



ADOPTION OF CASE TOOLS & UML: A LOCAL STUDY

OSAMA IBRAHEEM ASHOUR

SEPTEMBER 2019

ADOPTION OF CASE TOOLS & UML: A LOCAL STUDY



A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES OF
ÇANKAYA UNIVERSITY

BY
OSAMA IBRAHEEM ASHOUR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER ENGINEERING
DEPARTMENT
INFORMATION TECHNOLOGY PROGRAM

SEPTEMBER 2019


Title of the Thesis: Adoption of CASE Tools & UML: A Local Study

Submitted by: **Osama Ibraheem ASHOUR**

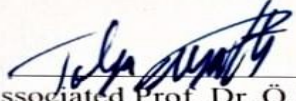
Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.


Prof. Dr. Can ÇOĞUN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.


Prof. Dr. Sıtkı Kemal İDER
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.


Associated Prof. Dr. Ö. Tolga PUSATLI
Supervisor

Examination Date: 12.09.2019

Examining Committee Members

Assoc. Prof. Dr. Murat KOYUNCU Atılım Univ.

Assoc. Prof. Dr. H. Hakan MARAŞ Çankaya Univ.

Assoc. Prof. Dr. Ö. Tolga PUSATLI Çankaya Univ.





STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Osama Ibraheem ASHOUR

Signature: 

Date: 23.09.2019

ABSTRACT

ADOPTION OF CASE TOOLS & UML: A LOCAL STUDY

ASHOUR, Osama Ibraheem

M.Sc., Information Technology Department

Supervisor: Assoc. Prof. Dr. Ö. Tolga PUSATLI

SEPTEMBER 2019, 35 pages

CASE (Computer Aided Software Engineering) tools include computer-based support in the software development process and support for the managerial, administrative or technical aspects of software engineering projects. CASE facilitates activities during the software development process, which as a result may increase productivity and improve quality. The use of UPPER CASE tools and UML diagrams for a number of small and medium scale selected local companies are studied. There are works in the literature which report companies that want to standardize their developments may use UML diagrams. UPPER CASE has been used to prepare UML diagrams in the early stage of the software life cycle. In this study, a survey was conducted in a number of selected companies in Ankara in which interviews are conducted given to information system managers and developers in six different organizations provided feedback throughout the survey. CASE and UML are reported as being considerably used in Ankara with CASE tools being used for project members' communication and documentations. Furthermore, CASE tools have been used for some companies for documentation as parts of contracts and to follow standards such as ISO and CMMI. However, they require a detailed and straightforward definition of company processes. Some standards have an indirect effect on CASE usage. To answer our research question, we can assert that CASE tools can be used to facilitate project developments, especially when enforced by the standards and methodologies required in the contracts. However, CASE tools have been adopted only for documentation and there is a lack of engineering when this

usage is left to developers.

This research presents the role of CASE tools in the software development process for local companies in addition to the importance of CASE in following a number of standards. This work relies on key informants from only six local companies; hence, the findings are limited.

Keywords: UPPER CASE tools, UML, small/medium size software developer companies



ÖZ

Bilgisayar Destekli Yazılım Mühendisliği Araçları ve Birleşik Modelleme Dili Kullanımının Benimsenmesi: Yerel Çalışma

ASHOUR, Osama Ibraheem
Yüksek Lisans, Bilgi Teknolojileri Bölümü
Tez Yöneticisi: Doç. Dr. Ö. Tolga PUSATLI

EYLÜL 2019, 35 sayfa

Bilgisayar destekli yazılım mühendisliği (CASE) araçları yazılım geliştirme sürecinde bilgisayar tabanlı desteği içerir ve yazılım mühendisliği projelerinin yönetsel ve teknik konularına destek olur. CASE, yazılım geliştirme süreçlerindeki etkinlikleri kolaylaştırır; bu da üretkenliği artırır ve kaliteyi yükseltir. Çalışmada, küçük ve orta ölçekli yerel firmaların üst CASE araçları ve Birleşik Modelleme Dili (UML) diyagramları kullanmaları işlenmiştir. Literatürde, süreçlerini standartlaştırmak ve bu standartları almak isteyen firmaların UML diyagramları kullandığı bildirilmiştir. Üst CASE araçları, yazılım döngüsünün başlarında, UML diyagramları hazırlamak için kullanılmaktadır. Bu çalışmada, Ankara'da, seçilmiş altı firmanın bilişim sistem yöneticileriyle görüşülüp geri dönüş sağlanmıştır. CASE ve UML'nin, Ankara'da, proje katılanları iletişiminde ve belge hazırlanmasında kullanılmakta olduğu rapor edilmiştir. Ek olarak, CASE araçlarının, sözleşmelerinin bir parçası olan belge hazırlamada ve Uluslararası Standartlar Örgütü (ISO) ve Bütünleşik Yetenek Olgunluk Modeli (CMMI) gibi standartları izlemede kullanıldığı bildirilmiştir. Bununla beraber, bu araçlar, firmaların ayrıntılı ve kesin süreç tanımları olmasını gerektirmektedirler. Bazı standartlar, CASE araçları kullanımını dolaylı olarak gerekli kılmaktadır. "CASE araçları endüstride kullanılıyor mu; öylese ne kadar kullanılıyor", araştırma sorusuna cevap olarak, projelerde kolaylık sağlamada kullanılabilir ve sözleşmelerde belirlenen standartlarla yöntemlerin zorlamasıyla kullanılıyor cevabını verebiliriz. Öte yandan, CASE araçları, sadece belge

hazırlamada kullanılıp, mühendislikte kullanılması geliştiriciye bırakılmıştır. Bu araştırma, CASE araçlarının, yerel firmalarda yazılım geliştirme süreçlerindeki rolünü ve buna ek olarak standartları izlemedeki önemini sunmaktadır. Çalışmanın en önemli kısıtlayıcısı, sadece yerel olması ve bilgi sağlayan sınırlı sayıda uzmana bağlı kalmasıdır.

Anahtar sözcükler: Bilgisayar Destekli Yazılım Mühendisliği Araçları, Birleşik Modelleme Dili, küçük/orta ölçekli yazılım firmaları



ACKNOWLEDGEMENTS

Thanks to God, the most compassionate and the most merciful for supporting me and supplying me with power throughout my life.

I would like to express my sincere gratitude to my supervisor Associated Prof. Dr. Özgür Tolga PUSATLI who helped me and dedicated a lot of his time to this project. I am indebted to his suggestions and valuable remarks.

I would like to also share my heartfelt thanks to my parents who supported me throughout my entire study period . It is also worth mentioning a bundle of thanks to and my friends .

I would like to thank everyone who helped in one way or another in bringing out this work.

TABLE OF CONTENTS

ABSTRACT	v
ÖZ.....	viii
ACKNOWLEDGEMENTS.....	ix
TABLE OF CONTENTS.....	x
LIST OF FIGURES	xii
LIST OF TABLES	xiii
LIST OF ABBREVIATIONS	xiv
CHAPTER 1	1
1.1. Introduction	1
1.2 Purpose of the Study	3
1.3 Scope of the Study	3
1.4 Research Questions	3
CHAPTER 2	4
2.1. CASE Tools, IDE and UML	4
Process Model Tools.....	7
Project Management Tools.....	8
Documentation Tools	9
Analysis Tools	9
Design Tools	9
Programming Tools.....	9
2.2. Literature Review	9
2.3 Summary	15
Standards	15
Diagrams	15
CASE Tools and IDE	15

CHAPTER 3	17
3.1 The Questionnaire	17
Part 1: Organization History and Size	18
Part 2: Standards	19
Part 3: Diagrams	20
Part 4: CASE Tools and IDE	21
Part 5: General Idea.....	21
3.2 Results.....	21
<input type="checkbox"/> Company 1.....	21
<input type="checkbox"/> Company 2.....	22
<input type="checkbox"/> Company 3.....	23
<input type="checkbox"/> Company 4.....	25
<input type="checkbox"/> Company 5.....	26
<input type="checkbox"/> Company 6.....	26
3.3. Discussion of Results	28
CHAPTER 4	32
4.1 Conclusion	32
4.2 Limitation.....	33
4.3 Future Work	33
CURRICULUM VITAE.....	37

LIST OF FIGURES

Figure 1: Classification of CASE.....	5
Figure 2: Diagram tools	7
Figure 3: Process Model Tools	8
Figure 4: Project Management Tools	8
Figure 5: Example 1 for Eclipse Process framework composer (EPF).....	22
Figure 6: Example 2 for Eclipse Process framework composer (EPF).....	23
Figure 7: Example 3 for Eclipse Process framework composer (EPF).....	24

LIST OF TABLES

Table 1: Organization history and size	18
Table 2: Standards	19
Table 3: Diagrams	19
Table 4: CASE tools and IDE	20
Table 5 A: Summary of Responses	29
Table 5 B: Summary of Responses	30



LIST OF ABBREVIATIONS

CASE	Computer Aided Software Engineering
CMMI[®]	The Capability Maturity Model Integration (CMMI [®])
ISO	International Organization for Standardization
MVC	Model – View – Controller
SDDs	Software Design Descriptions (SDDs)
SRS	System Requirement Specifications
UML	The Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1. Introduction

The software development process is a complex series of tasks starting with the early phases of design and analysis and ending with testing and maintenance. Among these factors, the analysis of the requirements and design requires considerable effort and resources.

To be able to develop a system, the requirements, analysis and design in addition to a number of software methods and notations have been devised and reported to the literature. Some of these, including Unified Modeling Language (UML), have become standardized thereby facilitating software engineers' adaptation to different organizations and tools. UML is a method of visualizing software programs using a collection of diagrams and standardized documentation. Thus, UML increases understanding among software developers and customers. Moreover, it helps to understand how software works, how a specific process may work in addition to decreasing software complexity during development. Architecture and design models also guide the implementation of a system. UML is seen as an important notation technique for software design documents such as Software Requirements Specification (SRS) and Software Design Descriptions (SDDs) to increase the quality of the final product and occasionally to track software development methodologies and standard.

To prepare UML diagrams, a number of tools are used to manage them effectively, including storing them in file formats that can be helpful when reusing them and easily maintaining them later. We call such tools Upper CASE tools

Related to software development, Computer Aided Software Engineering (CASE) tools are specifically developed tools to be used in design. These tools provide computer-based support in software development processes and in the managerial, administrative or technical aspects of software engineering projects. CASE facilitates activities during a software development process. As a result, CASE may increase productivity and improve quality. CASE tools can be used to apply a number of standards to organization management, such as CMMI (Capability Maturity Model Integration).

In the software development industry, it is quite common to discuss CASE tools, IDE and UML. These are also included in courses taught in related education programs at universities, hence new graduates are expected to be familiar with them. Being popular both in the industry and in education, we are motivated to study the use of CASE tools and UML diagrams by developers. Before we proceed, it would be beneficial to touch upon the study area. It is important for companies to standardize their developments through a number of methodologies and theories that make the use of UML diagrams necessary to gain acceptance for such standards. Moreover, UML explains business needs, technical solutions, the data flow of systems, data structures and the relations among them. These diagrams, such as SRS and SDD documents, must be prepared carefully and accurately during the documentation phase. CASE tools can be of great help when preparing UML diagrams through their features and their ability to document these diagrams. CASE can also standardize development and make it well defined to other team members and it is also important to enhance communications through project management. CASE tools, such as CMMI, can be used to apply a number of standards to organization management.

It is noteworthy that CASE tools, thanks to their various features and functionalities, improve the quality of software development plans. This, however, does not mean it also improves productivity [1]. Nor does the use of CASE tools for the generation of database schemas increase developers' productivity although it frees them from writing SQL code [13]. CASE can be affordable in comparison to a project's budget,

but some researchers emphasize that CASE tool expenses are typically high due to their steep learning curve. Moreover, before acquiring CASE tools, one should consider the total budget of a software project and compare it to be commensurate with the cost [14]. CASE can be one of the methods that helps to obtain certification for some standards [23].

The discussion continues with other research groups with findings saying that the staff of organizations have to be well trained to use CASE tools and this comes with another cost, namely that it is time consuming for a project and that the effect of CASE tools is limited to following project methodologies and standards [3].

1.2 Purpose of the Study

CASE tools play a specific role within the software release life cycle and it should be emphasized that these tools are divided into a number of categories or groups. We will focus only on the category of Upper CASE.

CASE tools are increasingly taking a leading role in the planning and execution of projects as far as we can follow from developers' websites while some researchers report reluctance to use these tools. Our purpose for this study is to determine whether or not companies are encouraged or compelled to use CASE tools and UML due to project requirements. Moreover, software standards are usually seen among project requirements; therefore, we will be investigating the position of CASE tools and UML in fulfilling the requirements for standards.

1.3 Scope of the Study

Our scope is to discuss the usage of UML notation and CASE tools. As the study does not aim to generalize, a quantitative survey for statistical analyses is not included. Instead, we investigated six companies through a detailed, semi-structured interview set.

1.4 Research Questions

Are UML and UPPER CASE tools being used by developers and to what extent are they being used?

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

2.1. CASE Tools, IDE and UML

Computer-Aided Software Engineering (CASE) is the use of computer-based support in the software development process, so it is a software tool to help produce software product systems or solutions. CASE is the application of automated technologies to software engineering procedures. CASE therefore encompasses all forms of computer-based support for any of the managerial, administrative or technical aspects of a software engineering project [3]. CASE tools, according to Pressman, are defined thus: “tools that give an automatic support for the process and methods, When tools are integrated so that the information created by one can be used by another” [1].

Upper-CASE Tools: This is a set of tools supports traditional diagrammatic languages such as Entity Relationship Diagrams (Figure 1). These tools are being used to draw diagrams through the early stage of a software development to facilitate the documentation.

Lower-CASE: These tools are being used to assist the developers through the stage of system generation, most of these tools facilitate rapid prototyping of specifications in terms of the functionality of the system. Lower-CASE (Figure 1) offer a powerful tool that can make the design of the system more effective and efficient.

Integrating CASE tools: This consists of Lower- and Upper-CASE tools and their characteristics [4].

Components of CASE Tools

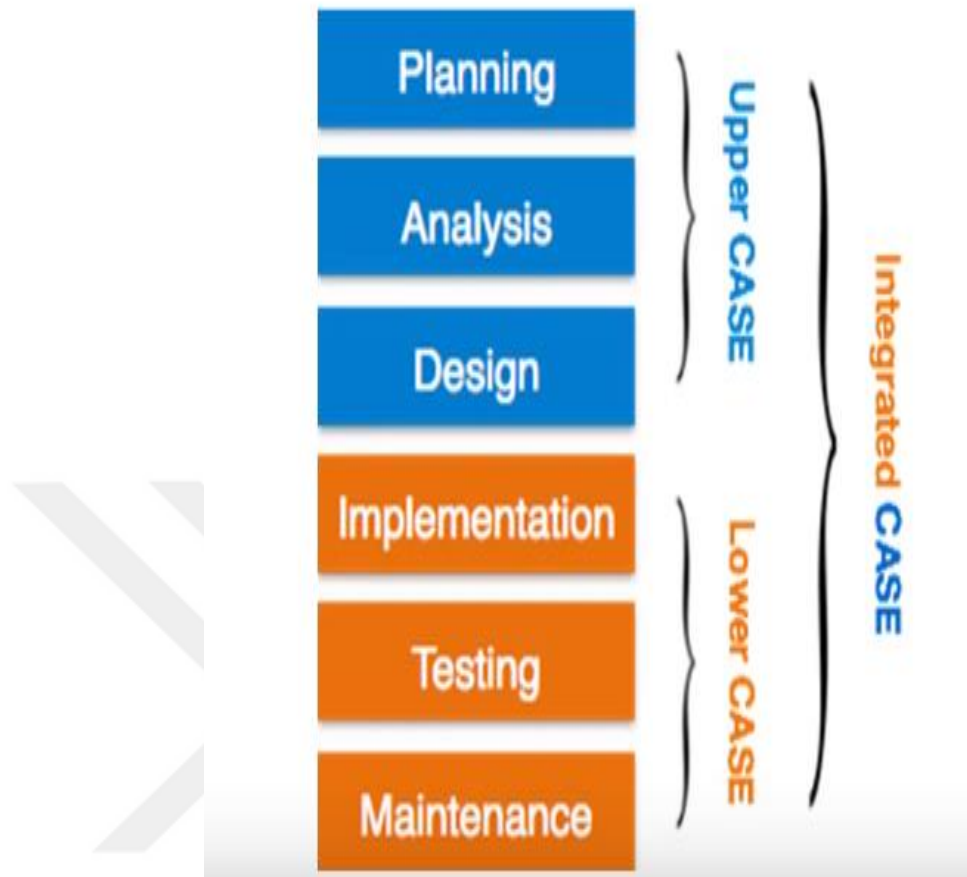


Figure 1 Classification of CASE[23]

Unified Modeling Language (UML) is a language for visualizing, specifying, constructing and documenting software systems. It is a general-purpose modeling language; however, is not only used for modeling software. It is commonly used for system engineering, for business process modeling and for the presentation of organization structures [10].

Software Development Methodologies: These provide a framework for planning, executing, and managing the process of developing software systems [2]. A number of methodologies exist, including waterfall, agile and incremental.

Some time ago, the de facto methodology was waterfall and similar methodologies which took the project as a whole. One would be analyzing, designing, developing, testing and finally delivering the product and these stages were respectively followed and one would not turn back except during the testing and developing stages which

were iterative. During the analysis and design stages, SRS and SDD documents were most commonly prepared. Moreover, UML tools and diagrams and ER diagrams and similar tools were used. Today, agile methodologies such as SCRUM are being used mostly. The idea is that at the beginning of a project, one would ask a number of questions to a client regarding the features the client would require from their software, what their needs were and any problems, etc. and a very general product backlog would be prepared. Then the client would be asked to prioritize the product features (backlog items) followed by splitting the product backlog items into sprints. At the end of the sprint, a usable chunk/module would be delivered to the client. Then we would begin with the next sprint. These other sprints may change previous sprints; therefore, the software constantly changes until the end of the project. In this methodology, one may use **use-case diagrams** or perhaps **ER diagrams** for that sprint only (the project in its entirety is at no time considered). However, these are no longer considered best practices. Moreover, SRS/SDD documents go completely against the soul of scrum. Instead, there are product backlogs, sprint backlogs, burn down charts and so on.

Software Requirements Specification (SRS) is a detailed document that describe the system's functional and non-functional requirements and it is also used by analyst to understand clients' system requirements prior.

Software Design Description (SDDs) is a detailed document that represents the software design information that communicate developer and stakeholders, and it contain various design concerns that play an important role of development and maintenance of software systems

International organization for standardization (ISO): This institution creates documents that presents the requirements, specifications, and guidelines of characteristics that can be used consistently to ensure that materials, products, processes and services are suitable for their purpose [20].

Capability Maturity Model Integration (CMMI®): This is a process improvement maturity model for the development of products and services [21]. Organizations that implement CMMI typically improve performance in terms of the quality of products

and processes, schedules and costs. Consequently, processes become more predictable and customer satisfaction increases [22].

CASE Tool Types

Diagram Tools

Such tools are been used to design diagrams that show system's components or to control the data flow or to represent data structure such as database systems for example **flow chart maker** in Figure 2.

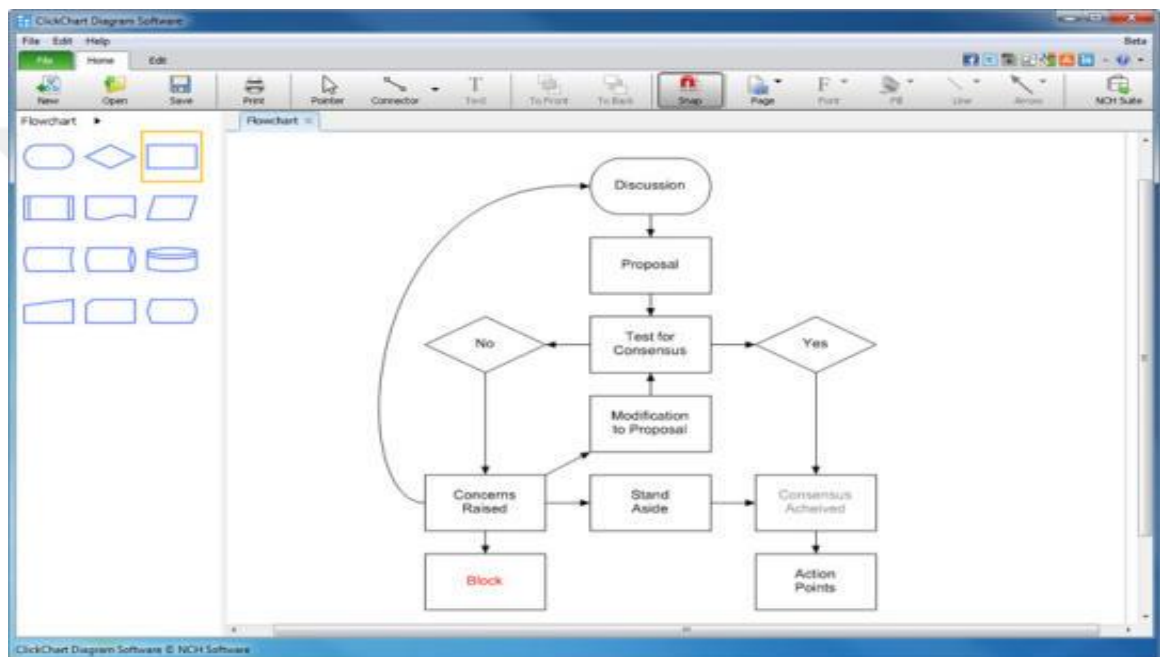


Figure 2 Diagram tools

Process Model Tools

These tools are being used to create a process model to help the managers to develop a part of the requirement by select a suitable process model for a software product. An example of this is *EPF Composer* (Figure 3).

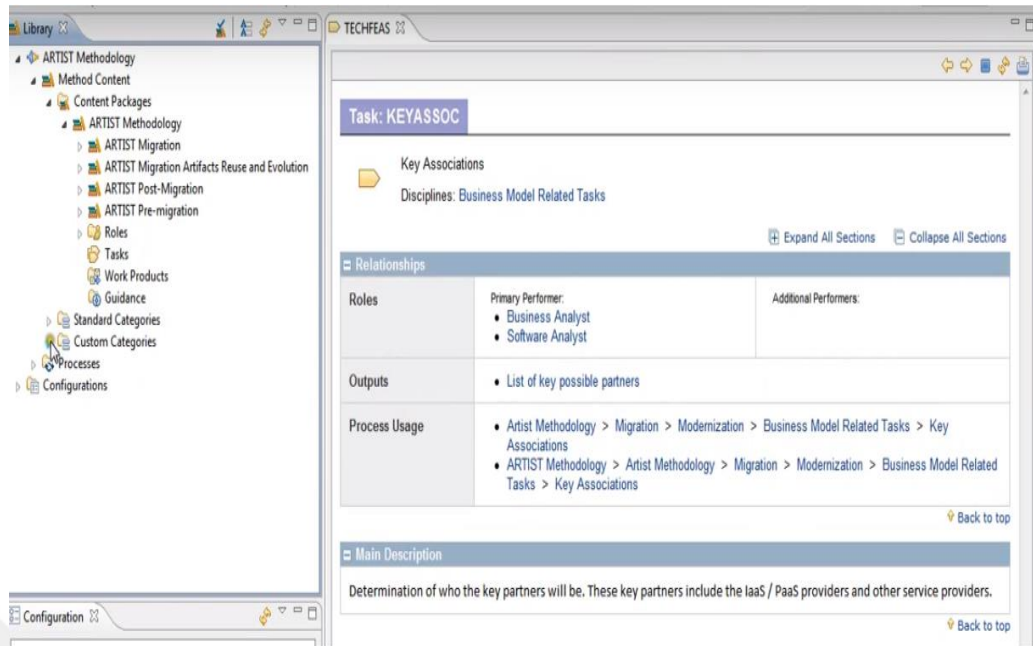


Figure 3 Process model tools

Project Management Tools

These tools are being used to manage the project cost, scheduling, time, and to share project data through the organization in real time. as shown in Figure 4.

trac
Integrated SCM & Project Management

Search

[Login](#) | [Settings](#) | [Help/Guide](#) | [About Trac](#)

[Wiki](#) | [Timeline](#) | [Roadmap](#) | [Browse Source](#) | **View Tickets** | [New Ticket](#) | [Search](#)

This report: [Edit](#) | [Copy](#) | [Delete](#) | [New Report](#) | [Custom Query](#)

{9} Time Tracking (7 matches)

Ticket	Planned	Spent	Remaining	Accuracy	Customer	Summary	Component	Status
#6	10h		10h	0.0	milestone1	asdf	component1	new
#5	2h	4h	0h	2.0	milestone1	234	component1	new
#4				0.0	milestone1	yxcv	component1	new
#3	4h	4h		0.0	milestone1	test3	component1	closed
#2	4h	2h	2h	0.0	milestone1	test2	component1	new
#1	8h	7.0h	3.0h	2.0	milestone1	test 1	component1	new
#7	1h			-1.0	milestone2	3452345	component1	new

Note: See [TracReports](#) for help on using and creating reports.

Download in other formats:
[XML](#) | [RSS Feed](#) | [Comma-delimited Text](#) | [Tab-delimited Text](#) | [SQL Query](#)

Powered by **trac** 0.9.pre
By Edgewall Software.

Visit the Trac open source project at <http://trac.edgewall.com/>

Figure 4: Project management tools

Documentation Tools

Documentation is an important stage in software development because it covers almost all phases of the software development, Documentation Tools generate documents for technical and end users. Examples of such tools include Doxygen.

Analysis Tools

Through the analysis phase some problems may occur such as inconsistencies or inaccuracies in designed diagrams or may be data redundancies, all these problem can be deal with CASE analysis tools for example CaseComplete, Accompa, etc.

Design Tools

Such tools are being used to draw and design the diagrams of software design stage. These tools providing details for each diagram and the connections among them.

Programming Tools

Code editors and IDE generate a good environment for programmers by providing them the libraries that provide a built function, further more they help them to write codes more efficient by suggesting the possibilities of right codes and errors. for example, Eclipse, PHP storm.

2.2. Literature Review

Zea Ordonez et al [1]. explain that software generally has many phases, starting from planning and ending with execution. These phases are difficult to be applied manually.

Yongzhuang Liet et al. [16] said that the software development cycle, after using CASE tools, reduces greatly and the detailed design stage and the coding are more easily performed.

Zea Ordonez et al. focused on Upper CASE tools. The aim of the study was to determine the most appropriate CASE tools products in the software design process by applying techniques of data collection through their assessment through its quality metrics. The researcher explained the usage of Upper-CASE tools in the software design process which is to transfer the requirements of the documented model, sketch

or conceptual schemes. In addition, they determined the limitation of the project and presented the entire view of the project in relation to other parties of the organization and improved the quality of the software development plan. However, it did not mean they had improved the productivity. Moreover, the situation would have been different at that point and it would depend on the issue that we are attempting to solve using CASE tools. For example, de Castro et al. [17] performed a study to present the role of CASE tools for Object-Relational Database Designs and stated that the use of CASE tools for the generation of database schemas increases developers' productivity since the tools free developers from writing SQL code. In the opinion of the author of this thesis as a web application developer, we use CASE tools to generate diagrams, especially activity diagrams, ER diagrams, SQL scheme diagrams with Microsoft Visio, and if we compare these with the traditional approach to drawing these diagrams and drawing them with the help of CASE tools, there is a vast difference especially in terms of the time of drawing and how to express the relations between two objects, thereby increasing the productivity of the development process.

Selecting an Upper-CASE tool should be considered a good choice if a tool meets the characteristics that come with benefits in software development. According to our choice of tool, it will have an effect on the final product and turn in the development process because if it is not flexible to the needs of a project, it can create problems such as lack of reliability, delays and higher than budgeted costs. They explain ISO/IEC 9126, which is a quality metric based on features and sub-characteristics and thus features will help us to select a suitable tool for the design process. In the methodology, they make an analytical and bibliographical investigation and claims that there are a number of metrics to assess the quality of software but not measure specific characteristics. He selects ISO/IEC 9126 as it is an international standard for software quality. The study area was the city of Machala as there are five active companies in that city. The definition of indicators, characteristics and sub-characteristics to determine the level of quality based on ISO/IEC 9126 include:

Functionality: the ability to perform important services in order to fulfil the functional requirements, such as the number of diagrams that can be designed, their quality and clarity, and the accuracy of these diagrams.

Reliability: performance over the time depending on different contexts, the detection of errors in the runtime and whether this leads to shutting down the program.

Usability: the high level of benefit that users obtain from a tool.

Efficiency: the ability to perform a task in less time and with fewer resources, such as the number of plots offered by each tool, the format of outputs, and programming languages.

Maintainability: the ability to find errors and deal with them as well as the configuration and modification of systems for new versions.

Portability: the running or installation of tools on different platforms as well as the formats that can be saved as documents and the ability to use or run these formats on other systems. However, ISO/IEC 9126 is no longer up to date and the newer ISO/IEC 25010:2011 is used. The quality model for this standard includes:

- Functional suitability;
- Performance efficiency;
- Compatibility;
- Usability;
- Reliability;
- Security;
- Maintainability; and
- Portability.

For the implementation of evolution criteria, they used a Likert scale. The collection of data occurred in the form of questionnaires and interviews with system analysts' managers, and for analysis, the Excel application was used. The results revealed that there were four companies that used CASE tools in the process of software design and one company that did not use any tool. 75% of companies were fully using the Enterprise Architect tool and 25% used ArgoUML. The survey showed the differences between the two tools and in terms of functionality, Enterprise Architect had more diagrams in the same design and the quality of these diagrams and tools was clearer, had more accurate diagrams and generated code for different

programming languages. In terms of reliability, Enterprise Architect was the best according to the surveys, which shows us that it supports any type of failure in the system and produce documents in the event of errors.

In terms of usability, ArgoUML was better than Enterprise. ArgoUML also has an interface structure and is comprehensive, which makes it not entirely clear what its user's needs are to be more effective, However, the user can use it with basic knowledge. In terms of efficiency, Enterprise is the best due to the fact that it has more complete UML diagrams and languages. However, ArgoUML has a huge reserve of resources, such as availability in multiple languages and the output of it as images. Both tools are very fast performing at millisecond or second times when they generate images, diagrams or codes. In terms of maintainability, Enterprise is more effective as it can track tasks, both are equal in terms of portability because both are multiplatform and can be run on different operating systems. They are easy to install and outputs can be run on other systems. The conclusion shows that selecting the appropriate CASE tool is essential for the success of the development of software, Case tools have the features which are necessary to focus on a model to show us the functional requirements [4].

Muhammad Naseeret et al. conducted a survey to analyze the importance of CASE tools for requirements management in Pakistani organizations, namely Ufone, IBM and Ciklum. Every question had three options: 'Yes,' 'No,' and 'I don't know.' In their survey, they asked about whether CASE tools were helpful in software requirements, whether it had increased productivity, whether they decreased the time of explaining requirements, and whether they had an impact on the documentation of the software development process. In their results, they showed that CASE tools were vital in the requirements engineering process. The Pakistani software industry was not mature because they depended on outsourced projects and every requirements document was well defined by other parts. In their survey, the main reason for not using CASE tools in Pakistan was cost and licensing [6].

Chiemeke et al. conducted a survey with 70 structured questions about the type of CASE tools being used in software companies and organizations in five states in Nigeria, namely Lagos, Edo, Rivers, Kaduna and the Federal Capital Territory. They selected these states due to the fact that most software development companies were

there and for commercial and financial reasons. In their survey, they found that software development organizations were involved in software activities and they used a particular tool for each activity or set of activities. They classified their survey into various sectors, the first of which was the characteristics of the CASE tools being used, and the second of which to discover the nature of CASE tools in organizations. In their results, they found that organizations were using CASE tools in the software development process and 21% were not. In their survey, they asked about whether there was more than one tool used in the firm, whether every development activity should include CASE tools, whether a single kind of CASE tool was being used for every IT project, whether CASE tools would increase productivity, quality and organization maturity, and whether managers were satisfied with the interface of CASE tools. In their conclusion, they found that they may not benefit from CASE tools when being used for development activities without a defined software process. They also stated that before purchasing CASE tools, the total budget of software projects should be considered and costs and budget should be compared for a justification of the cost. Moreover, the staff of an organization had to be well trained to use CASE tools. The effect of CASE tools was also limited in terms of different productivity gains, enhancement of documentation, project communication and following project methodologies and standards [7].

Sundaram et al. conducted a survey show to how CASE tools helped in risk and planning, control of software projects and how CASE tools would improve efficiency and reduce risk in a system. The researchers were motivated to show how software projects would suffer from problems such as delay times, high computational costs through the design of a project and how to solve these problems by using software risk management, which would allow software developers to deal with risk items such as analysis and identification. The researchers explained the role of software project development to reduce project complexity and software project latency while the maintaining process. Additionally, according to the researchers, CASE tools can improve the quality of a system being developed and reduce the cost of software development, increase productivity, facilitate maintenance and make any development more enjoyable for software developers. They state that software project planning and control using CASE tools to deal with risks related to software

planning and control are limited by software engineering for risk-free software products and for the reduction of the complexity of a project or multiple related projects. However, there is no certain support on this point, common sense would prevail [8].

Budgen et al. they made a systematic literature review on 49 publications to identify empirical studies on UML in terms of its methods. in their finding they said that the UML was used by researchers as given and they seem reluctant to ask questions that might help to make it more effective and the existing studies is simple and one-dimensional, furthermore most of the studies focused upon class diagram. Finally, they concluded that there are a few researches has been conducted about adoption of UML and most of them are laboratory-based, these do tend to be centered on one viewpoint [9].

Lang and Fitzgerald explored the processes, methods and techniques being used for Web/hypermedia system designs in practice. Their research focused on the rate of use for UML in web construction development. In their findings, they state that the responders were not able to communicate efficiently during the development using diagrams, moreover the responders seem not included in the design of mission-critical systems. The traditional techniques still been used such as Entity-Relationship Diagrams.

Panos Fitsilis et al. conducted an exploratory survey to identify pinpoint UML diagrams that were being used and their effectiveness, to discover whether CASE tools were being used and to record the perceived usefulness of the UML language.in their survey there was about 190 projects using UML diagrams out of 413. The result show that most of the responders was programmers and this led them to say that most of the Greek IT companies were using UML for every different IT role, which implies that UML was a significant communication vehicle within the software IT community.

Finally, in their conclusion they said that the UML was only general-purpose modeling language and the industry use it as standard technic.

Dobing and Parsons conducted a survey to find out to what extent UML is being used and for what purpose, their survey data was supported by 182 respondents, there were 171 respondents as UML users.

In conclusion the researchers state that the highest usage levels occurred for Use Case Diagrams and Class Diagrams and the lowest for Collaboration Diagrams [14].

2.3 Summary

The literature we have reviewed leads us to question on standards, system analysis and development diagrams and role of CASE tools and IDEs. Following references highlight the chapter.

Standards

Zea Ordonez et al. stated that ISO and IEC are organizations that allow systems to standardize or regulate or provide guidelines for quality assessment, safety, etc., for the software industry and computer science applied globally [5].

Dariusz Rogowski stated that developers could use some standards, strict rules or requirements, methods and tools in order to prepare their products for successful assessment [19].

Diagrams

Dobing and Parsons investigate the usage of UML and to what purpose may be used and their effectiveness on communications within software development teams [15]

Panos Fitsilis et al conducted an exploratory survey to find out what the UML diagrams that are being used, to know if the CASE tools being used and to discover the usefulness of UML.

Lang and Fitzgerald focused on usage of UML in web development applications [8].

CASE Tools and IDE

Chiemeke et al. asked about whether there was more than one tool being used in the firm, whether all development activities should include CASE tools, whether a single kind of CASE tool is being used for all IT projects, whether CASE tools increase

productivity, quality and organization maturity, and whether managers were satisfied with the interface of CASE tools [7].

Muhammad Naseeret et al. asked about software requirements, productivity, time for explaining requirements, documentation of software development processes, their quality in the development process, and traceability [6].

Nandana Sundaram et al. performed a survey to determine how CASE tools helped in risk and planning, control of software projects and how CASE tools would improve efficiency and reduce risk in a system [8].



CHAPTER 3

SURVEY AND ANALYSES

3.1 The Questionnaire

We prepared the first draft that we designed through the issues we covered in the literature review in Chapter 2. We discovered a number of issues relating to CASE tools, however it was not sufficient to know the real problems and needs for such tools; therefore, we used a key informative methodology to enhance our questions. We visited two large, old companies in Ankara and met with IT professionals to validate the questions. From the interviews, we were able to discover new problems and issues that required investigation. After the validation, the final form of the questions was ready. Ankara was selected as we had validated our questions in that city as well as for the issues being related to customers in that city.

Six companies with different IT services in different industries were visited. We met with IT professionals with good experience in the field. Of three of the companies we visited, one was large and two were small. Each interview lasted between 20 and 45 minutes. From the interviews with some of the questions (tables 1-4) being structured and others being free text related to some issues, we collected our data.

Part 1: Organization History and Size

Table 1 Organization History and Size

Q. No.	Question	Scope
1.1	How long has the organization been in the IS industry?	This question determines the experience of the organization and it is important as it determines what kind of projects they work on.
1.2	How many employees does the organization have?	This question determines the size of the organization.
1.3	How many IS/software developers does the organization have?	This question determines the need for CASE tools according to human resources that assign IT projects.
1.4	What software solution and services do you provide in the market?	This question from the key informative interviews. the services and products that company provide it, determine the programming languages that they use and that's effect on the selection of their IDE or text editor which is a part of CASE tools.
1.5	Do you standardize software development? If yes, Do you follow any development methodologies?	This question from the key informative , The development methodology determines the usage of some of the UML diagrams in Part 3.
1.6	What software development languages are you using?	This question from the key informative interviews, development languages determines the needs for CASE tools and IDE.

Part 2: Standards

Table 2 Standards

Q. No.	Question	Scope
2.1	Are you certificated with any standards?	This question aims to determine the profile of the organization and to know what kind of projects they are developing which effect on their motivations to use CASE tools, such as ISO 90003.
2.2	2.2 Do any of your standards encourage the organization to use CASE tools? If yes, then please tell us about it.	for example ISO/IEC 15408, CMMI, and in our literature we found a support to use for example the Common Criteria standard is a very difficult and complex task for the developers Without design patterns and supporting tools, applying the Common Criteria requirements in IT projects becomes costly and time consuming [19].

Part 3: Diagrams

Table 3 Diagrams

Q. No.	Question	Scope
3.1	Do you use use-case diagrams in your projects? Always. In some projects. We do not use them.	This question comes from the literature review [13, 14, 15].
3.2	Do you use data flow diagrams in your projects? Always. In some projects. We do not use them.	This question comes from the literature review [16, 17].
3.3	Do you use activity diagrams in your projects? Always. In some projects. We do not use them.	This question comes from the literature review [16, 17].
3.4	Do you use ERD/Class diagrams in your projects? Always in some projects. We do not use them.	This question comes from the literature review [9, 12, 13, 15].
3.5	Do you standardize software development? If yes, do you follow any development methodologies?	This question comes from the key informative interviews , The development methodology determines the usage of some UML diagrams in Part 3.

Part 4: CASE Tools and IDE

Table 4 CASE Tools and IDE

Q. No.	Question	Scope
4.1	<p>Do you use any CASE tools in the preparation of the diagrams in Part 3?</p> <p>Yes.</p> <p>In some projects.</p> <p>We do not use them.</p> <p>Can you explain your answer in terms of project size, requirements and human resources?</p>	<p>CASE tools can be used to prepare the diagrams in Part 3. This question has been used before in our literature review [14].</p>
4.2	<p>Do you use CASE tools to generate code from diagrams?</p> <p>Always.</p> <p>In some cases</p> <p>We do not use them.</p> <p>If the answer is “In some cases,” can you explain your criteria?</p>	<p>This question comes from the literature review [14, 17].</p>
4.3	<p>Which IDE do you prefer in coding and code generation?</p>	<p>This question comes from the literature review [5].</p>

Part 5: General Idea

5.1 Can you state the advantage(s) you have gained by using CASE tools?

5.2 Can you state the disadvantage(s) that have become apparent when using CASE tools?

3.2 Results

• Company 1

Company 1 was a middle-sized company with approximately 235 employees and 100 IS employees with nearly 21 years' experience working in the GIS field, mobile services and mapping services, which is good experience in the IT field. The company standardizes with CMMI 3 and a number of ISO standards and they follow the Agile methodology for software development. Some of the standards encourage

them to use CASE tools because of their use of diagrams in design and analysis reports and CASE tools is a good choice for accuracy of these diagrams in these reports as well as for the uniformity of the diagrams. For example, they are certified with ISO 90003:2015 the term 7.3.3 Design and development outputs, “the output from design and development process should be defined and documented in accordance with prescribed or chosen method. This output should be complete, accurate, and consistent with the requirements. design and development output may be expressed in textual form, by diagrams or using symbolic modeling notation. may be producing using computer design and development tools.”

From the term above, it can be said that CASE tools can be used to draw symbolic modeling notation and diagrams such as ERD diagrams, data flow diagrams, etc. for documentation purposes, and to make these diagrams understandable, well-defined, accurate with easy-to-follow relationships and notations among diagrams, and to be uniformed and standardized to others to read and modify , maintainable and saved in appropriate extension to be used and read later in the stage of design and development review in term 7.3.4.

They used Java, C++, C# and object C in addition to the respective frameworks of these languages. In some of their projects, they would use UML diagrams and CASE tools in the preparation of these diagrams. For large projects and for contracted agreements, especially governmental contracts, they have been required use CASE tools. However, CASE tools were not used to generate code from diagrams due to its limited ability to solve problems and business needs. From the interview, we can summarize the advantages of CASE tools such that they can simplify development activities by saving time for diagram preparations and having developers be more connected with other team members through the uniformity of any diagrams prepared using CASE tools. No disadvantages were recorded for this company.

- **Company 2**

Company 2 was a large company of approximately 1,200 employees and 500 IS employees with nearly 20 years of experience working in the field of HBYS HIS. Most of their projects were in the health sector and mostly in hospitals. The company standardized with CMMI 3 and a number of ISO standards. They followed the Agile

and Waterfall methodologies for software development. Some of the standards encouraged them to use CASE tools, especially ISO 9001 for the same reason as Company 1 as these companies would use 90003 to apply the ISO 9001 standard in the field of software development. This company used Java and C# and they would use the frameworks of these languages, especially .net. UML diagrams were used in all of their projects and they used CASE tools in the preparation of their diagrams. They were required to use CASE tools in all their projects as they would use UML diagrams in every project because of the nature of their projects being required to be standardized as a part of a contract, especially governmental projects. They did not use CASE tools to generate codes from their diagrams. Again, the same issue as Company 1 occurred due to their limitations in this matter. From the interview, we are able to summarize the advantage of CASE tools, which can help in configuration management and facilitate development activities and standardize development procedures and maintenance. Two disadvantages were the cost and the materials needed for training to use these CASE tools.

- **Company 3**

Company 3 was a large company with approximately 800 employees and 600 IS employees with nearly 22 years of work experience with command control systems, cyber security and big data. Their projects included research projects and industry. They used Java and Python for research projects, C++ and C# for industry, the .net framework in some large projects and Python machine learning libraries. The company standardized with CMMI 3 and a number of ISO standards. They followed the Agile Scrum and Waterfall methodology for software development. Some of the standards encouraged them to use CASE tools; for example, at the organizational level, they followed CMMI 3. In this standard, there were roles, tasks and products. Each task should be assigned to a number of roles or one role with tasks having inputs and outputs. Each role would have a number of skills and a description. This network needed to be followed precisely and for this, they used CASE tools such as the Eclipse Process framework composer (EPF) (figures 5-7). These tools provided the flexibility to follow the CMMI methodology through the planning of activities, tasks and roles, after which it would generate a report in an HTML format. This format showed the structure of the project and the details of each activity could

easily be followed. Moreover, one could see which document needed to be included in an activity or process. The figures below show a number of examples of such CASE tools.

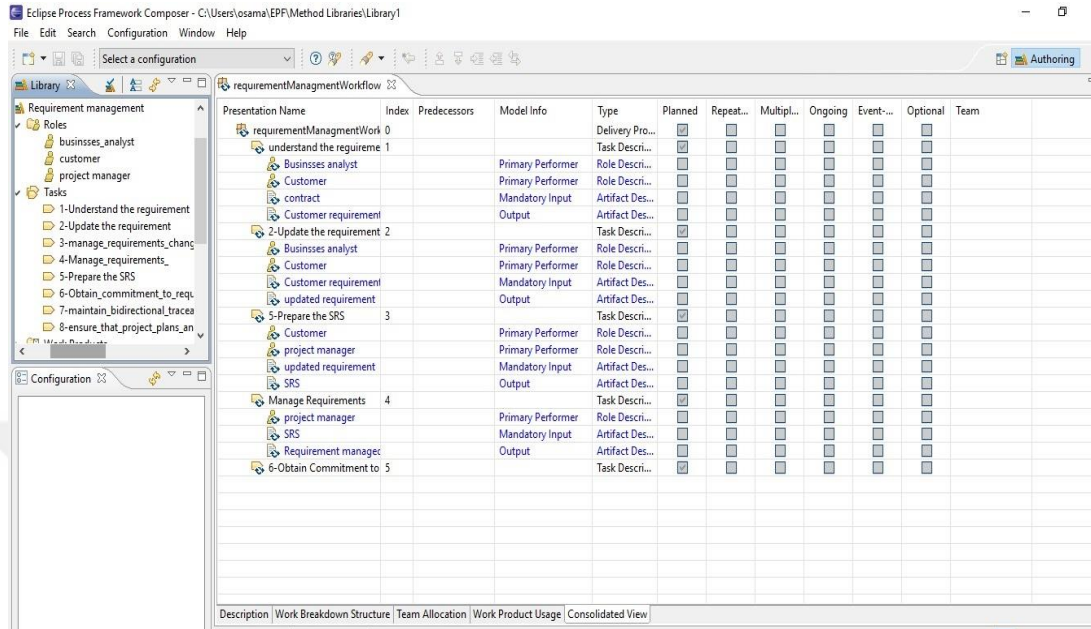


Figure 5 Example 1 of the Eclipse Process Framework composer (EPF)

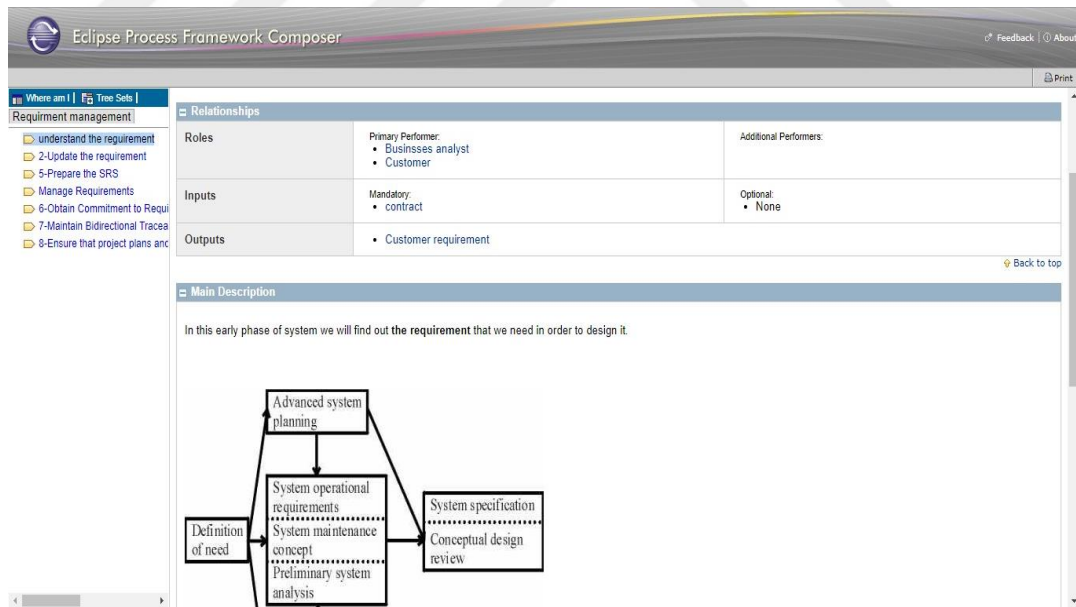


Figure 6 Example 2 of the Eclipse Process Framework composer (EPF)

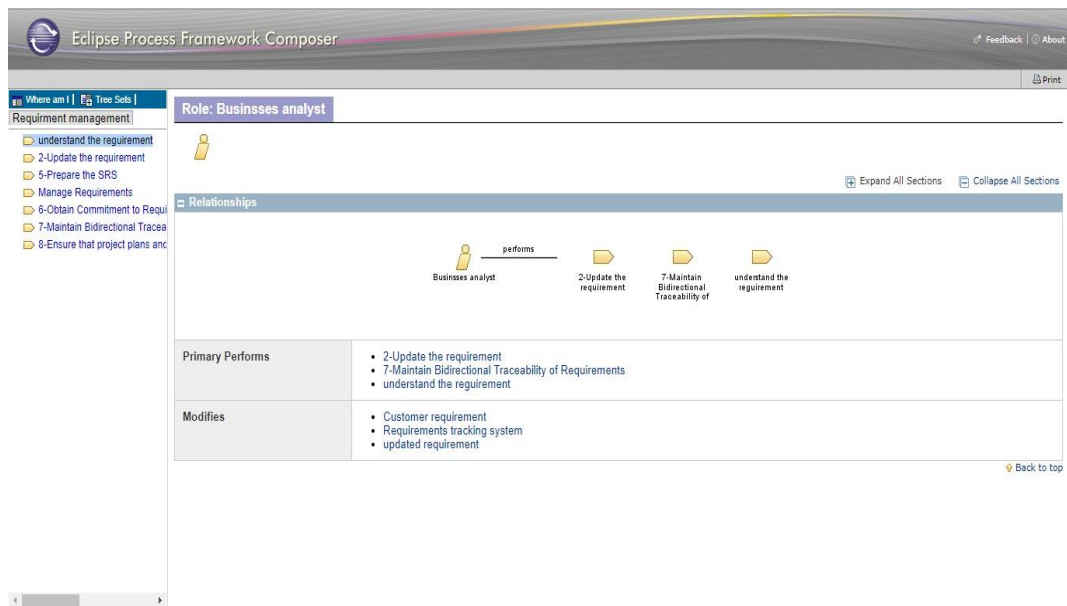


Figure 7 Example 3 of the Eclipse Process Framework composer (EPF)

They would use UML diagrams in some projects and CASE tools in the preparation of their diagrams. They did not use CASE tools to generate code from their diagrams. Again, the same issue as Companies 1 and 2 occurred due to their limitations in this matter. From the interview, we are able to summarize the advantage of CASE tools, facilitate software engineering activities, avoid human error and save more time for developers. One disadvantage was cost, the need for trained staff, and time consumption.

- **Company 4**

Company 4 was a small company with about three employees and three IS employees with nearly three years' experience working with mobile and web development. They used asp.net, PHP and the frameworks for these languages. The company did not standardize any standards, and they did not follow any methodology for software development. They used UML diagrams in a number of projects and CASE tools in the preparation of their diagrams. They did not use CASE tools to generate code from diagrams. From the interview, we could summarize the advantage of CASE tools such that they helped to prepare documentation, facilitate software engineering activities and saved more time for developers. One disadvantage was the cost.

- **Company 5**

Company 5 was a small company with approximately 15 employees and 13 IS employees with nearly 10 years of experience working with the Web, micro services, machine learning services, and big data analysis. The company did not standardize any standards and they followed the gangs of four methodologies for software development. They used Java, Python and UML diagrams in a number of projects as well as CASE tools in the preparation of these diagrams. They did not use CASE tools to generate code from their diagrams. From the interview, we can summarize the advantage of CASE tools such that it would help them to prepare their documentation, standardize development and occasionally provide for customer explanations. One disadvantage was that it was time consuming.

- **Company 6**

Company 6 was a small company with 15 employees and 14 years of experience. They provided e-government and e-university solutions. They also had a unique vertical search engine and thematic/business ecosystem management software as products which they sold as SAAS or on-premises models. The company provided consultancy services to their clients in middleware technologies. They mostly used Agile Software development methodologies such as Scrum. However, in some rare cases, usually in government projects, they also used Waterfall as a methodology where the customer's needs were very well defined and not subject to change. They mainly used Java with Spring/Hibernate frameworks, Ionic for mobile development and Python.

This company has ISO-9001:2016 and ISO/IEC 27001:2013 certifications. They stated that the standards for which they are certified do not specifically enforce or encourage the use of CASE tools. However, they require a very detailed and straightforward definition of company processes, standardization and documentation of these processes. Software development was one of their processes and they would define and follow a well-defined methodology for software development, including testing (security, unit tests and performance tests) and deployment procedures. They also had a process definition for change management and back-up policies. In these

processes, they would define relevant CASE tools and how and when they would be used.

They used every UML diagram in their projects and CASE tools to prepare them. They stated that they used CASE tools in most of their projects without considering project sizes, requirements and team sizes since even if a project were small, the outcomes of these tools would provide an organizational memory and make change management easy, especially if any team members were to leave the company.

They did not use CASE tools to generate codes. They used STS (Spring Tools Suite), Eclipse, Xcode and Pycharm IDEs for coding.

They spoke about the advantages of CASE tools such that they developed multi-layered enterprise software with mostly an MVC pattern. A well planning/requirement analysis/designing stage of these kind of software before starting coding process makes it easier when the actual coding starts as even a small change in the latest stages of a project may have a huge effect on the architecture of the product, which may mean great time and resource loss, and this situation may also lead to not being able to comply with the dead line of the project.

Having documentation generated by CASE tools in the planning stage would make it easier and quicker for a new team member to join and integrate into the project. Moreover, it would also lead to fewer bugs.

Change management would also be easier as one can follow which changes affect which parts of the product.

With regard to IDEs, it makes coding very easy and prevents bugs starting with real time coding as IDEs provide very advanced debugging tools to solve problems. They are integrated with sub-versioning tools which makes working as a team very easy and problem free.

According to Company 6, the disadvantage of CASE was that code generation (from class diagrams, DDL from ER, etc.) may result in loss of control in the later stages of a project. Secondly, it would affect less experienced team members as they do not have a full understanding of these parts and would thus avoid using this aspect.

3.3. Discussion of Results

For large and medium-sized companies, we can conclude that they all are certified with CMMI standards and a number of ISO standards, especially 9001. Most of their projects were governmental projects and for that reason, most of their projects had to be standardized. These companies used standardized software development methodologies, especially the agile methodology. CASE tools were of great help for them to follow their standards. Feedback we have collected is summarized on Table 5 A and Table 5 B.



Table 5 A : Summary of responses

Question	C1	C2	C3
1.1	21 years	22 years	20 years
1.2	235	800	1200
1.3	95	600	500
1.4	Mobile GIS mapping service, data collection, data cleaning	Command control system, big data, cyber security	HBYS HIS
1.5	Yes	Yes	Yes
1.6	Agile	Waterfall, agile scrum, increment	Agile, waterfall
2.1	C#, C++, Java, Object C	C++ , Java, Python, C#	.Net Java
2.2	CMMI 3, ISO 9000 ISO 8000 ISO 7000	CMMI, ISO 2770 ISO 230-1	ISO 9001, ISO 27001, ISO 2097
3.1	Yes	Yes	Yes
3.2	In some projects	In some projects	Always
3.3	In some projects	In some projects	Always
3.4	In some projects	In some projects	Always
3.5	In some projects	In some projects	Always
4.1	Yes	Yes	Yes
4.2	Not used	Not used	Yes
4.3	Eclipse Android Studio	Eclipse Visual Studio	Visual Studio Note ++ JKS
5.1	Free human errors, productivity Documentation avoid errors, configuration management	Free human errors, productivity Documentation avoid errors maintenance	save time, standardize development documentation configuration management maintenance for customer explanation, avoid errors
5.2	None	Cost, It needs for training	Cost, It needs for training

Table 6 B: Summary of responses

Question	C4	C5	C6
1.1	3 years	10 years	14 years
1.2	3	15	15
1.3	3	13	12
1.4	Web and mobile	Web, micro services, machine learning service, big data analysis	e-governme-nt and e-university solutions
1.5	No	Yes	Yes
1.6	---	Gangs of four	Agile scrum, Waterfall
2.1	Asp .Net PHP	Java and Python	Java and Python
2.2	No	No	ISO-9001:2016 ISO/IEC 27001:2013
3.1	No	No	Yes
3.2	Always	In some projects	Always
3.3	Always	In some projects	Always
3.4	Always	In some projects	Always
3.5	Always	In some projects	Always
4.1	Yes	Yes	Yes
4.2	Not used	Not used	Not used
4.3	Visual Studio Android Studio VS code	Eclipse	Eclipse Xcode and Pycharm IDEs
5.1	save time, standardize development documentation fast development process	Standardize development, documentation, fast development process for customer explanation,	Documentation is easier and quicker, fewer bugs in the development, Changing management is easier coding is easier prevents bugs.
5.2	Cost	time consuming	None

Now we can disagree with Chiemeké et al. [3] because in our literature review, they state in their findings that the effect of CASE tools is limited to following project methodologies and standards. Small companies do not follow any standards. However, all companies use UML diagrams in most of their projects and occasionally, they do not use them for internal projects or for small projects that do not need to be standardized as part of contracts with customers. CASE tools have been used to draw UML diagrams, which emphasizes the use of Upper CASE tools to help analysts in the early stages of development, such as design and analysis while working on prototyping and modeling. Java, Python, C# and C++ were the most used languages and they used the frameworks of these languages, which explains the reason for using Eclipse and Visual Studio. There are a number of advantages of CASE tools and we can conclude that they improve the quality of the industry through the standards. However, this does not mean they always improve productivity for organizations as some complain about time consumption through the developments. The disadvantages were related to the cost. The same findings were found by Ahmed Abd El-Rahman Mahdy et al. [14], who stated that CASE tool expenses were typically high due to their steep learning curve, and it is more important to ensure the CASE investment delivers value for the money invested. In other findings, Chiemeké et al. [3] stated that before one could purchase CASE tools, one should consider the total budget of the software project and compare it to be lower in cost. Another disadvantage was that these tools needed trained staff. In the same findings by Chiemeké et al. [3], the researchers stated that the staff of an organization would need to be well trained to use CASE tools.

CHAPTER 4

CONCLUSION

4.1 Conclusion

In this study, we discussed the use of CASE tools in local companies. This study has provided essential information related to the use of CASE tools and UML diagrams. We were informed in the interviews we had conducted with software developers and IT managers that the use and adoption of CASE tools and UML in Ankara is high, regardless of the size of the organization and software solutions. However, their use and adoption is variably related to organization sizes and standards.

CASE tools have been used to facilitate project developments and because of the standards and methodologies of enforcement. However, they have been adopted only for documentation and there was a lack of use in engineering activities.

Both literature surveyed and feedback from the interviewees inform that software organization without a defined software process may not benefit from the use of CASE tools for technical and development activities. For small organizations, they are used for communication between project members and in rare cases for documentation. For large organizations they are used for documentation as a part of contracts and to follow standards. However, they require a detailed and straightforward definition of company processes, standardization and documentation of these processes. We can conclude that some standards have indirect effects on CASE usage.

Software development methodology has an effect on the use of CASE tools due to the diagrams used in these methodologies.

CASE tools come with a number of advantages and we can conclude that they improve the quality of the industry through the standards. However, it does not mean

they always improve the productivity for organizations as a number of them complain about time being consumed through the developments.

Having documentation generated by CASE tools in the planning stage will facilitate and accelerate new team members joining and integrating with projects. Moreover, it also leads to fewer bugs.

Change of management will also be easier as one can follow which changes effect various parts of a product.

Management and technical staff having to be trained in the use of CASE tools and costs involved are two of the disadvantages of CASE tools.

It has been seen that CASE tools are not preferred to generate code as this may result in losing control in the later stages of a project, which affects less experienced team members as they do not have a full understanding of these parts. This finding basically opens a new discussion, namely on future work, regarding the use of CASE tools in software engineering.

4.2 Limitation

Primary limitations in this research include local research for six companies only and the findings relying on the previous experience of the developers; apparently one person's experience may limit the presentation of multiple viewpoints. Another limitation is about prices of the CASE tools and cost, both in time and human resources, of adopting them in organizations.

4.3 Future Work

In our study, we focused on UPPER CASE tools and in the future, we want to consider LOWER CASE or INTEGRATED CASE tools, such as project management tools and which ones to use (which will be an INTEGRATED CASE TOOL for use throughout the project). Moreover, we can focus on testing tools (which may be separated to unit tests, performance test, code quality and security) and what testing methodology to use as well as what tools to use for bug/issue tracking (Redmine, etc.) and what tools to use for changes in management.

REFERENCES

- [1]. Paola, Z. O. M., Rafael, V. P. M., Rolando, M. R. J., & José, C. S. M. (2016). Analysis of Upper CASE Tools in Software Design Process. *International Journal of Applied Engineering Research*, 11(18), 9377-9384.
- [2] Vijayasarathy, Leo R., and Charles W. Butler. "Choice of software development methodologies: Do organizational, project, and team characteristics matter?." *IEEE software* 33.5 (2016): 86-94.
- [3] Georgiades, M. G., & Andreou, A. S. (2010, November). Automatic generation of a software requirements specification (SRS) document. In *2010 10th International Conference on Intelligent Systems Design and Applications* (pp. 1095-1100). IEEE.
- [4] IEEE Standard for Information Technology--Systems Design--Software Design Descriptions," in *IEEE STD 1016-2009* , vol., no., pp.1-35, 20 July 2009 doi: 10.1109/IEEESTD.2009.5167255.
- [5] Paola, Z. O. M., Rafael, V. P. M., Rolando, M. R. J., & José, C. S. M. (2016). Analysis of Upper CASE Tools in Software Design Process. *International Journal of Applied Engineering Research*, 11(18), 9377-9384.
- [6] Naseer, M., Rehman, H., & Saeed, S. (2013). A Survey on Case Tools Adoption for Requirement Management in Pakistani Software Industry. *World Applied Sciences Journal*, 21(1), 94-98.
- [7] Chiemeké, S. C., & Egbokhare, F. A. (2005). The Use and Adoption of Computer Aided Software Engineering Tools in Software Development Organizations in Nigeria. In *The Nigerian Academic Forum, A Multidisciplinary Journal* (Vol. 9, No. 1, pp. 1-8).
- [8] Nadana Sundaram, P. V., & Iyakutti, K. (2015). A Survey on Software Project Planning and Control Using Case Tools. *IJCSIT) International Journal of Computer Science and Information Technologies*, 6(1), 326-331.
- [9] Budgen, D., Burn, A. J., Brereton, O. P., Kitchenham, B. A., & Pretorius, R.

- (2011). Empirical evidence about the UML: a systematic literature review. *Software: Practice and Experience*, 41(4), 363-392.
- [10] Grossman, M., Aronson, J. E., & McCarthy, R. V. (2005). Does UML make the grade? Insights from the software development community. *Information and Software Technology*, 47(6), 383-397.
- [11] Dobing, B., & Parsons, J. (2008). Dimensions of UML diagram use: a survey of practitioners. *Journal of Database Management (JDM)*, 19(1), 1-18.
- [12] Lang, M., & Fitzgerald, B. (2006). New branches, old roots: a study of methods and techniques in web/hypermedia systems design.
- [13] Wrycza, S., & Marcinkowski, B. (2007). Towards a light version of UML 2. X: appraisal and model. *Organizacija*, 40(4).
- [14] Fitsilis, P., Gerogiannis, V. C., & Anthopoulos, L. (2013). Role of unified modelling language in software development in Greece—results from an exploratory study. *IET software*, 8(4), 143-153.
- [15] Dobing, B., & Parsons, J. (2006). How UML is used. *Communications of the ACM*, 49(5), 109-113.
- [16] Li, Y., & Ye, C. (2011, August). The analysis of CASE tools to the efficiency of software development. In *Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology* (Vol. 1, pp. 67-69). IEEE.
- [17] de Castro, T. R., de Souza, S. N. A., & de Souza, L. S. (2012, June). CASE tool for object-relational database designs. In *7th Iberian Conference on Information Systems and Technologies (CISTI 2012)* (pp. 1-6). IEEE.
- [18] Mahdy, A. A. E. R., Ahmed, M. M. A. E. S., & Zahran, S. M. (2008). Flexible CASE tools for requirements engineering: a benchmarking survey. In *International Conference of Informatics and Systems* (No. 2, pp. 20-26).
- [19] Rogowski, D. (2013). Computer-aided tool based on Common Criteria related design patterns. *Business Informatics*, 3(29), 111-127.
- [20] International Organization for Standardization

<https://www.iso.org/standards.html> [visited 18.Sept.2019]

[21] Chrissis, M. B., Konrad, M., & Shrum, S. (2011). *CMMI for development: guidelines for process integration and product improvement*. Pearson Education.

[22] Goldenson, D. R., Gibson, D. L., & Ferguson, R. W. (2004). *Why make the switch? Evidence about the benefits of CMMI*. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.

[23] Software Case Tools Overview from
https://www.tutorialspoint.com/software_engineering/case_tools_overview.htm [visited 18.Sept.2019]



CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: ASHOUR, Osama

Date and Place of Birth: 10 April 1992, Sallah-aldeen

Marital Status: Single

Phone: +905369579586

Email: superstar8335@gmail.com



EDUCATION

Degree	Institution	Year of Graduation
M.Sc.	Çankaya University	2019
B.Sc.	University of Tikrit	2014
High School	Al-alam	2010

FOREIGN LANGUAGES

Advanced English, Beginner Turkish, Native Arabic