as illustration of the applicability of the proposed approach using A template-based implementation.

3.1. The results of the design process survey and indicators

The design processes allow software and system engineers to establish the steps essential to develop software and hardware solutions with timetable and price constraints [32-34]. The current design phase involves a

broad set of activities and tasks based on the identified requirements and prepared these designed requirements into design components for the products construction phase on the next step of implementations.

However, the current design phase built a design model of the product or system design structure; for instance, such designed model is used to evaluate the suitability of the proposed product and communicate the recommended product to others [35-37]. There are many proposed models for conventional design processes in academia and the industry; the most commonly used models are presented below:

- a) Generic design process: Recently, [38] defined six steps for a generic design process that are more general and can apply to any software project. These steps used as a baseline process for software design.
- b) Scientific design process: The scientific method [39] is a process for experimentation that uses to explore observations and answer questions. Engineering design process.
- c) Engineering design process: [40-42] is a series of steps that engineers follow in coming up with a solution to a problem using specific criteria to accomplish particular tasks.
- d) Software design process: The design phase of software development [3, 43, 44] transforms the customer requirements into a form of implementable using a programming language. The software design process can be divided into the following three levels of phases of design (interface, architectural and detailed design). Interface design [3, 45]: is the “interaction specification between the system and its system environment”. Interface design proceeds at a high level of abstraction concerning the inside mechanisms of the system. Architectural design [3]: is the specification of the "system components, concluding their properties, interfaces, responsibilities and components relationships". As well as system structure was chosen, the architectural design. Detailed design [3]: is the “internal elements of the primary system components, properties, relationships, processing algorithm and data structures”.
- e) Capability and Maturity Model Integration (CMMI): [24-25] have five maturity levels (initial, managed, defined, quantitatively managed and optimized). By contrast, the process in maturity level 1 (initial) is unpredictable, poorly controlled and reactive. In maturity level 2 (managed), the process can be characterized for the projects and is often reactive. In maturity level 3 (defined), the process tailored to the Organization and its proactive. However, in maturity level 4 (quantitatively managed), the process can be measured and controlled. Finally, in maturity level 5 (optimized), the focus is on the process improvements.

3.2. The mapping results of the design processes

This section presents the results of the mapping of the software design process in [3] with three conventional design processes from the industry. In this mapping, three design processes are chosen (generic, engineering and scientific design processes). These conventional processes present a variation in their definitions and concepts used at the level of software design. The results of this mapping w depicted in Figure 3. Where, the software design process, is considered as a baseline design process for the software. In contrast, the other three design processes are analyzed and w mapped as a software design process improvement for the baseline design process. In this mapping, one concludes that the software design process has some of the missing detailed steps after conducting the mapping from the generic, scientific and engineering design processes as discussed in the previous section. For instance, the software design process in [3] proposed to analyze the problem without enough details or design steps for the software designer to tackle. Whereas, in the improvement process, designers can follow the proposed and detailed steps to analyze their design problems.

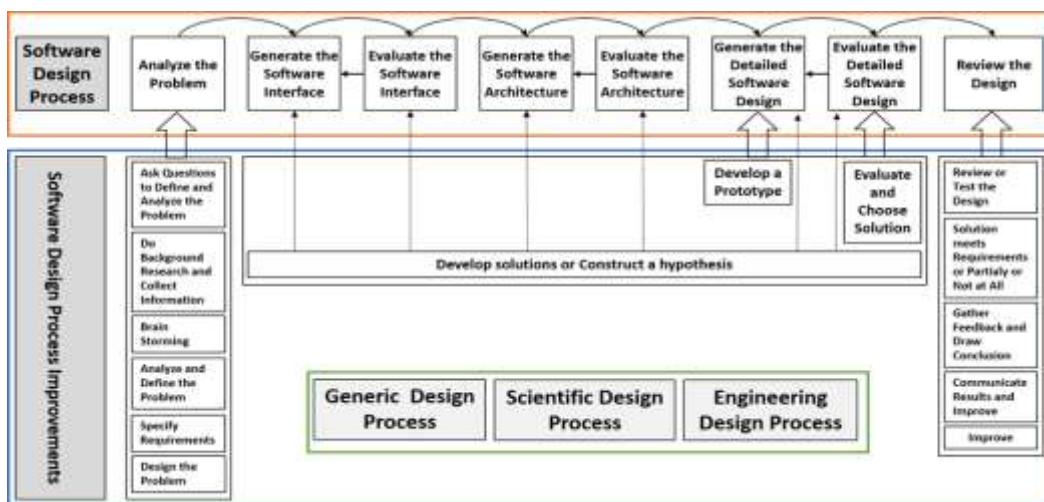


Figure 3. Software design process improvements

3.3. The sustainable Indicators for the software design process

In this section, the proposed procedural method for software process improvements presented in Figure 4. This procedural method consists of five individual levels as they are defined in the CMMI maturity models, these levels are: (initial, managed, defined, quantitatively managed and optimized), numbered from 1 to 5 respectively. Each level of the procedural method describes the detail of the software design process improvements within the baseline design process. The beginning of the procedural method of the software design process is considered as a baseline for this procedure-see Figure 4 (left-side). On the (right-side) of the same Figure 4, the software design process improvements are considered [46-47]. The proposed procedural method is described by using the following steps:

- 1) Initial Design Process: the process indicates that it is partially performing or not performing at all. Which means, one or more of the goals of the process area are not satisfied.
- 2) Managed Design Process: The project characterizes the process as a performed process.
 - a) Analyze the problem: understand the problem and what is precisely needed to do. Analyze the problem from different sides to discover the design requirements.
 - b) Generate and evaluate the software interface, architecture and detailed design. Develop possible solutions and choose the most appropriate one. Depending on the software designer's experience and available resources, use graphical representations, formal specification method or other notations to describe the components of the design and describe software design phases.
 - c) Review the software design: repeat the process for each identified abstraction until the individual expression of design. Defined Design Process: monitoring and controlling the software design process within organizations.
- 3) Defined Design Process: The simple process can monitor and controls the organization process to achieve given objectives and goals, such as project cost estimation and project schedule.
- 4) Quantitatively Managed Design Process: establish performance indicators for the software design process.
- 5) Optimization Design Process: standardized the software design process within the Organization.

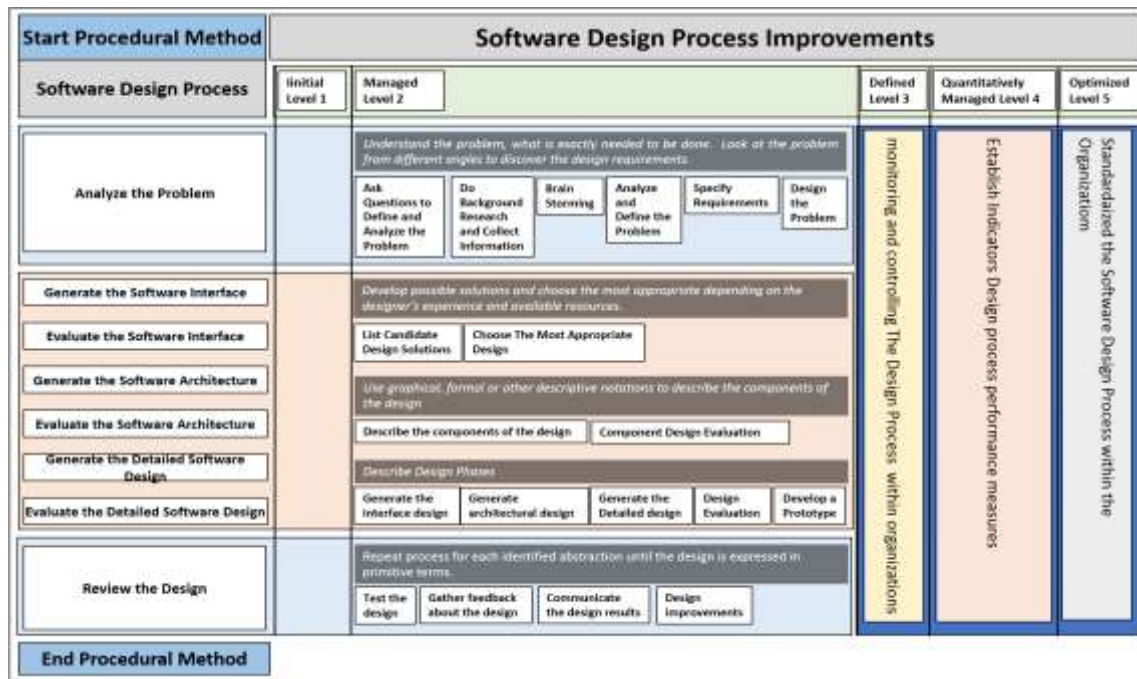


Figure 4. Procedural method for software process improvements

3.4. Template-based implementation

This section presents a template base implementation of the procedural method of the software design process improvements aligned with some design concepts of the CMMI standard models to be used by the manufacturing, businesses and commercial companies. In this paper, the procedural method is characterized and aligned with the industry conception of design processes. It offers a reference-based

solution for the software design phase on the software engineering life cycles (SDLC). The practical template used the following format:

- 1) Initial Design Process: Initial level: in this level 1, the process indicates that it is partially performing or not performed at all. Which means, one or more of the goals of the process area are not satisfied; thus, this level has no generic goal since there is no reason to institutionalize a partially performed process.
- 2) Managed Design Process: Managed level: in this level 2, the process is characterized by the project as a performed process that satisfies the goals assigned to that process.
 - a) *Define and analyze the design problem*: Software and system engineers can work without a problem defined; it happens all the time. Nevertheless, when an explicit declaration states what problem needs to be solved, all efforts are concentrated on its outcome. Understanding the design problem leaves the door open for creativity, but it eventually provides a clear view of each element of the project. Thus, designing and modelling the problem declaration and the design process steps help the Team understands and analyses the ideas correctly. As the design process improves, the Team would refer to the initial problem and confirm that what is designed quiet addresses the core problem declaration and is documented in the product design artefact.
 - b) *Background research about the design problem*: An essential part of the design process is the definition of a definite problem assertion, which the system and software engineers attend to solve. The problem is perhaps the most challenging part of the design process; it needs to generate explanations about the users from the first period in the design process. When the Team learn how to define the problem, problem declaration, or design challenge, it improves the design rational process and results because a clear definition of the problem declaration leads the system and software teams to define the process in the right direction.
 - c) *Collect information about the design problem*: Software and system engineers can collect information about the design process from the Internet research, previous research work, expert opinions, international standards and user interviews and they can use further techniques such as knowledge web and reverse engineering. Moreover, software and system engineering teams can improve the context for the design challenge; follows activities, and the design teams use this body of knowledge about the problem to generate product design ideas.
 - d) *Brainstorming of design problem understanding*: Brainstorming is a way of producing ideas to solve a design problem. It frequently contains a team under the direction of the team leader. The strong point of brainstorming is the prospective members who are always providing suggestions between their ideas in "a free-thinking environment".
 - e) *Specify design problem requirements*: Design problem requirements conclude the functional need that a specific design, product and process satisfy. It usually uses a formal sense in software and engineering design, it uses comprehensive concepts, functions, attributes and it may include the quality of the system to provide value to a customer, Organization or any other stakeholders. A specified requirement uses as inputs into the design stages of product development. Requirements are an essential input into the verification process since tester ought to trace back to specific requirements. Requirements illustration elements and functions are needed for the actual project.
 - f) *List candidate design solutions*: The best method to investigate the design solution alternatives is through trade design studies [48]. The trade design studies determine to confirm that the design and architectural results are using the best design solution process that can be accomplished with the available resources.
 - g) *Choose the most appropriate design*: Choose the possible solution that meets the design requirements, consider solutions that are better than others, and discard those that did not meet the requirements. Software and system engineers can use the following design criteria when choosing which possible solution to implement such as skills required, time and resources. The design criteria help to compare solutions in a decision matrix (i.e. a table with the requirements and criteria on one axis and the different solutions on the other).
 - h) *Describe the components of the design*: The software/system architecture defines the adaptable components that might be used to implement a work product family. Component design is a specification for one of these adaptable design components, and each component must be designed to fulfil significant aspects of the product requirements and all design structures of the product architecture. Each component design represents a family of components. Design component contains the following parts:
 - Adaptation Specification defines the components and components parameters with its variation and constraints. Each parameter has a name and type to specify its range of variation and a parameter constraint to identify the invalid combinations parameter variations.
 - Interface Specification defines the preferred features of the application for each component. The precise content of the interface specification is specific to the component type and the design

technique used. The interface specification parameterizes concerning the variations in the adaptation specification.

- i) *Component Design Evaluation*: Component design evaluation is a method to conduct qualitative component, and then combines the results data for each component with a specific design evaluation template. Occasionally these components may refer to different data sources (i.e. qualitative data from “interviews, and quantitative data from a questionnaire”) and from different evaluation aspects (i.e. qualitative data about context and processes and quantitative data about outcomes and influences).
 - j) *Generate architectural design*: Architectural design is a conception that concentrates on components or foundations of a design structure. System and software engineers are commonly responsible for the architectural design to create a comprehensible and functional design structure.
 - k) *Generate the interface design*: User interface design usually states the visual design of the product elements, and a user interacts via this visual design in the future. User interface designs have to be attractive to possible users and should be functional and created with users in mind.
 - l) *Generate Detailed design*: The detailed design describes the process of developing the approved design conception. The detailed design should be correct, synchronized and defines all the main design components. However, the technical features of the design may involve additional improvement. The detailed design should provide sufficient information for implementation. The detailed design may include the following: (General layout, functional and operational design flows and organizational policies and terms, build based on the first specified requirements and comprehensively describes all the design components for the application).
 - m) *Design Evaluation*: Design evaluation is a structure produced to provide a balanced assessment of design benefits. The result of evaluation design is influenced by the evaluation questions and the standards of effectiveness, nevertheless the available resources and on the degree of the accuracy required.
 - n) *Test the design*: The most crucial goal of the testing is to give substantial assistance on quality and risk assessment for the product. The tester should be building a testing strategy and start to collect all the information about product and product behaviour. The tester can build an executing test cases to cover all the product if possible. Designing the correct set of test cases is essential to link between the test strategy and the carrying out of the test strategy.
 - o) *Gather feedback about the design*: Productive feedback usually improves the effectiveness of the required design. Getting valuable feedback help developers to create a professional design and provide them with an alternative point of views, which help developers to improve both their current and future designed applications and projects. They increase the collaboration between key stakeholders, managers, developers and their potential users for the product.
 - p) *Communicate the design results*: Communicating the design results help to prove the design purpose, validating the complete solutions and confirm that they use a logical approach for the required design. A definite description of the design results expresses to industries that the outcome is a result of user research, product testing, and a well thought out design process.
 - q) *Design improvements*: Once evaluating the present situation of the product design and the improvement goals and objectives of the Organization, the results should be characterized and ordered according to which one is the most significant or have the utmost rigorousness. Developers have to observe the new target level of improvements and how they should look. Furthermore, the gap between the current and target design should plan in terms of a set of activities to achieve that design target. These activities should achieve with the arrangement of the elaborated stakeholders and the organization goals and objectives.
- 3) *Defined Design Process*: a process is a tailor to the Organization, and the process at this level is the proactive process. The simple process can monitor and controls the organization process to achieve given objectives and goals, such as project cost estimation, project schedule, and project quality assurance.
 - 4) *Quantitatively Managed Design Process*: the process can be measured and controlled. Software designers can define the process performance baselines and models using the monitored and controlled process at a defined level, and software designers can also link the baselines with the organization objectives. Then process performance quality from the elements of the quality assurance at level three such as (testing strategy, formal modelling and verification, and configuration management) and finally define their process performance measures.
 - 5) *Optimization Design Process*: this level focuses on process improvements. Developers or designers have the opportunity to determine process improvement. They can establish and appraise the organizational process needs, establish and implement the process action plans. Then they can deploy a standard

process and assets and monitor the process implementation and finally, incorporating their experience into the organization process.

4. CONCLUSION

The primary outcomes of this paper is a procedural method for the software design process improvements. This proposed method is built based on the analysis of common four different types of conventional design processes and aligned with the structure of the CMMI models. In literature, there are many proposed design processes; for instance, there are different design processes with different steps used in manufacturing tailored for each Organization. In software engineering, the design process is usually divided into three main steps (user interface design, software architecture and software detailed design). While in traditional engineering, the design process encompasses the initial and detailed design within their process. However, the scientific design process is used for innovative projects. Even these scientific projects may have software and hardware or a combination of both them. Furthermore, some of the completed products are built based on using a generic design process.

The suggested procedural method includes some of the consensual design process terms and concepts used by two sets of international standards (ISO and CMMI) and some other related works. They were analyzed and integrated using different design process views. The improvement design processes for describing the procedural method at a lower level was adopted.

There exists no such a procedural method for software design process improvements; the innovative work in this paper adapted from two sets of standards used by industry. The result of using this proposed method reduces the cost, efforts and time needed in the implementation phase. In this paper, The contribution of the proposed method is to provide a reference procedure for the software design process improvements for different types of projects in small companies under developments.

The industrial impact of this paper is sustainable improvements for the software design process. Therefore, this can enhance planning, management and development of software engineering or other types of products at the design phase in small companies. In this proposed work, some related issues are still not taken into consideration, and further work is required, such as the experiment of this proposed method on some real case studies. Small companies in the industry can use this work; to evaluate and to get feedback from their experience.

REFERENCES

- [1] S.R. Chidamber and C. F. Kemerer, "A metrics suite for object-oriented design," *IEEE Transactions on software engineering*, 20, no. 6, pp. 476-493, 1994.
- [2] A. Alain and K.T. Al-Sarayreh, "Standards-based model for the specification of system design and implementation constraints," *Industrial Proceedings, 17th European Systems & Software Process Improvement and Innovation, EuroSPI 2010 Conference*, Grenoble (France): Publisher: Delta, Denmark, 2010.
- [3] P. Bourque, and R.E Fairley, *Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0*, IEEE Computer Society Press, 2014.
- [4] K.T. Al-Sarayreh and K. Meridji, "Towards a Development of an Operational Process for Software Requirements: Case study application for Renewable Energy Software," *International journal of software engineering and its applications*, vol. 9, no. 7, pp. 11-26, 2015.
- [5] R. A. Majid, N. L. Noor, W. A. Wan, "Theoretical Perspectives of the HCD integration in the software development process," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 17, no. 3, pp. 1091-1095, 2020
- [6] IEEE Standard for Information Technology, "Systems design--software design descriptions," *IEEE STD 1016-2009*, pp.1-35, 2009.
- [7] K.T. Al-Sarayreh, K. Meridji, G. Issa and M. A. W. Hadi, "Procedural model of software quality improvements for sustainable applications process of renewable energy systems (Development & enhancement)," *2nd International Conference on the Applications of Information Technology in Developing Renewable Energy Processes & Systems (IT-DREPS)*, Amman, pp. 1-6, 2017.
- [8] A. Alain, and K. Meridji, "Analysis of software engineering from an engineering perspective," *European Journal for the Informatics Professional*, vol. 7, no. 1, pp. 46-52, 2006.
- [9] K. M. Niazi and M. Alshayeb, "Towards implementation of process and product quality assurance process area for Saudi Arabian small and medium-sized software development organizations," *IEEE Access*, vol. 6, pp. 41643-41675, 2018.
- [10] S. Kawamoto and J. R. de Almeida, "Scrum-DR: An extension of the scrum framework adherent to the capability maturity model using design rationale techniques," *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, Pucon, pp. 1-7, 2017.
- [11] J. T. Hackos, "Information Process Maturity Model," *2017 IEEE International Professional Communication Conference (ProComm)*, Madison, WI, pp. 1-8, 2017.

- [12] G. Rong, H. Zhang and D. Shao, "CMMI guided process improvement for DevOps Projects: An exploratory case study," *2016 IEEE/ACM International Conference on Software and System Processes (ICSSP)*, Austin, TX, pp. 76-85, 2016.
- [13] K. Meridji and A. Abran, "Software engineering principles: do they meet engineering criteria?," *Journal of Software Engineering and Applications*, vol. 3, no. 10, pp. 972-982, 2010.
- [14] J. Elhassouni, A. El-Qadi, M. Bazzi and M. El Haziti, "Modeling with ontologies design patterns: credit scorecard as a case study," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 14, no. 2, pp. 429-439, 2019.
- [15] N. Athirah, N. M. Idros, H. Mohamed and R. Jenal, "The use of expert review in component development for customer satisfaction towards e-hailing," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 18, no. 2, pp. 347-356, 2020.
- [16] A. Alain, K. Meridji, and J. Dolado, "Software engineering from an engineering perspective: SWEBOK as a study object," *Apoyo a la Decision en Ingenieria del Software-ADIS Workshop, Congreso Espanol de Informatica-CEDI Conference*, Zaragoza, Spain, pp. 1-6, 11/24/2007, 2007.
- [17] L. Peres and S. L. Meira, "Towards a framework that promotes integration between the UX design and SCRUM, Aligned to CMMI," *10th Iberian Conference on Information Systems and Technologies (CISTI)*, Aveiro, pp. 1-4, 2015.
- [18] S. Saito, M. Takeuchi, S. Yamada and M. Aoyama, "RISDM: A requirements inspection systems design methodology: Perspective-based design of the pragmatic quality model and question set to SRS," *IEEE 22nd International Requirements Engineering Conference (RE)*, Karlskrona, pp. 223-232, 2014.
- [19] F. Rainho and J. Barreiros, "Agile process optimization: An approach using the CMMI and GQM," *14th Iberian Conference on Information Systems and Technologies (CISTI)*, Coimbra, Portugal, pp. 1-6, 2019.
- [20] P. Örgün, D. Güngör, Y. Y. Kuru, Ö. O. Metin and M. Yılmaz, "software development overall efficiency improvement in a CMMI level 5 organization within the scope of a case study," *3rd International Conference on Computer Science and Engineering (UBMK)*, Sarajevo, pp. 258-263, 2018.
- [21] M. Jezreel, G. Marcos and M. Mirna, "Organization of the process areas of CMMI-Dev v1.3 level 2 through of its dependencies," *12th Iberian Conference on Information Systems and Technologies (CISTI)*, Lisbon, pp. 1-7, 2017.
- [22] H. Ying, Q. Peng and Z. Jiyou, "Maturity assessment model for aircraft collaborative design software solution," *Journal of Systems Engineering and Electronics*, vol. 29, no. 6, pp. 1228-1236, Dec. 2018.
- [23] B. Curtis, H. Krasner, and N. Iscoe. "A field study of the software design process for large systems," *Commun. ACM*, vol. 31, no. 11, pp. 1268–1287, 1988.
- [24] S. K. Najjar and K.T. Al-Sarayreh, "Capability maturity model of software requirements process and integration (SRP^{CMMI})," *International Conference on Intelligent Information Processing, Security and Advanced Communication (IPAC '15)*. Association for Computing Machinery, New York, NY, USA, Article 68, 1–5. 2015.
- [25] S.U. Team, "Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A, Version 1.3: Method Definition Document," *Software Engineering Institute, Carnegie Mellon University, Tech. Rep. CMU/SEI-2011-HB-001*, 2011.
- [26] M. Davis, *201 Principles of Software Development*, New York, NY, USA: McGraw-Hill, Inc., 1995.
- [27] K. Meridji and K.T. Al-Sarayreh, "mapping a knowledge areas of the SWEBOK standard with the CBOK in software engineering," *Advances in Software Engineering and Systems, Proceedings of the 14th International Conference on Software Engineering, Parallel and Distributed Systems (SEPADS '15)*, WSEAS, pp. 126-130, Dubai, United Arab Emirates, February 22-24, 2015.
- [28] K. Meridji, K. T. Al-Sarayreh, E. Fayyumi, and S. Idwan, "Integrated model of photovoltaic solar system with the sound biometric techniques," *IEEE 1st International Conference & Exhibition on the Applications of Information Technology in the Development of Renewable Energy processes & System IT-DREPS*, pp.: 43-48, Amman, Jordan May 29-31, 2013.
- [29] IEEE International Standard, "Systems and software engineering--design and development of information for users," *ISO/IEC/IEEE P26514/D1*, pp.1-87, March 20 2020.
- [30] R.E. Al-Qutaish and K.T. Al-Sarayreh, "Software Process and Product ISO Standards: A Comprehensive Survey," *European Journal of Scientific Research*, vol. 19, no. 2, pp. 289-303, 2008.
- [31] K. Meridji, A. Abran, K.T. Al-Sarayreh and A. April, "A Framework for the Measurement Concepts in the SWEBOK Guide," Book Title: Software-Engineering, German Title: Beiträge zum empirischen, *Shaker Verlag*, Andreas Schmiedentorf and Cornelius Wille, Germany 2012.
- [32] A. Ertas and J. Jones, *The engineering design process*, 2nd ed. New York, N.Y: John Wiley & Sons, Inc., 1996.
- [33] K. Al Makhadmeh, K.T. Al-Sarayreh, K. Meridji, " Towards A Design Measurement Context for Software Coupling and Cohesion Requirements," *Recent Advances in Computer Science, 14th International Conference on Applied Computer and Applied Computational Science (ACACOS '15)*, WSEAS, Kuala Lumpur, Malaysia, pp. 111-116, April 23-25, 2015.
- [34] A. Abran and K.T. Al-Sarayreh, "Measurement of Software Requirements Derived from System Operations Requirements," *20th International Workshop on Software Measurement (IWSM 2010)*, Springer, Stuttgart, Germany, Nov. 2010.
- [35] S. Tayal. "Engineering design process," *International Journal of Computer Science and Communication Engineering*, vol. 18, no. 2, pp. 1–5, 2013.
- [36] K.T. Al-Sarayreh, I. Al-Oqily and K. Meridji, "A standard-based reference framework for system operations requirements," *Int. J. Computer Applications in Technology IJCAT*, vol. 47, no. 4, pp. 351-363, 2013.

- [37] K. Meridji, K.T. Al-Sarayreh, A. Abran and S. Trudel, "System security requirements: A framework for early identification, specification and measurement of related software requirements," *Computer Standards & Interfaces, Elsevier*, vol. 66, 2019.
- [38] I. Sommerville, M. Fowler, K. Beck, J. Brant, W. Opdyke and D. Roberts, "Software Engineering," *Addison-Wesley*, 2019.
- [39] V.D. Bianco, V. Myllärniemi, M. Komssi and M. Raatikainen, "The role of platform boundary resources in software ecosystems: A case study," *IEEE/IFIP Conference on Software Architecture*, pp. 11-20, 2014.
- [40] R.F. Buckminster, "A comprehensive anticipatory design science," *Royal Architectural Institute of Canada*. 34. Retrieved 2016-09-14 – via *Google Books*, 1957.
- [41] K. Almakadmeh, K. Meridji and K. T. Al-Sarayreh, "Towards a reference model of software resources quality," *Journal of Computer Science (JCS)*, vol. 14, no. 2, pp. 182-198, 2018.
- [42] B. Meyer, *Object-oriented software construction*, 1st. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988.
- [43] R. E. Al-Qutaish, and K.T. Al-Sarayreh, "Applying Six-Sigma Concepts to the Software Engineering: Myths and Facts," *7th International Conference on Software Engineering, Parallel and Distributed Systems (SEPADS'08)*, University of Cambridge, Cambridge, UK, pp. 178-183, 2008,
- [44] K. T. Al-Sarayreh, "Dependability model for decomposition and allocation of system safety integrity levels of software quality," *International Review on Computers and Software (IRECOS)*, vol. 10, no. 11, pp. 1110-1119, 2015.
- [45] K.T. Al-Sarayreh, and A. Abran, "A generic model for the specification of software interface requirements and measurement of their functional size," *8th ACIS International Conference on Software Engineering Research, Management and Applications - SERA 2010*, Montreal, May 24-26, 2010, IEEE-CS Press, Los Alamitos, pp. 217-222, 2010.
- [46] M.S. Miki, T. Yamanouchi and M. Watanabe, "Software synthesis for trade-off design," *11th Knowledge-Based Software Engineering Conference*, Syracuse, NY, USA, pp. 144-153, 1996.
- [47] K.T. Al-Sarayreh, L. Hassan and K. Almakadmeh, "A Trade-off Model of Software Requirements of balancing between Security and Usability Issues," *International Review on Computers and Software (IRECOS)*, vol. 10, no. 12, pp. 1157-1168, 2015.
- [48] A. Abran, K.T. Al-Sarayreh and J. J. Cuadrado-Gallego, "A standards-based reference framework for system portability requirements," *Computer Standards & Interfaces, Elsevier*, vol. 35, no. 4, pp. 380-395, 2013.