



University of  
**Salford**  
MANCHESTER

# Requirements engineering process in the CIC development: The divercity project case study

Arayici, Y and Aouad, GF

<b>Title</b>	Requirements engineering process in the CIC development: The divercity project case study
<b>Authors</b>	Arayici, Y and Aouad, GF
<b>Type</b>	Conference or Workshop Item
<b>URL</b>	This version is available at: <a href="http://usir.salford.ac.uk/11526/">http://usir.salford.ac.uk/11526/</a>
<b>Published Date</b>	2004

USIR is a digital collection of the research output of the University of Salford. Where copyright permits, full text material held in the repository is made freely available online and can be read, downloaded and copied for non-commercial private study or research purposes. Please check the manuscript for any further copyright restrictions.

For more information, including our policy and submission procedure, please contact the Repository Team at: [usir@salford.ac.uk](mailto:usir@salford.ac.uk).

# REQUIREMENTS ENGINEERING PROCESS IN THE CIC DEVELOPMENT: THE DIVERCITY PROJECT CASE STUDY

Yusuf Arayici, Ghassan Aouad,  
School of Construction and Property Management, University of Salford, Greater Manchester, M7 1NU  
[y.arayici@pgt.salford.ac.uk](mailto:y.arayici@pgt.salford.ac.uk), [g.aouad@salford.ac.uk](mailto:g.aouad@salford.ac.uk)

**ABSTRACT:** Understanding the users' real requirements is absolutely critical to the development of successful information systems. To achieve a user-oriented and a high level quality of system and to increase the implementation of the integrated software systems, it is important that the user requirements must be captured and modelled in the right way. If done correctly, the software to be developed will meet the user' needs and lead to better user satisfaction and implementation.

This paper describes the case study of requirements engineering process adopted in the DIVERCITY system development. DIVERCITY was an EU funded project undertaken by a European consortium of researchers and practitioners from the construction industry. They were working together in an attempt to develop VR based software that enables the industry to better undertake the client briefing, design review, and construction planning phases of a construction project. It is the acronym for the project and the prototype: Distributed Virtual Workspace for enhancing Communication within the Construction Industry.

The techniques used for capturing the requirements, and modelling, verification and validation are explained. Furthermore, the need for user requirements, key criteria definition for the evaluation and mastering the requirements engineering process for the future integrated system developments are also described.

**Keywords:** Contextual design, distribution, integration, testing, use cases.

## 1. INTRODUCTION

Requirements capturing is the process where the users' needs in a software project are identified and modelled. This process is regarded as one of the most important aspects of building an information system because it is during this process that it is decided what is to be built.

Car (2000) defined the requirements as properties, attributes, services, functions, and behaviours needed in a product to accomplish the goals and purposes of the system to be developed.

Maguire (1996) emphasised that adopting a user centred design process leads to more usable systems and products. It reduces the risk that the resulting system will under-deliver or fail. User centred design implies the following:

- Early focus on users, tasks and environment;
- The active involvement of users;
- An appropriate allocation of function between user and system;
- The incorporation of user-derived feedback into system design;
- Iterative design whereby a prototype is designed, tested and modified

Carr (2000) addressed that the capturing of requirements seems like a simple matter that can be done quickly and put aside. But studies, surveys and experience demonstrate that poorly executed requirements engineering is the most contributor to system failures such as late delivery, over budget, and either poor or incorrect performance.

Superficially done requirements engineering causes major risks to software development project (Nikula and Sajaniemi, 2002). For example the software projects risk identification framework by Keil et al (1998) includes eleven risks factors and six of them are related to requirements. Furthermore, CHAOS survey (Standish group, 1995 and 2001) identified the

fourteen factors that can cause the failure in software development; six of these factors are related to user requirements risk factors, only one of which is not mentioned in the Keil (1998) risk identification framework. Furthermore, Carr (2000) also addressed the requirements engineering related risk factors that can also cause project failure. The combined set of RE related risks factors contains the following issues.

1. Misunderstanding and misinterpretation the requirements
2. Lack of adequate user involvement
3. Failure to manage end user expectations
4. Changing scope and objections
5. Lack of frozen requirements
6. Conflict between user departments
7. Incomplete and inconsistent requirements and specifications
8. Ambiguous and vague requirements

In the succeeding sections, these are explained; the need for requirements capturing, the DIVERCITY project, the DIVERCITY requirements engineering process including UML, contextual design, incremental prototyping with testing, and criteria definitions which is required to identify strength and weaknesses. Furthermore, the approach taken for mastering the DIVERCITY requirements engineering process and coming up with a generic guideline which can be used for the future CIC (Computer Integrated Construction) systems developments.

## **2. THE NEED FOR REQUIREMENTS ENGINEERING PROCESS**

CIC (Computer Integrated Construction) is an important solution for the integration of the processes through the construction supply chain. Research has emphasized some of the benefits of integration such as reducing the project costs and lifecycle period, removing the non-value added activities in the construction processes, encouraging collaboration, increasing client satisfaction (Sun and Aouad, 2000).

On the other hand, the industry has little awareness of how to use such systems effectively in their work environment because of unfamiliarity with such prototypes; that results in a gap between the developers of CICs and industrialists. One route to developing more user-centred systems is the use of appropriate systems development methodologies, which are suitable to the CIC systems. However, there is currently little debate within the research community as to what are the characteristics of a CIC systems development methodology.

To date CIC researchers have had little focus on the requirements capture issue in systems development, which is actually required to develop user-oriented and more practical CIC systems for the construction industry. Despite increasing interest by both academia and practitioners in CIC, there is little research to identify best practices in requirements capture.

Whilst there was hardly any attention and awareness about user requirements in the early ages of software system development, contemporarily user requirements issue is regarded as an engineering domain in system development. Because the CIC development is also a system development specifically for the construction industry, it is a must for the CIC community to identify and adopt appropriate and comprehensive requirements methodologies to provide successful CIC systems to the industry.

As a result, capturing and modelling the user requirements and developing the systems with the end users is becoming considerably more important.

Furthermore, a number of interviews for identifying the needs for requirements engineering process in the CIC systems developments were conducted in the CIC community. It has been asserted that there was consensus between the interviewees who were stressing the importance of user requirements as a key parameter and it is becoming more and more

important and vital for the CIC development. In addition, the subject of user requirements has influence on the implementation of the CIC systems. Although it can make certain contribution to the implementation, it cannot overcome all the problems of CIC relevant to the implementation. This is because; the capture of user requirements is not the only parameter that influences the implementation. There are several other aspects that also affect the implementation such as cultural issues, legal issues, fragmented nature, procurement, technological issues, contractual issues, processes and educational issues, etc.

However, the respondents have classified these aspects as uncontrollable parameters upon the implementation of CIC to the industry. Actually, process parameter is relatively classified also as controllable parameter compared to the other aspects. However, user requirements and technology issues are fully controllable parameters. In other words, the better requirements capture in the development will lead to the better implementation of the system.

CIC is an evolving concept in the construction industry and subsequently the subject of the user requirements for the CIC development is also an evolving issue. It is becoming a hot topic in system development for the integration of the construction industry. Furthermore, research studies to define the best requirements approaches and methodologies for system and data model development have already been undertaken. Therefore, the user requirements issue will be addressed in this paper.

### **3. The DIVERCITY Project**

Researchers have identified the need for an integrated construction environment, which acts as a project repository, during all stages of the life-cycle (Kiviniemi, 1999) (Aouad, 1997) (Alshawi, 1996) (Aish, 1999) (Grassi, 1999) (Sawhney, 1999). This will improve the communication between the different stakeholders and improve productivity. However, this environment has been difficult to implement. No commercial solutions exist (Shelbourn et al, 2000).

DIVERCITY is a large EU project in the area of construction IT undertaken by a European consortium of researchers and practitioners from the construction industry. It is the acronym for the project and the prototype: Distributed Virtual Workspace for enhancing Communication within the Construction Industry. The prototype presents the mechanism for smoothly and collaboratively conducting the construction projects from early briefing to the detailed design and even further to the end of the construction phase.

DIVERCITY allows users to produce designs and simulate them in a virtual environment. The designs are IFC based and can be viewed by all stakeholders within the project team. The project had the following objectives:

- Creation of a client-briefing workspace, which can facilitate interaction and communication of design ideas between client and the architect.
- Creation of an interactive design review workspace, which can facilitate multi-disciplinary design reviews involving different stakeholders of a construction project, i.e. planners, architect, designers, civil engineers, electrical engineers, contractors, facility managers and security personnel, etc.
- Creation of a virtual construction workspace, which can assess the buildability (the sequence of construction activities, scheduling, material handling etc) of a building.
- Specification and development of a software framework for integrating the above three workspaces and sharing them over networks to support collaboration between geographically distributed project team members.

The DIVERCITY system is comprised of Communication layer, Client Briefing, Lighting, Acoustic, Thermal and Heating Simulations, Visual Product Chronology, lastly Site planning & Analysis, each of which handles different aspects of a construction project in a construction supply chain and how they complement each others to constitute a seamless

integrated computer environment for the sake of excellence of briefing and design and construction planning.

The main strengths of the DIVERCITY system are as follows:

- Improving the co-ordination and communication between the client, design team members and construction professionals by using standard formats, and intuitive VR tools.
- Evaluating the design at the very early stages of the project lifecycle in terms of architectural, technical, financial and environmental aspects since VR tools allow design team to have a quick and high quality feedback on the project (Faraj and Alshawi, 1999).
- Doing what-if scenarios at the detailed design stage to assess the design solution in lighting, acoustic, thermal aspects.
- Closing the gap between the design team and construction team and providing them with an integrated platform for the best buildability and applicable construction planning.
- Support to allow virtual collaborative spaces at geographically distant sites to work together in either synchronise or asynchronise manner.

#### **4. The DIVERCITY Requirements Engineering Process**

Because the construction industry is a large multi-disciplinary industry, with multi-faceted perspectives and requirements, DIVERCITY needed to define broad industrial requirements, and expand into more detail to capture the briefing and design requirements.

DIVERCITY is a large-scale, highly innovative and interactive system. It is concerned with the development of interactive systems that cannot be treated simply as incremental improvements over existing products. In such cases it is not possible to identify user requirements on the basis of empirical techniques, as there are no instances of the use of the product (or similar products) from which to collect data. Consequently, the developers of innovative products must proceed by envisioning the use of the proposed product (Dearden and Howard, 1998).

DIVERCITY reviewed a number of methodologies and finally used a combination of techniques. The DIVERCITY requirements approach is relatively mature compared to the other CIC prototypes. It is detailed down into three category; contextual design, use case driven requirements analysis, incremental prototyping with user tests, each of which are strong techniques to undertake the aspects of requirements elicitation, requirements analysis and requirements validation respectively.

DIVERCITY's development and requirements capture teams spread across four EU countries. That added an additional layer of complexity to the requirements and change management. DIVERCITY requirements engineering process continued throughout the project lifecycle.

##### **The UML phase**

DIVERCITY initially intended to use use-cases and UML as means of requirements capture. CIT (Construct IT) group in the project team was familiar with UML and relying on its popularity in software development. Furthermore, it provides object-oriented development and fits with well some other technical aspects such as IFC. In addition, technical team also supported to use of UML for requirements capture. Therefore, use case driven analysis were selected for requirements elicitation and analysis and rational rose enterprise edition software tool of UML were selected for the development of high level use cases and for the

decomposition of these use cases into further detail object diagrams before committing them to code.

In use case driven requirements capture, the requirements workflow proposed by RUP (Rational Unified Process) was intended to use in order to communicate the complex top-level industry requirements and progressively break into more detail. Furthermore, to manage the large scale and evolving industry requirements, Boehm’s (1996) recommendation for a “win-win spiral model” were also taken on board, where the advises on creating three critical milestones, i.e. (i) lifecycle objectives (ii) lifecycle architectures and (iii) operational capability (shelbourn et al, 2001). The top level use case model for detailed design stage is shown below as an example.

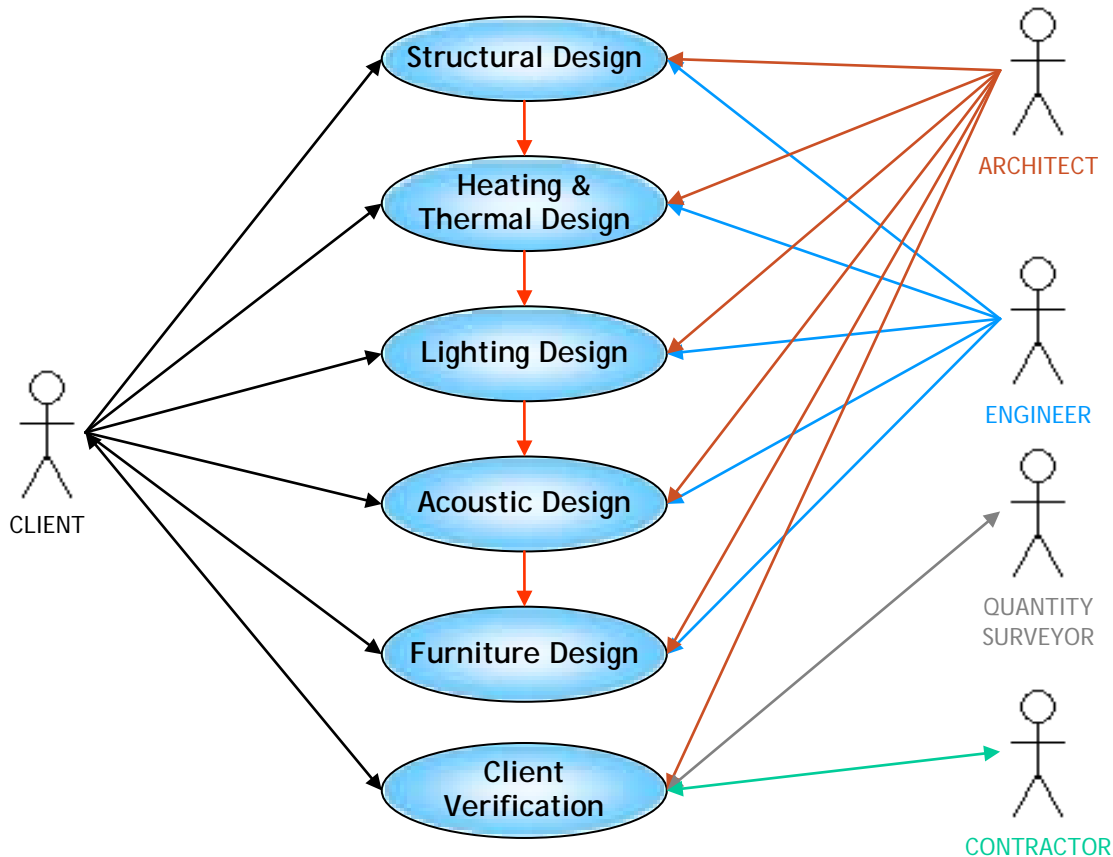


Figure 1: The top-level use case diagram for the detailed design modelling in DIVERCITY.

A number of techniques were used in order to communicate the complex top-level industry requirements and progressively break into more detail. In particular the following approaches, which were taken from RUP requirements workflow, were adopted

- Vision statement
- Stakeholder perspectives
- Use cases
- Systems requirements

### Contextual Design Phase

Often requirements modelling starts with UML (Unified Modelling Language) based on object modelling of the system with efficient tools such as Rational Rose

(<http://www.rational.com>). That results in taking too little consideration to understand user needs and user interface design as well as quality requirements (Regnell et al, 1995).

There are not so many well-formalised methods to support the entire design of a product like DIVERCITY. Due to its well worked out user centred approach the Contextual Design method (Beyer and Holtzblatt, 1998) is chosen to try to early take into account the end user work practice and interface requirements. Furthermore, because system developments which start with UML based object modelling with efficient tools like Rational Rose from Rational Software Corporation (<http://www.rational.com>) takes too little account to early conceptual analyses of user requirements and user interface functionality (Christiansson et al, 2001). The risk is obvious that important properties of the final product with regard to end use are overlooked. Besides that, use case models are loose collection of use cases (Regnell et al, 1995).

Contextual Design methodology has five different types of Work Models. These are as follows;

- Flow, representing communication and co-ordination necessary to do work (roles, responsibilities, actions/communication topics, and spaces which are regarded in DIVERCITY as project internal or project external memories and virtual/physical spaces).
- Sequence, showing the detailed work steps necessary to achieve intent. Sequence models can reveal alternate strategies to achieve the same intent.
- Artefact, showing objects created to support the work
- Culture, representing constraints on the work caused by policy, culture or values, formal and informal policy of the organisation, business climate, self image, feelings and fears of the people in the organisation, possibility for privacy.
- Physical, showing the physical structure of the work environment as it affects the work

These work models were developed in close collaboration between the user groups of the project team.

Afterwards, the storyboarding was conducted. To do so, the user group held a workshop, which ran very well and resulted in emerging a shared understanding amongst the end users. The user group focused on the further details how the DIVERCITY system should function and how the stakeholders such as client, architect, engineer, and contractor should act to conduct their duties using the relevant functions of the DIVERCITY system. 15 scenes were developed and in each scene the duties of the construction stakeholders and how they should interact to conduct their duties collaboratively were explained.

Furthermore, the storyboard enabled the technical team also to understand the requirements clearly and the shared understanding was extended amongst the technical team as well as the end users.

Initially, requirements capture efforts were chaotic and a learning process. However, subsequent to the workshop, the process became relatively more structured and turned into a shared process of requirements capture and subsequently a shared understanding evolved.

Because the storyboarding enabled a shared understanding amongst the project team, it was further developed to use in the testing process. The requirements specification was updated through storyboarding during the period of the DIVERCITY tests. Furthermore, storyboarding enabled close collaboration between the project team.

## **Testing Phase**

In development of innovative systems like DIVERCITY, the first thing that is very important is to have early prototypes to demonstrate in front of the end users. Therefore, the released

early prototypes were undergone black box testing conducted by the project user groups. A quite well documented strategy for the testing process was prepared. Besides, the well-structured testing documentation enabled the user group to conceive how to contribute to the testing process and interact and collaborate with each other throughout the testing process. From the very beginning of the testing process, the distributed team worked very closely in a collaborative manner. The testing process played a major role in evolving the requirements and providing a spiral development process.

In DIVERCITY, testing lifecycle incorporates three main testing phases: Alpha, Beta and Final testing. The DIVERCITY tests were also called as black-box tests. This is because, DIVERCITY tests centre on testing the programs against the written specifications and test observes the programs as black-boxes and is totally unconcerned with the internal structure of the programs (Lewis, W.E., 2000).

Furthermore, there was a symbiotic relationship between the end users and the developers in the DIVERCITY project. The entire tests in the lifecycle were conducted by the end users not the developers. The developers released the prototypes and the end users, who were distributed across Europe, tested the prototypes continuously in a collaborative manner with respect to the defined test criteria above. They continuously provided feedback to the developers throughout the test phases.

The most effective way to reduce risk is to start testing early in the development cycle and to test iteratively, with every build. With this approach, defects are removed as the features are implemented (rational white paper, 1999). Each testing phase-Alpha, Beta, Final- is conducted in an interrelated and continuous and iterative manner.

The user group employed the UML test workflow to conduct the black box tests in a stage wise manner. A typical UML workflow encompasses the stages: plan test, design test, implement test, execute test in integration tests stage, and execute tests in system test stage.

## **5. CRITERIA DEFINITION FOR THE EVALUATION OF THE DIVERCITY REQUIREMENTS ENGINEERING PROCESS**

Central to understanding and improving the requirements engineering processes is subject to the ability of measuring the requirements engineering success (Emam and Madhavji, 1995). Requirements engineering has paid little attention to the problem of analysing requirements in large-scale systems involving people and technology (Sutcliffe, 2001). However, software systems of today are characterised by increasing size, complexity, distribution, heterogeneity, and lifespan. They demand careful capture and modelling of requirements and architectural designs early on, before the low level system details begin to dominate the engineers' attention and significant resources are expended for system construction (Grunbacher et al, 2001).

An attempt to identify a number of criteria for evaluation of the requirements engineering processes applied in system developments was conducted. Whilst this attempt forms the basis for the evaluation and assessment of the DIVERCITY requirements capture process, it also provided a contribution to requirements engineering community with a number of key criteria for the selection of the requirements engineering techniques in software system development.

The attempt started with introducing a number of approaches to improve the requirements engineering processes by means of some key aspects in the RE process. For example, process modelling, product modelling, goal-orientation in RE, complexities in requirements models, quality requirements, testing the requirements, reconciling software requirements and architecture, identifying stakeholders and barriers to the implementation of the requirements engineering processes and so on. A number of key issues were extracted and related to the system development for the construction industry.



On the other hand, a broad research study conducted by Emam and Madhavji (1995) about measuring the success of the requirements engineering techniques was introduced. The research provided with 34 key criteria that addresses quality in the requirements engineering process from a variety of faces including cost/benefit analysis, stakeholders involvements, quality of the products developed, quality of RE services etc.

Lastly, the key issues and criteria are comparatively matched to the each other. It is observed that the criteria have broader scope than the key issues defined. Furthermore, this analysis has supported and strengthened the validity of the criteria identified by Emam and Madhavji (1995).

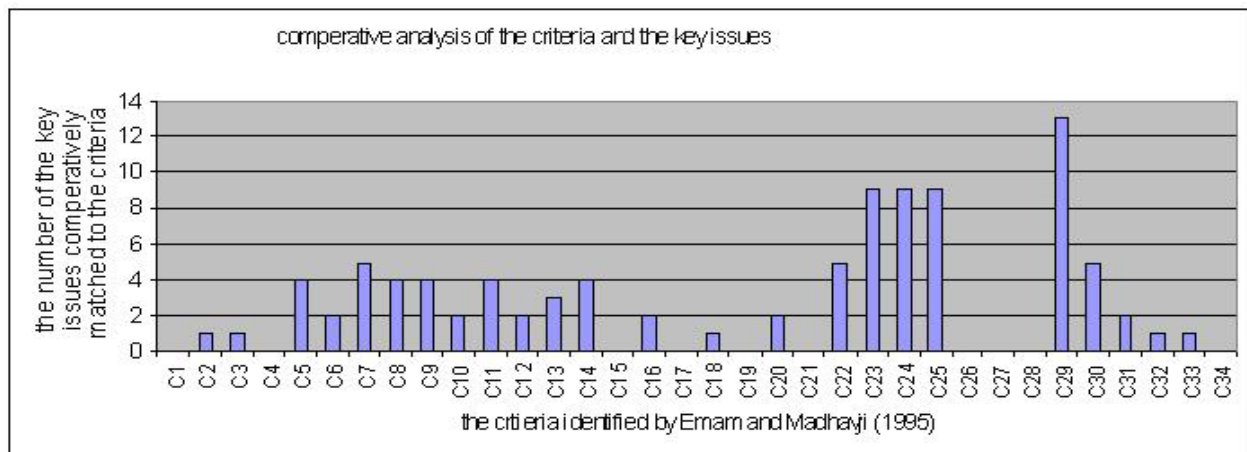


Figure 2: the level of fitness in comparison between the criteria and key issues

## 6. DISCUSSION ON LESSONS LEARNT

It is hard to find well-formalised methods to support the entire design of collaborative and distributed VR systems like DIVERCITY (Christiansson, 2001). Initially user requirements processes were not well planned in advance. Rather, they evolved as the team learned more about the DIVERCITY environment. Since the groups were from different backgrounds, there was no shared understanding and agreements of the requirements capture process. Initial requirements capture and modelling was not regarded as process that requires normally interaction and communication and a shared understanding between the end users as well as the developers. At the outset, there were some isolated requirements capture activities. Subsequently, a shared understanding between the end users did not emerge. This lack of shared understanding reduced the effectiveness of the requirements activities.

Some specific results were conveyed below in bullets

- Although communications and networking will be a major topic over the next ten years Aouad (et al, 1998), the emphasis was less for the communication and networking aspects than the specific applications in the DIVERCITY requirements engineering process.
- One of the important results from the DIVERCITY project is to emphasise the need for parallel integrated efforts on User Environment and System design and the greater emphasis on early user needs and requirements capture.
- Although it was a research project, it was not a blue-sky project and it is an instance prototype of the CIC concept that is very likely to dominate the construction industry in near future. Therefore, the fit between the system and the industry should have been taken into account in the very early stages definition in the requirements process.
- In regard to the fit between the system and the industry in the category the DIVERCITY requirements process has become relatively successful. This is attributable to the

techniques used in the half way through the project such as storyboarding and incremental prototyping with the end users tests as an agile development process, which increased the process awareness and entailed the users' involvement intensively.

- Due to the lack of completeness of the requirements documentation, the users were not sure in many respects until the final stage.
- Since insufficient capability of handling the complexities in many respects through the development process, the users involved in the project were lost and could not pursue the development properly until the storyboard enhancement and users testing.
- From the outset, requirements management was conducted with lack of process and product awareness.
- Testing the requirements through storyboarding against the system enabled relatively better, more consistent and accurate requirements specification.
- The DIVERCITY tests were not only done for verification and validation of the system as well as for verification and validation of the user requirements but they were also done to introduce it to the end users who are not well familiar with such an innovation of computer-integrated construction.
- A process of workflow called storyboard was derived from the user requirements, which implies construction supply chain for lean construction work environment. All tests were conducted according to the storyboard by the end users not the developers. Based on the consecutive test results, the developers released subsequent prototypes until the end users were satisfied in the defined test criteria in every phase.
- The tests were undertaken in different countries at the same period of times in a distributed manner, which brought an extra novelty to the testing approach but also brought about extra difficulty in managing the test activities. Because the test teams were geographically distributed, the test plans were communicated and a consensus should be reached prior to and during every testing phase. Under this circumstance, communication and interaction between the testers distributed across Europe were extremely important for effective tests. Another dimension of the difficulty of distributed testing was that test teams in different countries conducted the tests independently that resulted in different interpretations in discrete countries. These various interpretations were required to be harmonised and a shared interpretation of test results needed to be provided to the developers.
- Business strategy for marketing and exploitation were executed more effectively due to the DIVERCITY tests. To be precise, business strategy and exploitation were not clear enough before the tests started. The DIVERCITY tests enabled the project team to clearly define the business strategy for marketing and exploitation of the system.
- As a result of the testing approach, the followings, which were already proved by researchers and engineers in the software testing community, were confirmed by means of the DIVERCITY tests.
  - Initially, testing started early in the development lifecycle, and
  - Tests were conducted iteratively and test teams continuously interacted with each other and the developers. These led us to close the quality gap in the focused testing objectives.
  - Possible risks were identified and reduced earlier. Repairs to problems were less costly than repairs at the closing stages. Project was completed on time as desired.
  - There was an iterative process between the requirements capture and software testing. Many of the detailed requirements, which were ill understood during the requirements capture, became clearer, and expanded during the testing.

## **7. FUTURE STUDY**

Despite a constant improvement in the implementation of the user requirements techniques in the CIC developments, the level of guidance for user driven and also industry driven CIC development is still very low. A more effective guidance to requirements engineering staff need to be determined, which enable to deal with a large variety of situations.

It is noteworthy to highlight the high similarities between the results from DIVERCITY and the results of Macaulay research study (1996), which identified three common type failure; process failure, expectation failure and interaction failure and five different causes that result in failure. These are; (i) the requirements engineering process itself, (ii) human communication within requirements engineering process, knowledge development, documentation of requirements and management. The evaluation results of the DIVERCITY requirements engineering activities can be clustered in these five groups easily. Furthermore, these five causes are also much related to the key issues and criteria identified in the research.

The main purpose, afterwards, is to link the DIVERCITY requirements engineering activities into a process framework that can be used a reference for driving concrete requirements engineering activities in the CIC development. By remedying the above causes in the DIVERCITY requirements engineering process using specific techniques and approaches in the light of the strengths and weaknesses identified by means of the evaluation according to the identified criteria. Therefore, in the requirements engineering framework to be proposed will describe how to do the activities as well as what to do them. Lastly, because the empirical validation is not possible within the scheduled research period, the theoretical evaluation and validation will only be conducted.

## 8. CONCLUSIONS

In this paper, the DIVERCITY project and the requirements approach in the DIVERCITY project are described. It included mainly the contextual design and storyboarding, UML approach, incremental prototyping with the end users tests. Furthermore, the need and importance of the requirements engineering is addressed in some respects such as the awareness of people about the CIC systems and requirements engineering, and its contribution to the implementation in the construction industry and so on.

Before proposing a generic requirements engineering process, it is required to analyse and find out the strengths and weaknesses of the requirements engineering approach taken in the DIVERCITY project. To do so, the analyses approach is explained. In addition, some issues are also highlighted from the results of the analysis. Lastly, after identifying the strengths and weaknesses, it is required to improve the requirements engineering process and make it ready-to-use for the future system developments. Therefore, mastering the requirements engineering process is aimed at future study.

## 9. REFERENCES

Aish, R., (1999) "*Migration from an Individual to an Enterprise Computing Model and its Implications for AEC Research*", Berkeley-Stanford CE & M Workshop: Defining A Research Agenda, Stanford, 1999.

Alshawi, M. (1996) "*SPACE: integrated environment*" Internal Paper, University of Salford, July 1996.

Aouad, G., Marir F., Child T., Brandon P. and Kawooya A. (1997) "*Construction Integrated Databases- Linking design, planning and estimating, The OSCON approach*";

International Conference on the Rehabilitation and Development of Civil Engineering Infrastructures. American University of Beirut, pp 52-60, June 1997

Aouad G., Hinks, J., Cooper, R., Sheath M.D, Kagioglou, Sexton, M., “An IT map for a generic design and construction process protocol”, *Journal of Construction Procurement*, 1998.

Beyer H, Holtzblatt K, 1998, *Contextual Design. Defining Customer-Centered Systems*. Morgan Kaufmann Publishers, San Francisco. (472 pp).

Boehm, B.W.,(1996) “*Anchoring the Software Process*”, *IEEE Software*, July 1996.

Carr, J. “Requirements Engineering and Management: the key to designing quality complex systems, *The TQM Magazine*, Volume 12, Number 6, 2000, pp 400-407, MCB University Press, ISSN 0954-478X

CHAOS Report, 1995, Software development report, The Standish Group, published at [www.standishgroup.com](http://www.standishgroup.com)

CHAOS Report, 2000 The Software development report, The Standish Group, published at [www.standishgroup.com](http://www.standishgroup.com)

Christiansson, P, 2001, "Capture of user requirements and structuring of collaborative VR environments". AVR II & CONVR 2001. Conference on Applied Virtual Reality in Engineering & Construction Applications of Virtual Reality. (eds: O. Tullberg, N. Dawood, M. Connell. 201 pp.) Gothenburg October 4-5, 2001. (pp. 1-17).

Christiansson, P., Svidt, K., Skjaerbaek, J.O., Aalholm, R., (2001), “User requirements modelling in design of collaborative virtual reality design systems”, CIB W78 - Information Technology in Construction - Mpumalanga, South Africa, 30th May - 1st June 2001

Dearden, A., Howard, S., 1998, *Capturing User Requirements and Priorities for Innovative Interactive Systems*, Australasian Computer Human Interaction Conference, November 29 - December 12, 1998, Adelaide, S. Australia

Emam, K., Madhavji, N., 1995, “Requirements Engineering problems in Practice”, Second IEEE International Symposium on Requirements Engineering, York, 1995, IEEE Computer Society Press.

Faraj I. and Alshawi M. (1999), *A Modularised Integrated Computer Environment for the Construction Industry: SPACE*, 1999, University of Salford, UK, <http://www.itcon.org/1999/3/>

Grassi, M.D., Zorgno, A.M., (1999) “*The Virtual Construction Enterprise*”, Berkeley-Stanford CE&M Workshop, Stanford 1999.

*Grunbacher, P., Briggs B., Surfacing Tacit knowledge in Requirements Negotitation: Experiences using EasyWinWin, in proceedings Hawaii International Conference on System Sciences, IEEE Computer Society, 2001*

Keil, M., Cule, P.E., Lyytinen, K., Schmidt, R.C., 1998, "A Framework for Identifying the Software Projects Risks" Communication of the ACM, vol. 41, pp 76-83.

Kiviniemi, A., Laitinen, J., Lautanala, M., (1999) "*Defining a Research Agenda for AEC*", Berkeley-Stanford CE&M Workshop, Stanford 1999.

Lewis, W. E., 2000, *Software Testing and Continuous Quality improvement*, Software Testing Associates, Inc. Plano Texas, 2000

Macaulay, L., (1996), Requirements for Requirements engineering techniques, Second International IEEE Conference on Requirements Engineering, pp 157-164, Proceedings of ICRE'96.

Maguire, M., (1996), Prototyping and evaluation guide, HUSAT Research Institute, the Elms, Elms Grove, Loughborough, Leicestershire, LE11 1RG, UK, October 1996.

Nikula, U., Sajaniemi, J., 2002, A Lightweight Combination of Proven RE Techniques" IEEE Joint International Requirements Engineering Conference, 2002, Essen Germany.

*Rational white paper, 1999, "the rational Approach to Automated Testing", USA,*  
<http://www.rational.com/products/whitepapers/100581.jsp>.

Regnell, B., Kimbler, K., Wesslen, 1995 A., *Improving the Use Case Driven Approach to requirements Engineering*, Proceedings of Second IEEE International Symposium on requirements Engineering, York, UK, March 1995

Sawhney, A., (1999) "*Research and Development Plan for the AEC Industry*", Berkeley-Stanford CE&M Workshop, Stanford 1999.

Shelbourn, M., Sarshar, M., Aouad, G. (2001) "*DIVERCITY, A Construction Industry Case Study of User Requirement Capture*", *Proceedings of ninth IEEE International Symposium on Requirements Engineering, Toronto, Canada, 2001.*

Sun, M., and Aouad, G., (2000), Integration Technologies to Support Organisational Changes in the Construction Industry", 7<sup>th</sup> ISPE International conference on Concurrent Engineering, Lyon, France, 17-20 July 2000, PP. 596-604.

Sutcliffe A., 2001, Requirements engineering for complex collaborative systems, IEEE Joint International Requirements Engineering Conference, 2001