Benefits of using the MVC architecture design pattern in agile development environment

Laila.A.H. Bubaker, AISHA.M.YOUSEF, Asma Abd Aljalil

Department of Computer, Faculty of Science, Omar Al-Mukhtar University, Albayda, Libya.
Benefits of using the MVC architecture design pattern in agile development environment

Abstract
In the field of software development, there are two areas constantly evolving during the software engineering process, which are software development methodologies and software architecture design patterns. It has become indispensable to shed light on them in terms of the suitable selection of developers to employ the appropriate types of them to obtain more efficiency of their work together, improving the quality of the software applications, and to produce software that has a high level of accuracy.

The aim of this research is to determine and define the features that MVC architecture design pattern and agile methodology work by software engineering and development process, and to investigate how those features, that existing in both, could assist them to work in harmony without conflict in the way of work, and whether they support each other when they employ together on one software project. Also, how that compatibility will increase their ability to improve the quality of software products.

Two types of research methods have been used in this paper. Firstly, we used previous studies and experiences of researches to collect and determine the most important criteria that contribute to the success and excellence of software products, which are the advantages granted by the use of both Agile methodology and MVC architecture. The second method is distributing two questioners to participants that their experiences are up to 5 years, the first one is for the MVC framework and the other is for the Agile process, both of them are distributed to investigate the extent of the existence of these criteria from the point of view of experts who participated in answering these questionnaires. We used the results of both questionnaires and previous research to study how to obtain the advantages of using the two technologies, employing them, and applying them together to increase the efficiency of the resulting software product and that they support each other in practice.

Keywords: software development methodology, software product, Agile process, software architecture, (Model, View and Controller) design pattern.
المملوء:

ليلي عبد القادر حدوث، عائشة محمد يوسف، أسماء مصطفى محمد

في مجال تطوير البرمجيات، هناك مجالان يتطوران باستمرار أثناء عملية هندسة البرمجيات، وهما منهجيات تطوير البرمجيات ونظام تصميم هندسة البرمجيات. كان من الضروري تسليط الضوء عليهم من حيث الاختيار المناسب للمطور لتوظيف الأنماط المناسبة، معهم للحصول على مزيد من كفاءة عملهم معًا، وتحسين جودة تطبيقات البرمجيات، وإنتاج برمجيات ذات مستوى عالي من الكفاءة.

هدف من هذا البحث هو تحديد الميزات التي يتعامل بها نظام تصميم معماري MVC (Model, View and Controller) من خلال هندسة البرمجيات وعملية تطوير البرمجيات، والتحقيق في كيفية مساعدة الميزات الموجودة في كليهما على العمل بانسجام دون تعارض في طريقة العمل، وما إذا كان كلاهما يدعمان بعضهما البعض عندما يتم تطبيقهما معا في نفس المشروع.

تم استخدام نوعين من طرق البحث في هذه الورقة، أولاً، استخدمنا الدراسات والخبرات البحثية السابقة لتجميع وتحديد أهم المعايير التي تساهم في نجاح وتميز منتجات البرمجيات والتي يمنحها استخدام كلا من منهجية Agile ومنهجية MVC. يتم استخدام نوعين من طرق البحث في هذه الورقة، أولاً، استخدمنا الدراسات والخبرات البحثية السابقة لتجميع وتحديد أهم المعايير التي تساهم في نجاح وتميز منتجات البرمجيات والتي يمنحها استخدام كلا من منهجية Agile ومنهجية MVC.

word:

منهجيات تطوير البرمجيات، المنتجات البرمجية، عملية Agile، هندسة البرمجيات، نمط تصميم MVC (Model, View and Controller)
1. Introduction

Recently, there is strong competition in the software industry to produce professional and efficient software products in less and appropriate time. A successful design and implementation of any software system is based on complete user requirements, system requirements specification, and appropriate selection of software methodology to manage and implement those requirements accurately.

It was desirable to think about the study of selecting the appropriate methodologies for working with Software architectures of all types.

In this paper, we discussed whether the Agile methodology as one of the most popular software methodologies [1] could work more effectively with a specific type of architecture like MVC architecture to produce efficient software system.

1.1 MVC (Model, View and Controller) design pattern

It is Software architecture which divides the software into three functional layers which are (Model, View and Controller ) and it is commonly used as design pattern, particularly in the object-oriented application system[1].

The MVC concept helps to write more structured code and provides variety of object-oriented libraries, this facilitate the tasks of programmers and help to make the process of developing large software applications speeder and easier with high quality achievement[2].

This architecture depends mostly on the relationship between three layers of its structure:

- (Model) The Date Business layer.
- (View) The User interaction or client side.
- (Controller) The Business logic (the server side).

The Model layer represents the business logic of the system and access to the relational database, It encapsulates the data access methods (databases, files, etc.) and provides a reusable class library. It is usually built with data abstraction, validation and authentication in mind. In addition, the model consists of classes of the domain, these domain objects often encapsulate data stored in databases, but also include the code used to manipulate that data.
The View layer is the user interface that allows users to interact with the system, and it manage the display of the data and the interaction of the user with. The technologies mainly used in views are HTML, CSS and JavaScript. As a general rule, a view should never contain elements belonging to the application logic, in order to facilitate the work of the designer with it[1, 3].

The Controller layer is a component on a web server that is responsible for the management of the event, these events can be triggered either by user interaction with the application or by a system process. The controller accepts the requests and prepares the data for the response. It is also responsible for building the form of this response. The controller interacts with the model to retrieve the required data and create the view [1, 3].

![MVC architecture](image)

**Figure1:** MVC architecture.

### 1.2 Agile Methodology

Agile methodology is a set of software development methods based on iteration and increment development, it uses iterative development that is widely used in a variety of industrial projects as a lightweight development method that can meet the changes of requirements. The four main characteristics that the fundamental principles of all agile methodologies are: Planning, iterative and scalable development, fast and flexible response to change, and improved communication[4, 5].

The short iterations required for efficient product delivery are used; moreover, the use of agile practices is the most effective to improve the quality of a software project[3].

The agile methodology includes different types of methods which include Scrum, Extreme Programming (XP), Adaptive Software Development (ASD), Crystal Clear, Lean Software
Development, and Dynamic Systems Development Method (DSDM) Crystal, Feature Driven Development (FDD), etc. [4].

Figure2: Agile development methodology.

The paper is organized as follows: Section 1 Introduction: provides an overview of agile methodology and MVC design pattern. Section 2 Research Methodology: explains the different methodologies used in this research. Section 3 Literature review: specifies the features of both agile process and MVC framework and illustrated different types of previous studies on software architecture in agile methodology. Section 4 Questionnaire analysis: analyses in detail the data and information obtained from the two distributed questioners. Section 5 Discussion: discusses and explains in detail how researchers obtained the results of this paper Section 6 Result: illustrates all results obtained from the research methodologies and discussion section. Finally Section 7 Conclusion and further research of this work.

2. Research Method

In this study, the researchers followed two research methods, the first one is determining and studying standards that help in raising the efficiency of software such as (Flexibility, Changeability, Modularity, Supporting teamwork, etc.) and understanding the mechanism of its work in both technologies, and this is through previous research, studies and practical experiences. Moreover, this method assisted us to understand and clarify the respondents’ brief answers and comments that we obtained from distributing two types of questionnaires, and it is the second method we used in this research.
The second method is distributing questionnaires, there are two types of questionnaires that have been used, the distribution was to two types of samples that have been chosen from participants who are professionals or from an academic environment, and this selection was based on two criteria. The first criterion with the questionnaire for MVC (Model View and Controller) architecture design pattern, it distributed to participants who have considerable experience of development of software applications utilizing MVC architecture in different software methodologies, the sample consisting of 23 developers.

The second questionnaire was distributed to specialists of using agile methodologies, the sample consisting of 26 participants.

The reason behind distributing two types of questionnaires is to investigate the ability of both MVC and Agile to support the most important criteria of developing an effective software system and the capability of both when they work to gather to produce high-quality software. These criteria such as flexibility, reusability, modularity, changeability, Teamwork support, maintainability.

3. Literature review:

3.1 MVC Architecture features

3.1.1 Changeability and Flexibility:

One of the important features that MVC architecture provides is changeability, this criteria is a result of the separation of the presentation and business logic layers because this separation makes the modifications and the improvement changes in the application possible and easy because of the ability to make changes on each component separately and without making any changes on the other by reusing and modifying the existing codes[2]. Moreover, there is a group of researchers, who supported that vision when they observed that the same model could be used for multiple views because the separation between the view and model allows multiple views to use the same model. As a result, the application models are easy and flexible to make changes, execute and maintain because all models can be accessed through these components[6]. In addition, according to other researchers MVC framework is able to build high flexible applications. This design pattern has flexibility in creating different views for one application, otherwise it enables programmers to use different types of view which can be accessed by the same code on the server side. Those benefits were resulted and
obtained by this researcher when he applied MVC design pattern in the J2EE platform and Stratus framework[3, 7].

In other study, authors supported the above features when they proved that MVC could increase the flexibility and reusability of the application used. This architecture enables developers to assign multiple views to a model that provides different presentations; they can also create new views for a model without rewriting it[8].

3.1.2. Supporting Teamwork contribution

What makes the MVC framework so useful for a rapidly changing environment is that it allows for parallel development.

This means that a developer can work on the presentation layer (the view), while working again on the logic of the console and a third can work on the logic of work in the model. This decomposition allows backend and frontend developers to simultaneously develop their code without having to wait for the other to finish.[8] In an environment that must perform tasks quickly and efficiently, while maintaining a relatively small team, this is of great value. This benefit is a result of dividing the developers’ works focus and responsibilities between the three components; for, example, the controller programmers concentrate on only transactions and the view controller concentrates on the interfaces; consequently, that can reduce the development time and increase the team work efficiency [7].

3.1.3 Increasing customer satisfaction

With MVC it could be easy to make changes according to customer feedback, this architecture is flexible enough to accept customer new and changing requirements and add these modifications rapidly and without difficulties because of as mentioned above to the ability of MVC to accept frequent changes and add new requirements all the time of development and apply them easily[1, 9]. This flexibility assists in reducing the risks and increasing customer satisfaction on the last version of the software product [9].

3.1.4 MVC Efficient modularity:

MVC framework could enable software applications to obtain this feature because the interface and the transaction layer are clearly separated to allow developers to make changes and modifications in the interface design of the application without making any changes to the model or to the controller layers[1]. To add further evidence on availability of this benefit
in MVC applications, there are studies stated that managing components such as adding or
deleting in MVC architecture can be efficiently done without a negative impact on the
application. Making changes to one component does not affect the other components[3, 6,
10].
The requirements of software applications can be changed and expanded over time, it’s
important to be able to try and building modular applications, this is a feature that MVC
provides to developers could easily separate all of the components of the application and
allow programmer to activate or disable them depending on client needs. This separation
allows developers to write more modular and maintainable code, which by definition will be
completely separate from the parent application, other modules and components[1, 6, 10].

3.2 Agile methodology features

3.2.1 Changeability and flexibility:
An agile process is typically lightweight and changes to the system-under-development.
Using agile methodology during developing software could help developers to make changes
over the time of development. Requirements can change over a period of development and
this agile process takes these changes into account. This is one of the most significant
advantages of an agile process over traditional software development processes[11]. The
agile process is iterative, it allows projects to be developed incrementally, and it greatly
reduces the risk of the end result to be significantly different from what was envisioned at the
beginning of the project[12].
New functionality can be added easily using multiple iterations, this is one of the most
important advantages of an agile process over traditional software development processes[11]. The Agile s is an iterative process, it allows the progressive development of
projects, after every iteration developers send an initial release of the application to obtain
client feedback, if the feedback requires changes in the system requirements(modifying or
adding functions), agile application has the flexibility to respond to requests for any
reasonable Changes of all stakeholders[4].

3.2.2 Agile support teamwork Contribution:
Teamwork in agile processes works differently from the traditional software methodology
such as waterfall methodology in which the team members work separately. Also, agile
process involves developers, testers and business people to work together during all project stages[5].

supporting teamwork is one of the Agile manifesto principles Agile methodology typically requires a team of developers with high responsibilities and the excellence of combine new requirements, integrate, changes and make improvements to enhance the system[13].

In the projects that have changes in requirements or customer feedbacks, also, testers could send modifications; all of these required changes need quick development and response. As agile is an iterative approach and divides the project into short cycles, so the team can collaborate together quickly to take these changes into account and produce responses effectively in suitable time[14].

3.2.3 Increase customer satisfaction

Agile process is an iterative process in which changes can be made easily to increase customer satisfaction. It focuses on efficiency and speed of workflow development and relies on reliable and timely feedback between clients and programmers[15].

and it dependents mainly on the regular meeting and frequent interaction between developers and the stockholders, this increases customer satisfaction because the teamwork release and deliver the system frequently and take the users feedback to make changes and integrate new requirements, therefore this enhances the efficiency and the quality of the final release of the system[12].

In agile methodology, the software development team not only responds to end-user feedback and requests, but also provides the end-user with a frequent and realistic view of the features in every iteration that can be included in the product within a specific timeframe. These flexible communications, feedback and practical modifications help to achieve a high degree of success compared to the traditional cascading approach to software development[15].

3.2.4 Modularity

The modularity of projects can be achieved successfully by using one of the different types of agile, that could be applied to any project with strict deadlines, complex requirements, and a degree of exclusivity[15].

Agile types have been used for every levels from developing traditional software to designing complex new integrated systems, they mainly depend on the process of incremental
development[5]. In scrum method which is the most commonly used type of agile methods, the entire development cycle is divided into a series of iterations where each iteration is called as a sprint in every new iteration there are new functionalities that need to be added and combined with the functionalities of the system[12]. In addition, scrum focuses on increasing the team's ability to quick delivery and fast responding to integrating functionalities[15].

3.3 Previous studies on Software architecture design in agile methodology

In one of previous researches, the authors reported the design and implementation of a time management software (TMS), they indicated that the system was built by best practices of model view controller-based software engineering and agile implementation methodology as appropriate choice and they also pointed this system has been successfully implemented, tested and has practical use in both industry and academia in terms of, efficiency, usability, and cost-saving[1].

In other study, a software implementation has been applied and achieved by apache Struts, an open source java web framework which is, a package of Java classes and JSP tag libraries which use a model-view-controller (MVC). The search concluded that, Struts can certainly be used for technological applications of the agile methodology. Struts support facilitates code development for an agile methodology process and it can be adapted into an iterative and incremental development method. Because of many reasons illustrated clearly in the article, such as the ability to respond to changes easily and adding new requirements because of frequent delivery of agile application, for these frequent modifications, the business logic of the application is encouraged, view and logic are divided to separate layers, actions are configured and the application can be modular[16].

And in a third research, The authors conducted a detailed comparison between variant types of architecture design patterns, one of them was MVC architecture. Based on the results of the evaluation, the MVC architecture was considered a suitable type for use in the development of agile methodology[17].

In another type of papers, the researchers focused on agile methodology as a popular software methodology because it is goal is quick efficient delivery of software products and its ability
to achieve frequent and quick changes on time, as they mentioned. Also they provided an overview on software architectural design and its specification that is related in agile methodology and compatible with it. These specifications such as changes in software requirements are accepted, effective teamwork cooperation to handle these changes because the agile process is frequent and incremental development, and because of agile ability to work on large and complex software projects, it needs an organized and managed software architecture[18].

4. Questionnaires Analysis:
This section will illustrate the questionnaires’ data analysis and results. The answers gathered from both questioners will be analysed in this part of this research, all the questions in both questionnaires were centred on the main criteria of developing and maintaining a high quality and efficient software system.

Firstly, we will begin with the analysis of the MVC questioner, which has 23 participants. The respondents’ opinions, as clear in table 1, were supported strongly with eighteen of them agreed on the ability of the MVC framework to build flexible application and five did not refer to it.

The result in table 1 showed obvious consensus with 19 agreements among developers on making the required changes easily on any component without any risk of affecting the other components of the MVC application; though 4 respondents disagreed with them. The most Participants in this table supported strongly the existence of modularity in MVC applications with 19 agreements.

Teamwork supporting: MVC framework contributes effectively to increasing and enhance teamwork collaboration, this is strongly obvious from the answers as you can see in the table 1 20 out of 23 answers were agreed on this point.

Other question for contributors was about whether the development speed and performance of MVC architecture are affected by the software methodology used with, the answers, as illustrated bellow, 15 were agreed and 4 were disagreed.
Table 1: The number of answers obtained from the MVC questionnaire.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Yes</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexibility</td>
<td>18</td>
<td>5</td>
</tr>
<tr>
<td>Changeability</td>
<td>19</td>
<td>4</td>
</tr>
<tr>
<td>Modularity</td>
<td>19</td>
<td>4</td>
</tr>
<tr>
<td>Teamwork support</td>
<td>20</td>
<td>3</td>
</tr>
<tr>
<td>Dependency on software methodology</td>
<td>Mostly Agree</td>
<td>Slightly Agree</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>6</td>
</tr>
</tbody>
</table>

Secondly, we will illustrate the analysis of the Agile process questionnaire to test its ability to meet these important criteria. 26 subjects participated in this study. A basic criterion for selecting this sample was that, they were worked and developed software application using agile process with different software architecture such as (web form and MVC).

As illustrated in the chart below (figure 3), all of participants have more than three years of experience of working with agile, 56.3% of them have 5-10 years of experience and 25% more than 10 years.

Figure 3: years of experience for agile questionnaire participants.

Flexibility and changeability features are strongly available during the life cycle of agile process, according to the answers of the participants as obvious in table 2, this methodology is flexible to help developers to accept new requirements, add or update functionalities and make changes on a software system easily on any stage of the software life cycle, 16 of the answers were agreed on both features, while the answers against these opinions were very less by 7, 6 answers as shown in table 2.
Also, one of the questions the inquiry was about the ability of agile process to support working on teams during the software life cycle and achieve cooperation between them effectively. The answers were supported strongly the availability of this feature with approval of 15 of the participants between mostly and slightly agreement as mentioned in table2 and only 2 repliers gave negative answers.

As well as, the other question was about the availability of modularity feature in terms of the great flexibility of agile to adding new requirements and features to the application easily because as most repliers commented, its nature of work accepts the changes without negative effects because of dividing the project into small parts with continuous and overlapping iterations, the agreement answers were 12 in compared with the disagreement answers which were 7 answers.

Using agile process contributes to increasing customer satisfaction, this was confirmed by a large number of repliers agreed on that, by 21 agreement answers, 14 of them mostly agreed.

Dependency on software architecture: The question was whether agile process effectiveness was influenced by software architecture work with, 20 answers approved that while one answer denied it.

Table2: The number of answers obtained from the agile questionnaire.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexibility</td>
<td>10</td>
<td>6</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>changeability</td>
<td>12</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>modularity</td>
<td>7</td>
<td>5</td>
<td>7</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Teamwork support</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Dependency on software architecture</td>
<td>9</td>
<td>11</td>
<td>5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Increase customer satisfaction</td>
<td>14</td>
<td>7</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>improve the accuracy of software system</td>
<td>13</td>
<td>11</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
5. Discussion:

In this section, we will interpret and combine research methods results to obtain a clear interpretation and clear results, and discuss how MVC architecture and agile process are related and can be used together effectively and how to combine their features. The researchers will demonstrate the results were extracted from the analysis of both questionnaires also use the explanation and clarification obtained from previous studies, that helped us to understand the questionnaires repliers’ answers and to give more explanation to assist us to study how can combine these benefits of both frameworks to improve software engineering and development stages and gain and deliver a reliable software product.

Firstly, from the analysis of MVC questionnaire, these graphs below in figures 4, 5, 6 illustrate the great agreements of the repliers on the availability of these features (flexibility, changeability and modularity) in MVC applications through their experiences of using this architecture, these points also clarified in details in the analysis of the MVC questionnaire section. To confirm that, from previous studies and several of the participants’ comments that supported their answers, we came to this result, these features can be easily carried out because of the clear lines that separate the responsibilities of each component, for example, it is easy to create different views on the same system without introducing any change in the other components[7, 9]. Another reason behind these confirmations was the possibility of adding more services that might be needed according to the users’ feedback or the companies’ wants, and the ability to modify the application functionalities and the ability of conducting rapid and easy response on the changes that come from customer feedback regarding to application functionalities or user interface. Those two points could be behind the increase in application flexibility, in the literature review those points are supported strongly and explained in detail[8].
On the other hand, from the analysis of agile questionnaire these graphs illustrates the variance level of agreement answers from the lowest level (1) to the highest (5), also it shows clearly and strongly of the availability of (flexibility, changeability, modularity, team work support) features when using agile process, with noting that there are some differences in the levels of availability of these criteria. And to clarify more as you can see the availability of flexibility and changeability is the highest rate of strong agreement responses, whereas team work support and modularity is the lowest.

It is clear from the above analysis and explanation the compatibility between these two technologies because they provide almost the same features in them that supports each other, which will lead to an increase in the software that will be built by both of them. For instance; Agile model accepts change and the adding of new requirements, throughout all the project development period, and in the same context, the MVC architecture supports the same feature and can be easily implemented without causing harm to the rest of other components or affecting it negatively.
Secondly, in figure 8 the result was obvious and strongly supported by their approval of the presences on supporting agile to team work during their experience with agile process by 92% of agreements that divided between 52% were strongly agree and 40% were agree. Comparing with the result of MVC questionnaire as you can see in figure 9, we found strongly this feature is also available, and it is one of the important features of MVC architecture by 65% of agreements.

Also, to increase proof to this point the researchers in the literature review mentioned to the importance of this point in both agile process and MVC framework, as they said agile is supporting working on a team because the agile environment where tasks are required to complete quickly and effectively, according to its nature, the project is divided stages to small iterations the developers work to gether in all iteiraions, and in each iteration there are new requirements need to be added and new tasks have to be done[5,13,14]. That could be one of the reasons to make MVC framework so beneficial for agile environment, if MVC is the architecture applied with agile process in one project, it will help developers adequately because it strongly supports this point as well, that was mentioned obviously in the figure 9, in the questionnaire analysis section and in the literature review section, because MVC allows parallel development, this means one of the developers can work on the view layer, the second one can work on controller layer, and the third one can work on model layer, this separation allows back-end and front-end developers to develop their code simultaneously without the need to wait for the others to complete their tasks[7, 8].
thirdly, the both frameworks, as clearly explained in the literature review, how they can increase customer satisfaction, because MVC application is able easily to accept any changes comes from customers feedback\[1, 9\]. Also, during agile application life cycle the customer is a member of the teamwork to give the feedbacks on every iteration and the developers apply the required changes. In addition, agile process concentrates on involving customers as contributors in all stages of the agile processes, and this increases the ability to respond to their feedbacks and meet their needs. Because of this involving of the customer and frequent contacts assists to increase customer satisfaction ,because of the high accuracy of the last software version, this leads to an improvement in the software functionalities and quality \[5, 12, 15\]. Also the percentages in figure10 below shows the great agreements upon that.

Fourthly, as illustrated in figure11, on this question, which was about the efficiency of MVC framework, whether it depended on the software methodology that the team of developers follow during the life cycle of the application they work on, the responses were mixed between “strongly approve”, “slightly approve”, “neutral” and “disapprove”, but the greatest number strongly supported this opinion.
However, when we asked the experts about whether applying agile would be easier and more effective with applications that have a clearly divided structure for their components and the architecture that supports the change in requirements, the answers were illustrated in figure 12 with great agreement on this point.

In addition, to confirm this result, we asked them, which of these software architectures (one tier architecture or N-tier architecture) Agile can work better with. As it is obvious from figure 13, the experts agreed strongly on the N-tier architecture, of which MVC is one of its types.
Finally, in the previous studies on software architecture design in agile methodology, the studies confirmed that, using Agile methodology in MVC architecture is an appropriate choice to develop quality software systems [1, 16-18].

6. Result:

This section summarized the results of the discussion section which presented and analysed these results in details. These consequences of this research are illustrated below:

- This paper defined the main criteria and features that assist to build efficient software application, which are (Flexibility, Changeability, Teamwork support, Modularity).
- This research proved the availability of these features in both the agile process and the MVC framework, but with different mechanisms.
- The researchers verified that the agile methodology is a suitable methodology to apply with MVC architecture, and both together can bring many features to software applications.
- They confirmed that agile methodology performs more effectively with N-tier architecture such as MVC architecture, also confirmed that the effectiveness of MVC framework depends on the methodology that is applied with.
- Both of them accept the customer feedbacks even in the last stages of the project, so that increases defiantly the customer satisfaction on the software product.
- One of the most important results demonstrated by this research is that both technologies support each other strongly, and when applied together, they will provide these advantages more strongly and work to produce high quality and efficient software.
7-Conclusion

To conclude, this research focused on the importance of employing suitable software methodology with the appropriate software architecture that resonates with it, because the right and reasonable choices could have a positive and significant effect on the software product. This paper defined separately the features granted by the use of agile methodology and MVC architecture, and showed how they are compatible when they work together, to increase the benefit of their advantages to deliver timely and accurate products.

Future research:

This research was carried out to explore the efficiency of using MVC architecture with agile process in developing software applications. For extending to a research there are several ideas could be researched; for example, applying MVC architecture with different software methodologies and comparing the results to obtain the most effective type of them, that can facilitate the selection process to reach the best results.
Reference


6. Selfa DM, Carrillo M, Boone MDR, editors. A database and web application based on MVC architecture. 16th International Conference on Electronics, Communications and Computers (CONIELECOMP'06); 2006: IEEE.


10. Kristály DM, Craciun AV, PELCZ A, TRUICAN I. MVC ARCHITECTURE IN WEB APPLICATIONS DEVELOPMENT.


