

# **A NEW APPROACH FOR JOINT APPLICATION DEVELOPMENT BY USING GROUP SUPPORT SYSTEMS**

**A Thesis Submitted to the  
Graduate School of Natural and Applied Sciences of  
Dokuz Eylül University  
In Partial Fulfillment of the Requirements for  
the Degree of Master of Science in Computer Engineering,  
Computer Engineering Program**

**T.C. YÜKSEKÖĞRETİM KURULU  
DOKÜMANTASYON MERKEZİ**

**by  
Kökten Ulaş BİRANT**

**July, 2002  
İZMİR**

119597

## Ms.Sc. THESIS EXAMINATION RESULT FORM

We certify that we have read the thesis, entitled “A NEW APPROACH FOR JOINT APPLICATION DEVELOPMENT BY USING GROUP SUPPORT SYSTEMS” completed by KÖKTEN ULAŞ BİRANT under supervision of PROF. DR. ALP KUT and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

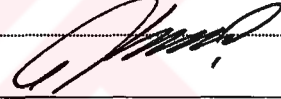


Prof. Dr. Alp KUT  
Supervisor

119597

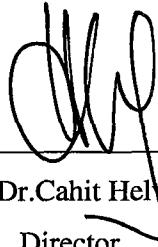


Committee Member  
Doç. Dr. Yalçın ÇEBİ



Committee Member  
Prof. Dr. Ahmet KAŞLI

Approved by the  
Graduate School of Natural and Applied Sciences



Prof. Dr. Cahit Helvacı  
Director

---

## ACKNOWLEDGMENTS

---

I'd like to thank all who helped me to make this work. In particular, I want to express gratefulness to the following people:

Firstly I should thank to my wife, Derya for her love and support. When she didn't be with me, I couldn't be so ordinary and this thesis can't be completed.

To my mother, Gönül DENİZERİ for her support and for teaching me to see, to read, to learn and to think. To my brother Çağdaş BİRANT for cheering me when I was exhausted from hard work.

To my thesis supervisor, Prof. Dr. Alp KUT, for his guidance during my studies and trust in my work.

To one of my best friends Mustafa KASAP, for his invaluable helps.

To my friends from the group of the supporters of Fenerbahce SK. When they wouldn't be, I can't be so relaxed and can't finish the thesis.

To Evren TEKİN for his help on formatting this thesis report.

Kökten Ulaş BİRANT

---

## ABSTRACT

---

Developing software was not so problematic in the past. Before 1970's, there are not so many computers and not so many needs for new software. Computers were only usable for academic and military researches. After PC (Personal Computer) revolution, the users and the needs are starting to have variety. This means giving more importance on the user satisfaction, because computer science is not an academic research, but a business after personal users. To create the satisfaction for development time and usability, the developers started to define some methodologies. The Computer Science departments also begin to prepare controlled methodologies for this growing software industry and the "Project Management" concept is started to be used.

After 1980, the quality concept was grown, the team approach's advantages are started to discuss, the users become smarter and more experienced about the computer usage and user – oriented design methodologies were trendy. The Joint Application Development (JAD) idea was prepared in such a manner. And after the Internet / networking trend (after taking frontiers off the map), this development methodology should also be started to be used in "Wide-Area" project via Group Support Systems (GSS). And the aim of this project is to re-define the methodology for such usage.

---

## ÖZET

---

Yazılım geliştirme işlemi geçmişte daha az sorunları olan bir işlemdi. 1970'ler öncesinde bilgisayarlar çok yaygın değildi ve yazılım gereksinimleri de buna bağlı olarak daha sınırlıydı. Bilgisayar kullanımı akademik ve ordunun tek elinde kabul edilebilirdi. Bilgisayar dünyasındaki PC (Kişisel Bilgisayarların geliştirilmesi) devriminden sonra kullanıcılar ve buna bağlı olarak ihtiyaçlar değişim göstermeye başladı. Artık kullanıcı memnuniyetine daha fazla önem verilecekti, çünkü bilgisayar dünyası bilimsel bir alandan ticaret düşüncesine geçiş yapmaktaydı. Geliştirme zamanı ve kullanılabilirlik kavramlarının yerleşmesi ile de geliştiriciler bu kavramlarda kendilerini yeterli kılacak (veya piyasada onlara avantaj sağlayacak) sistematiğe geliştirme durumunda kaldılar. Bu durumda bilgisayar bilimleri de gelişen bilgisayar sektörüne desteklerini, kontrollü metodolojiler geliştirerek gösterdiler ve "Proje Yönetimi" kavramını oluşturarak bu kavram altında çalışmalarını yürütmeye başladılar.

1980'lerden sonra, kalite kavramının ortaya çıkışı, takım olgusunun yararlarının farkedilmesi, kullanıcıların bilgisayar sistemleri konusundaki gelişmeleri ve kullanıcı – odaklı sistematik akımının etkisi yeni metodolojilere yön verdi. Birleşik Uygulama Geliştirme (JAD) fikri bu kavramlar üzerine oturdu. Ve dünyada Yerel Ağ / Internet çalışmaları ile sınırların kalkmasının etkisiyle bu sistematik Dağıtık Projelerde de Grup Destek Sistemleri (GSS) aracılığıyla kullanılması gündeme geldi. Çalışmamızda bu fikrin geliştirilmesine yönelik olarak bir sistematik ile JAD'in yeniden tanımlaması yapılmaktadır.

---

# CONTENTS

---

	<b>Page</b>
Contents .....	VI
List of Figures .....	X

## Chapter One

### INTRODUCTION

1 INTRODUCTION .....	1
1.1 History and Software Development .....	1
1.2 Group Support Systems (GSS) .....	3
1.3 Previous Works .....	4
1.4 Thesis Organization .....	5

## Chapter Two

### JOINT APPLICATION DEVELOPMENT / DESIGN

2 JOINT APPLICATION DEVELOPMENT / DESIGN .....	7
2.1 History .....	9
2.2 Members of JAD Session .....	10
2.2.1 Executive Sponsor: (Project Sponsor, Sponsor) .....	10
2.2.2 IS – Project Team: (Technical Personnel, Development Team) .....	11

2.2.3	Full – Time and On – Call Participants: (Potential Users, Client Representatives).....	12
2.2.4	Observers .....	13
2.2.5	Scribe (Secretary, Documenter, Record Keeper, Modeler) .....	14
2.2.6	Facilitator (Session Leader, Meeting Leader, JAD Leader) .....	15
2.3	Implementation Details .....	18
2.4	Advantages.....	21
2.5	Disadvantages / Pitfalls.....	22
2.6	Performance Statistics.....	24

### Chapter Three

#### COMPARISON OF JAD WITH RAD, PD

3	COMPARISON OF JAD WITH RAD, PD .....	26
3.1	Rapid Application Development (RAD).....	27
3.2	Participatory Design (PD).....	28
3.3	Comparison between RAD, JAD and PD .....	29

### Chapter Four

#### JAD - NW

4	JAD – NW.....	32
4.1	Basic First Agenda .....	33
4.1.1	Interview .....	33
4.1.2	Defining User Profiles .....	34
4.1.3	Distribution and Education Phase .....	34
4.1.4	First Meeting – Discussing Agenda .....	35
4.1.5	System Discussions.....	35
4.1.6	Technology Discussions .....	35
4.1.7	Discussing Data Model and Interface .....	36

4.1.8	Last Overview .....	36
4.1.9	Reporting .....	37
4.2	Team Members .....	37
4.2.1	Meeting Administrator .....	38
4.2.2	The Software .....	39
4.3	Solutions to Existing JAD Problems.....	40
4.3.1	Costs.....	41
4.3.2	Domination.....	41
4.3.3	Overloaded Facilitator.....	42
4.3.4	Attention.....	42
4.3.5	Misunderstanding and Not-Heard .....	42
4.3.6	Standardization.....	43
4.4	Pitfalls .....	43

## Chapter Five

### UCHE PROJECT

5	UCHE PROJECT .....	45
5.1	Programming Environment.....	45
5.2	General Design.....	45
5.3	Capabilities.....	47
5.3.1	Step By Step Guideline .....	48
5.3.2	Discussion .....	48
5.3.3	Member Information .....	48
5.3.4	File Transfer .....	49
5.3.5	Picture Show .....	50
5.3.6	Messaging .....	50
5.3.7	Slide Show .....	51
5.3.8	Alert .....	51
5.3.9	Questionnaire .....	52
5.3.10	Analysis of Questionnaire .....	52



5.4	User Guides.....	53
5.4.1	Guide for Meeting Administrator.....	54
5.4.2	Guide for Clients .....	73

## Chapter Six

### CONCLUSION & FUTURE WORKS

6	CONCLUSION & FUTURE WORKS.....	84
6.1	Testing.....	85
6.2	Improvements on Tool .....	85
6.2.1	Using Multimedia .....	86
6.2.2	Security Improvements .....	87
6.2.3	More Parameterized Manner.....	87
6.2.4	Automatic Documentation with Artificial Intelligence.....	87
	References.....	88

---

## LIST OF FIGURES

---

Figure 2.1 A typical JAD room.....	9
Figure 2.2 Drexler/Sibbet Team Performance Model (Drexler/Sibbet, 1990).....	18
Figure 3.1 Development Methodologies according trends .....	26
Figure 5.1 General View of UCHE Project .....	54
Figure 5.2 UCHE Center .....	55
Figure 5.3 Creating a new project.....	56
Figure 5.4 Opening an existing project.....	57
Figure 5.5 Guideline (or Agenda) to follow JAD .....	58
Figure 5.6 Accepting a step as completed.....	59
Figure 5.7 Configuration of steps in guideline .....	60
Figure 5.8 Opening a discussion .....	61
Figure 5.9 Discussion platform.....	62
Figure 5.10 Message Box.....	63
Figure 5.11 Writing a new message.....	64
Figure 5.12 Picture Show .....	65
Figure 5.13 Slide Show .....	66
Figure 5.14 Next Meeting Alert .....	67
Figure 5.15 File transfer platform .....	68
Figure 5.16 Team Information platform .....	69
Figure 5.17 Preparing Questionnaire .....	70
Figure 5.18 Adding a new question .....	71
Figure 5.19 Questionnaire Analysis.....	72
Figure 5.20 General View of UCHE project.....	73
Figure 5.21 UCHE Center .....	74

Figure 5.22 Creating a new project .....	75
Figure 5.23 Opening an existing project .....	76
Figure 5.24 Discussion platform .....	77
Figure 5.25 Message Box .....	78
Figure 5.26 Writing a new message .....	79
Figure 5.27 Picture Show .....	80
Figure 5.28 Slide Show .....	81
Figure 5.29 Answering a questionnaire .....	82
Figure 5.30 Answering an information request .....	83



---

## CHAPTER ONE

# INTRODUCTION

---

This thesis is devoted for explaining a new approach on Joint Application Development (/Design) by using the Group Support Systems. A new tool is also developed to use as a Group Support System with checklists for the new approach of JAD.

In the following chapters, an introduction to Joint Application Development will be done and an overview of other methodologies developed will be given. After that, the theoretical design of the new Joint Application Development System is explained. At last, architecture and the implementation details of the Group Support System developed in Computer Science Engineering Department of Dokuz Eylül University will be described.

### 1.1 History and Software Development

Software development is not so problematic in the past. There are only academic and military people who use computers. As a result of the usage of the computer, they need this tool only for faster calculations and analysis. This means, they don't need new software. Their only need can be preparing some software to run their calculations faster. In these years, the computer science departments were working on better computers (faster hardware or some algorithms to let the computers work faster). The computer industry works only on the high performance computers for universities or military needs.

And in 1948, the first transistor was developed instead of tubes. After this year, the computers begin to be small-sized. This means that the computer can be used also in industry and government. According to these new usage areas, the software development industry begins to earn money. These new users need new software always. The only need is storing some information and checking in a better way than papers. For better software, the computers hardware should be developed. There is no quality concept or developing better software opinion in the Computer industry. The only work is on hardware development.

After another revolutionary invention, "Integrated Circuits", the computer's internal pieces become smaller. This means that the computers are starting to be more compatible to use without large spaces and there is a hint to develop the performance of the computers by using silicon technology and integrated circuits.

The big revolution was the development of the Personal computers (PC's) with lower prices and smaller place requirements. This means that each person can buy his own computer and after buying a computer, each user becomes a customer for the software industry. As a result of this, the software development industry begins to grow. The software developers think on the needs of the users and develop software for public market. However, there is also a boundary for developers: The boundaries of the performances of the computers. The computers are maturing slowly and the developers cannot realize their all ideas. The good thing is the customer size in the market. The people can buy their own computers to be the customers for software developers. And the developers are not so many. This cause that the developers make what they want, not what the users want.

In the near past, the number of developers increases geometrically. This gives a selection chance to the users. The users can choose the software developer, who will satisfy him. There is no performance problem from the computer and the only choosing criterion between software developers is the satisfaction.

In this manner, software engineering departments were started to build the differences between the software developers. The differences are started to be written like quality concepts, deadline managements, user interactions, validations, etc. To check these criteria, the development companies should run on some methodologies. With this way, they can be ensured on these criteria.

The Joint Application Development is developed for such purpose. The only aim of JAD is increasing the user satisfaction in the project development operation. There are more than one names of the Join Application Development, like Joint Application Design, Accelerated Design, Facilitated Design, Facilitated Sessions, Facilitated Team Techniques, Facilitated Work Sessions, Group Design, Interactive Design, Interactive JAD, Joint Requirements Planning, Joint Sessions or User – Centered Design. The Joint Application Development is prepared under the trend of the team approach. The team approach is a new concept, which is based on the idea “There is somebody wiser than any of us, and that is all of us (Napoleon BONAPARTE)”. The team approach, which can be supported by the Brainstorming idea, can be too effective. The team approach was not crashed, because the users, nowadays, are smarter and more experienced about the computers. The users know something about the computer technology and they can also understand a low-level jargon of computer science. This increases also the communication between the project developers and customers.

## **1.2 Group Support Systems (GSS)**

The computer science development has also one more revolutionary invention: Network / Internet. Connecting two machines together to exchange information is a big idea for the computer world. This means that there is no boundary and distance between the countries. The technical meaning is that two people will be together, when they have computers connected to a network (We can call also Internet as a network where all of the computers are connected via existing telephone lines.). As a result, the information is started to be exchanged over lines. And the people from different countries have the chance to work together.

The next step is working together effectively. To make this process easier, the Group Support Systems are developed. The starting aim of the GSS is creating a platform to realize the meetings on computers via networks. The main idea is taking all of the facilities in the real world into the computer as realistic as possible.

The first idea on Group Support Systems came from Nunamaker and Dennis (1991) with the definition of “the software tools, each of which uniquely focuses on enhancing specific team dynamics or processes” (Nunamaker, Dennis et al. 1991). The developed GSS concept has some tools in it like brainstorming, idea generation, open discussion, idea organization, issue analysis, alternative analysis, questionnaires, polling, group writing, and process modeling. A good GSS application includes not only these abilities. Taking the control of a well-trained facilitator is one of other important points, who can set rules, maintain group memory, maintain agenda and check effectiveness of the usage of the GSS modules.

A well-trained Group Support System session can have many advantages on the project management and development. Because the development process goes on a virtual place, the anonymity possibility occurs and this causes to produce new ideas easily. The electronic environment also makes the communication easier. The group size can be enlarged, without thinking over the place constraints. The parallel communication may be done. Meeting Domination problems decrease. A written task structure can support more discipline on the team to follow.

### **1.3 Previous Works**

The previous works can be organized in three parts. In the first phase, the works on JAD can be discussed. In the second phase, the GSS improvements should be thought. And at last, some researches about Joint Application with Group Support Systems can be read.

Wood and Silver from IBM prepare the Joint Application Development at late 1970's. After common usage, some versions of the JAD are also developed. These versions are developed only according to the specifications of the projects and work

structures of the development groups. And now, we can find own JAD definition for each company based on the general rules. Another work over Joint Application Development is also implementations on different project types, not only onto the projects about computers.

In the Group Support Systems definition, there is a starting point by Nunamaker and Dennis at 1991. They defined the basic conditions and requirements of a GSS. And after that, some concepts are added, which are supported by the new technologies, which are results of the researches. Some researches are also for measuring the capabilities of the Group Support Systems under some conditions.

The Group Support Systems are new technologies for the software development process and there are also some researches about the usage of them under some development methodologies. Liou and Chen specify the union of JAD and GSS firstly at 1994. After their first hypothesis, which exposes the improvements of GSS on JAD sessions, some researches are also prepared. For example Dennis and Hayes tried to run this hypothesis on business at 1999. And some numerical results were published according the experiments.

#### **1.4 Thesis Organization**

In the second chapter, the brief definitions about the Joint Application Development can be found. This chapter also includes history of this development methodology. The team member's roles are written in subtitles, which are specific to the methodology. The advantages, disadvantages and the numerical proofs can be also found in the chapter.

In the third chapter, a comparison between the major software development methodologies can be found. The software development methodologies; Rapid Application Development, Participatory Design are described and the comparison between the Joint Application Development and other methodologies were written. In this chapter, also the opinions about the choice of the JAD, not PD or RAD, can be found.



In the fourth chapter, the theoretical part of the thesis can be found. The theoretical definition of the JAD-NW (New name to describe the approach) was written here with brief definitions of the roles (which are revisited) and task divisions. The solutions to the Joint Application Development methodology problems, which couldn't be solved and the problems, which are newly formed, are also in this chapter.

In the fifth chapter, the details about the tool, which is developed as a Group Support System specialized for Joint Application Development (UCHE Project). The development phases, environment, used technologies, capability descriptions and a user guide are in this chapter.

The thesis ends with the conclusion of the definition and defined future works on the theoretical part or on the tool's capabilities. The following references include the materials that can be used for developing this project or for preparing some new ones about the Joint Application Development or Group Support Systems.

---

## CHAPTER TWO

# JOINT APPLICATION DEVELOPMENT / DESIGN

---

In a software development process, the development operation goes in such a manner: The customer wants something from the developer. The developer prepares a system according to the requests of the customer. And finally the customer does not like the system with a reason of “I didn’t mean such a thing!”

For solving such a problem before this situation, a “Requirements Engineering” process is required. Here, “Requirements Engineering” is a process, which works for better interpretation between customer and developer on the project.

At this time, we can start to work over user-oriented analyze and development. There are some methodologies under this point like Rapid Application Development (RAD), Participatory Development (PD) and Joint Application Development (JAD). These 3 have different benefits and disadvantages under some conditions. Because we tried to use JAD in this project, we must know something about JAD. You can also find a brief comparison between the 3 methodologies in the following chapters.

We have some definitions for Joint Application Development (JAD):

“....Joint Application Development (JAD) is a management process, which helps IS work effectively with users to develop information technology solutions that really work....” (Human Resource Services, 1998)

“....Joint Application Development is a technique for developing business systems requirements....” (Matthews, 1999)

“....Joint Application Development is to quickly build a consensus regarding to business requirements and solution options, then identify and document the requirements....” (Kara, 1999)

“....Joint Application Development, or JAD, is a process originally developed for designing a computer-based system, which brings together business area people (users) and IT (Information Technology) professionals in a highly focused workshop....” (Creative Data Corporation, 1999)

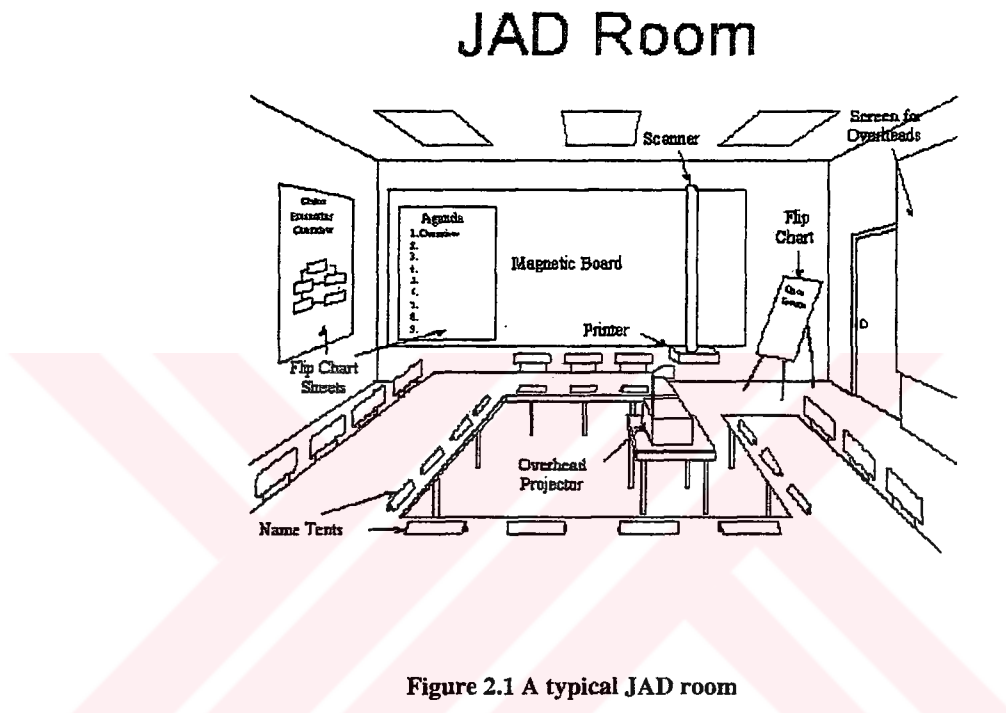
“....Joint Application Development is bringing together representatives with management authority and accountability into a structured workshop to foster timely decision-making....” (Cristel and Kang)

“....Joint Application Development is creating a common language, which helps all the participants communicate and understand each other's needs so that IS can build systems that more effectively support the company's higher level information needs....” (Jennerich)

“....Joint Application Development methodology takes end-users, executives, tech lawyers and developers; off-site, and away from distractions, for meetings where they work out the details of what the developers will create and the focus is on business objectives rather than programming details....” (Grossman, 2001)

As a result of these definitions, the definition can describe JAD, as an analysis and design methodology, which unites the customer and the system developer in meetings under some rules and schedules while the development process runs. The methodology defines roles of the customers and developers, (Or we can say also; the meetings must include people from different stages of customer side and developer side.) prepares some deliverables after meetings, (Or we can say also; the developers

can bring some information onto the paper easily in the meetings.) controls the development process of the system under conditions (Or we can say also: the customer can easily control the process meeting-by-meeting.). (Figure 2.1 A typical JAD room)



### 2.1 History

Joint Application Development/Design (or in common name; JAD) is firstly developed for IBM in late 1970's. Chuck MORRIS (from IBM Raleigh) and Tony CRAWFORD (from IBM Toronto) are prepared this new methodology as a derivative of another IBM's methodology Business Systems Planning and in 1980; they started to teach JAD in workshops in Toronto. In workshops, the companies were satisfied about the methodology and with these fine results, the JAD methodology were started to be used in several companies.

## **2.2 Members of JAD Session**

The JAD methodology is a team-based methodology as explained. The disciplined structure of JAD is built on several defined roles. According to the base methodology, the JAD team (or meetings) should contain some people with the following roles:

### **2.2.1 Executive Sponsor: (Project Sponsor, Sponsor)**

The Executive Sponsor is the manager in the project, who is the main decision-maker. His workplace is not about the project directly. For example, the E. Sponsor controls the project funding. This is not the first thing about the project, but it is also one of the important points about project.

Other works of an Executive Sponsor can be also said as controlling the project on a higher point, like giving a vision or direction for the project development. The empowering of the people is also a function of the E. Sponsor. He has the most power over the project (he is not in development side, but his power is more than the others) and he should also use his power to motivate the team members. He is the member, who is most near to the top management of the team. And he should also show the top director's feelings (he is the representative of the company in the team). Because of his representative role in the team, he can sign the deliverables from the JAD sessions. He can act here as the system owner.

According to these works, we can say that the E. Sponsor is the main bridge between project team (the project) and the directors. He should not attend all of the meetings. In the first meeting, he can specify the team leader and boundaries, purposes, direction of the project. In the last meeting, he can decide whether the result is acceptable or not. In other meetings he is not an inevitable in the team, but he can be also necessary for go - not go decisions or signing the reports.

The work of an Executive Sponsor may be summarized as:

- Specifying the group members.
- Controlling the technical support of the team.
- Checking the on-going development process.
- Determining the go-not go decisions.
- Inspecting the documentations and signing.
- Ensuring that the JAD meetings are going without disciplinary problems.

#### **2.2.2 IS – Project Team: (Technical Personnel, Development Team)**

The Analysts, database specialists, user interaction (interface) experts and some other technical (development) personnel should also attend to the meetings for several times. They are not needed for all meetings, but they can be called for design phases.

The purpose of calling these people is because of their specialties. The Analysts should attend to the earlier meetings for describing the project detailed. After the descriptions are ready, the database specialists and some other development personnel should attend to the meetings. These people can now define how the project can be realized. (The abilities of these technical people should be chosen according to the type of the project. A database project needs different information and a web site project needs different types of specialties.) The experts about the user interaction can be called in the last meetings. After the functionality of the project is accepted, these people can work on the project for better and easier usage of these functionalities.

The work of an IS – Project Development Team may be summarized as:

- Analyzing the project.
- Deciding for the technical requirements of the project.
- Preparing the theoretical and practical background of the project.

### **2.2.3 Full – Time and On – Call Participants: (Potential Users, Client Representatives)**

The customers should be also in the meeting team. The users are finally, whom we will work for. (These participants can be defined as the people, who will use the results of the project or who will be affected by the results of the project.) The main start point of doing these discussions is also finding the best solutions with taking the solution providers and the customers, who define the problems together and developing the solutions faster.

The choice criteria for these participants are one of the most important operations of the JAD team selection and meeting facilitation. (The first one may be seen as Facilitator selection.) The choice of the right persons will result with better and faster knowledge about the problem. In the meetings, these participants should try to explain their problems to the developers and answer the key questions of the IS professionals as clear as possible.

The criteria of this selection should contain the following attributes:

- *These participants should have enough information about the past system.*
- *They should include the real users of the system.*
- *They may not have any worry about their words.*

- *They have to have a speaking ability. (They should have the ability of clear and good explanations. The listeners should understand them as easy as possible)*

The list can have more objects to higher the ability of understandability. These can be about experience on the work or information on human relations.

The work of the full-time and on-call participant can be summarized as:

- Explaining all of the needs.
- Specifying solution possibilities for the known problems.
- Determining improvements for the going systems.
- (If it is possible) Explaining the solutions and improvements with running sample systems. (They may be used by another companies)

#### 2.2.4 Observers

These team members cannot be said as necessary people for meetings. However, they can be in the meetings for a controlled JAD meeting. The ability of an Observer may be having experience about meetings and Joint Application Development process.

These people may be seen necessary for the people who are amateur for a JAD meeting. The Project Leader may not have a good experience about a meeting facilitation. In such a situation, the observers can analyze the meetings and with discussing these analysis results, the meetings can be more effective.

Another work for Observers is taking notes about the meetings for improving the JAD according to the project or according to the nature of the development team. This development methodology has also standard rules but without distracting these



rules, each project team (each developer company) can prepare its own JAD rules and documentation. To prepare such specifications, the company can select some participants from outside of the project to analyze the effectiveness of the going meetings and current documentations. After these analyses, the observers can find specifications for JAD sessions according to the project types or according to the nature of the company.

The work of an observer may be summarized as:

- Checking and improving the meetings for a well-defined JAD meeting.
- Defining the documentation techniques (for a better documentation) for the specified project or team.
- Defining the changes for JAD process to improve the efficiency on the project development.

#### **2.2.5 Scribe (Secretary, Documenter, Record Keeper, Modeler)**

The Scribe is in the team to document the meetings. (The scribe is the “group memory” of the meeting.) According to the nature of a Joint Application Development session, the participants should discuss everything as easy as possible. In this situation, a member(s) should take notes about the meeting to document the discussions and events in a session.

The work of a scribe is also too important. The documents about the meetings are the only things that we can use for the project. Because the scribe is the only history of the project development meetings, the work of him is very hard.

The Scribe has too many things to remember. He should take notes about discussions, accepted ideas. The presented things are also should be taken by the scribe, because they can be very important for the results of the meeting. The

scribe's work is collecting this information in a well-described documentation. The presentation of information is also as important as the ingredient of the information. In this situation, the scribe should publish the documents, where each reader can find what he needs and how he wants. These results require a good analyzing capability.

There are too many tools to document the information especially for software design. For a better publishing, the scribe could have the ability to use these tools such as CASE tools, UML notation, word-processing tools or presentation preparing tools. These all tools can be necessary for collecting, forming or representing data. Because of this, the scribe should have the background for documenting a software design process with several notation tools.

The work of a Scribe may be summarized as:

- Collecting the discussions in the meeting without having anything unclear.
- Analyzing and documenting the presentation materials (slide shows, diagrams, pictures, etc.) that are used in the meetings.
- Documenting all of the information about the meetings, where all of the team members can find.
- (If necessary) Preparing the information in some (well – known, such as UML) notations for better understandability.

#### **2.2.6 Facilitator (Session Leader, Meeting Leader, JAD Leader)**

The meetings can be defined like markets with customers and sellers. The sellers have information about the project. The customers need this information to develop the project. The important thing is here, that the conversation between the sellers and the customers. The quality of this conversation affects the quality of the JAD meeting and changes directly the results of the project development. The Facilitator

manages the meetings for a more effective session. The Facilitator is here an interface between the developers and users.

The Project sponsor will select the Facilitator for leading the meeting teams and controlling the information traffic. Clearing the information about the project and encouraging/guiding the team for providing the solution is the main responsibility of him. Also, the facilitator can be said as an interface between the Project Sponsor and the team, because the team will not be necessary for business operations and the project sponsor will not be forced to learn about the technical requirements of the project.

The Facilitator should have an experience about the Joint Application Development process. With this experience, the Facilitator may control the team more easily. With some skills about human affinities, the Facilitator should be able to higher the group dynamics. These skills may contain information from psychology to sociology.

The work of a Facilitator may be summarized as:

- Controlling the unclear objectives in the meetings to correct them.
- Providing the required information and tools before the meetings to prevent the meetings from the unexpected delays.
- Guiding the team member to provide (better) solutions. (Not developing the solutions)
- Checking the members, whether they are able for their roles or not.
- Planning the meeting agenda and controlling the planned walkthrough.
- Controlling the scribe for (better or correct) documentation.

- Controlling the tasks of the users.
- Controlling the disciplinary in the team with establishing rules for team.
- Administrating the meetings for forming a consensus on an idea.

These 6 persons construct a team for developing a system using Joint Application Development process methodology. (In some teams, more than one person may be assigned to one role.) In the first meeting, they are 6 persons, who will work as a group for a time. The work of the Facilitator begins here to form a team from these people. After taking the members in a team soul, the performance of the team will increase.

To higher the team performance, some stages may be defined. The DREXLER/SIBBET Team Performance Model (Figure 2.2) describes the step-by-step improvement stages for increasing the team performance. The first step, Orientation step, is the meeting between the members. After this step, the members of team are already know, what the team will do. After clearance of the general aim of the team, the second step, The Trust Building, begins. The team members have some personal information from other members. And they can now have the knowledge, what a team member can do or cannot do. The more knowledge about a person can help to build the trust on him more easily. Now, the team is ready for task clarification, because they know now something more personal (abilities, etc.). In the third stage, the Goal and Roles of each person can be clarified. After this stage, each member knows, what he should do and what his team partners will do. After a time, the team members will know what they should do and they can now have consensuses for the implementation methodologies. Constituting consensuses on ideas is easier now between people, who know each other, rather than people, who don't know anything about other team members. After building consensuses, the team should decide on implementation teams. The implementation stage details can be developed according to the personal information (abilities, knowledge, etc.)

easily. If the previous stages are successfully done, the team can divide the work, without overloads and mismatches and this causes high performance from the team. At the last stage, renewal, the team should decide to continue with the group or not. Here, the team may change some persons or can go with the current group.

If the formed group continues to work, the first steps may be passed easily and the performance may be started. However, after some team works, the team can decide to change the team content to refresh the team's group dynamics or the company can force them to change the team to work with other people in the company. (For a bigger group dynamic idea)

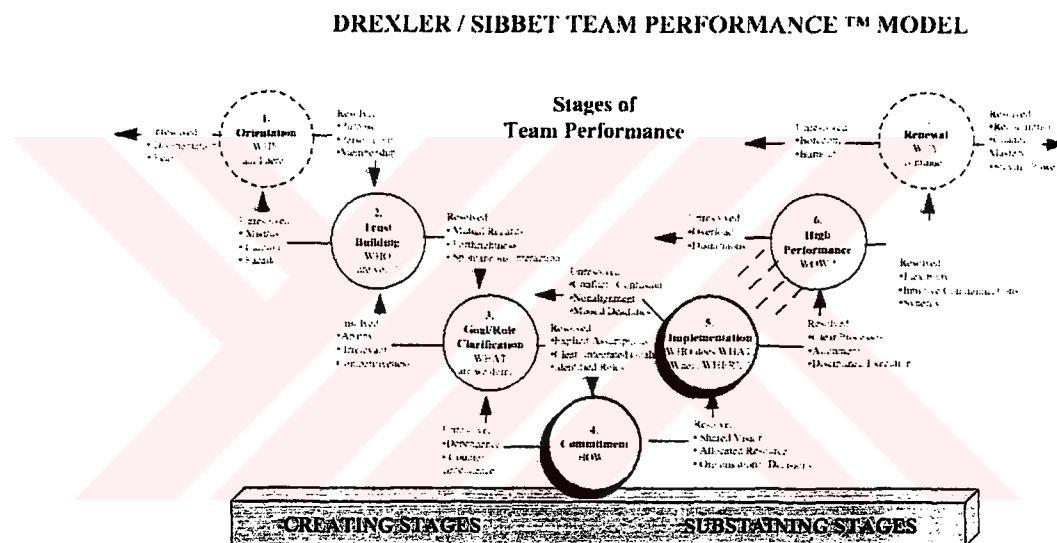


Figure 2.2 Drexler/Sibbet Team Performance Model (Drexler/Sibbet, 1990)

### 2.3 Implementation Details

Joint Application Development is based on a much-disciplined structure of continuous sessions. The JAD team for the project is formed and their mission is working together to find the problems (the projects are always some solutions for some problems or some solutions for better usage). After describing the problems, the team with different experience and abilities can solve them in an easy way.

The first important step is planning the main agenda to follow. The system is based on the discipline and there must be an agenda to follow. This agenda should be prepared by the Executive Sponsor and (if available) Facilitator. The agenda should contain the main objectives with some deadlines. This agenda may change through the meetings. However, the ideal thing is keeping the agenda static and preparing the project according to these deadlines. Here, defining the agenda is really important, because assigning too long time for a task may disturb the teamwork or assigning too short time causes unnecessary stress on the team.

After preparing the agenda, the executive sponsor may select the team members. The selection mechanism may be subjective, but the Sponsor can do the work as democratic as it can be possible. He can use questionnaires or helps from the management sides by selecting the participants. Through the JAD process, the participants may be also changed, but the main idea is here defining the main requirements about the team and selecting the members according some concrete things.

After planning the team, the remaining process is up to the Facilitator and the team members. According to the time lines in the agenda, the team will work. They will find the problems. And after that, the team should provide the solution. The solution's implementation will be developed with requirements and results, in the next phase. The next stage is the implementation stage and while implementation, the team can also re-define the specified information (problem or solution). However, the ideal is not to change the documents after implementation. In the implementation stage, the meetings also continue to take potential errors into account before newly occurring errors. After implementation, the product will be also tested and documented by the team. If the deliverables (documentation and the product) are ready, then the team's task is over.

The formal needs for a JAD process is shortly as explained. There are also defined suggestions for a better JAD process. They can be said as needs also, but not formal. A quite place should be selected. Especially, the workplace should be away from the

company place. A conference room with high facilities (facilities for several presentation capabilities) will be good for the team's performance. Some breakout rooms may be useful for special works in the meeting. Disciplined brainstorming sessions should be supported by the Facilitator. The agenda and the goals in it should be defined clearly. The meetings should be prevented by the dominations of one participant. The technical words should be kept minimal in the discussions and documentations. (If it is necessary, a dictionary should be in the documentation)

The important thing, what is not to be forgotten is the deliverables of the process. The main aim of using Joint Application Development methodology is not to developing a project with these rules. The main target is well defined and quickly prepared requirements documentation. As a result of well-defined requirements documentation, the product will be also bug-proof and highly acceptable by the customers. The documentation should include the scope, purpose, objectives, the working agenda, the discussions about the problem definitions (at least main points), the discussions about the solution definitions, constraints, resource requirements, assumptions, the unclear points, meeting reports and participant information. This produced document will be helpful for correction operations on the product (if necessary), a guide for users, a guide for the future developers (for better analyzes, when this system becomes old.), a treaty between the developer and customer for the delivered system.

We can say the application of the Joint Application Development is successful, when the meetings of the agenda are well attended. If the main point of the system is the group dynamic, the attendance percent will be one of the first criteria. Another important thing is the selection of the participants. The criterion is checking whether all of the effecting users (at least user representatives) approve the decisions (attend to the sessions) or not. One of the criteria is checking whether the base problem is solved or not. (This is also related to the finding and understanding the real problem.) As a last thing, we can check the deadlines on the agenda for completing the project. In this situation we can decide on the success on the application of the JAD.



The success of the methodology is another problem. One of the factors is whether the team members know, what they developed or not. A project charter should be prepared and accepted by all of the members, when we want to talk about success of the methodology. All of the user representatives, who will be effected by the results of the project, should be informed about the project and development phases. The team approach is important at this point. Every team member should be listened by all of the team and have equal responsibility in the team decision. The recording operations specify also the success of the project in this manner. The team members should be in the meeting place, but not for being there. The aim has to be there, to talk something about the project. The breaks between meetings should not be too long or too short. This means loosing the group manner or coming to the meeting with incomplete tasks. And also, when the decisions are prepared after a consensus, we can say that the Joint Application Development methodology was the best choice for the prepared project.

## **2.4 Advantages**

There are some advantages of using Joint Application Development methodology by developing a project. The advantages of the JAD methodology come from the trendy development ideas such as Brainstorming, User-Centered Design, etc. The numerical proofs of the advantages of the Joint Application Development can be also read on (Chapter 2.6). The main advantages can be listed as follows:

- Improved Quality: The quality of the product increases, because the developers and customers work on the project together with helps of experts on the implementation requirements and the customers can define anytime, the not-good things on the project, before the product is ready.
- Reduced Cost: The customers are always in the development side and this reduces the costs of re-producing some stages, after constructing. The customers can see most of the problems, before the prototyping. And as a result, the costs of time or money reduce.



- Reduction on Life-cycle time: In a project development cycle, after each stage the developers should return to the prior stage to correct something. These corrections will be reduced, because the customers can check each stage's results on time. And this effect the life cycle time naturally.
- Ownership Opinion: The customer has an ownership opinion after such a development cycle. And this opinion prevents the customer, to see some unimportant problems on the product.
- High Concentration: The JAD methodology has a structure to complete. And the developers are not only responsible to the customers, also responsible to their team and their friends. This worry on uncompleted agenda and responsibility idea improves the concentration of the team members.
- Brainstorming: Brainstorming, which is one of the silver bullets for the meeting, can be done in this methodology. This is one of the best ways for the meetings to define solutions for problems and may be used in the sessions to build new ideas.

## 2.5 Disadvantages / Pitfalls

Listing the disadvantages and pitfalls of using Joint Application Development in a chapter is more correct, because the disadvantages occur when the pitfalls occur. In such situation, we cannot divide the disadvantages and pitfalls. Also, advantages and disadvantages can be discussed in the same problematic. The advantages of the system can be also the disadvantages of the project development, when the methodology is not applied according to the rules. Because of this, the following list does not include such things. (For example: We can say: "If the Brainstorming sessions does not supported as it requires, the idea production will not be good as it said". However, it won't be necessary to say.) We can list the disadvantages or pitfalls as follows:

- Cost, because of the meeting times: The meetings are required to be attended. And this means, some workers from the companies should be in another place, rather than in their offices. The reduced total project time, and time, which is spend for meetings should be compared. The workers should prepare their selves for meetings, should go from their offices to the meeting places (in ideal, the place have to be far from the work place.), and should go from the meeting place to their offices back. This calculation should be also taken into account. Another cost is training scribe and facilitator for a better Joint Application Development. (This can be said a one-time cost for the development team, if the facilitator is not from outside of the company for each time.)
- Sponsor Problems: When the sponsor is not authorized as described, there can be problems in the team. An Executive Sponsor, who is not authorized as needed, can decrease the performance of the team. (For example, the team may want to decide on some resources, but the Executive Sponsor can say, that he should ask to the manager. In such situation, the team dynamic can be disturbed.)
- Too many or too few members: The team members should be selected by the Facilitator and Executive Sponsor after a short discussion about the project. After a false discussion, the members can be too many and there can be problems in the sessions, because of the crowd. And if there are too few members, the work – division may have problems.
- Domination problems: When a member dominates the meetings, this prevents the creation of the new information. The other members are let to listen to him and they cannot discuss the ideas or not talk about their own ideas.

- Facilitator and Scribe role: The most dangerous pitfall is the facilitator and scribe. The facilitator is the main controller of the success of the meetings. A good facilitator can force the team to work. But a facilitator, which is not good enough, cannot motivate the team to work. And a bad scribe also may be a pitfall, because he is the only one, who will record all of information.

## 2.6 Performance Statistics

According to the reports from several projects around the world, we can clearly write the following statistics. These numbers are collected from the project documents of the developers, who used Joint Application Development methodology in the project development:

- Repair effort per defect is only 10% in JAD phase as compared with system test phase.
- Timesaving is to 30-40% in design and 20-30% in implementation.
- 15% Cycle Reduction.
- 8 hours / per Function Point for traditional method vs. 2.5 hours/Function Point for Joint Application Development.
- 4 to 6 weeks time saved in a project at Western-Southern Life.
- Cost avoidance of \$500.000 in a project at Texas Instruments.
- Joint Application Development removes 50% of the defects of the requirements phase and 25% in design phase (percentages are not cumulative.).
- 25% increase in number of Function Points in a project at CNA Insurance.

- 99% of users would do it again at Texas Instruments.
- 94% of users said they had a better understanding of the system at American Airlines.
- 100% of the users said the system would be at least “good” at American Airlines.



---

## CHAPTER THREE

# COMPARISON OF JAD WITH RAD, PD

---

According to the development in the computer industry, some methodologies were developed to ensure on the completeness and quality of the projects. The methodologies were prepared on the technologies and the needs at their days. Rapid Application Development, Joint Application Development and Participatory Design were developed in this order to satisfy the user needs. (Figure 3.1 Development Methodologies according trends) Each one has its own properties and superiorities in some conditions.

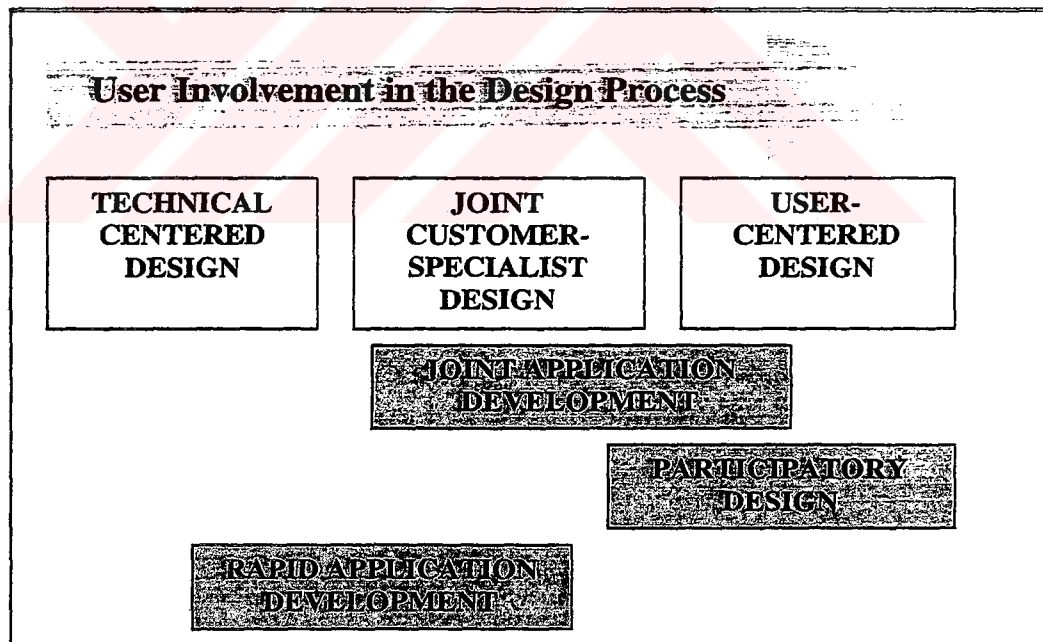


Figure 3.1 Development Methodologies according trends

### **3.1 Rapid Application Development (RAD)**

Rapid Application Development is a methodology, which has more technical - based manner. With the help of CASE tools, prototyping tools and code generators, the project development team tries to develop the system. The approach is also targeted to low cost, high speed and high quality.

The method to reach these goals is using some tools. In the way, the developers collect some information about the system and the scribe documents all of this information after an analysis. The produced documents will be always published to look at from all participants.

The team includes here the scribe and the developers. The developer team should have enough Information Systems experience to collect right information from right people. After that, the developer team tries to represent the information with some tools. These tools include CASE tools, UML, etc. In all stages, the objectives, description, prerequisites, deliverables, guidelines, techniques used, participants and roles and subtasks should be documented.

A rapid application development process may be defined stage – by – stage: Requirements Planning, User Design, Rapid Construction and Transition.

Requirements planning stage is the general understanding phase. The team tries to understand and define the problem. To realize this, the developers specify the problem, checks the similar (or old) systems. In the next sub-step, after understanding the situation, the developers will define the functional requirements of the new system.

In the User Design Phase, the activities of the system users should be analyzed. This means the usage characteristics of the new system. According to this new information, the system structure, screen layouts and functions can be defined. The construction approach can be also specified, because now the base requirements (like

performance, deadline, relevance, etc. requirements) can be handled in the scribe's documents.

The detailed design of the system can be now prepared. This is called the Rapid Construction. The software will be created and tested. The performance tests should be included into this stage. The project documentations can be prepared also in this stage. After verifications of the system, the development can go to the next stage.

In the Transition phase, the system installation has place. After the installation, the project will be tested in the real environment. After having the information about enhancements, the project team can think that the project is finished.

### **3.2 Participatory Design (PD)**

Another design methodology, Participatory Design is developed at late 1970's by Ehn and Kyng. The main approach for PD is thinking socio-technical and more focus on user involvement. The concept is based on user – centered design, with some prudence; like the qualification of the users.

The Participatory design prepared by a confidence on the enough qualification of the users on the computer systems. The main idea of the PD approach: "The users have got spend enough time on using computer. They also want to make their life easier. And the users know, what the computer can make, to solve problem. Only think, they don't know, how the computer can solve the problems."

In this Scandinavian design methodology, (This is another formal name of PD, because the methodology is based on the Scandinavian worry, where quality and usability is the most important feature and the creators are from Scandinavia.) the process begins with selecting the users from the possible user group. After that, the development team and the user representatives will start to discuss on the system. The discussions can include brainstorming sessions or discussions over existing systems. The tools may be slideshows or video.

In the Participatory Design methodology, the operation can be divided into three stages also: Visualizing the current situation, Visualizing the possible workplace. Prototyping and Completing.

In the first phase, the users and the developers will work on the abilities, knowledge, and information of possible users of the system. The current system may be infiltrated at this operation. After that, the developers and users can see more clearly the current system and the needed corrections. We need here some users, who have the analysis capability and information.

Secondly, the improvements can be defined. The users and the developers will have now the same vision, because they analyzed the current system together. At this step, they will do critique, fantasy and implementation. This step goes mostly as brainstorming sessions. The team will critique the systems problems. After the mistakes and improvements are pointed out, the fantasy session can begin. The possible (realistic or not realistic) solutions should be written now. At last, the solutions may be checked by discussions.

Thirdly, the specified operations will be implemented as prototypes. On prototypes, the team can re-discuss the solutions and think over the improvements or corrections on the developing system. After a totally accepted prototype, the system may be done.

### **3.3 Comparison between RAD, JAD and PD**

Firstly, the similarities will be defined. All of these 3 development methodologies include users in the development team. These methodologies believe on the user's qualification. However, the level of the confidence may be changed method to method. The three methodologies are using the presentation technologies as many as possible.



When we want to compare the Participatory Design and Joint Application Development, the role of the users may be varied. The users have to be seen as information sources (as participants) in Joint Application Development. However, the Participatory Design will see the users in a more critique role. The users are in development team and they are also as important as other developers. The Joint Application Development takes the savings (economical, time, etc.) into account. But, the Participatory Design thinks over democracy, mutual learning, and conflict resolution. The Joint Application Development is a structured software engineering process. However, the participatory design is a group learning social process. The JAD methodology names the target as a "System". The PD method names the product as a "Workplace". The JAD comes from Canada, USA (the concept may be seen all for business). The PD comes from Scandinavia (the concept may be seen all for quality). The meetings in the JAD are delimited by time (highly - structured meetings are the main point of methodology). The meetings of PD are delimited by satisfaction of the users. The Joint Application Development is an example of "Structure". Yet, the Participatory Design is based on "Creativity".

If we compare Joint Application Development and Rapid Application Development, we can firstly see the target of the two methodologies. The Rapid Application Development focuses on the result, where the Joint Application Development gives more attention to the application of the methodology and software engineering. The RAD's system is based on the speed of the implementation of the project. The main thing is finishing the project as soon as possible. However, Joint Application development tries to prepare documentations about the project, while preparing the project. This means, maintenance and re-build of the project, which is planned with Rapid Application Development is harder than a project, which is prepared with Rapid Application Development, because the documentation is the only help in such operations.

When we think over the choice reasons of the Joint Application Development, the selection objectives can be listed as follows:

- The users are not so ready to attend in a project development process as developer, unless the project is highly specific project, on a technical theme.
- The project development is not a school to learn something about the project development. The first aim should be finishing the project as structured.
- The project should not be thought as a stand-alone product. The system is integrity of product, defined requirements and improvements on it. Here, the product should be well documented to easier the process.



---

## CHAPTER FOUR

### JAD – NW

---

The Joint Application Development is a session – based structure, as explained in the previous sections. The constructed team should be met in a session room to discuss the project from different views. A JAD team here has some members, who have also some other works than the on-going project. Especially, the participants have work in the company of customer.

There are some meeting and project management systems to realize the meetings on the Internet or network. The Group Support Systems can help to reduce the consuming time in an on-going project development.

An integrated Group Support System can produce some solutions to the problems on the traditional JAD. The whole process can be divided into new mission definitions. The roles should be re-defined according to the features of the meeting systems. The GS System can be defined as a new role in the JAD team. The features of the system can take some of the responsibilities of the team members. According to this approach, the task definitions of the whole search process can be found in the next section. Secondly, the team members (or roles) with their responsibilities will be re-defined. At the third section, the solutions of the JAD-NW to the previously defined problems can be found. At last section of system definition, the problematic in the system, which can be new problems or the not solvable problems, will be written.

## **4.1 Basic First Agenda**

The first rule of application Joint Application Development is preparing an agenda and following the tasks over it. Because of its critical situation, the agenda should be defined realistic. A basic agenda includes the tasks to complete and the deadlines for each task.

In this approach, the defined tasks can be checked by the JAD-NW software. This means, that the software has a role of project management tool. The defined tasks can be checked according to the deadlines and the work of the facilitator can be reduced, because controlling the tasks to complete is a work also.

The tasks below are the basic application steps of the JAD-NW. After the first steps, the facilitators can also use this phases as starting agenda. After assigning deadlines, they can be used as a starting point. Some additional phases can be found and added to the list.

Here, the tasks can be started to look at briefly.

### **4.1.1 Interview**

This task cannot be done by electronic systems. The project Sponsor (Executive Sponsor) specifies the Facilitator and the Scribe for the project. (The Facilitator can also select his scribe, because he can want to work with a scribe, with whom he can work easily) After that the Facilitator can define an interview to specify the basic outline of the project.

To prepare the basic outline, the interview should be applied to the possible users face to face. The answers can be useful by defining the approximate deadlines for the tasks in the project management. The work is only for facilitator.

### **4.1.2 Defining User Profiles**

The prepared outline gives a chance to the facilitator and the scribe to write the basic requirements for a successful Joint Application Development process on the project. These requirements are the members of the sessions.

The team members can be defined in two types: The development members and the information source members. The development members are defined according to the specified needs. For example, in the first requirements, the facilitator can decide to the number of IS Professionals needed (according to the wideness of the project) or the number of Database experts (according to the need to the databases in the project). The source members are the possible users. These can be changed according to the project type. When the project is public, the facilitator should define the user profiles for the coming system. When the project is a private system, the facilitator should also define the user profiles from the company, who want the system from the development team.

After the number and required abilities specified, the Executive Sponsor, who is from the employer company and the Facilitator can select participants.

### **4.1.3 Distribution and Education Phase**

The team members are now selected according to the needs. In the first stage after that, the facilitator will distribute the group support system tool and gives also some education/practice documentation.

At this phase, the most important thing is that, all members can use software and computers easy. (The experience on the computer usage can be thought as a selection criterion.)

This phase is also not electronic. The hard job is only for administrator. He should distribute the software and be sure on the knowledge of the users to a more easygoing project development process.

#### **4.1.4 First Meeting – Discussing Agenda**

An existing agenda is ready from the Facilitator. However, in the first meeting, the members of the team can better discuss the situation and re-design the agenda with more realistic data. The development experts can re-analyze the system according to their experiences. And the users can think about their abilities and their wishes from the system. This causes also a better result on time tabling the job.

Another work in this phase is meeting with each other. As more information the users have about each other, as better team dynamics can be build. As explained in the section 2.3, the group opinion is the most important thing for expecting better results from the team.

The Facilitator will control the sessions. And the information collection is the software's work.

#### **4.1.5 System Discussions**

This step can be defined as the first step, where the requirements definition starts. The main discussions about the system will be done in these meetings. The developers try to get information about the structure at this step.

The members can use presentations or draw charts to represent their ideas. The Facilitator and the scribe have too much work in this phase without GSS software. However, the software can collect now the discussions or presentation materials. And it can do this without disturbing the presenter.

#### **4.1.6 Technology Discussions**

After specifying the need of the company and the requirements about the project, the specifications may be listed. The problem is now, specifying the right technology to use. At this stage, the developers and their experiences have more important role.

However, the user's experiences about the old systems or about the similar systems can be used.

By technology discussions, the facilitator should control sessions. However, the most important thing for these sessions is checking the discussion history for many times. An information collecting software can make easier this operation.

#### **4.1.7 Discussing Data Model and Interface**

A long break may be required before this step. The requirements and the usable technologies are ready after discussions. Now the experts prepare some preliminaries about the usage of technologies in the system.

These preliminaries may include all information from database structures, modeling systems to data structures, which will be used in the system. The functions or the functionality requirements should also be validated in these sessions. This session may be said as the last session before the implementation.

Before finishing the system, the interface discussions may be done in this stage. While the implementations run, there will be also sessions to check the going of the project programming. In these sessions, the team cannot only check the prototypes for the validation of the functionalities and also discuss on the interface or non-functional requirements. These discussions may be more realistic, because they are on some real products. (At least prototypes)

#### **4.1.8 Last Overview**

After preparing the product, the team may check the whole product. This operation includes also the testing operation. The result of the "last overview" stage is the wholly validation of the product. In this overview sessions, some new participants may attend to the team. With new users, the validation phase can get some new critiques about the project to be better.

#### **4.1.9 Reporting**

There are prepared reports and presentations about the project during the project development process. These discussion reports, logs, presentations, drawings, charts will be prepared as a requirements document and development document.

This report preparing work is the scribe's job. He should collect all of the information and materials. However, logging may be hard. Because of this, the scribe only takes notes about the important things. In such a system, the software will log all of the discussions and the scribe can prepare the report form these logs.

Also in the reporting session, the team members can attend to the session and check the important things from the discussion logs to write into the reports.

#### **4.2 Team Members**

The team members should be reviewed according to the new number of member types. The members can be defined as follows:

- Executive Sponsor
- IS Project Team
- Full – time and on – call participants
- Observers
- Meeting Administrator
- Software



The Executive Sponsor, IS Project Team and Full – time/on – call participants are not changed from the original roles in the Joint Application Development. (The brief descriptions can be found in section 2.2.1, section 2.2.2, and section 2.2.3)

The observer's selection criterion can be changed. The observers have the same job as explained in section 2.2.4, but they should change the software. To establish this job, there are three possibilities. One is communicating with the publishers of the software according to the new documentation specifications. The software's documentation function can be changed according to the desires. Another possibility is having the source code of the system and coding the new documentation function. The last possibility of the correction is working with a scribe and re-defining the created documents after each reporting stage.

#### **4.2.1 Meeting Administrator**

The first new member role is Meeting Administrator, which is a revisited version of the Facilitator from the original Joint Application Development (explained in Section 2.2.6)

The responsibilities are similar to the original one. But, some of his works can be done automated by the software.

The meeting administrator's first job is controlling the discussions. The discussions are the most important parts of a Joint Application Development and providing the discipline in the discussion will specify the success of the project. The discussions should be the place where new ideas can be developed. And the meeting administrator should guide the team by improving their ideas to some solutions. By doing this, the administrator should always think over his experiences and the logs of the old discussions.

Another job is specifying the task list and controlling the reality of the deadlines. According to the feedbacks from the other team members, the task list and properties can be revisited.

The most important work of a meeting administrator is providing the order in the project development (especially in discussions), as explained before. To help the meeting administrator in this job, the system is defined as a centralized system. The system's administrator can specify what he wants. This means, giving too many rights. However, the system is imaginary. And controlling the people on a virtual system may be too hard. The only way is giving some rights to the meeting administrator to control the meeting.

In this centralized system, the administrator is the only member, who can give permissions for action or permit the actions. The actions, the messages, the files, etc. will go to the meeting administrator. And when the administrators want to send received information to anyone else, he can send to the selected members. In such a system, controlling and logging the operations is also easier. And these logs will be used to report the sessions.

#### **4.2.2 The Software**

The system will be completed with the meeting and management tool. This tool is a unifying and reporting program according to the methodology. The tool will have some roles in it like controller, observer and reporter. (These jobs were written according to the project management tools in the market.)

The software can be thought as a controller, because the specified tasks, meetings and warnings for specified members will be checked by the tool. The administrator should only to assign the works to the members. The tool can warn the member and force him to do his job. When the tasks were not completed, the tool can also warn the meeting administrator or log the situation to report.

The software can be thought as an observer, because one of the observer's jobs is checking the JAD application on the project development and the first objection on the application is running over the steps of a successful Joint Application Development. A system, which is specialized on Joint Application Development, can

prepare the standard agenda on the task list. And force the meeting administrator to follow the instructions.

The software can be called as the reporter of the team. When we analyze the work of the scribe; we can see that he should log the discussions, collect the presentations / drawings and report them after analyzing. After the improvements on summarizing systems, an intelligent summarizing system may be integrated to the tool to analyze the discussions and report the important things. However, with a simple system, the tool may report the presentations, brainstorming sessions and member logs. In the simple system, a scribe should be required only to analyze the discussions.

To realize this tool, the templates should be required. The tool developer will prepare some templates to specialize the tool for some methodologies. The task list will be prepared by the tool automatically, and the administrator will not be allowed to go another task, without completing the task. And in a system, where the meeting administrator is has too many rights, the acceptance means that a phase is totally accepted. Another template is required to prepare a report. A well-prepared report template can be enough to write down the reports. By analyzing the discussions, the members can help to the software by checking the discussions and choosing the required discussion parts. Or in the best case, the tool may contain an intelligent report-summarizing tool to summarize the discussions and will prepare the document with these summaries of the discussions.

#### **4.3 Solutions to Existing JAD Problems**

Using Group Support Systems in Joint Application Development process (or with the name in the thesis, JAD-NW) brings some solutions to the problems in the classical JAD approach.

These solutions will be effective in the ideal situation, like the approach, that we couldn't think a benefit as disadvantage, when the rules didn't followed. The solved problems (or the disasters of the problems decreased) may be listed as follows:

- Costs
- Domination
- Overloaded Facilitator
- Attention
- Misunderstanding and not-heard
- Standardization

#### **4.3.1 Costs**

The main problem of the Joint Application Development is the costs of the sessions (meetings), which should be done often. And the costs of a JAD meeting are described in the previous sections. An electronic meeting system, which can take the team members in a virtual hall, can easily decrease this cost. The transportation times should not to be calculated now and the time, which is spending for something in the meeting, which is not related to the project can be easily controlled.

When a member has another job, he can do also this job, while the meeting goes. The only thing is that the member should follow the meeting. When this object is ready, then the member can do what he wants, without disturbing other members.

#### **4.3.2 Domination**

One or more members can dominate with their talking styles or with their authority in the company. When a member has a manner, like he knows everything, he will not allow other members to talk about. And therefore, some good ideas will be dead, before they born.

But, in a virtual system the users can be defined anonymous to say their opinions without a fear.

The facilitator should follow everything in the project development. Some automated processes may be helpful for him. For example, controlling the tasks may be thought as automat able process. A task controller program may inform all users about their tasks, may warn the members about their deadlines warn the meeting administrator about the incomplete tasks and prepare some information to evaluate the members about their works. Some operations like these can be automated by the help of a tool and the overloading problem will decrease.

This problem is highly related to the “costs” problem in JAD. Attending to a meeting will be very costly for people and when the people have another jobs to do, they cannot worry about to attend meetings. However, attending to a virtual meeting will not disturb their other works and these people also can do their work and attend to the meeting at the same time. When they don't have too many works about the project at this meeting, he can only follow the meeting and may be informed about the development process.

In the meetings, the members cannot follow all of the spoken information. And in the JAD methodology, the team includes a scribe to solve this problem.

However, the members cannot take information from the scribe real-time. Another problem is the ability of the scribe. The scribe cannot follow and note the discussions.

When the meeting system can include a recorder to record all of the discussion, the problem seems to be solved. But, recording all of the conversation is also not effective, because some conversations may not be about the project or when more than one people talks, the data may be disturbed. In a virtual meeting system, the meeting system and administrator will only give permission to one person to say something at one time. And every member can control the discussion history, when he wants.

#### **4.3.6 Standardization**

The electronic meeting system should use templates to prepare documentation. And using some document templates causes to standardize the operations. When the standard templates of the program are changed, then the standard document of the system also changes.

#### **4.4 Pitfalls**

The attention to the project may be decreased. When the team members will not see someone, who is speaking (Speaking and writing is not the same thing.) to explain something, they cannot concentrate on the project. This may cause to doing something, which is not related to the project. To avoid such problems, the leader should believe on members not to think something more important then the project.

Another pitfall may occur in the anonymity rights. When the person takes the anonymity right, he can use this right for not-good things, because everyone can write everything without be worried to be catch. One of the solutions is controlling all members by the meeting administrator, but not listing the name when the anonymity is on. Another solution is trusting to the members.

Writing is not so easy to explain something like speaking. Some people cannot explain with writing, what they think. Additionally, we cannot see also the mimics or stresses of the speaker, while explaining the idea. The mimic or stress problems may be solved with font variations or with little pictures. However, they will not be same.

All of the members should have the software and should be informed about using the meeting software. When the team members are changing often or when some persons want to attend to some meeting as visitor, this may be a problem.



---

## CHAPTER FIVE

# UCHE PROJECT

---

Utilized Central Hall Environment Project (or UCHE project) is designed as a tool to realize the needs and features of the Joint Application Development – Network Approach (JAD – NW).

The programming environment can be found in this chapter in the next section. After that, the capabilities and their programming systems were explained. Additionally, there are also two “User Guide’s” for administrator and client usage to make easier to use the tool.

### 5.1 Programming Environment

System is developed under Borland Delphi 6.0 programming environment. The system that the software is implemented on is an Intel Pentium III 800 MHz, 128MB RAM computer that runs Microsoft Windows XP Professional as the operating system.

For tests, the system on which the implementation is done and a second system in the same domain are used. The second system is an Intel Pentium III 600 MHz, 256 MB RAM computer that runs Microsoft Windows XP Professional as the operating system.

### 5.2 General Design

The UCHE Project has some capabilities to realize the JAD-NW. These capabilities are designed to work on direct connections over sockets. And a brief



work over the functional requirements was done. In the following paragraphs, the requirements on the first prototype can be found.

The system is constructed on several forms, which are communicated over the main form. All of the operations go to the main form and connections form. Other forms are only for show or collect data. After collecting data, they send the data to the main form and the main form process the data. Or the main form fills the information show forms and the user can only work on the filled data in the form.

All of the communication is done over sockets. One member runs his tool in administrator mode and other users connect to the server tool with sockets. Some socket ports are dedicated for some operations. These socket ports will be explained in the Capabilities sections. Via socket programming, the information exchange may be done. The general messaging port is specified as 777. The tools of the members are communicating at port 777 with each other for system messages.

A directory structure is defined for each project. This project directory contains all of the information related to the project in a structured way to bring an order to the project. The directory structure can be defined as follows: (Variables are defined between '<' and '>')

- <Project\_name>: *Directory*. To define the project
  - <Project\_name>.u\_g: *File*. To store general information for project
  - CompletedSteps.u\_s: *File*. To store information for completed steps
  - Team.u\_i: *File*. To store information about team members
  - CurrentTasks.u\_t: *File*. To store the current tasks of the user
  - Drawings: *Directory*. To store the drawings related to the project
  - Slides: *Directory*. To store the slides related to the project
  - Reports: *Directory*. To store the documentation related to the project
  - Questionnaires: *Directory*. To store the questionnaires in the project
  - Discussions: *Directory*. To store the logs of discussions in the meetings.

- Others: Directory. To store the files not related one of the specified directories

### 5.3 Capabilities

The capabilities of UCHE Project are designed for realizing the JAD-NW under the defined rules and test the usability of the new approach. For an easier implementation, the needs from the meeting system are divided into parts:

- Step by Step Guideline
- Discussion
- Member Information
- File Transfer
- Picture Show
- Messaging
- Slide Show
- Alert
- Questionnaire
- Analysis of Questionnaire

### 5.3.1 Step By Step Guideline

*Only for Administrator.* This guideline is to follow the development methodology without passing an operation. The meeting administrator can check the process situation by checking the process guideline.

Another function of the guideline is taking the process in an order. When an objective is completed, the meeting administrator can define the related files to the completed step to arrive the information about one step easier. And after the acceptance, the date and the related files are recorded to prepare documentations.

### 5.3.2 Discussion

*For Administrator and Client.* The discussion platform is defined to allow the members to communicate with each other.

Before the discussion, the administrator should allow the users to discuss. This property is to open new sessions and log files for new discussions.

The communication port for discussion is 778. The tools are using this to write and read discussion messages. Firstly, all of the discussion messages are going to the administrator to log. And after recording it, the administrator broadcasts the message to all members.

### 5.3.3 Member Information

*For Administrator and Client (limited).* Some information about the team members can be required. And this capability is to record some information about the members as follows:

- Name
- IP

- Responsibility in the team
- Sex
- Age
- Nickname

The system allows only the administrator to see the information of team members. The Meeting Administrator can also request the member information from the members via a message. However, the client can only see the member information form, when some information from him is requested.

#### 5.3.4 File Transfer

*Only for Administrator.* The file transfer capability is designed for two reasons. Firstly for sending files according to the inner operations like picture show or messaging system, the file operations are needed. Secondly, the capability can be used for sending some files according to the requests.

When a file is required for an operation (slide show, picture show, etc.), it will be put to the required directory of the user's computer by the administrator. When a file is requested from a user, the user can send a message to the administrator, which explains his requirements and the administrator can answer him. This feature is prepared according to the centralized structure of the tool.

The File Transfer operation goes through the port 779. After a handshaking over messaging port 777, the sending file operation goes on the port 779.

### 5.3.5 Picture Show

*For Administrator and Client (limited).* The Picture Show capability is designed for drawing shows in the meetings. The drawings may include pictures, interface sketches, tables or UML draws.

The user, who wants to show the drawing, can take attention on some points, with the help of a pointer. Other users will also see the pointer on their machines also.

The administrator can determine a user to control the pointer or change while the show runs. And also the administrator can change the picture, which the users see on the frame.

The client can only see the picture on his frame, with the pointer. When the administrator gives permission, the user can also use the pointer, to point something on the picture.

For messaging in the picture show system, the tool uses port 780.

### 5.3.6 Messaging

*For Administrator and Client (limited).* The messaging system can be called as a small version of mailing system. In this system, the administrator is the mail server and other users are clients. According to the Centralized manner, the users can only send messages to administrator and administrator can send messages one or more users.

The Messaging system includes a Message Box, which is same for all team members and a panel to create new messages, which has different capabilities for Meeting Administrator and Clients.

The Message Box shows all of the messages, the main part of the selected message and buttons to see attachments. A message can contain only two attachment types, which are a questionnaire or member information request.

The Message Creation for Meeting Administrator contains a receiver list, message definition and attachment parts. However, other participants can only define the message body and caption.

The messaging system runs over file transfer operation. As a result, there is no port dedication required.

### **5.3.7 Slide Show**

*For Administrator and Client (limited).* The Slide Show is prepared for showing slides to other participants of the meeting. The Meeting Administrator saves the slide files to the machines of the participants before the show begins. And the member, who wants to show slides, then changes the slides by selecting the no of the slide.

The Administrator can here specify the director of the slide show and the name of the slide show. The director of the slide show has a list of the slides to change the current slide. Other participants have no functions to change something.

The slide show messages are going over the port 781. The messages like slide changing are going over this port.

### **5.3.8 Alert**

*Only for Administrator.* The administrator can determine an alert to warn everyone in the system about the next meeting. After the administrator specifies a date and time for the next meeting, the system begins to send messages about the next meeting periodically. The alert runs until a new date is assigned or until the meeting time passes.

### 5.3.9 Questionnaire

*For Administrator and Client (limited).* The aim of creating a questionnaire can be defined as taking the opinions about a theme or making a prioritization in some works. Therefore, a questionnaire platform is developed in the system. The Questionnaire system has two sides. One side is creating the questionnaire and the other side is answering the questionnaire.

The administrator can create a questionnaire and save into a file to let the participants answer. After creating a questionnaire, the administrator can store it to use later. And to apply a questionnaire to the participants, the administrator can prepare a message to all of the members with the attachment of the specified questionnaire.

Other team members can only see the questionnaire, when the administrator sends a questionnaire request to them. The message will contain a button to see the questionnaire and (because the administrator puts the file into the user's machine, before send the message) after pressing this button, the user can send his comments about the questions.

### 5.3.10 Analysis of Questionnaire

*Only for Administrator.* At any time, the administrator can want to see the results of a questionnaire. To show the results, answers of the users are saved into files, which are named with the same name of the Questionnaire.

After selecting the questionnaire from the list, the percentages for each answer can be seen as numbers and pie charts.

## 5.4 User Guides

The tool is prepared for two types of the users: Administrator and Clients. According to the type of the using person, the tool hides its features or shows. As a result of this, two user guides were prepared.

First user guide is prepared for Administrators. This guide contains the features that an administrator will use and how the features can be used. It will be given to the Administrator of the project with the tool software.

Second user guide was written for a standard client user. This guide is a guide for a limited usage with limited information not to disturb the client with unusable properties. It will be given to all of the meeting participants to use the software easily.





### 5.4.1 Guide for Meeting Administrator

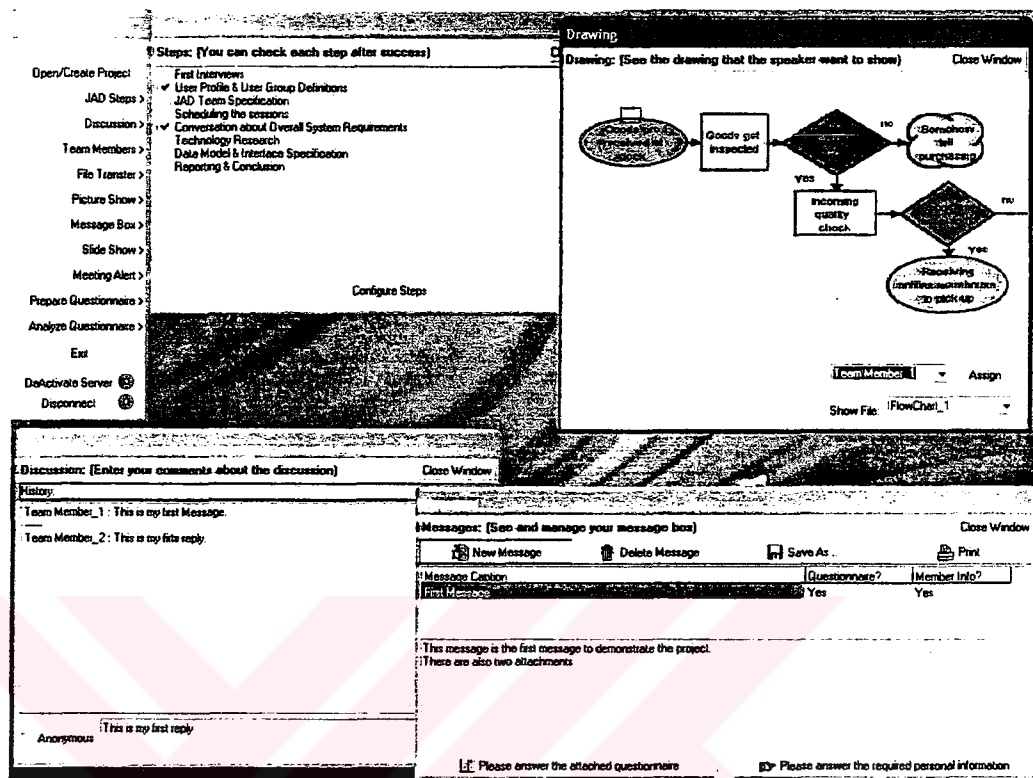


Figure 5.1 General View of UCHE Project

The Utilized Central Hall Environment (UCHE) project is software for management of tasks and realization a virtual meeting. This guide is prepared for administrative use of the software. In the next sections, the explanations of the features and usage of these features on interface can be found. (Figure 5.1 General View of UCHE Project)

The common feature of all windows is the explanation and exit panel on the top. Each form in the system has a panel at top which contains a description about the work of the form. Next to the description, there is a button to close the selected form, which is named "Close Window".

### 5.4.1.1 UCHE Center

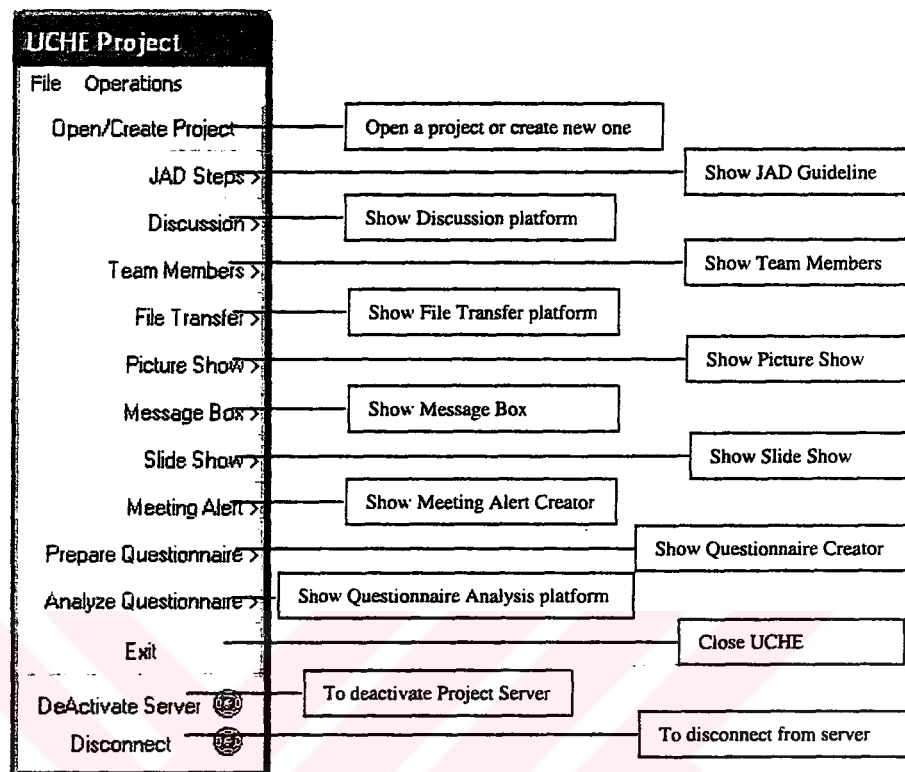


Figure 5.2 UCHE Center

The buttons, which are seen on the panel, are the entrance buttons to the platforms. The Meeting Administrator can press the button to see the related form to use the specified feature. After the new form is opened, the button will be lowered to show, that this feature is ready to use. The raised buttons are the features that should be pressed to use. (Figure 5.2 UCHE Center)

### 5.4.1.2 New Project

Enter the New Project Information

Project Name: My\_First\_Project

in

C:\

- Documents and Setting
- Downloads
- Inprise
- My Music
- Personal

c: []

IP of Meeting Administrator:

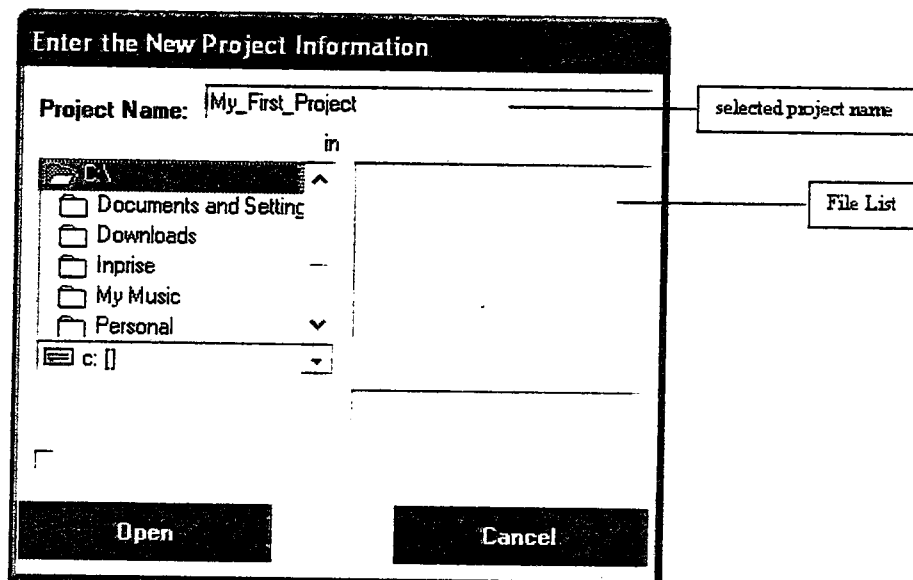
☐ Meeting Administrator for this Project

Create Cancel

Figure 5.3 Creating a new project

The form is prepared to take the parameters from the user to create the project. The Project Name box is required and defines the identical name of the project. This name is represented in the documents and on the directory names in the computer. After that, the directory can be selected, in which the project will be created. Turn the “Meeting Administrator for this project” check to ON, because you will create a server for a project. The “IP of Meeting Administrator” input box will be invisible. After the required fields are filled, press “Create” button to start the creation operation. (Figure 5.3 Creating a new project)

### 5.4.1.3 Open an Existing Project



**Figure 5.4 Opening an existing project**

The Meeting Administrator can open an existing project, which is defined before. To open the existing project, the project file should be pointed from the file list. If the Project Name describes the project that you want, then you can press "Open" button to open the project. (Figure 5.4 Opening an existing project)

#### 5.4.1.4 JAD Guideline

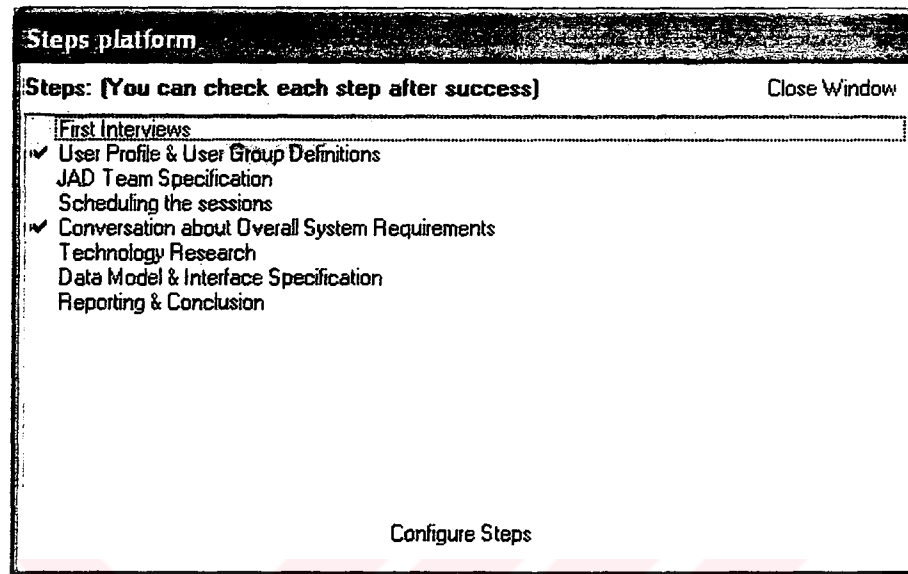


Figure 5.5 Guideline (or Agenda) to follow JAD

The software is designed for an ordered and standardized Joint Application Development implementation. Some standard steps are listed at the agenda list. To specify a step as completed, the administrator can mark it with the check. To add new steps into the list, or to configure the order of the steps, the "Configure Steps" button may be also used. (Figure 5.5 Guideline (or Agenda) to follow JAD)

### 5.4.1.5 Step Completion

The screenshot shows a dialog box titled "Informations" with the subtitle "Informations: (You must confirm the Step Completion)". In the top right corner, there is a "Close Window" button. The dialog contains several fields:
 

- "The Confirmation Date:" followed by a date field showing "15.07.2011".
- "The Confirmation Time:" followed by a time field showing "13:38:25".
- "The Drawings:" followed by a list box containing "Drawing\_1", "Drawing\_2", "Drawing\_3", "Drawing\_4", and "Drawing\_5". "Drawing\_3" is selected and highlighted.
- "Questionnaires:" followed by a list box containing "Quest\_1".
- "The Slides:" followed by an empty list box.
- "Others:" followed by a list box containing "List of Todobit", "Req: doc", and "Doc1 doc". "List of Todobit" is selected.

 At the bottom of the dialog, there are two buttons: "Ok" (with a checkmark icon) and "Cancel" (with an 'X' icon).

Figure 5.6 Accepting a step as completed

When the administrator tries to check a step, a form will appear to set the parameters about the completed step. The parameters to set a step as completed are firstly the confirmation date and confirmation time. These parameters are automatically defined from the system. If there is no problem, it is not recommend changing them manually. After that, the related files to the step can be defined from the lists of Drawings, Questionnaires, slides and other files. After pressing "Ok" button, the step will be checked as completed. When the administrator presses to the button "Cancel", it will mean that the step is not completed and the operation will be canceled. (Figure 5.6 Accepting a step as completed)

#### 5.4.1.6 Step Configuration

**Add New Step**

Add New Step: (And place with arrows) Close Window

- First Interviews
- User Profile & User Group Definitions
- JAD Team Specification
- Scheduling the sessions
- Conversation about Overall System Requirements
- Technology Research
- Data Model & Interface Specification
- Reporting & Conclusion

Delete

New Step ✓ Add

Box for new step

**Figure 5.7 Configuration of steps in guideline**

The steps or the agenda can be configured according to the needs of the project and after selecting the “Configure Steps” button from the Guideline form, a new form will appear for adding a new step or changing the order of the existing steps. To change the order of the steps, a step will be selected and moved with the help of the arrows, can be found right. To delete a step from the guideline, the step, which is wanted to delete will be selected and the “Delete” button will be pressed. For adding new step, the definition of the new step can be written into the “Box for new step” and “Add” button can be pressed. The new step will be added as the last step into the guideline. After adding it, a replacement operation may be required. (Figure 5.7 Configuration of steps in guideline)

#### 5.4.1.7 Opening Discussion

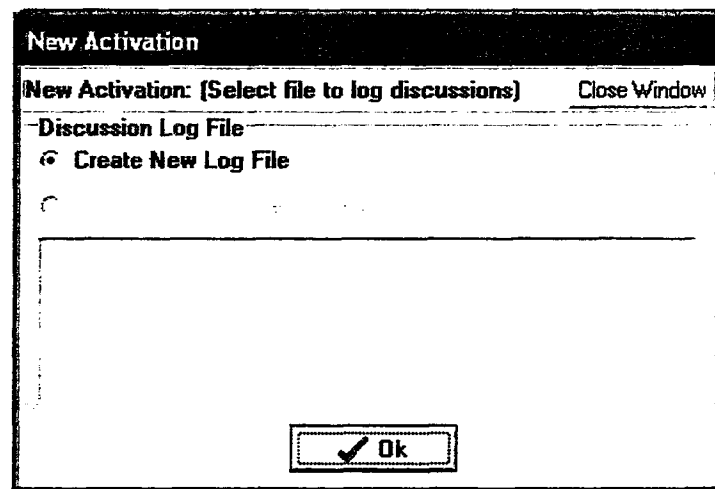
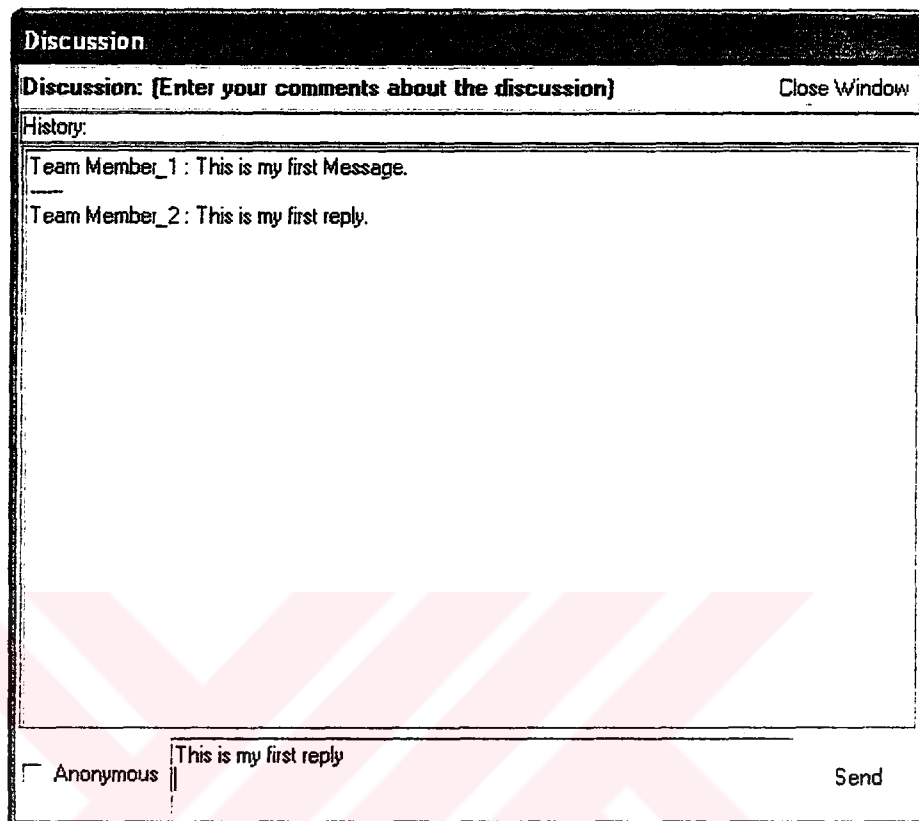


Figure 5.8 Opening a discussion

To open a discussion the administrator should select a log file to remember the discussions. The previous discussion log files can be selected from the list or new log creation choice may be selected to start a new discussion. (Figure 5.8 Opening a discussion)



#### 5.4.1.8 Discussion



The screenshot shows a window titled "Discussion". At the top, there is a text label "Discussion: (Enter your comments about the discussion)" and a "Close Window" button. Below this is a "History:" section containing two messages: "Team Member\_1 : This is my first Message." followed by a horizontal line, and "Team Member\_2 : This is my first reply." Below the history is a large text input area. At the bottom of the window, there is a checkbox labeled "Anonymous" which is currently unchecked, followed by a text input field containing "This is my first reply", and a "Send" button.

**Figure 5.9 Discussion platform**

The discussion platform is prepared only for simple text discussions. The administrator can see the history of the current discussion in the “History” box. To send a message to the platform, the sender can write his message into the box, which can be seen at the bottom, and should press, to the “Send” button. To send a message anonymously, the “Anonymous” should be checked. After disabling the check box on the “Anonymous”, the send messages will go to all members, with the identification of sender. (Figure 5.9 Discussion platform)

### 5.4.1.9 Message Box

The screenshot shows a window titled 'Messages'. Below the title bar is a subtitle 'Messages: [See and manage your message box]' and a 'Close Window' button. A toolbar contains four icons: 'New Message', 'Delete Message', 'Save As ...', and 'Print'. Below the toolbar is a table with three columns: 'Message Caption', 'Questionnaire?', and 'Member Info?'. The first row of the table shows 'First Message' in the first column, 'Yes' in the second, and 'Yes' in the third. Below the table is a large text area containing the message: 'This message is the first message to demonstrate the project. There are also two attachments'. At the bottom of the window, there are two checkboxes: 'Please answer the attached questionnaire' (checked) and 'Please answer the required personal information' (unchecked).

Message Caption	Questionnaire?	Member Info?
First Message	Yes	Yes

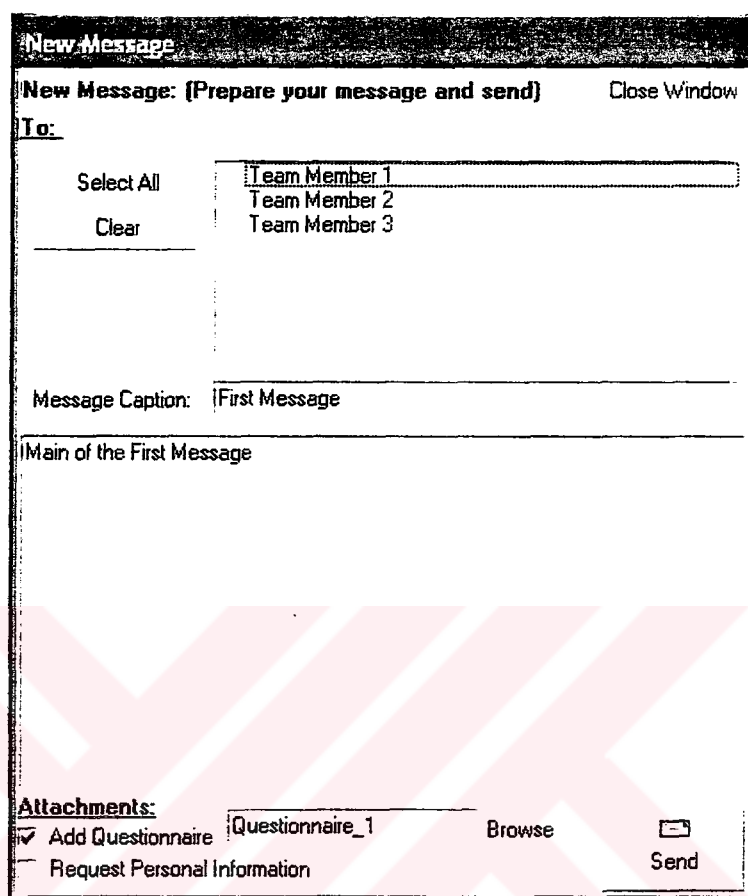
This message is the first message to demonstrate the project.  
There are also two attachments

☒ Please answer the attached questionnaire ☐ Please answer the required personal information

Figure 5.10 Message Box

A message box can be used to read the coming messages from the project group. On this small version of the mailing systems, the user may create new messages, delete the selected message, save the message to another location or print with the help of the buttons. On the list, we can see the new coming messages, with message caption and specification of the attachments. The selected message from the list can be seen on the memo. At the bottom of the form, we can see the attachments of the message. When the message has an attached questionnaire, the left will be visible to take the answers from the user. When the message has a personal information request, the right button will be seen on the form to take the information. (Figure 5.10 Message Box)

#### 5.4.1.10 Writing a New Message



The screenshot shows a window titled "New Message" with a subtitle "New Message: (Prepare your message and send)" and a "Close Window" button. The "To:" field contains a list of members: "Team Member 1", "Team Member 2", and "Team Member 3". To the left of this list are buttons for "Select All" and "Clear". Below the "To:" field is a "Message Caption:" field with the text "First Message". The main body of the window is a large text area labeled "Main of the First Message". At the bottom, there is an "Attachments:" section with two options: "Add Questionnaire" (checked) and "Request Personal Information" (unchecked). The "Add Questionnaire" option has a text field containing "Questionnaire\_1" and a "Browse" button. A "Send" button is located at the bottom right of the window.

Figure 5.11 Writing a new message

The administrator can send a message to which he wants with selecting the names from the members list. The buttons "Select All" and "Clear" can also help to the administrator by selection, with capabilities of selecting all of the members in the list and clearing the all selections. The message caption box should be used to assign a message caption to the mail. After writing the message main, the administrator can add attachments to the mail, by taking checks on the Questionnaire or request of personal information. After checking the checkbox to add a questionnaire, the administrator should also attach the file of previously prepared questionnaire. (Figure 5.11 Writing a new message)

#### 5.4.1.11 Picture Show

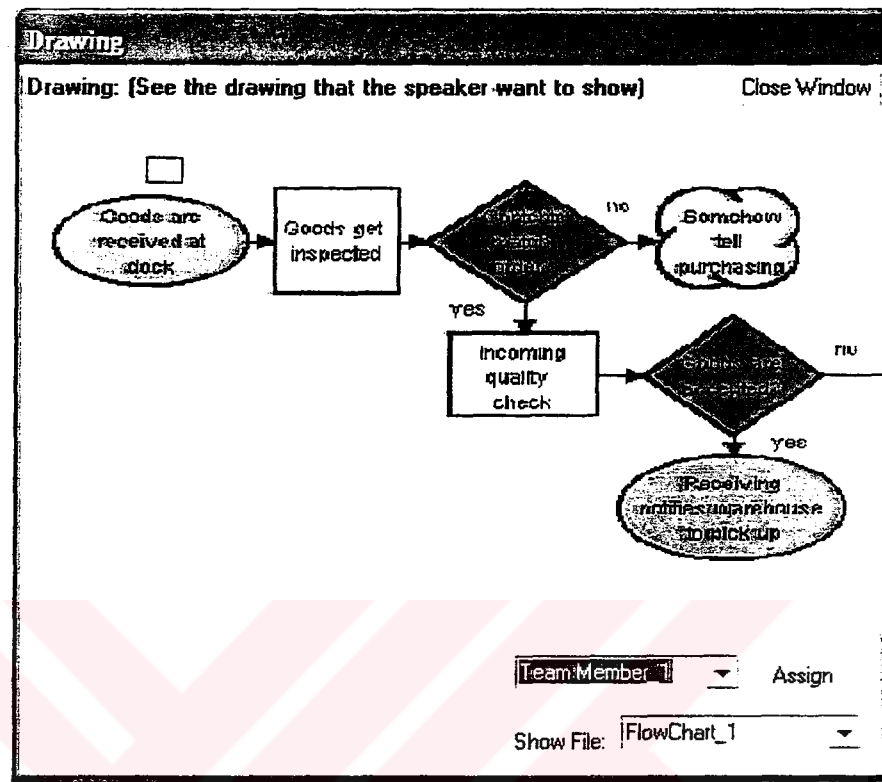


Figure 5.12 Picture Show

The picture show form is to show the picture that a user wants to show. The selected team member is assigned as the shower of the picture and can point something on the picture with rectangles. Other selection is the picture, which is on the screen. The administrator can select the picture that the users will see. Therefore, the presenter should send the picture, which he wants to show, to the meeting administrator before the presentation begins. (Figure 5.12 Picture Show)

#### 5.4.1.12 Slide Show

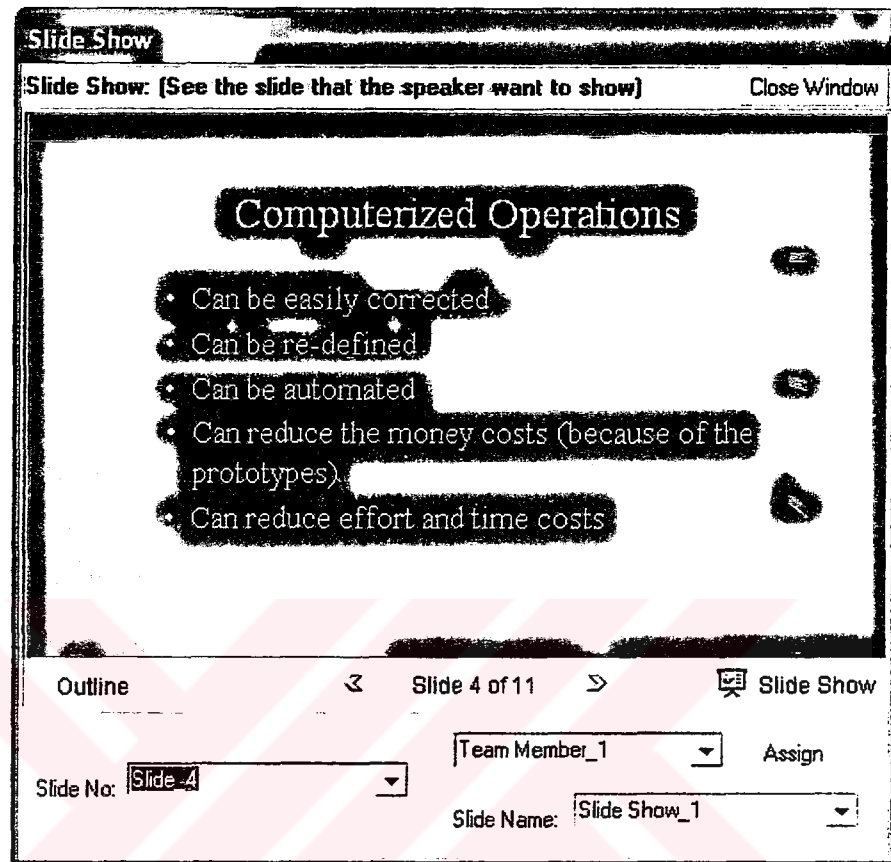
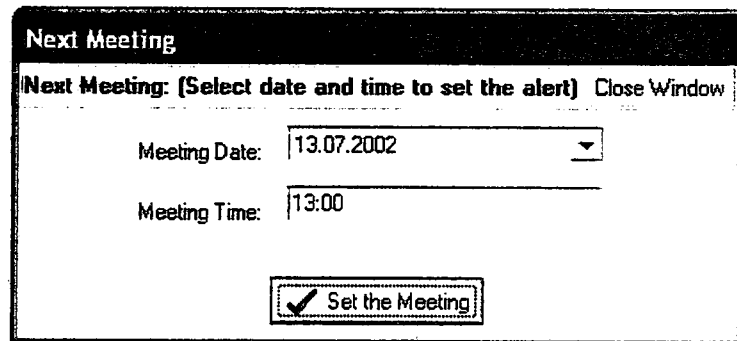


Figure 5.13 Slide Show

The members can also show slides for idea presentation. To show a presentation, the procedure runs like the picture show. The administrator can assign the presenter to change the slides and the name of the slide show, which the users will see. Also the administrator has permission to change the slide on the screen. (Figure 5.13 Slide Show)

### 5.4.1.13 Meeting Alert



**Next Meeting**

Next Meeting: (Select date and time to set the alert) Close Window

Meeting Date: 13.07.2002

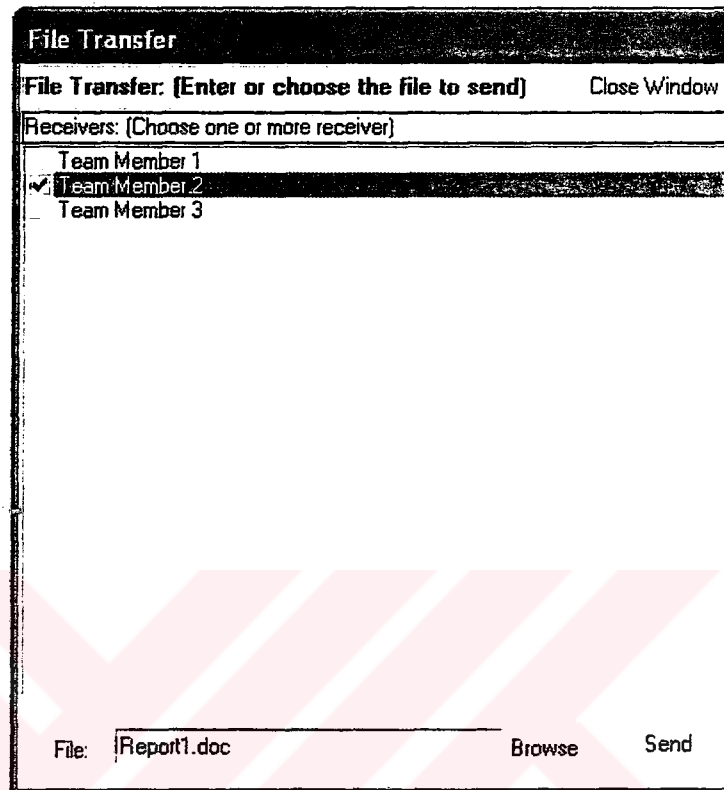
Meeting Time: 13:00

☒ Set the Meeting

Figure 5.14 Next Meeting Alert

The administrator can assign an alert for the next meeting. To specify the alert, the administrator will set the meeting date and meeting time to the fields and press to the “Set the meeting” button. The system will begin to send messages to the boxes periodically to remember next meeting. (Figure 5.14 Next Meeting Alert)

#### 5.4.1.14 File Transfer

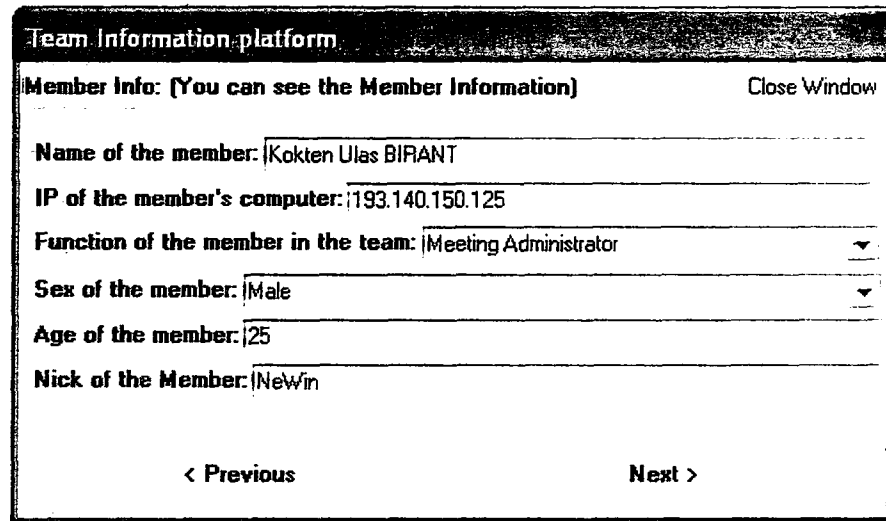


The screenshot shows a window titled "File Transfer". Inside the window, there is a header bar with the text "File Transfer: (Enter or choose the file to send)" and a "Close Window" button. Below this is a section labeled "Receivers: (Choose one or more receiver)" containing a list box with three items: "Team Member 1", "Team Member 2" (which is selected with a checkmark), and "Team Member 3". At the bottom of the window, there is a "File:" label followed by a text input field containing "Report1.doc", a "Browse" button, and a "Send" button.

**Figure 5.15 File transfer platform**

The system has also file transfer capability. To send a file to some members, start by selecting the receiver members. After selecting the members, specify the file to send by writing its name or specifying by the browse button. After these specifications, press to the "Send" button to send the file to the specified members. (Figure 5.15 File transfer platform)

#### 5.4.1.15 Personal Information of Team



The screenshot shows a window titled "Team Information platform". Inside, there's a header "Member Info: (You can see the Member Information)" and a "Close Window" button. Below this, several fields display member information: "Name of the member:" with the value "Kokten Ulas BIRANT", "IP of the member's computer:" with "193.140.150.125", "Function of the member in the team:" with "Meeting Administrator" (indicated by a dropdown arrow), "Sex of the member:" with "Male" (indicated by a dropdown arrow), "Age of the member:" with "25", and "Nick of the Member:" with "Newin". At the bottom, there are two buttons: "< Previous" and "Next >".

Field	Value
Name of the member:	Kokten Ulas BIRANT
IP of the member's computer:	193.140.150.125
Function of the member in the team:	Meeting Administrator
Sex of the member:	Male
Age of the member:	25
Nick of the Member:	Newin

Figure 5.16 Team Information platform

To see personal information of the members of team, the administrator can select the "Team Members" section. In the form appeared, the administrator can see the personal information of the team, which is working on this project. To browse all of the members, the "Next" and "Previous" buttons can be used. (Figure 5.16 Team Information platform)



#### 5.4.1.16 Preparing Questionnaire

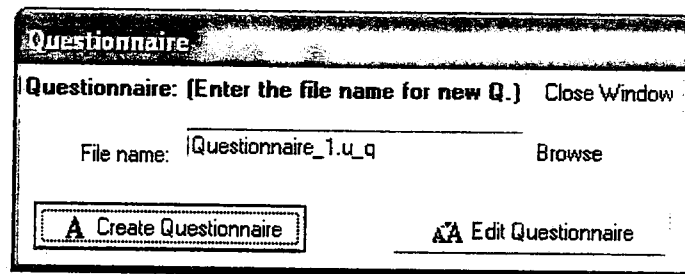
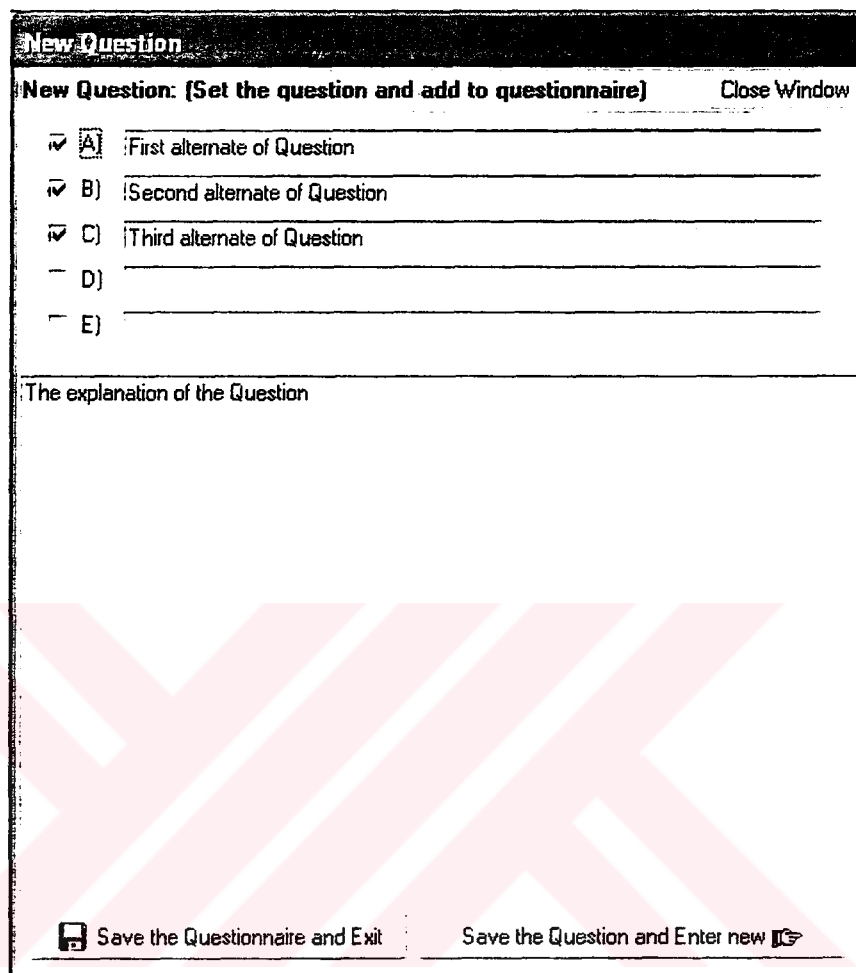


Figure 5.17 Preparing Questionnaire

When the meeting administrator wants to prepare a questionnaire, the Questionnaire form will appear firstly. After specifying the questionnaire name, the administrator can start to create a new Questionnaire or edit the existing questionnaire. (Figure 5.17 Preparing Questionnaire)

#### 5.4.1.17 Adding a New Questionnaire



**New Question**

New Question: (Set the question and add to questionnaire) Close Window

☒ A) First alternate of Question

☒ B) Second alternate of Question

☒ C) Third alternate of Question

☐ D)

☐ E)

The explanation of the Question

Save the Questionnaire and Exit Save the Question and Enter new

**Figure 5.18 Adding a new question**

After specifying the file name of the questionnaire, adding new questions to the questionnaire begins. The alternate count for a question can be maximum 5. The user can select some of the alternates and fill the information as answers. After writing the explanation of the question, the meeting administrator can save the questionnaire and exit the operation or can continue to enter new questions. (Figure 5.18 Adding a new question)

### 5.4.1.18 Questionnaire Analysis

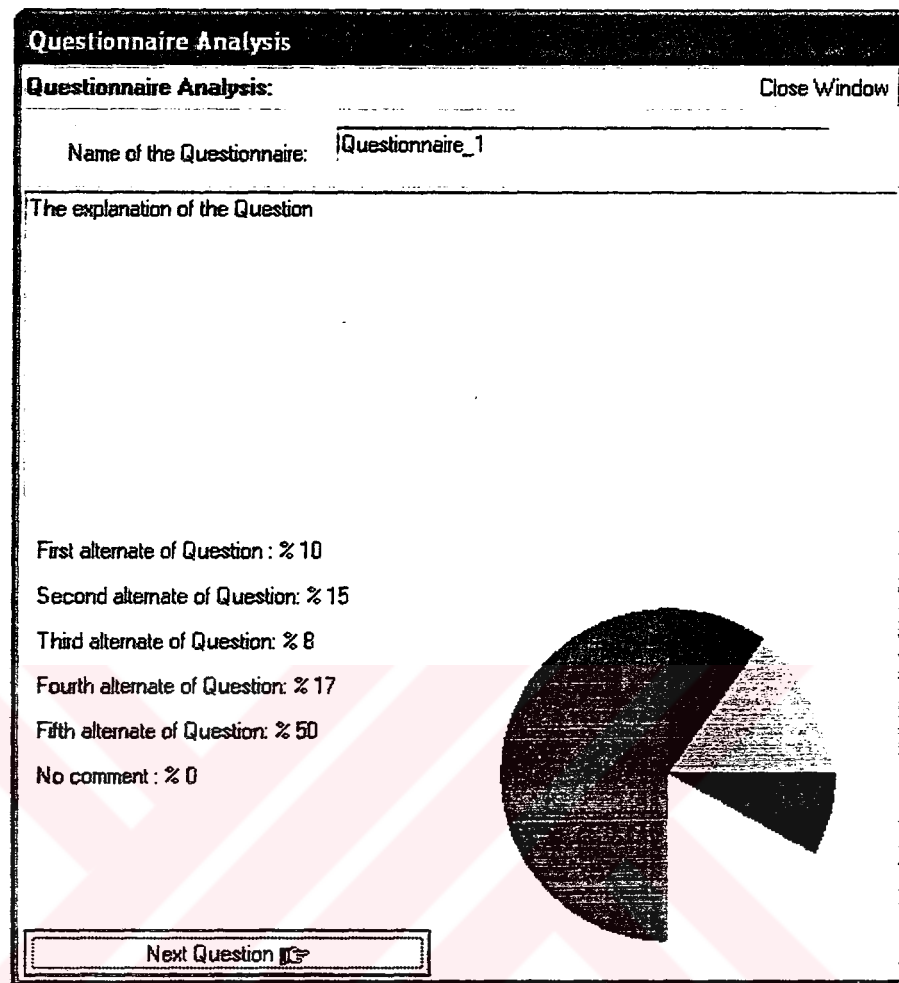


Figure 5.19 Questionnaire Analysis

The meeting administrator can also analyze the results of a questionnaire. After specifying the name of the questionnaire, which will be analyzed, the results of the questionnaire will be loaded into the form and the user can use the next button to browse the questions. On the left pane, the percentages of the answers can be found and the representation with pie-chart is on the right pane. (Figure 5.19 Questionnaire Analysis)

## 5.4.2 Guide for Clients

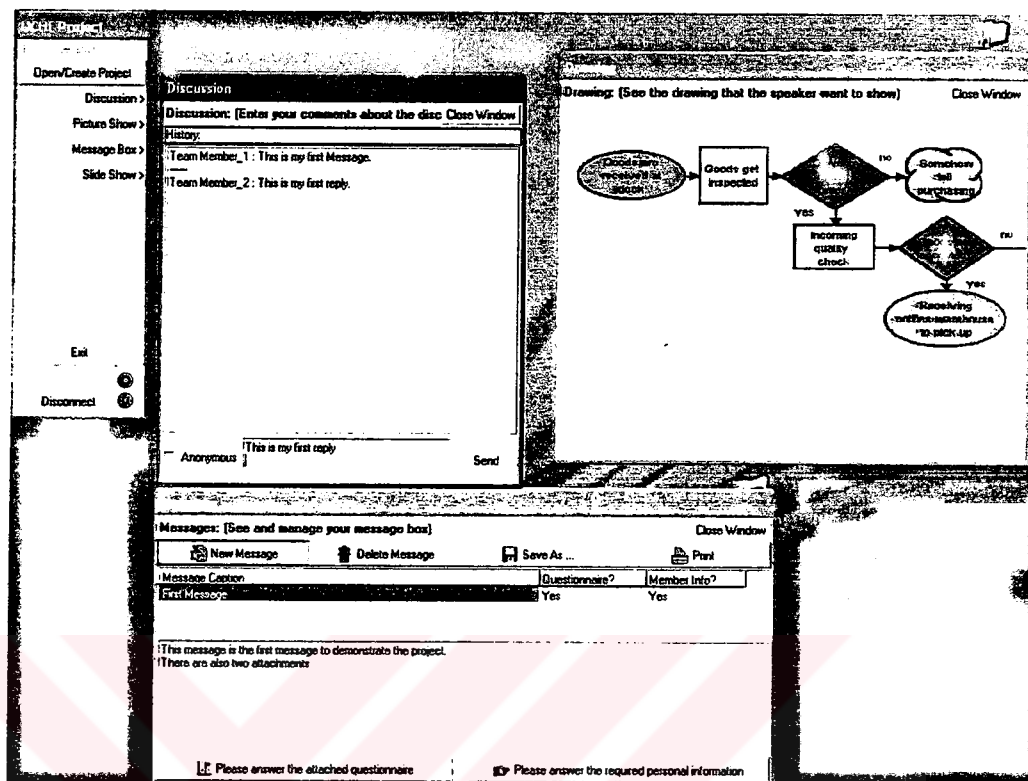


Figure 5.20 General View of UCHE project

The Utilized Central Hall Environment (UCHE) project is developed as software for management of tasks and realization a virtual meeting. This guide is prepared for client use of the software. In the next sections, the explanations of the features and usage of these features on interface can be found. (Figure 5.20 General View of UCHE project)

The common feature of all windows is the explanation and exit panel on the top. Each form in the system has a panel at top which contains a description about the work of the form. Next to the description, there is a button to close the selected form, which is named “Close Window”.

### 5.4.2.1 UCHE Center

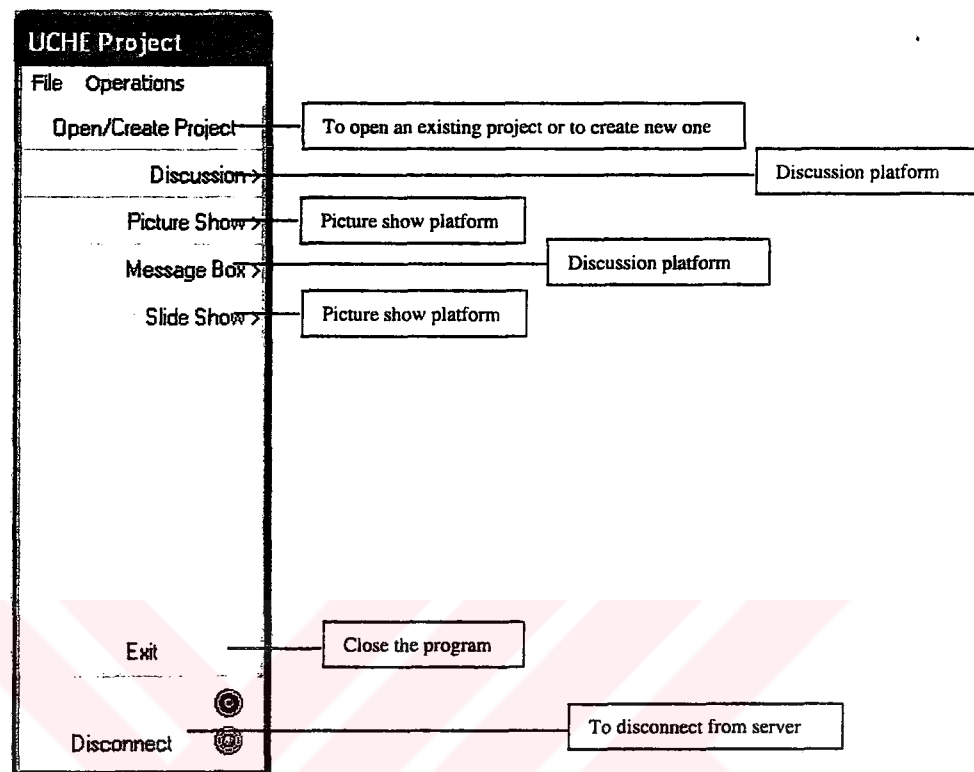


Figure 5.21 UCHE Center

The buttons, which are seen on the panel, are the entrance buttons to the platforms. The user can press the button to see the related form to use the specified feature. After the new form is opened, the button will be lowered to show, that this feature is ready to use. The raised buttons are the features that should be pressed to use. (Figure 5.21 UCHE Center)

### 5.4.2.2 New Project

**Enter the New Project Information**

Project Name: Project4

in

- Personal
- VeryImportant
- Tez
- Program
- Images**

c: \

IP of Meeting Administrator: 193.140.150.125

☐ Meeting Administrator for this Project

Create Cancel

Figure 5.22 Creating a new project

The form is prepared to take the parameters from the user to create the project. The Project Name box is required and defines the identical name of the project. This name is represented in the documents and on the directory names in the computer. After that, the directory can be selected, in which the project will be created. Leave the “Meeting Administrator for this project” check to OFF, because you will not create a server for the project. Fill the “IP of Meeting Administrator” input box with IP number of administrator. (This data can be taken from the network administrator or from the Meeting Administrator). After the required fields are filled, press “Create” button to start the creation operation. (Figure 5.22 Creating a new project)

### 5.4.2.3 Open an Existing Project

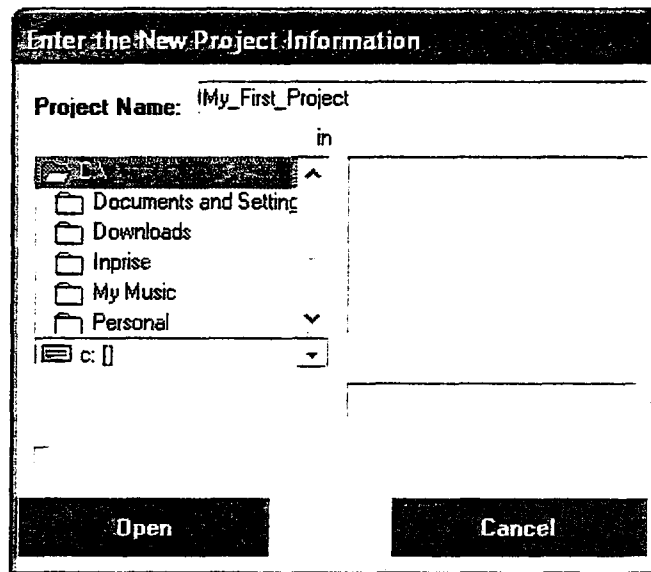


Figure 5.23 Opening an existing project

The user can open an existing project, which is defined before. To open the existing project, the project file should be pointed from the file list. If the Project Name describes the project that you want, then you can press "Open" button to open the project.

#### 5.4.2.4 Discussion

The screenshot shows a window titled "Discussion". At the top, there is a text input field with the placeholder "Discussion: (Enter your comments about the discussion)" and a "Close Window" button. Below this is a "History:" section containing two messages: "Team Member\_1 : This is my first Message." followed by a dashed line, and "Team Member\_2 : This is my first reply." Below the history is a large text input area. At the bottom, there is a checkbox labeled "Anonymous", a text input field containing "This is my first reply", and a "Send" button.

**Figure 5.24 Discussion platform**

The discussion platform is prepared only for simple text discussions. The user can see the history of the current discussion in the “History” box. To send a message to the platform, the sender can write his message into the box, which can be seen at the bottom, and should press, to the “Send” button. To send a message anonymously, the “Anonymous” should be checked. After disabling the check box on the “Anonymous”, the send messages will go to all members, with the identification of sender. (Figure 5.24 Discussion platform)



### 5.4.2.5 Message Box

The screenshot shows a window titled "Messages" with a subtitle "Messages: [See and manage your message box]" and a "Close Window" button. The window contains a toolbar with "New Message", "Delete Message", "Save As ...", and "Print" buttons. Below the toolbar is a table with three columns: "Message Caption", "Questionnaire?", and "Member Info?". The first row shows "First Message" with "Yes" in both the "Questionnaire?" and "Member Info?" columns. Below the table is a large text area containing the message: "This message is the first message to demonstrate the project. There are also two attachments". At the bottom of the window, there are two buttons: "Please answer the attached questionnaire" and "Please answer the required personal information".

Message Caption	Questionnaire?	Member Info?
First Message	Yes	Yes

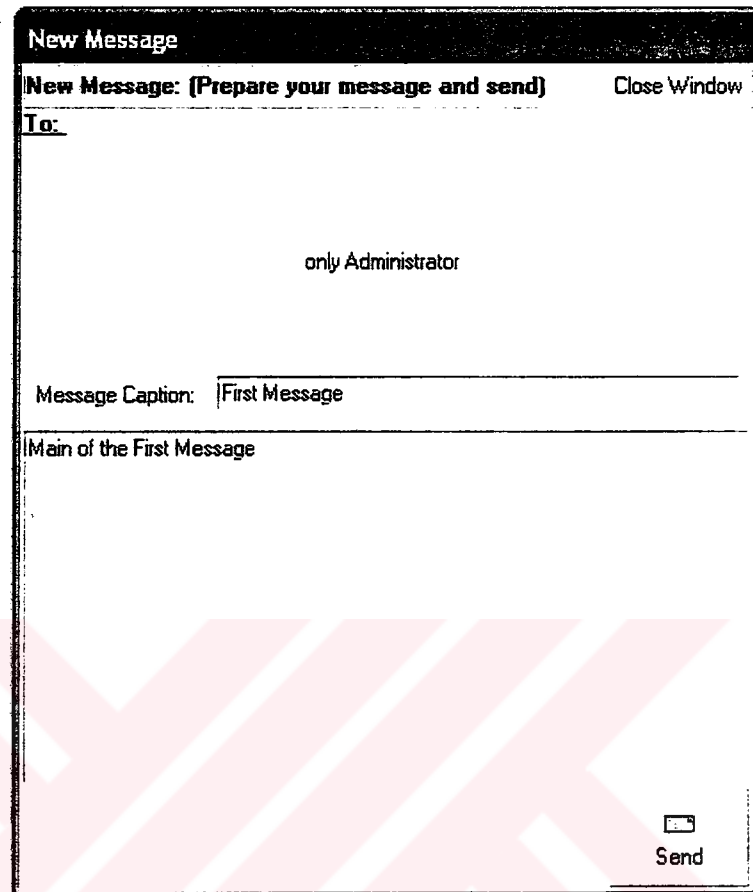
This message is the first message to demonstrate the project.  
There are also two attachments

☐ Please answer the attached questionnaire ☐ Please answer the required personal information

Figure 5.25 Message Box

A message box can be used to read the coming messages from the project group. On this small version of the mailing systems, the user may create new messages, delete the selected message, save the message to another location or print with the help of the buttons. On the list, we can see the new coming messages, with message caption and specification of the attachments. The selected message from the list can be seen on the memo. At the bottom of the form, we can see the attachments of the message. When the message has an attached questionnaire, the left will be visible to take the answers from the user. When the message has a personal information request, the right button will be seen on the form to take the information. (Figure 5.25 Message Box)

#### 5.4.2.6 Writing a New Message



New Message

New Message: (Prepare your message and send) Close Window

To:

only Administrator

Message Caption: First Message

Main of the First Message

Send

Figure 5.26 Writing a new message

The user can send only messages to administrator as a result of the centralization approach. The message caption box should be used to assign a message caption to the mail. After writing the message main, the user can send the written message. (Figure 5.26 Writing a new message)

### 5.4.2.7 Picture Show

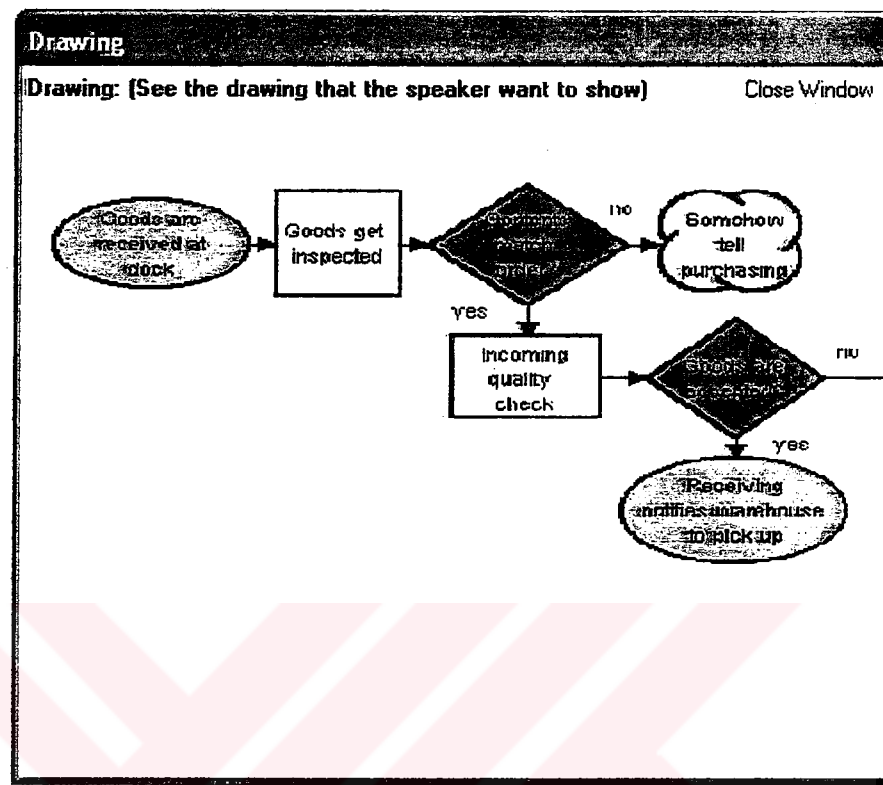


Figure 5.27 Picture Show

The picture show form is to show the picture that a user wants to show. When the user is selected to present the picture by the administrator, then he can move his mouse on the picture to move the pointer. However, if the user is not selected by the meeting administrator, he can only watch the moving pointer on the picture, which is directed by the selected user (presenter). Here, the meeting administrator can select the picture that the users will see. Therefore, the presenter should send the picture, which he wants to show, to the meeting administrator before the presentation begins. (Figure 5.27 Picture Show)

#### 5.4.2.8 Slide Show

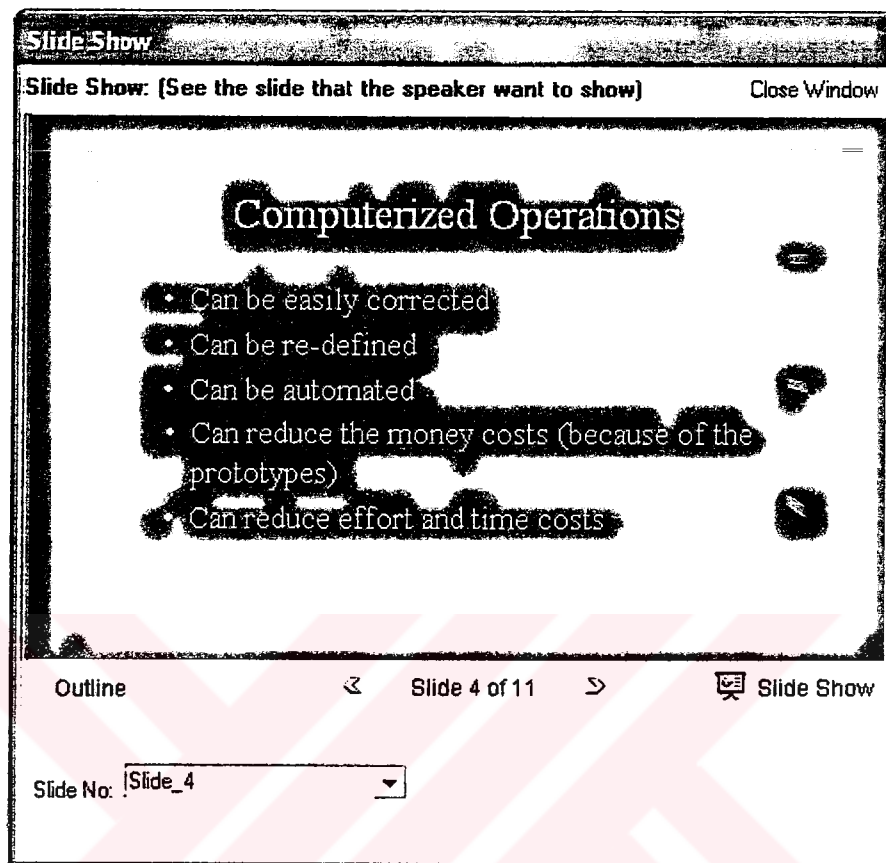
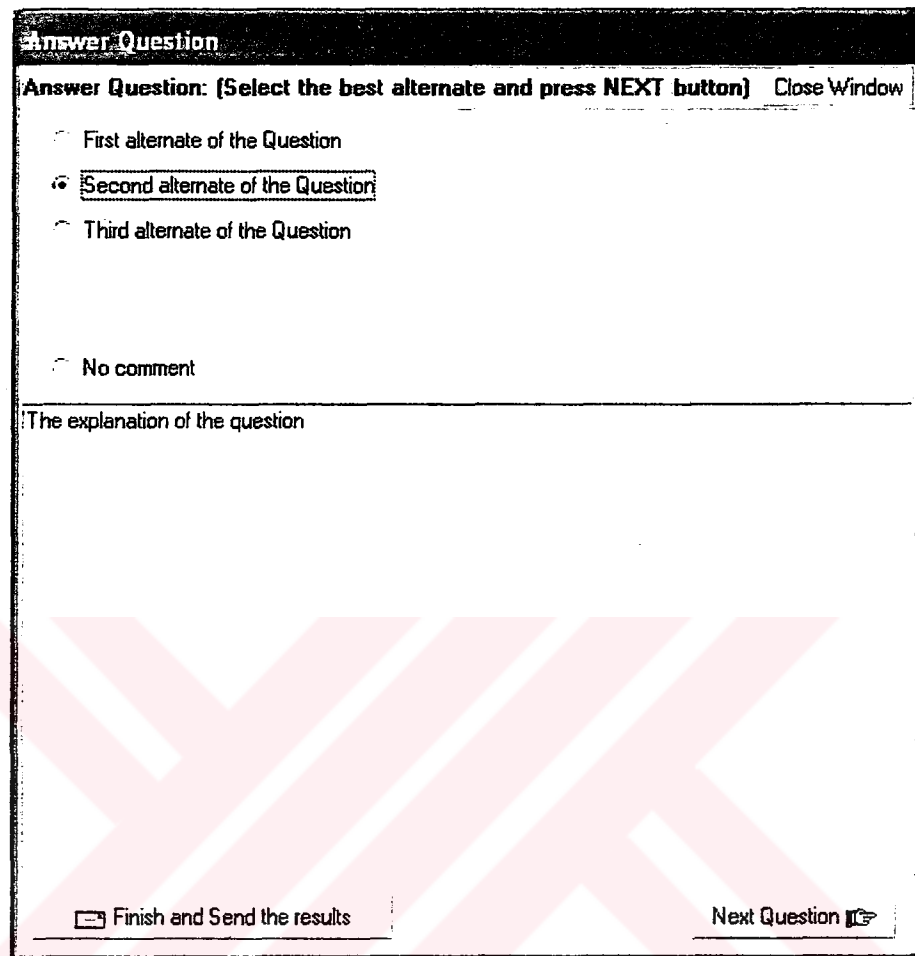


Figure 5.28 Slide Show

The members can also show slides for idea presentation. To show a presentation, the procedure runs like the picture show. The administrator can assign the presenter to change the slides and the name of the slide show, which the users will see. And the selected user can change the current slide on the member's screen with the help of the list. However, when the user is not the presenter, then he will also not see the slides list. (Figure 5.28 Slide Show)

#### 5.4.2.9 Answering a Questionnaire

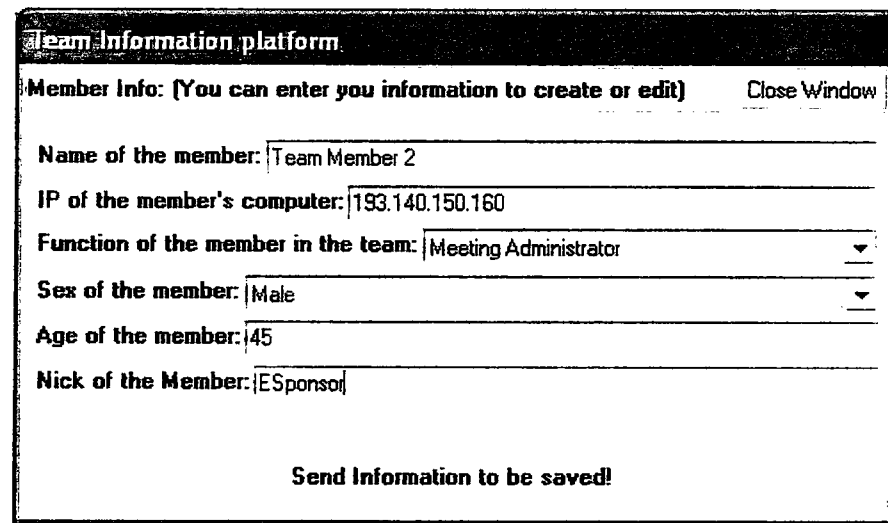


The screenshot shows a window titled "Answer Question". At the top, it says "Answer Question: (Select the best alternate and press NEXT button)" followed by a "Close Window" button. Below this, there are four radio button options: "First alternate of the Question", "Second alternate of the Question" (which is selected), "Third alternate of the Question", and "No comment". A horizontal line separates the options from a large text area labeled "The explanation of the question". At the bottom, there are two buttons: "Finish and Send the results" on the left and "Next Question" on the right.

Figure 5.29 Answering a questionnaire

When the administrator sends a questionnaire to answer, a button to show the questionnaire can be seen on the message box. After pressing this button, the Questionnaire will be shown to be answered. On the Questionnaire show platform, the explanation and the alternates of the question can be seen. After answering all of the questions, the user can send his answers to the server. (Figure 5.29 Answering a questionnaire)

#### 5.4.2.10 Answering an Information Request



The screenshot shows a window titled "Team Information platform". Inside the window, there is a header bar that says "Member Info: (You can enter you information to create or edit)" and a "Close Window" button. Below the header, there are several input fields with labels and values:

- Name of the member: Team Member 2
- IP of the member's computer: 193.140.150.160
- Function of the member in the team: Meeting Administrator (with a dropdown arrow)
- Sex of the member: Male (with a dropdown arrow)
- Age of the member: 45
- Nick of the Member: ESponsor

At the bottom of the form, there is a button labeled "Send Information to be saved!"

Figure 5.30 Answering an information request

If the meeting administrator needs the personal information about the member, then he can send a personal information request. When a message is appeared in the box with information request, then the button for information request will be visible. After selecting this button, the Team Information platform appears on the screen. After filling the required information, the user can send the information to the server, by pressing the button. (Figure 5.30 Answering an information request)

---

## CHAPTER SIX

# CONCLUSION & FUTURE WORKS

---

The software development becomes more standardized nowadays. The software developer companies were prepared their teams according to the development rules. After giving the problem to solve, the management of the development company knows that the solution will be prepared in time with some additional outputs to improve the standards.

Naturally, these companies are the best software developers. (When we think over the Capability Maturity Model (CMM), these are companies in fourth or fifth level.) And this show, how important is going on a disciplined structure. Importance of defining the Joint Application Development in a project development process starts from here. The young software development firms can use such start points.

The most important part of a successful JAD implementation is facilitation and the documentation of the process. And these two works of the JAD session should be given to the experts. Another solution is giving an external help to the facilitator and scribe. This help may be defined as a Group Support System that is specialized for Joint Application Development. And to give more help to the scribe, we can also take the meetings, which are the most important point in JAD, into the virtual world.

When we looked from this point of view, then we can see that the importance of the standardization of Joint Application Development over networking (JAD-NW) more than we think. The approach in this thesis is also defined as specifying a

standard for JAD over Internet. And a tool is developed also to realize the prepared system and to test it in the real world. (UCHE Project)

The quality for the software projects in Turkey is a new idea. Most of the software developers begin newly to standard their projects. This means that there are not so many facilitator experiences for quality checking or project development. And an approach to help the developers for the first steps of learning project management and standardization may be useful.

### **6.1 Testing**

It is explained that, the software development industry is a new concept for Turkey and it is required that some researches may be done on the software development in Turkey. Some researches about the current situation may be done to infiltrate the development process in the companies. After taking some hints about the general manner, than some specific methodologies for Turkish thought structure and Turkish software market can be prepared.

After preparing some results about the current situation, the software developers can be forced to use the system, which is explained in the thesis with or without the prepared tool. And the tool and system can be revisited according to the market needs of the Turkish software development industry.

### **6.2 Improvements on Tool**

The prepared tool (The UCHE Project) can be always said as incomplete, because the computer technology improves everyday. Some improvements can be done on the tool or a new tool may be developed to operate the methodology. The improvements can be classified in two classes: Technical improvements and functional improvements.

The UCHE's development system is Borland Delphi 6.0. According to the technological improvement, the development tool can be changed with a more



powerful and the system may run faster on this technology. Another technical improvement may be using some more efficient algorithms instead of the current algorithms.

Functional improvements can be explained in separate sections and may be listed as follows:

- Using multimedia
- Security improvements
- More parameterized manner
- Automatic documentation with Artificial Intelligence

#### **6.2.1 Using Multimedia**

The computers and Internet technology is growing everyday. And after using multimedia (audio and video capabilities) on the computers, the information exchange earned a new dimension. Using audio and video over networks becomes easier after the new technologies on the networking technologies. And the first improvement on the tool can be using these high technologies.

One of the problems of using Group Support Systems were the communication problems with text discussions. Text discussions were not as easy as real world. To prepare the system more effective, video discussions can be taken. After using the multimedia capabilities, new problems will be like face and sound recognition for automatic documentation or higher network requirements for a good performance from the system.

### **6.2.2 Security Improvements**

The prepared tool is not checked for security attacks. The information goes over socket programming and this cannot be defined secure. To provide the security in the system, the communication manner can be changed or the messages can be sent as encrypted. However, the important objective is that the used tool for a real world application should have a security feature for a usage over Internet.

### **6.2.3 More Parameterized Manner**

The tool is prepared with some built-in templates. These templates would be changed according to the requests of the customer. There are two ways to realize this: The templates can be defined as text documents and after the user changes the template document, the system can use the new one. The second system is preparing a new tool to access the template documents and be able to change them. By using one of these systems, the system's inputs or outputs may be changed like the properties of personal information or the discussions format.

### **6.2.4 Automatic Documentation with Artificial Intelligence**

The current tool is only available for collecting the information and preparing the connections between the related documents. The system should be supported by a good designed and intelligent reporting system.

The system has to define the important things from the discussions with an intelligent summarization capability. After that, the collected data (drawings, slide shows, etc.) may be inserted into the related sections of the summary from the discussion. To realize such a complete system is also a large project and can be done in another thesis.

---

## REFERENCES

---

- Bardram, J.E. (2000). Scenario Based design of Cooperative Systems Re-designing a Hospital information System in Denmark. Group Decision and Negotiation 9. Kluwer Academic Publishers.
- Carmel, E., Whitaker R.D., & George, J.F. (1993). PD and Joint Application Design: A Transatlantic Comparison. Communications of the ACM, June 1993/Vol.36, No: 4.
- Cline, A. (2000). Joint Application Development for Requirements Collection and Management. Carolla Development
- Crawford, A., & Soltys R. (1993). The ten tough questions to ask your training vendor. The Facilitator.com. last accessed: 16/07/2002 11:37.
- Cristel, M., & Kang, K. (1992). Issues in Requirements Elicitation. Software Engineering Institute, Carnegie Mellon University.
- Christie, A.M. (1999). Organizational and Social Simulation of a Software Requirements Development Process. Software Engineering Institute.
- Creative Data Company. Development Methodology – Joint Application Development (JAD). <http://www.creativedata.com/research/jad.html>. last accessed: 16/07/2002 10:50
- Garner R. (1994). Why JAD goes bad?. Computerworld.

Geier, J. (2000). Improving Requirements with Joint Application Design (JAD).  
[http://www.wireless-nets.com/whitepaper\\_jad.html](http://www.wireless-nets.com/whitepaper_jad.html). last accessed: 16/07/2002  
 11:39

Gregory S.& Phong T. (2000) JAD, RAD and PD. University of Calgary.

Grossman, M. (2001). Joint Application Development. The Miami Herald.

Hayes, G.,& Daniels Jr., R.M. (1996). Topic Analyzer: A GSS tool for Idea  
 Generation and Idea Analysis. University of South Alabama. last accessed:  
 16/07/2002 11:21

Hirschberg, M.A. (1998). Rapid Application Development: A brief overview.  
Software Tech News. Vol.2 Number.1. last accessed: 16/07/2002 12:01

Human Resource Services (1998). Joint Application Development (JAD): "What do  
 they really want?". The University of Texas in Austin.

Jennerich, B. (1990). Joint Application Design – Business Requirements Analysis for  
 Successful Reengineering. <http://www.thebeenet.com/bluebird/jaddoc.htm>. last  
 accessed: 16/07/2002 10:57

Jones, P.H. (1997). Handbook of Team Design: "A Practitioner's Guide to Team  
 Systems Development". Mc-Graw-Hill Publications, New York.

Klenc, M.W. (2001). The Effective Methodology for System Requirement Analysis,  
[http://www.umsl.edu/~sauter/analysis/488\\_f01\\_papers/Klenc/](http://www.umsl.edu/~sauter/analysis/488_f01_papers/Klenc/). Last accessed:  
 16/07/2002 11:45

Kock, N. (2001). Asynchronous and distributed process improvement: the role of  
 collaborative technologies. Information Systems Journal. 2001 Vol. 11

- Kuchmistaya, S.B. (2001). Incorporation of Joint Application Design (JAD) in Systems Requirement Determination. Information Systems Analysis.
- Lee S., Qian M.Z., & Ford S. (2001). JAD, PD, RAD. Software Engineering Research Network.
- Lowry, P.B., & Wilson D.C. (2000). The potential of Group Support Systems (GSS) to enhance Systems Analysis and Design Processes and Outcomes. Proceedings of the 3<sup>rd</sup> Annual Conference of the Southern Association for Information Systems, Atlanta, Georgia, April 2000.
- Maner, W. (1997). Rapid Application Development. <http://csweb.cs.bgsu.edu/maner/domains/RAD.htm>, last accessed: 16/07/2002 11:55.
- Matthews, J. (1995). JAD to rescue. Computerworld.
- Paulson J., Wong R., Hong M., & Springl M. (1998). JAD & Participatory Design. Software Engineering Research Network.
- Romano Jr. N.C., & Briggs O.R. (1999). Distributed GSS Facilitation and Participation: Field Action Research. Proceedings of the Hawaii International Conference on System Sciences, IEEE.
- Sommerville, I., & Sawyer P. (1997). Requirements Engineering, A good practice guide. John Wiley & Sons Publication, New York.
- Villegas, A. (1992). Joint Application Development. <http://www.usfca.edu/~villegas/classes/992-3281/3281modD/tsld003.htm>. last accessed: 16/07/2002 11:26

Yatco, M. (1999). Joint Application Design/Development. University of Missouri - St.Louis.

