

# Plagiarism declaration literature review examples

[Business](#), [Customers](#)



\n[[toc title="Table of Contents"](#)]\n

\n \t

1. [Critical Thinking and Learning](#) \n \t
2. [Literature Review](#) \n \t
3. [Comparison.](#) \n \t
4. [Conclusion](#) \n \t
5. [References](#) \n

\n[/toc]\n \n

## **Critical Thinking and Learning**

### Introduction

Software and system development can be achieved through several means. The information system development process is largely determined by the need for creating the system. However there are two major techniques that are generally used in software development; the agile software development and the structured techniques. This paper thus provides a review of the two techniques and how each operates. This review is based on research work of scholars in the field of software development and also credible sources that can be used for academic purposes in the field of system analysis and design and also software development.

The paper is composed of several sections this being an introduction, there will be a literature review section that will provide an in-depth analysis and review of both agile software development and the structured techniques.

This section will entail several contributions to the field of software development that has been made by several people with regard to agile

software development and structured techniques. Additionally this section will outline the differences and provide a balanced perspective to software development.

The conclusion section will explore how the agile software development relates with structured techniques and provide a clear distinction based on the findings in the literature review section. The section will conclude by providing how the two techniques can be used in the process of software development. Finally a reflection section on the information on the literature review and conclusion will mark the end of this paper. A reference section on the source of data and information will be appended to this document.

## **Literature Review**

Software development.

This section will provide an understanding of the different methods of software development and indicate the differences of each and finally provide a balanced perspective of the two techniques. There are different methods and approaches that are used to develop software. Structured techniques have been used and utilized by organizations for a long time. A software development model can be defined as being a simplified, abstracted description of a process of software development. The software development process can be described as being the process of translating the needs of a user to a software system/product. This process entails transforming the needs of the users to software requirements and then translating these user requirements to software design (Highsmith, 1999). The user design is then translated to software code. The code is then tested and if it is found to solve the problems of the user, the system is then

installed. Whereas the different development approaches to software have characteristics which are general, like the need to have strong documentation, and the mandate to be traceable, it is worth noting that software development approaches have specific practices which they have to be followed. One of the specific practices is the need to have inspection of the code. The relationship between the software processes and the software development approaches can be taken to be like the relationship that exists between abstract super class and subclass relationship (Sharp, Robinson, & Segal, 2004). In this case therefore, the software development model is used to give specifications of the general techniques and philosophies that will be followed while the methodology overrides and specializes the general specifications with details of the specific practices that need to be followed.

- agile software development.

Previous studies have suggested several factors that contribute to failure to deliver a substantial information system . However there are factors that can be singled out as affecting software development which include shortcomings of personnel, either owners developers or users of the system . According to (Highsmith, 2002) therefore, a development process and approach should focus on the management of interactions and not entirely technology; collaboration among the software development team members and the communication skills through out the development process and the implementation stage of the project. The modern software development has immensely grown from the centralized through the open system to the current commodity system. This has wielded two forces that is business and technical forces that have forced a change in the software development

process to a more dynamic and flexible process.

This saw the emergence of agile software development methods. This process according to Koch ( 2005) is an informal analysis and design approach. Through this approach the design and analysis of software development is done in close collaboration with the client, therefore shifting emphasis that was initially placed on modeling, substantial planning and artifacts.

According to Highsmith (2007) the Agile software development approach manifesto states that "" we are uncovering better ways of developing software by doing it and helping others to do it. Through this work we have come to value:

Individuals and interactions over processes and tools.

Working software over comprehensive documentation.

Customer collaboration over contract negotiation.

Responding to change over following a plan.

That is, while there is value in the items on the right, we value the items on the left more" (Highsmith, 2002).

Herbsleb & Mockus ( 2003) Further notes that to capitalize on this manifesto the following aspects of software development play a major role:

- Individuals and interactions.

The individuals involved in the development process must have multiple skills so as to become a generalist and not only a specialist. However the strength and specialization of these individuals in particular areas cannot be

ruled out. The individuals assembled for software development projects need not be specialized in the various sectors but they can learn through interaction with one another in the development process. The level of interactions and communication between the development team, the client and the end users of the system will ensure development of good software.

- Working Software.

The agile development technique ensures a working software throughout the process of software development this can be used for additional tasks as conferences, presales and customer demonstrations this indicates the confidence of the method used in the software development and this cannot be surpassed by any other means. It is the interactive and the incremental approach that enable the delivery of working software for the customers.

This approach allows the development of software through small and individual working groups and software components that work together in collaboration and high level of coordination to produce complex and large software and systems.

- Customer collaboration.

The unique aspect of agile development is that the specific requirements that are needed by the client to be incorporated in the software cannot be determined beforehand unlike in other development methods and techniques. This thus calls for more collaboration with the client or customer throughout the development process of the software. This facilitates timely delivery of software, easy decision making that is based on consensus, sharing of information and knowledge among team members and the ultimate development of a quality product.

- Response to change.

Following a plan predetermined for a software development will often lead to the creation of obsolete software that may not meet the user requirements. Therefore it is agreeable that no software development can be completely planned from the beginning. Deviating from the original plan is always normal and this arises from new information and discoveries arising during software development process. The users, clients or customers are generally responsible for what gets out whilst the development team is responsible for how this gets out. . However notes that the client can as well dictate how the final product is implemented.

- structured development techniques.

According to Karat and Dayton (1995), agile methods continuously address the changes in the design and analysis process and also technological and business conditions. These factors are critical to business success.

Structured development technique is based on capability maturity model integration (CMMI) and people CMM . Additionally it is also based on heavy methods that are usually based on waterfall lifecycle frameworks.

The structured development technique basically consists of several procedural phases which include: requirements analysis, design, testing, development and implementation. These phases are mostly iterative and thus should include each and every aspect and understanding of the user requirements and tasks, context, and the environment in which the application is to be used (Sharp, Rogers, & Preece, 2007).

An important aspect of software development is understanding how the users of the system will use it practically and embodying this aspect through

various mechanisms. The methodological approaches to structured development have been based on human computer interface. Provided and early reference to structured development, they described three steps including identifying user tasks, testing and interactive design. Others have contributed to these general framework among them Button and Dourish who describe how structured techniques have been developing through three important pivotal influences on software development this include addition of cognitive scientists to design teams to give psychological input on design, redesigning the design process on usability and methods, and inserting engineering methods and techniques with specific written documentation.

### **Comparison.**

The agile method of software development basically refers to the repetitive process where software is developed and given to the owner (customer) in developmental increments. In this approach the system functionalities of individual increments are not planned prior to system development. What to include in an increment entirely depends on the customer's priorities. Due to the likelihood of the change in the owner requirements it is important to put in place a flexible plan that house the changes.

There are two approaches to agile planning according to Schwaber (2004) and Beck (2000) corresponding to the phase in plan-driven development and development planning. These are:

- Release planning that looks ahead for a couple of months and determines the features that the system should be in the release of a system.
- Iteration planning that is short-term and concentrates on the planning of



the subsequent system increment which is normally two to four weeks of team work.

The two software development approaches differ in that the plan driven approach requires that one gets to know what the project will achieve way before going on with the project. In this approach, the client would like to have all the requirements beforehand. There are tools which are used to estimate the cost of the project and the duration that the project will take. This process is done and achieved by making use of software that is available in the market. The tasks are allocated way before the project is undertaken.

On the other hand, agile development process is one where there is an interaction in the processes as the development is carried out (Highsmith, 2002). In this method there is comprehensive documentation.

Comprehensive because there is need to have all the processes and activities which take place in each phase. In the course of the development process, the customer is updated and shown what the project will get. There is collaboration with the customer over the given period of time. This approach is seen to be better in managing change as the changes are integrated in the course of project implementation. It is easier because the client is involved in the development phase (Sharp, Robinson, & Segal, 2004). With this, therefore, the client is updated and is involved in every stage of the development.

When selecting for an approach to make use of, it will depend majorly on the situation that is being worked upon. The complexity of the problem will also be looked into. In the structure development approach, there is the use of

structure to control and mitigate risks that arise in the course of the project. On the other hand, agile development approach makes use of flexibility to control risks. This approach is because as the project is underway, the client might not accept some processes and in this case, the project team will just change the whole issue.

The two methods are continuously related with each other. They are found in the two extremes. It can be very agile or very controlled. The general information that is available for these two approaches is that in structured technique driven, there is a formal communication where the process tries to get what will be done at the end of the project. It tries to predict what the whole project will achieve. On the other hand, the agile development tries to have some informal communication in the process where the requirements of the project are communicated in advance.

With structured development, there is the emphasis that is given to the use of artifacts and milestones. Milestones are the achievements that will be seen in the course of the project. The communication that is seen here is structured. This is where there are meetings that take place in the course of implementing the project. In this regard, the project is heavily governed and controlled. There is separation of roles and well defined duties for the various people in the project.

Agile development is people oriented. This is because there is constant communication with the client. The client is constantly updated of what is happening in the project. Although the communicating is informal, but there is constant flow of information to the client. There is low project ceremony taking place in the project.

## **Conclusion**

Agile software development ensures there is good management of user communication. This is because the user is involved in all the stages of the development and will raise any questions that are required before the completion of the project. This approach requires that the user gives their requirements. There is much flexibility seen here as the user makes sure that the right product is delivered. The involvement of the users is also important as this ensures that the right product is delivered. The parts which are not part of the requirements are eliminated.

There is testing of the project throughout the development phase. This serves to make sure that the right software has been achieved and also to serve as a sign of quality assurance. With the testing, the right quality is assured.

Structured approach allows the whole project to be predicted. The fact that all the requirements and the entire process is planned for at the beginning is a good process which assures that the budget will be adhered to. The right project information is attained and documented at the beginning of the development process. It therefore allows for the cost to be well budgeted.

In conclusion it is evident that there is a prevalence of large and distributed projects that are characterized by great deal of change. This nature of projects that are large and distributed and involve a great deal of user involvement pose a great challenge to both agile and structured software development methods. Neither of the methods alone can effectively address this challenge. Applying the methods simultaneously is possible but challenging to any organization. This is attributed to the different and often

conflicting assumptions and principles that are inherent in the agile and structured approaches. Even though there has been the urge to balance agile and structured approaches, how the approaches can be effectively balanced in the management of complex, large and dynamic projects is still a puzzle. Therefore there is need for structured and agile software development approaches to complement one another. Structured approach controls and coordinates to provide a disciplined organizational infrastructure that is needed for the agile approach to be effective. On the other hand the agile approach with its iterative and fast turn around circles enables the structured approach and the control processes to learn and efficiently adapt to the changing conditions of the development environment. Structured and agile can be harmonized within the same project, especially when the project is large, strategic, time-sensitive, and distributed.

## **References**

- Cockburn, A. (2002). *Agile Software Development*. Boston: Addison-Wesley.
- Fernandez, D., & Fernandez, J. (2008). Agile Project Management—Agilism versus Traditional Approaches. *The Journal of Computer Information Systems* , 49 (2), 10 - 17.
- Fitzgerald, B., Hartnett, G., & Conboy, K. (2006). Customizing Agile Methods to Software Practices at Intel Shannon. *European Journal of Information Systems* , 15 (2), 43-53.
- Gould, J., & Lewis, C. (1985). *Communications of the ACM*. Designing for usability: Key principles and what designers think , 360-411 .
- Hansson, C., Dittrich, Y., & Randall, D. (2004). Agile processes enhancing
- <https://assignbuster.com/plagiarism-declaration-literature-review-examples/>

user participants for small providers of off-the-shelf software. Fifth International Conference on Extreme Programming and Agile Processes in Software Engineering (XP2004) (pp. 6-10). Germany: Garmisch-Partenkirchen.

Herbsleb, J., & Mockus, A. (2003). An Empirical Study of Speed and Communication in Globally Distributed Software Development. *IEEE Transactions on Software Engineering* , 481-494.

Highsmith, J. (1999). *Adaptive Software Development*. Cambridge: Dorset House.

Highsmith, J. (2002). *Agile Software Development Ecosystem*. Boston: Addison-wesley.

Jiang, J., Klein, G., & Balloun, J. (1998). Perceptions of System Development Failures. *Information and Software Technology* , 39, 933 - 937.

Karat, J., & Dayton, T. (1995). Practical education for improving software usability. *the conference on Computer-Human Interaction*, (pp. 162 - 169).

Koch, A. (2005). *Agile Software Development: Evaluating The Methods For Your Organization*. Boston: Artech House. .

lyytinen, k. (1987). problems and solutions. *different perspectives on information systems* , 19 (1), 5-46.

Sharp, H., Robinson, H., & Segal, J. (2004). *eXtreme Programming and User-Centred Design*. 18th British HCI Group Annual Conference: HCI2004 Design for Life (pp. 122- 145). UK: Leeds Metropolitan University.

Sharp, H., Rogers, Y., & Preece, J. (2007). *Interaction Design: Beyond Human-Computer Interaction* (2nd edition). Wiley.

Vinekar, V., Slinkman, C., & Nerur, s. (2006). *Can Agile and Traditional*

Systems Development Approaches coexist? Information systems management , pp. 31-42.

### Reflection

Software development is a process that is conventionally achieved through a set of steps that must be followed to the latter to realize the final product. These steps vary depending on the approach taken to build the software but all aim at producing software. This objective can be achieved through various means depending on the need and availability of resources. Another major contributing factor is the need for user involvement in the entire process so as to achieve user friendly software that will be acceptable by its users.