

# The Effect of Using Model Driven Architecture Technique in Reducing the Cost of Information System Development

MOHAMMAD HUSSEIN ALSAIDOMAR

Higher Institute of Energy, The Public Authority For Applied Education and Training, KUWAIT

---

**Abstract:** The cost of information systems development is one of the basic factors that the company considers when making a decision for developing these systems. The use of traditional methods in the system development is costly for the company. So that using the Model Driven Architecture (MDA) technique will reduce the cost of building the information system of the company on the one hand and reduce the cost of developing the system as a result of the rapid change in the system requirements on the other hand. The research focuses on studying the amount of reduction of the cost that can be achieved as a result of using the model driven architecture technique. The research found that the use of this technique will reduce the phase of writing code by up to 70% of the cost of writing code. Several factors control this rate including the size and complexity of the systems and the degree of completeness of the tool used in the transformation as well as the experience of the work team to deal with this tool. Also, the research found that it is possible to reduce the cost of documentation and testing.

**Keywords:** System Development Lifecycle (SDLC) - Model Driven Architecture (MDA) - Unified Modeling Language (UML) - Model Transformation Tools.

---

## 1. INTRODUCTION

Z. Mansor et al. [1] see five factors that are most influential on the success of systems development: engagement of users in the development process, executive management support, clear articulation of requirements, appropriate planning and logical expectation. Noor Habibah et al. [2] see that previous work and technical skills of the work team are one of the most important factors that influence the success of system development. While many specialists agreed that the project manager (team leader) plays the most vital role and he is the most important factor in the success of the system development [3, 4,5,6]. In order for the project manager to make the appropriate decision regarding systems development, he must decide on the tools to be used during the system development lifecycle, starting from identifying the tools used in the analysis and design to the system platform, and thus he shall determine whether the used analysis and design tools will be traditional or modern, like the tools used by model driven architecture.

Before talking about the phases of the system development lifecycle, the time, effort and cost required to estimate the rate of each phase, and before talking about the methods used in estimating the cost of system development, and at what phase this cost will be reduced when using MDA technique, and before talking about MDA technique, it is necessary to illustrate the concept of software for large, medium and small enterprises.

Large enterprise software is an integrated software product that perform all related phases, steps and processes of the business in order to enhance internal coordination and collaboration of work within the company, while small to medium-sized enterprises software is that intended for use by a certain department in a company or by a number of individuals whose work is not linked or interdependent. This research will focus on software for SMEs.

This research will identify the different ways of estimating the cost of developing information systems and the factors influencing the estimation of this cost. The focus will be on the cost of developing the implementation, testing and documentation phases. When using traditional methods in the information systems development, we find that the cost of

the implementation and testing phases is high, and to reduce this cost the researcher will use MDA technique for developing the system.

The researcher will present a number of tools used in the model transformation process (code generation). Three tools will be selected and used in the code generation process and a new transformation tool under development will be proposed as a transformation tool.

In order to know the amount of savings in cost achieved by the use of the three tools, six developed information systems will be selected and the number of lines the programmer can save writing due to using the three transformation tools will be counted, in addition to studying the possibility of generating system documentation automatically. The results of the three tools will be compared to assess the effectiveness of using the model driven architecture technique.

## 2. RESEARCH PROBLEM

Information systems development decision mainly relies on the cost of developing information systems. Furthermore, the use of traditional methods to develop the system is expensive to the company or organization for the following reasons:

- High cost of information systems development.
- Ongoing change in the requirements of companies or organizations and thus the need for new information systems to meet the new requirements, resulting in high cost to cover these requirements.
- Rapid technological development and the consequent emergence of modern techniques and tools that benefit and evolve the system if it is utilized.

The development of a low cost information system on the one hand, which can be easily developed to meet the requirements of companies and organizations on the other hand is a key and essential need for companies. So, what technology can help meet these needs? and what is the effectiveness of using this technique?

## 3. RESEARCH IMPORTANCE

The importance of this research signifies in its treatment of the information system development cost reduction issue. Cost reduction in general is of interest for any organization or company for the resulting increase in profits, in addition to providing a competitive merit that enables the company or organization to meet all the requirements that may arise in the future, as well as keeping pace with the technological developments.

## 4. RESEARCH OBJECTIVES

This research aims to study the cost of information systems development in traditional ways and to determine the cost of each phase of the system development, and to define the phases which cost can be reduced using the model driven architecture technique. The research also aims to study the effectiveness of using model driven architecture technique in reducing the cost of information systems development through using more than one MDA tool and to study the amount of savings achieved by each tool separately.

## 5. RESEARCH METHODOLOGY

In this research, the cost of information systems development in traditional ways will be presented, then the model driven architecture technique will be presented. Next, the benefits of applying MDA technique to reduce the cost of information systems development will be presented.

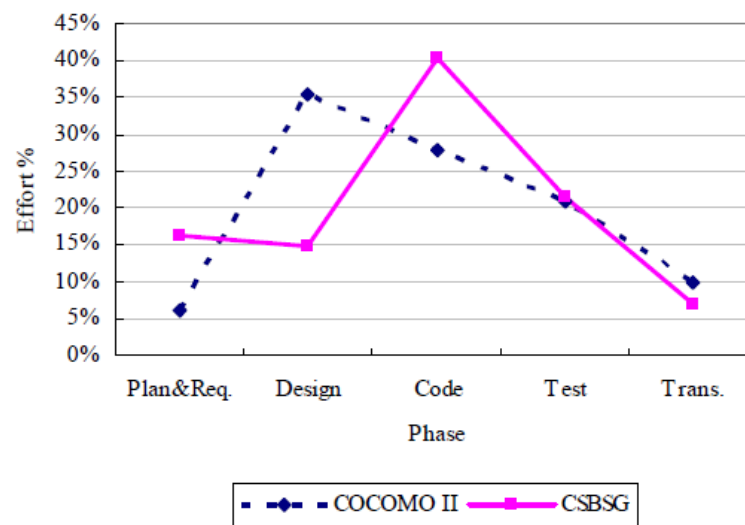
## 6. TIME AND COST NEEDED FOR INFORMATION SYSTEMS DEVELOPMENT IN TRADITIONAL WAYS:

Literature differs in determining the rate required to complete each phase of the system development lifecycle (in terms of cost, time and effort). Basavaraj and Shet [7] believe that the effort required for the analysis phase is 3% while the design phase needs 9%, the implementation phase needs 39%, the testing phase needs 27% whereas the rest of the system development lifecycle needs 22% (phases for documentation, review, training and other sub-phases). According to Stephen Haag [8], the rate of time and cost for each step of the information system development is given in the following table:

**Table (1): Time and effort required for each phase of the system development lifecycle**

Phase	Planning	Analysis	Design	implementation	Testing	Setting-up	Maintenance
Time %	11%	30%	11%	10%	11%	10%	17%
Cost %	3%	9%	11%	14%	22%	40%	1%

The Chinese Software Benchmarking Standard Group (CSBSG) collected data on 75 information systems development projects from various sources, with the aim of studying the effort required for each phase of the system development lifecycle. To study the differences in the distribution of each phase between the CSBSG dataset and the recommendations of COCOMO II (Constructive Cost Model), the average distribution of the CSBSG dataset was compared with the waterfall distribution quantities of the COCOMO II method. Figure (1) shows the effort required for each phase of the system development lifecycle. [9]

**Figure 1: Estimation of effort for all development phases as seen by CSBSG and COCOMO II**

Based on the previous studies we see a clear difference in estimating the time, effort or cost required to accomplish each phase of the information systems development lifecycle. The researcher argues that this difference is attributable to several reasons, including:

1. The varied degree of the system complexity. The analysis phase rate in complex systems will be large, while the analysis phase rate in systems with simple complexity will be small.
2. The programming language used in the development process. We often find a difference in the time period that we need to write the same code but using two different programming languages. The reason is attributable to the varying degree of difficulty in dealing with programming languages.
3. The experience of the system development team. The team with more experience is able to accomplish tasks in less time compared to the development team with a little experience.

The factors determining the cost of project development can be divided into four types [10]:

1. Product-related factors (software): required reliability degree, product complexity degree, the volume of the used database, reuse possibility and appropriateness of documentation for all needs in the project lifecycle.
2. Computer-related Factors: time requirement needed for implementation, basic storage requirement, transformation to using the computer requirement and platform specifications.
3. Personal factors: capabilities of the system analyst, capabilities of the programmer, the experience gained by application, the experience gained by the platform, the experience gained by the programming language and tools used in development and personal scenarios.

4. Project-related factors: the development in more than one site, the use of programming tools and the request of the agenda.

Most models used to estimate the cost of information systems development attempt to estimate the effort which can be transformed into a period of time and cost required to develop the project. Also, effort and cost are closely related, but not necessarily there is simplicity in the relationship in which effort can be converted into cost. The effort is usually measured by the number of people needed to complete the project in one month. The people are programmers, system analysts and project managers. The project cost is estimated by calculating the average salaries of the people employed in the system development and then this average is multiplied by the estimated effort for the project as a whole.

For many years, managers have relied on the expertise and ready-made models that are widely used as a basis for estimating development costs. The most commonly used method of estimating cost is the estimation or judgment of experts. However, the cost estimation based on experience remains inaccurate; i.e. remain probable, and therefore this method has the following problems:

- We rarely see two identical (or semi-identical) systems. This method is not iterative and the method of deriving or concluding estimation is not clear.
- It is difficult to have a highly experienced person to estimate the cost of developing new projects.
- Sometimes management manipulates the budget to avoid any over expenditure that makes the experience and data taken from previous projects questionable.

Many estimation models rely on size measures such as the number of lines and the number of function points obtained to estimate the project size. There are obstacles to each type of common measure used in estimating the size. However, it is possible for the organization to avail any measure, as long as it uses an appropriate, consistent and correct counting method.

The researcher chose the cost estimation that is based on the number of lines of the code. The main reason for this choice is the basic advantage of MDA technique that generates the bulk of the code automatically.

## 7. MDA TECHNIQUE

Model Driven Architecture technique is a method by which the functional specifications of the system can be separated from the implementation of these specifications so that these specifications work on a specific platform. In this way, the system structuring and modeling will be independent of any programming language or intermediate layer between the system and the platform on which the system will operate.

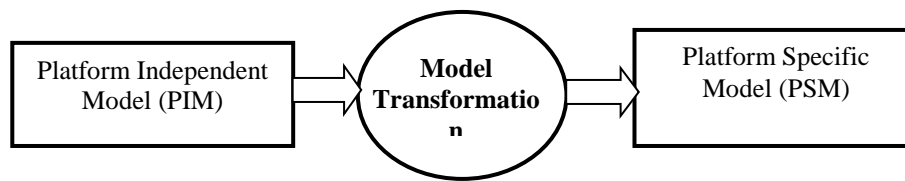
What distinguishes MDA technique from other techniques used in the development of information systems is the way it uses in modeling and generating (implementing) the application. Through this technique we see that the cornerstone of the system development lifecycle is the model rather than the code. Through this technology, the bulk of the code can be generated in early phases of the system development lifecycle.

MDA technique differentiates between two types of models. First type focuses on the functions performed by the system and is expressed through the Platform Independent Model (PIM), while second type focuses on modeling the functions performed by the system relying on the Platform Specific Model (PSM).

MDA technique has gradually become more important because of the maturation of the information technology used by the transformation tools on which MDA technique is based. This has led many large and well-known IT companies to approve this technology in the development of their systems, including IBM, Oracle, Unisys, IONA, and many other companies.

MDA technique depends on three basic parts:

1. Platform Independent Model (PIM).
2. Platform Specific Model (PSM).
3. Model transformation (MT).



**Figure 2: Key components of MDA technique**

MDA technique is an initiative launched by the Object Management Group (OMG) team. Through this technique, Platform Independent Model (PIM) is built that can be transformed to one or more Platform Specific Model and then into a code by a programming language.

Object Management Group defined MDA technique as a method for organizing and managing the organization structuring. This method is supported by tools and services to identify models and facilitate their transformation automatically to different types. [11]

MDA technique has contributed to two fundamental improvements in information systems development, the first is enhancing abstraction, while the second is enhancing reuse level. [11]

1. Enhancing abstraction level by mainly depending on the model, while basic independence of tradition methods was on machine languages, and then the development of high level programming language, and then visual programming languages.
2. Enhancing reuse level by mainly depending on the domain.

### **7.1 Basic principles of MDA technique**

Basics of MDA technique rely on three integrated ideas: [12]

1. Direct representation (modeling): In this way, attention is transformed during information systems development from the focus on implementation technology (the platform on which the system will operate) to the focus on the ideas and concepts related to the domain under development.
2. Automatic transformation: In this way the ideas and concepts expressed during the direct representation (modeling) are automatically transformed to the code required to be obtained which has specific technological specifications. In such a way the gap between the design and implementation phases has thus been closed.
3. Open standards: Standards are one of the most effective enhancements of progress throughout the history of technology.

### **7.2 The difference between MDA technique and traditional methods in systems development:**

There are several points that make up the basic differences between MDA technique and the traditional methods used in systems development [13]:

1. MDA technique starts at a high abstraction level (model). This is not achieved by traditional methods of systems development. Platform Independent Model (PIM) has a high level of abstraction. It contains entities and relationships between these entities in addition to the services needed by the system and how they are performed.
2. Platform Specific Model (PSM): through this model, the design can be enhanced and improved through specifications of specific technology (e.g. Entity Java Beans “EJB”) with no need to make modifications to the code written in the specific programming language (Java code).
3. The code generated from PSM is very close to the complete code. Many of the tools used in the traditional methods of systems development generate part of the code from specific types of models, which are not understood because they do not start from a whole system model.
4. The algorithm through which PIM is transformed to PSM or to code can be identified or formed by a specialized systems developer.

Thus, it can be said that MDA technique differentiated between the details of the implementation process (usually associated with a specific platform) and the system functions and the model used to describe them. In this way, there is no need to re-model the system functions when new technology emerges (e.g. new programming language). In such a manner, the system functions (PIM) are modeled only once, while the tool that will be used in the transformation process from PIM to PSM and from PSM to the code is built only once, and used by different parties, and to perform the transformation for multiple types of systems (can be used more than once).

### 7.3 Benefits of using MDA technique

The most important benefits of using MDA technique are [14]:

- Reduce cost.
- Reduce the time needed for development.
- Improve the quality of the information system under development.
- Increase the return on investment (ROI).
- The possibility of availing any new technology in developing the system.

In addition to the above benefits we find that productivity has improved during the information system development because PSM is obtained automatically from PIM, and the code is obtained from PSM or PIM, and accordingly, the time required to build PSM and the code (when using the tradition methods in developing the system) will be shortened and utilized by focusing more on building PIM with accuracy and excellent specifications.

The amount of cost reduction is inaccurate and is governed by several factors, which the research will try to focus on and clarify.

### 7.4 Steps of MDA technique

Figure 3 shows MDA technique steps and the obtained outputs from each step [15]

The research mainly focuses on the system generation phase and consequent cost reduction.

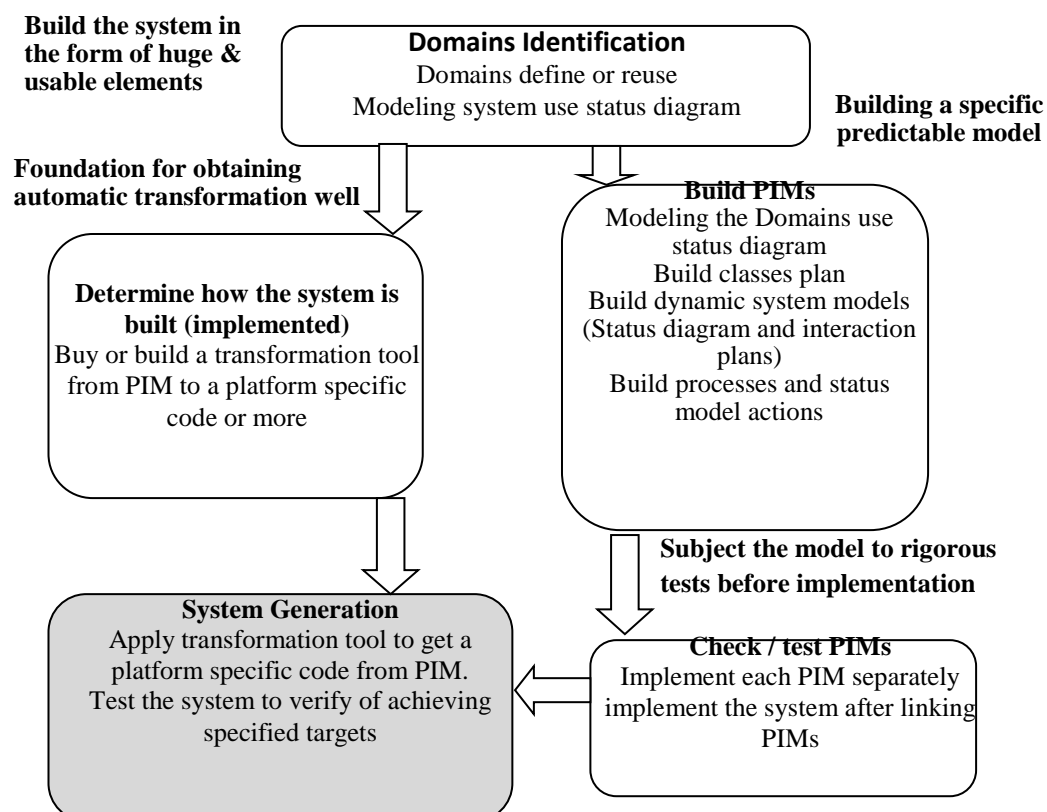


Figure 3: MDA technique steps

### 7.5 MDA technique tools

Several companies introduced tools that support MDA technique. The coming years will see a growth in the market of MDA technique tools due to their adoption in the information systems development by large and well-known companies. Some of the most important tools are:

Astah UML - Star UML - Argo UML - Software Ideas Modeler - Visual Paradigm - Enterprise Architecture - UMT - MTL - ATL - GMT - BOTL - OptmialJ - ArcStyler - Eclipse Modeling Framework - AndroMDA - Integranova Model Execution System - Objecteering - Visual Studio 2012 - CodeFluent - Adaptive Framework - Gen it Architect - iUML and iCCG -...

## 8. MDA LITERATURE REVIEW

Stanley [16] argues that the use of MDA technique can reduce the cost of information systems development. Here are the main findings of his study:

1. Analysis and design phase: The cost of development phase will be increased by 20%. The researcher argues that this increase is attributable to:
  - a. Focus on adding all the details in the analysis and design phase, for example, information used in the system documentation phase is added. This addition explains the function of some elements, how they work and the conditions to be met.
  - b. Addition of some elements that may be unnecessary to the current system, but the reuse feature in MDA technique necessitates this addition.
  - c. Testing built models (PIMs).
2. System building phase: Reduction rate is 60%. The researcher believes that the reason is generating the code automatically either from the PIM or from PSM using one of the transformation tools.
3. Test phase: Reduction rate is 80%. The researcher believes that the main reason for this reduction is the early test performed at the analysis and design phase, after generating the code we find that the remaining test is a simple part.
4. Documentation phase: Documentation cost reduction rate is 70%. The researcher argues that the reason for this is the automatic documentation of the system development.

In another study conducted in the Smart Village in Egypt [17], researchers found that when using the MDA technique, we need to increase the time required for the analysis and design phase by 20%, while the program implementation phase (code writing) is reduced by 80%. The total time that can be saved during the system development lifecycle when using MDA technique is 28%. The team hopes this rate to reach 40% after increasing their experience in using MDA tools.

Several large, renowned and leading organizations managed to demonstrate the success and excellence achieved by using MDA technique, including Siemens Electronics, Deutsche Bank Bauspar AG; which is a leading bank in Germany and Austrian Railways; one of the largest transport companies in Europe [18]. But the question remains: What is the rate of savings achieved by using MDA technique?

The answer to the previous question is very difficult, especially in large systems because there are certain acquired advantages which material benefit cannot be determined directly at the time. Thus, we will try to study the savings achieved by using MDA technique in the development of small and medium applications only. There are two types of economic savings that can be achieved; saving that can be measured directly, and strategic saving (achievable in the long term).

1. **Direct Saving:** The researcher argues that this saving can be achieved through three basic points during the system development, as follows:
  1. Cost reduction of the application implementation phase (code writing).
  2. Cost reduction of the project documentation phase.
  3. Cost reduction of testing and maintenance.

2. **Strategic Saving (on the long term):** This savings is achieved through the reuse feature. In addition to the possibility of making any modification or development to the system which is easily developed using MDA technique, and at low cost because the modification and development is done on the model and then the modified model is automatically transformed to code with some manual intervention by programmers.

In general, the effectiveness of using MDA technique tools to generate code depends on the effort the tool can reduce or generate automatically. There are two basic factors that control how much effort can be reduced:

1. The nature of the transformation engine that generates the code within the MDA tool.
2. The amount of information in PIM. Focusing on building PIM and incorporating all details related to each function, how to perform this function and the requirements to be taken into consideration would increase the programming effort can be reduced.

The researcher will try to study the cost reduction related to the system implementation phase by comparing the number of lines that make up the program code without using MDA technique, and the number of lines of code that will be automatically generated for this program when using MDA technique, and compare them to find the rate of saving that can be reduced when using MDA technique.

## 9. RESULTS

Six information systems development projects were selected, along with three MDA tools. Each of the three tools was used to develop two projects. Two ready-to-use tools were used: Microsoft Visual Studio 2012 and Visual Paradigm for UML 8.0. A new tool was also developed during the search which is the Stored Procedures\_Classes Generator.

### 9.1 Stored Procedures\_Classes Generator:

This tool was created using C# .Net and WPF technique. This tool assumes that a database has been properly normalized by the database programmer. It analyzes the database and the tables within it to create:

1. Classes

Database tables are transformed into classes. Each class contains the necessary fields, properties, procedures, and necessary basic programming functions.

2. Databases Stored Procedures

The tool links such procedures with classes that are automatically created. This tool is able to automatically generate a database. Table (2) shows the rate of saving achieved in the implementation phase as a result of using this tool.

**Table 2: Using StoredProcedures\_Classes Generator tool in systems development**

	Number of lines of code in the traditional method				Number of lines of code automatically generated	Code automatic generation %
	DB	Classes	Remaining code	Total		
HR Dept.	2229	4152	3493	9874	3457	35%
Charitable Society Dept.	1653	1473	9992	13118	3126	24%
Platform	Visual Studio 2012, MS SQL Server 2012					

From above table we see that the rate of code automatic generation in the first system is 35%, and 24% in the second system. Here, we note two basic things:

1. The code generation rates vary in the two systems due to the complexity of the second system which is higher than the complexity of the first system.
2. The rate of the code generated automatically is low due to the nature of the system platform, as XAML language is not supported by the transformation engine used by the tool which has been built, so the team had to write XAML language code manually.



The research also found that above tool is unable to generate documentation automatically due to the fact that this tool is new and the work on this tool is continuing to reach better capabilities for this tool.

### 9.2 Microsoft Visual Studio 2012:

As a result of the research, it was found that this tool is unable to generate the code of the database. Moreover, it is unable to document the system automatically. Table (3) shows the information of the implementation phase:

**Table 3: Using Microsoft Visual Studio 2012 in Systems Development**

	Number of lines of code in the traditional method			Number of lines of code automatically generated	Lines of code generated automatically %
	DB	Program	Total		
Medical analysis lab	1345	4445	5790	1879	32.5%
Supplementary system of Syriatel	58	423	481	341	70.9%
Platform	Visual Studio 2012, Crystal Reports, MS SQL Server 2012				

From above table we see that the rate of code automatic generation in the first system is 32.5%, and 70.9% in the second system. Here, we note three basic things:

1. The code generation rate varies in the two systems due to the complexity of the first system which is much higher than the complexity of the second system. In addition, the second system is very simple and contains no complex requirements or constraints.
2. The rate of the code generated automatically is high in the second system because its size is small and simple.
3. The rate of the code generated automatically is low in the first system because its size is big and the high degree of its complexity.

### 9.3 Visual Paradigm for UML 8.0:

As a result of this research, it was found that this tool is able to generate database code, in addition to its ability to automatically document the system. Table (4) shows the information of the implementation phase.

**Table 4: Using Visual Paradigm in Systems Development**

	Number of lines of code in the traditional method			Number of lines of code automatically generated	Lines of code generated automatically %
	DB	Program	Total		
Al-Bab Hospital Dept.	3001	4498	7499	4533	60.5%
Medical Fitness Center	493	1517	2010	1238	61.5%
Platform	Visual Studio 2012, Crystal Reports, MS SQL Server				

From above table we see that the rate of code automatic generation in the first system is 60.5%, and 61.5% in the second system. We note that the rate of the code generated automatically is high due to:

- d. The ability of Visual Paradigm to generate database code (or generate the database itself).
- e. Completeness (maturity) of this tool and its distinctive abilities through supporting it to add much information in the analysis and design phase, in addition to the nature of the advanced transformer used by this tool.

From the above three tables, we find that the use of MDA technique reduces the cost of the implementation phase at varying rates, but the best saving that can be achieved for systems of medium-sized and complexity is by using the Visual Paradigm tool.

The research also found that first tool StoredProcedures\_Classes Generator (under development) and the second tool Microsoft Visual Studio 2012 were unable to document the system, while Visual Paradigm tool is able to automatically document the system by more than 80%.

The previous six systems development work teams were also able to test the system in early phases of the system development lifecycle.

## 10. CONCLUSIONS

The study shows that there is a difference in the programmatic effort that the three tools can reduce. This difference is attributable to several reasons including:

1. The size and complexity of the system.
2. The completeness (maturity) of the used transformation tool.

Through this study, we also observe that reduction rate of the code in the implementation phase is low. This is due to the lack experience of the development team relevant to the transformation tool used in the development process.

The researcher recommends MDA technique to be adopted by information systems development companies. The researcher also recommends teaching MDA technique to students of the Department of Management Information Systems in the Faculty of Economics and students of the Faculty of Information Engineering. Moreover, the researcher recommends increasing the effort made by the development team to fully study and understand the tool used in the transformation process.

## 11. FUTURE WORK

The researcher recommends identifying the different transformation tools that depend on Model Driven Architecture technique, and to conduct a comparative study between these tools to know the pros and cons of these tools and thus know the appropriate tool for each type of information systems.

## REFERENCES

- [1] Mansor Z., Yahya S., and Arshad N.H.H., 2011 - Success Factors in Cost Estimation for Software Development Project, ICSECS, Part I, CCIS 179, pp. 210–216.
- [2] Noor H., A., Azlinah, M., Zaiha, M.N., 2007 - Software Development Projects: Risk Factors and Occurrences, WSEAS Transactions on Computer Research Journal 2(2).
- [3] Ostvold, K.M., Jorgensen, M., 2005 - A Comparison of Software Project Overruns – Flexible versus Sequential Development Model, IEE Transactions on Software Engineering 31(9), pp.754–766 .
- [4] Zulkefli, M., 2009 - Cost Estimation Tool for Web-Based Application. Master Dissertation, Department of Software Engineering, Universiti Malaya. Kuala Lumpur.
- [5] Yang D., Qing W., Mingshu L., Ye Y., Kai Y., Jing D.A., 2008 - Survey on Software Cost Estimation in the Chinese Software Industry, In: ESEM, Kaiserslautern, Germany.
- [6] Terry C., Davies 2001 - The real success factors on projects," International Journal of Project Management 20(3), pp. 185–190.
- [7] Basavaraj M.J, K.C Shet, 2008 - Empirical validation of Software development effort multipliers of Intermediate COCOMO Model, JOURNAL OF SOFTWARE, VOL. 3, NO. 5.
- [8] Stephen H., 2010 - Management Information Systems of the information age, McGraw-Hill/Irwin.
- [9] Zulkefli, M., Zarinah, M.K., Habibah, A., Saadiah, Y., 2010 - E-Cost Estimation Using Expert Judgment and COCOMO II, In: ITSIM, vol. 3, pp. 1262–1267.
- [10] Sommerville I., 2004 -Software Engineering, Pearson Addison Wesley, 7th edition.
- [11] Brown A., 2004 - an introduction to model driven architecture part1: mda and today's systems, Developer Work, IBM Corporation.

- [12] Booch G., Brown A., Lyengar S., Rumbaugh J., and Selic B., 2004 - An MDA manifesto. The MDA journal.
- [13] OMG - Model Driven Architecture (MDA) FAQ, Accessed: 11-06-2014 [http://www.omg.org/mda/faq\\_mda.htm](http://www.omg.org/mda/faq_mda.htm)
- [14] OMG - The Architecture of Choice for a Changing World, executive overview, Accessed: 11-06-2014 [http://www.omg.org/mda/executive\\_overview.htm](http://www.omg.org/mda/executive_overview.htm)
- [15] Riastrick C., Francis P., Wright J., Carter C., Wilkie I., 2004 - model driven architecture with executable UML, Cambridge University Press.
- [16] Stanley J. S, 2003 - Executive Justification for Adopting Model Driven Architecture (MDA), OMG. Accessed: 11-06-2014 <http://www.omg.org/mda/presentations.htm>
- [17] Smart Village, 2012 - Model Driven Software Development – A reality check, 2nd RECOCAPE Event ITIDA, Giza, Egypt <http://www.secc.org.eg/RECOCAPE/Resources/Events/Model%20Driven%20Software%20Development%20-%20A%20Reality%20Check.pdf>
- [18] Interactive Objects, 2005. MDA Success Story, ePEP successful with Model Driven Architecture. [Online]. Available at: < [http://www.interactiveobjects.com/customers/success-stories/Daimler chrysler-tss-success-story](http://www.interactiveobjects.com/customers/success-stories/Daimler%20chrysler-tss-success-story).