# ITERATIVE AND INCREMENTAL DEVELOPMENT ANALYSIS STUDY OF VOCATIONAL CAREER INFORMATION SYSTEMS

Isyaku Maigari Ibrahim[1], Ogwueleka Francisca Nonyelum[2] and Isah Rambo Saidu[3]

[1&2]Department of Computer Science, Nigerian Defense Academy, Kaduna,
Nigerian Defense Academy, Kaduna, Nigeria
[3]Department of Cybersecurity and Interdisciplinary studies,
Nigerian Defense Academy, Kaduna, Nigeria

***ABSTRACT***

*Software development process presents various types of models with their corresponding phases required to be accordingly followed in delivery of quality products and projects. Despite the various expertise and skills of systems analysts, designers, and programmers, systems failure is inevitable when a suitable development process model is not followed. This paper focuses on the Iterative and Incremental Development (IID)model and justified its role in the analysis and design software systems. The paper adopted the qualitative research approach that justified and harnessed the relevance of IID in the context of systems analysis and design using the Vocational Career Information System (VCIS) as a case study. The paper viewed the IID as a change-driven software development process model. The results showed some system specification, functional specification of system and design specifications that can be used in implementing the VCIS using the IID model. Thus, the paper concluded that in systems analysis and design, it is imperative to consider a suitable development process that reflects the engineering mind-set, with heavy emphasis on good analysis and design for quality assurance.*

**KEYWORDS**

*Iterative, incremental development, vocational carrier, system development process*

## 1. INTRODUCTION

Over the years, the focus of researchers and developers in computing is on how to develop suitable systems, software and services that can be used as key asset for competitive high technology operations. This can be achieved with a good analysis and design practice that breakdown complex systems into their structural parts, and provides a working blue print upon which an acceptable implementation and deployment is made. This is also the reason the field of software development, systems analysis and design are not diffident of innovating, following and using new methodologies that can provide the expected quality and satisfy the stakeholders involve. Thus, in the analysis and design of software systems, information system, it is paramount to follow a suitable development process that reflects the engineering mind-set, with heavy emphasis on up-front analysis and design (Feiler and Humphrey, 1993; Wastell, 1999; Fuggetta, 2000; O'Connor, 2008).

The development process provides the necessary guidance between different stages of a project, and should be beneficial to the stakeholders (ISO, 2008). These development processes are characterized by fundamental activities or phases in accordance with the method or model called software development models (SDM) or software development life cycle model (SDLC). Despite the various expertise and skills of systems analysts, designers and programmers, systems failure is inevitable when a suitable SDM is not followed. Such failure can only be avoided when systems analysts and designers followed a typical process model in producing quality software products.

The SDM contains several phases that a system analyst followed in the analysis and design of quality software product or system (Lamassoure, 2004; Iqbal and Idrees, 2017). The SDM helps in determining the order of the stages and to establish the transition criteria (Boehm, 1988). Examples of such models include Sequential models (for example, Waterfall, V model), Rapid Prototyping, Iterative models (for example, spiral), Phased models (for example, incremental, evolutionary). Common to the aforementioned SDM is that they are all data driven (Selic, 2003; Völter,2013).

This paper focuses on the iterative and incremental development (IID) models in a case study and justified its role in the analysis and design software systems. The iterative model prescribes the construction of initially small but even larger portions of a software project to help all those involved to uncover important issues early before problems or faulty assumptions can lead to disaster (Jakeman, 2006). Today, the agile software development processes(Abrahamsson, 2017) are built on the foundation of iterative and incremental development that emphasizes technical, communication, and teamwork skills for effective delivery of software products of good quality (Zhou and Mockus, 2011; Päivi, 2017; Magana, 2018). The iincremental development allows the addition of more functionality in each release(Greer and Ruhe, 2004).Manifestly, the paper preached that the IID are important subjects in the area of systems analysis and design and other related disciplines. It is concerned with improving the approach to software quality through analysis and design.

The incremental development is a method that supports the design, implementation and testing of software systems incrementally (Larman and Basili, 2003). It provides supports for the addition of requirements iteratively until the software product is finished thereby involving both development and maintenance. a constant communication is a fundamental necessity for successful realization of IID method (Kern, 2002). Particularly uncertainties in the form of shifting requirements may require more communication and interaction between the software engineer and client for better understanding and problem solving (Larman, 2003). IID works by identifying the software requirements, analysing the requirements, and based on the analysis, a design specification for the software is produced, and then coding follows based on the design specification. When the process of gathering requirements, specifying the design and coding is completed, feedbacks from stakeholders are gathered are assembled before proceeding to the next software product increment phase or cycle. However, the gathered feedbacks and information are considered subsequently. This process continues until the product is perfectly delivered (Larman, 2003).

Complexity of software development requires adequate training and good understanding of Systems Analysis and Design which may be obtained through proper academic degree from universities or tertiary institution. Time and cost are the major constraint for Vocational carrier software developers to obtain such training and be skilled enough and break even in the software development market and to get a quick return and earn a living. The objective of this research is to educate Vocational carrier software developers by analysing the process using iterative and incremental development approach. This is considered as the approach that is most advantageous to less skilled developers based on its simplicity in terms of testing and debugging of iterations. In section two, a review of related work is presented, while the methodological design approaches used are highlighted in section three alongside the detailed explanations on the data collection, analysis processes, and the system design specifications. The general discussion on the results was addressed in section four. Section five finalise the paper and future work with implications for the adoption of the iterative and incremental development analysis was also discussed.

## 2. REVIEW OF RELATED LITERATURE

In the literature, the generally successful software/system development requires a development process that can facilitate the development of right products within a desired time and being able to develop the right products (Trott, 2005). Table 1 showed the success factors on the applications of the development processes followed in engineering quality software product development derived from key research (Daria, 2018). Some researchers distinguish parameters such as the quality of the product, cost and time of development (Mäkela, 2008); or more precisely technical performance, innovation degree, manufacturing and design costs, level of service and attractiveness of the product or service (Krisnan and Ulrich, 2001). All of these borders having a suitable process model that will aid good analysis and design. This process is a social learning process (Pressman, 2005).

Table 1: Success factors on the applications of the development processes followed in engineering quality software product development derived from key research (Daria, 2018)

| **Authors** | **Success Factors** |
|---|---|
| Colby, 2015 | Culture, feedback, communication, staff-ing, collaboration, time/budget |
| Sudhakar, 2012 | Top management support, communication in the project, clear project goal, user in-volvement, team work, reliability of output and project planning |
| Chow and Cao, 2008 | Correct delivery strategy, proper use of ag-ile software engineering techniques, strong team capability, adaptive management style |
| Baskerville, 2006 | Quality, cost, and development speed |
| MacCormack, 2001 | Product quality (reliability, technical per-formance, breadth of functionality), superi-ority to the competitors, project resource productivity |
| Curtis, 1988 | Software productivity and quality |

However, existing models consist of several sub processes called phases, stages or activities (Dara, 2018). A number of different software development approaches, methodologies, established best practices that ensure the completion of final software product or framework that has been developed through history. A timeline of the evolution of the software development methodologies is presented in Figure 1.

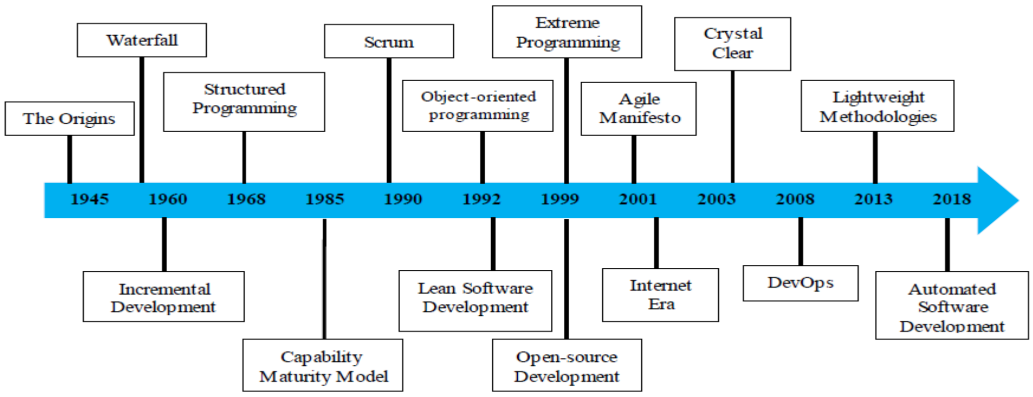

Figure 1;Timeline of software development methodologies (Daria, 2018)

In the Waterfall model for example, these phases go one after another, while in other models their sequence changes. In an agile model that supports and ensure iterative and incremental development, the process itself is more iterativeas describe in Figure two. According to Gao and Xiong (2015), there are only four main sub processes: specification, designing, coding, testing which a system analyst can follow during a development process.

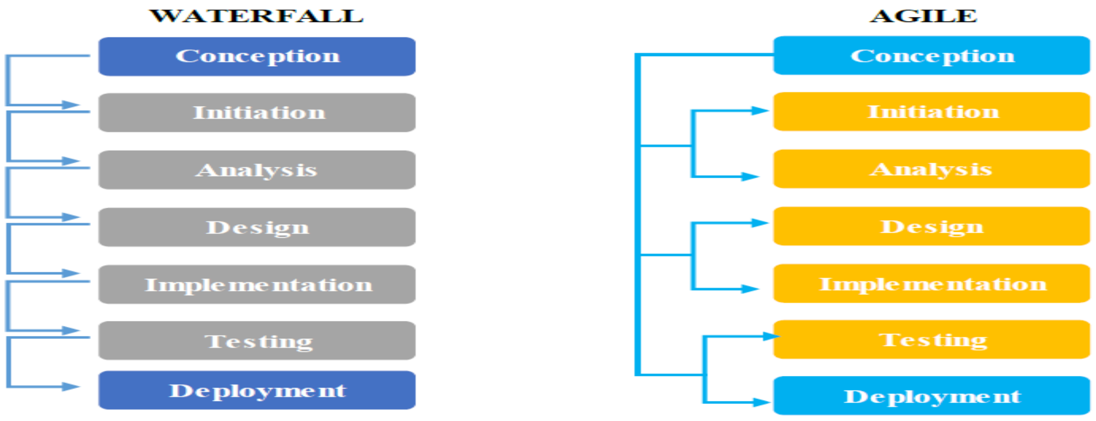

Figure 2:Process differences between the Waterfall and Agile methodologies Gao and Xiong (2015)

According to Soriyan (2004), IIDis a revolution process accomplished by series of agents that perform a variety of activities and whose connotation results in the production of a software system.The IID process outline the sequence of steps that are required to develop or maintain system iteratively and incrementally. The essence of IID include ensuring that high quality systems are delivered, providing strong management controls over the projects, and maximising the productivity of the development team (Bender, 2003).

The review conducted by (Mitchell&Seaman 2009) compare software cost, duration, and quality for waterfall and IID. The work Mitchell and Seaman (2009) viewed the traditional waterfall approach, iterative and incremental development as the two broadly defined competing approaches. The reason behind their comparism is to assist software project managers to make conversant choices of software development model for their projects and the results evidenced the superiority of IID over the waterfall model. This justified the acceptance of the IID approach in practice and reality for a software development project. Again, it was evident from the review that further empirical studies, both quantitative and qualitative, on the IID need to be undertaken using a case study to ascertain its effectiveness, and to reach a solid consensus on the standard sets of comparable parameters that can best assess cost, duration and quality of delivery. Päivi (2017) provided an in-depth study on the future trends and development methods in software quality assurance. Emphasis was more on the iterative and incremental development, focusing on the views and expectations customers in a case study have, thereby impacting the development methods in the future.

Emphasis on students been exposed to hands-on iterative and incremental software development was made with the aim of allowing students plan for project release and iteration delivery; and also develop, test, and deliver software in an iterative and incremental manner.

## 3. METHODOLOGY

In this paper, the qualitative research approach and a case study that justified and harness the relevance of iterative and incremental development from the perspective of systems analysis and design was adopted. In the qualitative approach, the iterative and incremental development model was taken as a change-driven software development process model. Instead of full planning and documentation from the planning stage of a development project, the incremental and iterative model are open to new requirements and encourages feedback from end users regularly . Thus, the phases of the iterative and incremental development model were strictly followed, but emphasis were made on the analysis and design phases, and the followed process were meant to take care of changes in requirements.

The case study was based is a Vocational Career Information System (VCIS) that will provide accurate information and guides on vocational career decision making. The goal of this system is to analyse and design a system that will to close gaps between vocational career and academic career, thereby allowing students to make prompt and appropriate decision on a vocational career choice of interest using the iterative and incremental model.

In order to understand meaning ofiterative and incremental developmentin analysis and design, a typical definition and application of the methodology was considered and adopted, and then an understanding on how it generalizes at the system development process level was presented. Difference the iterative and incremental models was made in order todifferentiate it from other process models.

Therefore, the requirementsfor the VCIS were elicited and analysed in between the planning and analysis phase to determine the targeted users, expected functionalities of the system from users, the system functionalities, the necessary data for the system and the likely feedback from the system. To accomplish the aforementioned, aface-to-face interview and a focus group approach was used. The system designed was specified using the Unified Modelling Language (UML) tools to describe the relationship between users and the system operations. A typical model of the iterative and incremental development analysis process (Figure three).

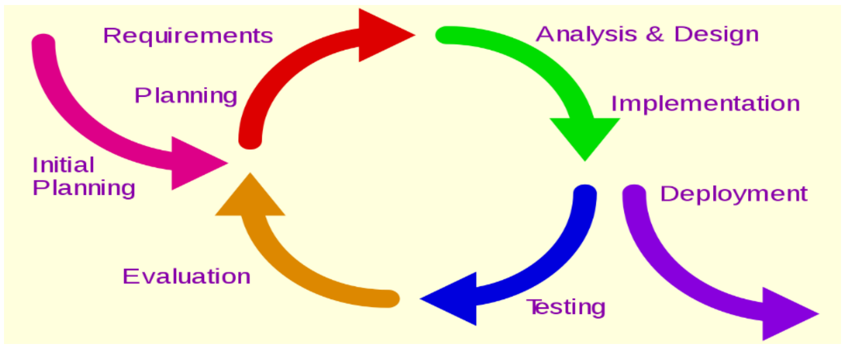

Figure 3: Iterative and Incremental Development Model(Jevon, 2009)

## 3.1. Data Collection and Analysis

Interview and focus group methods were used for collecting data. The target users are students from some secondary schools, Kaduna Polytechnic and Apprentices as contained in table 2. They were interviewed and discussed to know their view on vocational career information system. Few out of the questions asked during the interview are (1) what are expected systems requirements, (2) what are expected functional requirements, (3) what are your security expectation.

Table 2: Analysis of data collected

| **S/NO** | **Group /interviewee** | **System requirements** | **Functional requirements** |
|---|---|---|---|
| 1 | 22 | Username and password for authentication, 2nd factor authentication, dabase design, data indexing user interfaces designed | System to authenticate users, users to suggest career test, user to request career data , system to have administrator |
| 2 | 4 | Username and password, biometrics 2nd factor authentication, dabase design, user interfaces designed | System to authenticate users, users to suggest career test, user to request career data , system to have administrator |
| 3 | 4 | No response | No response |

## 3.2. System Design Specification

The expected users during the face-to-face interaction discussion process articulated their expectations for the system, and the various concern when making a vocational career choice. Tables 3 and 4 shows the system and functional specifications of the system respectively. Generally, the user is authenticated into the system and given access to either take a career test in accordance with the defined criteria for vocational choices or select a career path and view the selected career information.

Table 3: System Specification

| S/No | Specification |
|---|---|
| 1 | Username and password for identification and authentication |
| 2 | An administrative and user interface accessible via secure authentication thus ensuring permission is granted only to those entitled. |
| 3 | Database design and implementation as a means of having all the projects efficiently stored in a particular place. |
| 4 | Data indexing to enhance the search operation |

Table 4: Functional Specification of System

| S/No | Specification |
|---|---|
| 1 | User is authenticated into the system and given access to either take a career test based on Holland theory or select a career path and view the selected career information. |
| 2 | User can suggest a career and request for a career data which will be granted approval by the system administrator. |
| 3 | User can also search for available career choices. |
| 4 | The system administrator grant approval to the suggested career choices and requested career data. |
| 5 | The system administrator can add, update and delete a career, change login data and update any changes in the system. |

In specifying the design, a 3-tier architecture (Figure 4), was used to depict the structure of the system. The architecture comprises of the standard graphical user interface (GUI), the logic layer and the data tier layer. Mostly, the application requires the use of GUI such as a web browser to have access to the applications functionality to aid interaction with the system. The browser transmits the user's request to the web server, sending the requests using the HTTP protocol. The database is can be accessed with PHP script or a suitable language. This is meant to contain the system functionalities in terms of operations and dynamism.In conclusion, a web server with the likely proceeds from the database server to generatee the HTML pages that will be returned to the user.

In Figure 5, activity diagram is depicted to show the activities flow to be used by the system the system. At the initial stage, the user logs on to VCIS website, the system request for username and password. If the username and password is valid, the page will display all the services of the system. The user can view career information or take a test. If no more information is to be processed, the user exits the career system, else he or she performs more task. Figure 6shows the activity diagram on how the user specifies the search queries, the system searches the database for the specified search query, and the result is display to the user if match is found. Again, these design specifications can be implemented iteratively to provide a working prototype of the system during development process

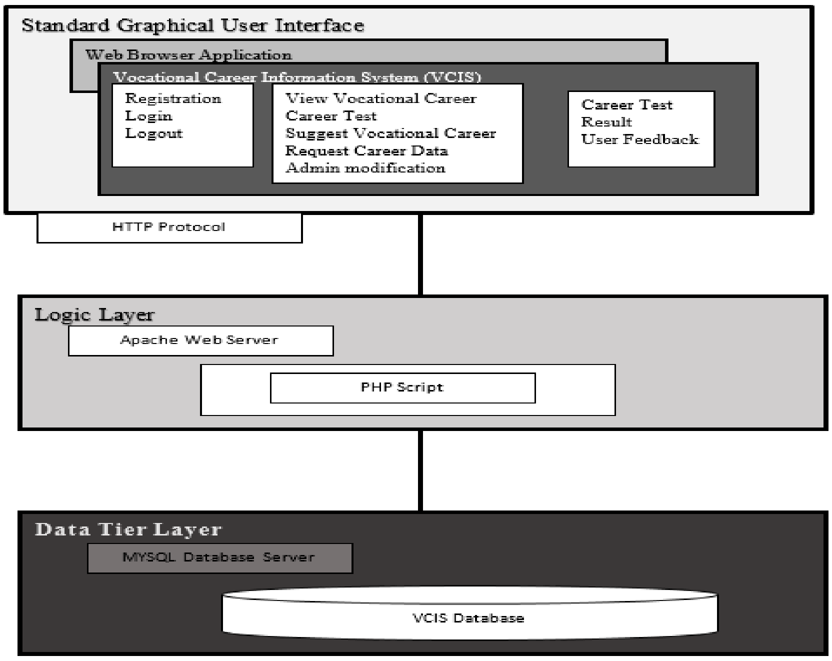

Figure 4: System Architecture

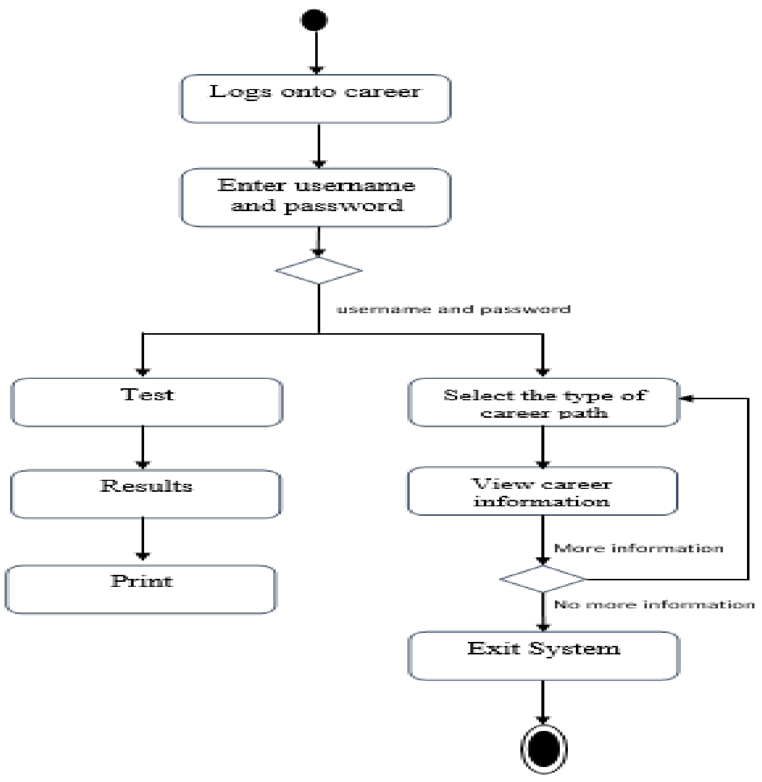

Figure 5: Activity diagram for the vocational career choice system

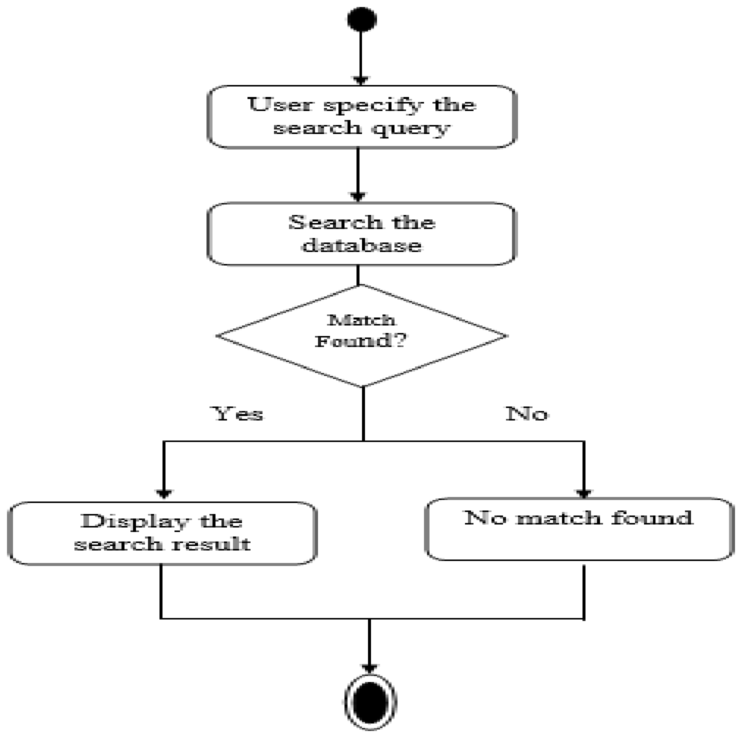

Figure 6: Activity diagram for search in the system

## 4. RESULTS AND DISCUSSION

The results showed some system specification (Table 2), functional specification of System (Table 3) and design specifications (Figure 4, 5, and 6 respectively) that can be used in implementing the VCIS using the IID model. Furthermore, it is possible with the IID model to focus on delivering tested small features of the VCIS and then reviewing these interactively with the users, which means progressively implementing the actual system in detached component parts. These design specification at the requirements level can be supported with a modelling process (Figure 7). The modelling process here can support bridging different levels of abstractions. For the IID, the modelling process plays essential role that focuses on the aspects (abstraction from multiple details) system analysts are expected to be observed. In addition the modelling process provide supports for understanding, explanation and communication with stakeholders. It also provides support for early analysis where alternatives are identified and accommodated by the IID.

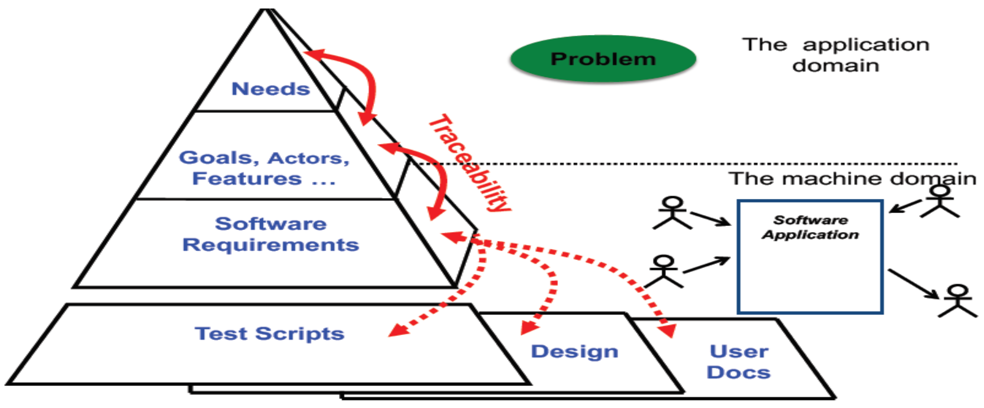

Figure 7: Proposed modelling process for IID

Accordingly, the design can be modified with each iteration, which means that the VCIS can evolve as new functional capabilities are developed. This can assist when he/she wants to test one level of the system in action before making full decisions on other functionality. It is not expected that the development be limited to single modules. In actual sense, it is possible to have a good number of iterations in progress with the cycles in IID model at any time. It will be necessary to testing and validation of each version of the VCIS based on the defined criteria within the model cycles. The advantages of the IID for the VCIS includes:

a) It permit the use of the VCIS and supports fast development time
b) It creates a functional VCIS in good time during the system life cycle.
c) It is more flexible and less expensive whenever the scope and requirements of the VCIS changes. That is it can accommodate changes at any development process stage with less expensive.
d) It is easier to test and debug the VCIS as smaller changes are made during each iterations.
e) The customers or clients can easily respond to each product developed.
f) Faulty elements of the VCIS are easily and quickly identified during the testing phase that comes up after each iteration.

Going further, working with the IID does not require detail requirements specification. The process can start with the design and then implement a limited part of the software. In this regard, the IID justifies that a developer can always return to expand and enhance the requirements specifications and design details until the entire system is fully implemented and deployed. In practical terms the following are ideal scenarios a developer and system analyst for the VCIS can consider in the use of IID model (Balaji and Murugaiyan, 2012):

- Comprehensive and informed requirements have been defined
- Defining main requirements at the beginning, but other functionality may change from time to time in collaboration with the client
- Time-to-market is considered as a constraint in the entire project
- Involvement of innovative technology
- Subjectivity of changing goals over time.

## 5. CONCLUSION

In conclusion, this paper opined that it is paramount to follow a suitable change-driven and data-driven development model process during software development, system analysis and design. The paper presented the IID as an ideal process model for a VCIS that can be used to implement a fast system for making career choices.

When working with stakeholders who have interest in VCIS (e.g. clients, end-users, customers), there can always be a trade-off between time of product delivery, budget (in terms of cost) and functionality (in terms of users and system expectations). With other SDMs, one of these areas needs to be compromised or suffer, thereby hindering on-time delivery of products, while also incurring high cost. However, the IID is presented in this paper as a perfect solution for ambitious development projects where budgets are limited, as it provides the opportunity to bring functioning software to the market quickly, and then gradually improve it over time. Once the right development resources are on board, IID can spell success through quick return on investment. Short iteration phases required early disclose of issues especially when the working condition is uncertain (Janzen& Saiedian,2005).The importance of the communication can be vital for integration part when the iteration cycle is longer. Without a proper communication and understanding between the parties, it is impossible to successfully implement IID.

For future work, it will be necessary to consider using the IID in a large-scale development of a life-critical system in order to determine its reliability.

## REFERENCES

[1] Abrahamsson, P., Salo, O., Ronkainen, J., andWarsta, J. (2017). Agile software development methods: Review and analysis. arXiv preprint arXiv:1709.08439, pp. 72 - 79

[2] Bakir, A., Turhan, B., andBener, A. (2011). A comparative study for estimating software development effort intervals. Software Quality Journal, 19(3), 537-552.

[3] Balaji, S., andMurugaiyan, M. S. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. International Journal of Information Technology and Business Management, 2(1), 26-30.

[4] Bender RBT Inc. (2003). Systems Development Lifecycle: Objectives and Requirements.Bender RBT Inc, Queensbury, New York, pp, 22 - 53

[5] Benediktsson, O., Dalcher, D., Reed, K., and Woodman, M. (2003). COCOMO-based effort estimation for iterative and incremental software development. Software Quality Journal, 11(4), 265-281.

[6] Boehm, B. W. (1988). A spiral model of software development and enhancement. Computer, 21(5), 61-72.

[7] Colby, C. L., Mithas, S., Orlando, T., and Norman, E. (2015). "What Drives Successful Product Development and Innovation in the Software Development Process? The Product Development Success Index (PDSI) “Frontiers in Service Conference, San Jose, CA [online]. Available at: https://www.slideshare.net/ccolby/frontiers-2015-by-3-pillar-cesrockbridge-50735368 and http://productdevelopmentsuccess.com/about [Accessed 28 Apr. 2018].

[8] Daria, K.(2018). New Product Development Process in Finnish Software Start-Ups and University Spin-Outs. Master’s Thesis, LUT School of Business and Management, Lappeenranta University of Technology, Finlandpp.1-95.

[9] Feiler, P. and Humphrey, W. (1993) “Software process development and enactment: Concepts and definitions," in Software Process, 1993. Continuous Software Process Improvement, Second International Conference on the. IEEE, pp. 28-40.

[10] Fuggeffa, A. (2000). Software Process: A Roadmap. In Proceedings of the Conference on the Future of Software EngineeringACM pp. 25-34.

[11] Greer, D., and Ruhe, G. (2004). Software release planning: an evolutionary and iterative approach, Information and Software Technology, 46 (2004) 243–253.

[12] Iqbal, S. Z., and Idrees, M. (2017). Z-SDLC Model: A New Model For Software Development Life Cycle (SDLC).International Journal of Engineering and Advanced Research Technology (IJEART), 3(2), pp. 1-8.

[13] ISO/IEC, Amendment to ISO/IEC 12207-2008 - Systems and software engineering Software life cycle processes. pp 34 - 35

[14] Jakeman, A. J., Letcher, R. A., and Norton, J. P. (2006). Ten iterative steps in development and evaluation of environmental models. Environmental Modelling & Software, 21(5), 602-614.

[15] Janzen, D., and Saiedian, H. (2005). Test-driven development concepts, taxonomy, and future direction. Computer, 38(9), 43-50.

[16] Jonathan, A. (2018). A comparative analysis on small-group task-based learning between software engineering and dance studies at the University of Auckland. Master Thesis, University of Auckland, 1-94.

[17] Kern, A., Kuhlmann, M., Schaad, A., and Moffett, J. (2002, June). Observations on the role life-cycle in the context of enterprise security management. In Proceedings of the seventh ACM symposium on Access control models and technologies ACM, pp. 43-51.

[18] Lamassoure, E., Wall, S., and Easter, R. (2004, September). Model-based engineering design for trade space exploration throughout the design cycle. In Space 2004 Conference and Exhibit (p. 5855).

[19] Larman, C., and Basili, V. R. (2003). Iterative and incremental developments. a brief history. Computer, 36(6), 47-56.

[20] Magana, A. J., Seah, Y. Y., and Thomas, P. (2018). Fostering Cooperative Learning with Scrum in a Semi-Capstone Systems Analysis and Design Course. Journal of Information Systems Education, 29(2), 75-91.

[21] Mitchell, S. M., and Seaman, C. B. (2009). A comparison of software cost, duration, and quality for waterfall vs. iterative and incremental development: A systematic review. In Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement IEEE Computer Society, pp. 511-515.

[22] O'Connor, R. V. (2008) “Human aspects of information technology development," International Journal of Technology, Policy and Management, Vol. 8, No. 1.pp.633-648

[23] Pinto, G., Wiese, I., and Dias, L. F. (2018). How do scientists develop scientific software? An external replication. In 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)IEEE pp. 582-591

[24] Pressman, R. S. (2005) “Software Engineering: A practitioners approach.” Fifth Edition. MacGraw-Hill, pp 429 -430

[25] Salo, O. (2006). Enabling Software Process Improvement in Agile Software Development Teams and Organisations. VTT Publications, pp. 153.

[26] Selic, B. (2003). The pragmatics of model-driven development. IEEE software, 20(5), 19-25.

[27] Soriyan H. A. (2004) A conceptual Framework for Information System Development Methodology for Educational and Industrial Sectors in Nigeria. PhD Thesis. Obafemi Awolowo University, Ile-Ife Nigeria, pp. 53 -54

[28] Trott, P. (2005). Innovation Management and New Product Development. Pearson, England. Pp 23.

[29] Völter, M., Stahl, T., Bettin, J., Haase, A., and Helsen, S. (2013). Model-driven software development: technology, engineering, management. John Wiley &Sons, pp 262 - 263

[30] Wastell, D. (1999) “The Human Dimension of the Software Process," Software Process: Principles, Methodology, and Technology, pp. 165-199.

[31] Zhou, M., andMockus, A. (2011). Does the initial environment impact the future of developers?. In Proceedings of the 33rd International Conference on Software Engineering ACM pp. 271-280