J. Basic. Appl. Sci. Res., 6(9)26-37, 2016

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

A Usability Oriented Software Development Process with Constant Usability Evaluation

Muhammad Awais Gondal[1], Majed Aadi Alshamari[1], Hafiz Farooq Ahmad[1] and Khalid Latif[2]

[1]College of Computer Sciences and Information Technology (CCSIT),
King Faisal University, Alahssa 31982, Kingdom of Saudi Arabia.
[2]School of Electrical Engineering and Computer Science,
National University of Sciences and Technology, Islamabad, Pakistan.

Received: June 28, 2016
Accepted: August 25, 2016

ABSTRACT

Usability has become an important aspect of software applications and it is gaining importance increasingly. A number of approaches have been presented to develop usable applications. However, their main focus is on usability tasks and not on software engineering principles, resulting in failure to adopt them. Moreover, usability evaluation which is very critical for developing user-friendly applications is also neglected by usability models. This paper proposes a complete software development process that aims to achieve higher standards of usability. It focuses on constant usability evaluation and eliminates usability issues early in the development process. The proposed process is close to general software development practices which helps include usability aspects in software applications in a seamless way. We have evaluated the effectiveness of our approach through different case studies, and the results show that it helped in handling most of the usability issues and requirement changes, in early design phase. Moreover, the projects which followed our approach attained better levels of usability and user satisfaction.

KEY WORDS: Usability, Software Development Process, Usability Evaluation, User-centered Design

1. INTRODUCTION

The concept of usability in software applications is not new and has its roots back in 1980's. The field got maturity in 1990's and in the last two decades it has gained adequate significance. Demand for more friendly, easy to learn and usable software has increased.

However, usability has not been adopted as an essential part by software organizations and engineers. The major reason is that software engineers tend to follow defined software development processes that lack concepts of usability. The approaches that consider usability as an essential part of software applications have their focus on usability activities only and there are not sufficient details for software engineers to adopt them. Moreover, these approaches are difficult to understand which adds in the reasons that make it hard to develop software applications using them. Additionally, usability evaluation which is considered very important for achieving better standards of usability, is also neglected in general development models.

The main contribution of this paper is the simplified representation of traditional usability oriented development processes and their modification for use by typical software development units. In this paper, we present a software development process that integrates usability tasks and evaluation methods in a general arrangement of software development activities. Usability tasks included in the process are close to general practices of software engineers that makes it an easily adoptable option. In the later sections of this paper, we have discussed the details of the proposed method.

It is aimed that through our proposed approach, development units will be able to handle usability issues at early stages of software development. It will be easier for software engineers to adopt our process, than adopting other usability oriented approaches. In addition to this, better levels of usability can be achieved by using our proposed method as a software development process.

First, we discuss the concepts of usability and its various definitions. Than we present an account of different approaches and methods found in literature, which focus on achieving usability. We also present different methods that are used to evaluate usability of software applications. Afterwards, we present our method and provide details of its different phases. Evaluation of the approach using different case studies is also included in this paper.

Corresponding Author: Majed Aadi Alshamari, College of Computer Sciences and Information Technology (CCSIT), King Faisal University, Alahssa 31982, Kingdom of Saudi Arabia.

Usability has been defined by a number of standards and studies [1, 2], which consider it as a software quality attribute and specify certain characteristics of it. ISO 9241-11 defines usability as, "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use [3]." IEEE standard 1061 is related to Software Quality Metrics and defines usability as, "Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component [4]." Jakob Nielsen, a prominent usability expert, defines usability as a quality attribute that depends on five components; Learnability, Efficiency, Memorability, Errors and Satisfaction [5]. Other components of usability can also be found in literature, which include utility, safety, accessibility and some others, and they may vary according to the project.

Various process models and approaches for achieving usability can be found in literature. Here we present a number of such approaches and processes. A detailed review of existing models for attaining usability helped us in presenting a suitable usability development model relevant to the needs of software engineers. The basic issue that was found after the survey was that most of the models and approaches are considered as new paradigms for developing software applications. However, to help software development units in adopting usability activities in the process, it is better to offer a method that is there in operation. Rather than proposing a totally new model, presenting usability oriented approach that is closer to general software development process is a better idea.

2.1 ISO 13407: Human-centered design processes for interactive systems

ISO 13407 can be considered as a basis for other usability focused development models [6]. It gives a very general representation of an iterative approach to be followed for developing systems with a focus on usability. It is based on a repetitive process where context of use and requirements are specified first, followed by producing and evaluating multiple design solutions, until the requirements are met. ISO 13407 can be used as a baseline with other development processes; however, it is not possible to adopt it alone as a complete development process.

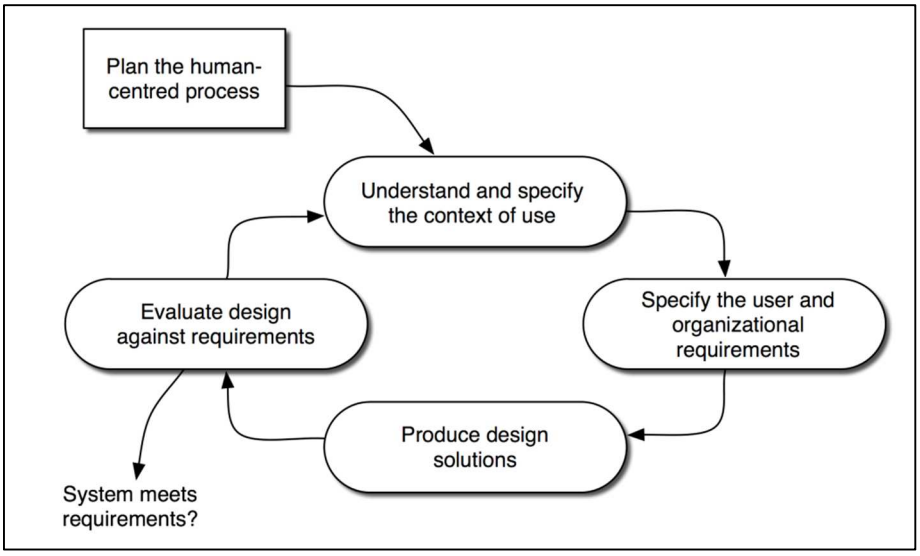

Figure 1: Human-centered design processes for interactive systems (ISO 13407)

2.2 Usability Design Process

Usability Design Process (UDP) follows the principles of User-Centered Design and has three main phases [7] including requirements analysis, growing software with iterative design and deployment. Requirements Analysis focuses on elicitation of business goals and documenting them. Users are analyzed on the basis of their skills, knowledge and abilities. In growing software phase, the system is designed and grown in three iterative loops as mentioned by User-Centered Design i.e. conceptual design, interaction design and detailed design.

An important concept that we have borrowed from Usability Design Process is making of Usability Design Guide containing UI specification and guidelines for achieving usability. We also advocate UDP's flow of activities that supports different levels of designing with a focus on usability.

It has been admitted that UDP is not a suitable development process [7] as it focuses mainly on concepts from usability engineering which are difficult to understand for software engineers. It has been

suggested that this process should be framed in a more particular and detailed representation. Our approach adopts the basic structure of UDP; however, we have included activities that can be more easily adopted by general software development units.

2.3 Usability Evaluation

Usability Evaluation methods and techniques are used to determine the extent to which a software is usable. These methods are usually divided in three main categories; testing, inspection and inquiry [8], as discussed below.

Usability Inspection: In inspection techniques, professionals examine a system and evaluate the extent of its usability. The professionals can be trained usability experts, software developers or software quality engineers. Cognitive walkthrough and feature inspection are the examples of usability inspection. Heuristics evaluation is also an inspection technique where evaluators/experts evaluate a system against a predefined set of heuristics/principles. Jakob Nielsen's famous 10 heuristics are widely used for this purpose [10], however, other heuristics set should also be studied to choose most appropriate one for the product being developed. In our approach we have focused on the issue that the 10 famous heuristics of Jakob Nielsen are not always helpful [9], so separate attention should be given to select a suitable set.

Usability Testing: In these approaches, users use the system (or prototypes) according to defined tasks or procedures, and the evaluators assess the data and results from this specific usage. Coaching method and question asking protocol are examples of this category. Performance measurement is another way of usability testing which provides quantitative data that can be used to assess usability goals. We consider this method essential for the assessment of usability goals once the product is in operation.

Usability Inquiry: In these methods, users are inquired about the usage experience regarding a system to get information about usage satisfaction, usability problems and requirements etc. Users may also be involved in answering a set of different questions related to usability of the system. For usability inquiry, some standard questionnaire sets are there which provide a good indication of user's experience and satisfaction. We have used System Usability Scale (SUS) and Single Ease Question (SEQ) for inquiring the users about case studies used in evaluation of the proposed methodology.

3. **METHODOLOGY**

In this section, we propose our representation of a usability oriented software development process. It is based on the structure presented by Usability Design Process and involves the concepts listed in User Centered Design. The main theme of this representation is integration of usability engineering (UE) and software engineering (SE) activities with constant usability evaluation throughout the process. This reduces the chances of usability issues in later stages as well as after different levels of the process. Usability evaluation and improvement tasks are split over phases of Requirements Analysis, Design, Development, Validation and Deployment. These are usually used by general software development houses, thus it is aimed that the representation cab be adopted without much difficulties. Instead of using usability related concepts and terms, we have tried to use terms that are more easily acceptable by software developers. Moreover, a complete development process cycle is there that guides development units throughout the process of development. This section provides details of different activities of our proposed representation on the basis of its salient features and characteristics.

A general representation of the proposed approach has been shown in Figure 3 and notation used for the representation is shown in Figure 2. The process starts with analyzing user's requirements and extracting usability requirements. Design, development and validation are merged in the next phase that can be iterated as well. The most critical and important stage is the early design phase where most of the usability issues and changes in requirements are uncovered. We have used a very simple representation here so that software developers and units can easily adopt it. Instead of using purely usability engineering related notions, we use 'workflow' and 'prototyping' terms that are familiar to all members of software development teams.

3.1 Requirements Analysis Phase

The first phase deals with user requirements, their tasks and goals on the basis of which user profiles and usability goals are determined. These activities are very much related to requirements analysis phase of other usability processes, however, we strongly recommend doing a competitive analysis to gain knowledge about applications of similar domain and genre. This helps in setting high standards of usability as well as getting guidelines for a better design. We support specifying the requirements through use cases as they are from user's viewpoint and are better understood by software engineers as well.

The documented artifacts of this phase include user profiles, usability goals, requirements specification and heuristics specification documents. A heuristics specification document should contain specific heuristics to

be followed and it should also describe which heuristics should be evaluated in design stage and after development. This set of heuristics should be selected according to the domain of software being developed. It has been proved through different studies [9] that Jakob Nielsen's famous heuristics set [10] is not always helpful in every domain. A number of new sets of usability heuristics have been presented for specific domains like healthcare information technology [11], video games [12], augmented reality applications [13] etc. Moreover, for creating heuristics set for specific domain or project, a methodology has been presented in [14], which suggests defined stages for establishing usability heuristics. So, we recommend that heuristics according to specific domain should be selected and analyzed accordingly in evaluation process.

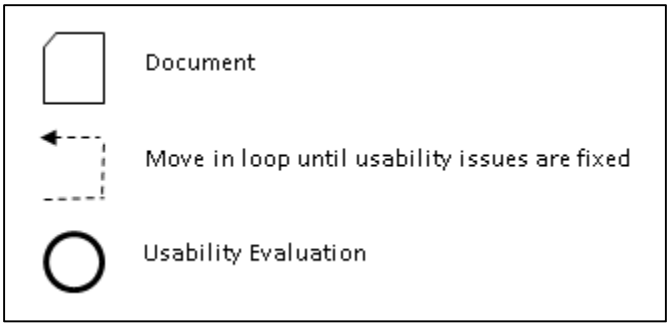

Figure 2: Notation used for proposed approach

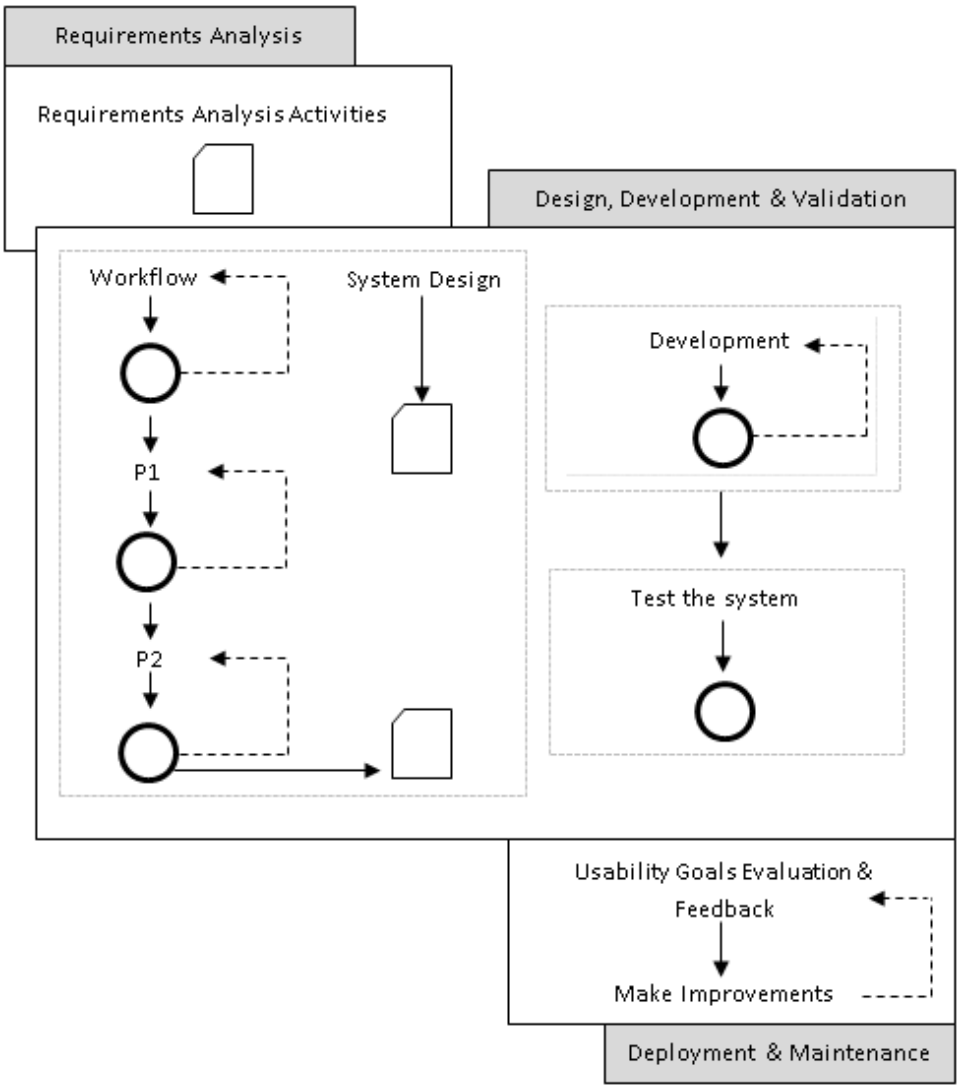

Figure 3: General representation of proposed approach

3.2 Design Phase

The design phase is divided into two sub phases which are UI design and system design, the former of these is user focused while system analysts drive the latter which can be executed side by side with UI design if possible. Here, we discuss UI design phase only and consider it as most critical in the whole process. This is where most of the usability issues are uncovered and fixed at a very early stage. Moreover, this phase helps in dealing with changing requirements in an efficient way. Requirements phase can be revisited with the changes in requirements during this phase.

3.2.1 UI Design Phase

Initially, a workflow of the system is designed which represents an overall concept of navigation. It is important to specify a workflow at this stage as users have predetermined ways of performing certain tasks. A workflow can be created on paper which can be evaluated by users through usability walkthrough or any other evaluation technique. Users go through the paper based navigational flow to identify any problems in it, and designers fix them accordingly. It should be assured that before moving forwards an overall workflow of events should be finalized with the user.

The next step is to design Prototype 1 (P1) on the basis of user requirements, usability goals and defined navigation. The main objective of P1 is to uncover maximum requirements anomalies and new user needs. It also helps in uncovering some basic usability problems related to on-screen text, general layout and screen components. We recommend using paper mockups for P1 as they take less time and are easy to modify in evaluation sessions with the user. For usability evaluation of these prototypes, we recommend using 'Coaching Method' where users attempt to perform tasks by using the mockups and ask questions from UI designers or usability coaches. Usability experts or UI designers analyze the users and determine the areas which are problematic.

Once Prototype 1 has been sufficiently improved with user feedback and usability evaluation, Prototype 2 (P2) is to be created. P2 focuses on detailed aspects of user interface where working prototypes are constructed that represent the actual software application to be developed. Representative data should also be used to get actual flavor of usage. In addition to Coaching Method, Feature Inspection can be carried out for evaluation of P2, to check organization of different features of the application, their understandability, friendliness and other aspects. The main objective of P2 is to uncover as many usability issues as possible. These are spread over multiple aspects like colors, size of elements, data representation, user experience, and behavior and so on. Changes in requirements can still be there, but case studies used for evaluation show that the number of changes decreases as we proceed in the process.

As mentioned earlier, the UI design phase and continuous usability evaluation involved in it, ensures that major usability issues are screened out at a very early stage. User-focused evaluation techniques confirm constant user involvement. This also helps in creating real expectations from the software application being developed and makes learning process easy once the software is deployed. A user interface specification document is created at the end of this phase which contains workflows and high fidelity prototypes, to be used in later stages of development. This document combined with artifacts from requirements analysis phase collectively make the usability design guideline.

3.3 Development Phase

The development phase focuses on system design and usability design document for developing the product according to usability goals and requirements. It is recommended that development should be carried out in various iterations after each of which formative usability evaluation must be carried out, in order to ensure that development phase is leading towards the right way. Any problems found in formative evaluation should be fixed before moving towards the next iteration.

3.4 Validation Phase

Throughout the process usability is evaluated by a number of techniques, however, it is still necessary to assess whether the product met usability goals and is free of any possible issues. A summative usability evaluation has to be carried out in this phase with an objective of evaluating usability goals and requirements. Moreover, according to heuristics specification document (created in requirements analysis phase), usability features that are to be evaluated in the developed application should be analyzed at this stage. Furthermore, other software testing activities are also conducted to find possible software bugs. Issues related to usability and others are fixed and the software is deployed after validation.

3.5 Deployment and Maintenance Phase

Deployment can be done in incremental way or otherwise, depending upon the nature of the product. Usability evaluation and improvement should not be neglected even after deployment. We suggest performance measurement techniques for this phase where error rate, task time, learning difficulties etc. should be examined and the product should be improved accordingly. User feedback and satisfaction measurements help in improving the product likewise. In case of users who have been directly involved in UI design and usability evaluation, it gets easy to use the software in better way. Moreover, already developed realistic expectations result in higher levels of satisfaction.

4. RESULTS

In this section, we present the case studies in which our proposed approach was implemented. Multiple software development projects were taken as scenarios, in order to present reliable and concrete results. To prove the effectiveness of proposed approach, different perspectives of the process were tested across these projects.

4.1 Case Studies

The case studies were divided in to three categories:

Full Process (FP): the projects which used our approach and results were gathered from them in all phases of the process.

Partial Process (PP): the projects which used our approach, but the results were gathered only up to design phase.

Null Process (NP): the projects which did not use our approach.

Personal Health Record Insurance Module (PHR): An application for insurance plan providers to manage offered insurance services. General users can select an insurance plan according to their specific requirements.

Home Health Agency (HHA): An application for nurse practitioners that provide health services at patient's home.

Electronic Prescribing Application (sRx): An application for doctors to create and manage prescriptions.

Restaurant Booking Application (RBA): A smart phone application for booking tables in restaurants and place orders.

Same projects (having same requirements specification) were developed using NP as well as FP or PP in order to present comparable results. Table 1 shows overview of projects used in case studies.

Table 1. Overview of case study projects

Project	Process Category	Platform
PHR Insurance Module (PHR-NP)	NP	Windows
PHR Insurance Module (PHR-FP)	FP	Windows
Home Health Agency (HHA-FP)	FP	iOS
Electronic Prescribing Application (sRx-PP)	PP	Android
Restaurant Booking Application (RBA-NP)	NP	Android
Restaurant Booking Application (RBA-PP)	PP	Android

4.2 Variables

This section presents the variables against which data was captured during the development of mentioned software projects. Different aspects and characteristics were recorded to cover all aspects of the hypothesis. Following are the variables that were observed and measured in the subject projects.

4.2.1 Usability Problems

Usability problems indicate the quality of a software application. The finished product should contain only a few or ideally no usability problems. This can be attained if usability problems are uncovered and eliminated earlier in development process. The records of usability problems, their description, number of occurrences and stages at which they are found will help us in determining that either or not the proposed process helps in uncovering usability process in earlier stages of project development.

4.2.2 Changes in Requirements

Stable requirements are considered ideal in software development; however, changes in requirements are always there in almost every software project. These changes have a direct effect on cost, quality and schedule. As the hypothesis proposes that the process will help in identifying changes in requirements, so recording the changes with respect to the stages at which they are identified will help in analyzing the hypothesis.

4.2.3 Development Time

The proposed process focuses on spending time in usability related activities throughout the process. Participants were requested to carefully record the time they spent on different activities of the process. Recording the time spent in usability activities as well as in the overall process will help in investigating the effect of usability activities on overall development time. It will help on determining that either or not an early usability focus helps in reducing overall development time.

4.2.4 Complexity of proposed approach

For measuring the complexities of particular tasks involved in the process, and the whole process itself, we asked the participants who used the process in their projects. Participants had to rate the complexity of the process and its particular tasks on a scale of 1-5.

4.2.5 Usability Level

To evaluate usability of the projects included in the evaluation of this thesis, we used Single Ease Question (SEQ) for task level satisfaction, and System Usability Scale (SUS).

Single Ease Question (SEQ): Single Ease Question helps in determining how easy or difficult it is for users to perform certain tasks while using a system. It is a scale that focuses on different levels of difficulties associated with tasks. Although, SEQ is a simple method for measuring usability of a system, yet it is found to be equally well or better [15, 16] than other standard questionnaires for measuring task difficulty.
System Usability Scale (SUS): System Usability Scale was originally developed by John Brooke [17], which is a ten points questionnaire that evaluates effectiveness, efficiency and learning aspects of a system. This scale can be used on small sample sizes and is considered as reliable and valid [18]. Comparisons can be made on the basis of SUS scale, so it is definitely helpful in comparing usability levels of subject projects.

4.3 Data

This section presents the data captured during the development of the subject projects. The data was captured according to the variables discussed in the previous section. In order to evaluate usability in different phases of the proposed process, at least five (5) users participated in the process activities, as these much users are sufficient according to Jakob Nielsen's mathematical model of usability problems [19].

For usability problems, changes in requirements and development time, comparable results were required, so this section shows the data related to these variables in general software development phases i.e. Analysis, Design, Development, Testing and Deployment. However, it must be noted that usability problems and changes in requirements were uncovered during specific evaluation activities workflow design, Prototype 1 & 2 etc. Moreover, for projects with PP category, as the data was recorded up till design phase only, records for development, testing and deployment phases are not applicable.

4.3.1 Usability Problems

Tables 2 and 3 respectively indicate the number of problems according to particular applications and with respect to the process used. Usability problems, for the projects which used our proposed process, were uncovered mostly in design phase and a few in later stages. However, usability problems were uncovered at late stages for the projects which did not use the proposed approach.

Table 2. Number of usability problems in different case studies

Phase of development	PHR (NP)	PHR (FP)	HHA (FP)	sRx (PP)	RBA (NP)	RBA (PP)
Analysis	0	0	1	0	0	0
Design	0	7	8	7	2	6
Development	1	2	3	-	5	-
Testing	1	0	1	-	7	-
Deployment	4	0	1	-	4	-

Table 3. Number of usability problems with respect to process used

Phase of development	NP	PP	FP
Analysis	0	0	1
Design	2	13	15
Development	6	-	5
Testing	8	-	1
Deployment	8	-	1

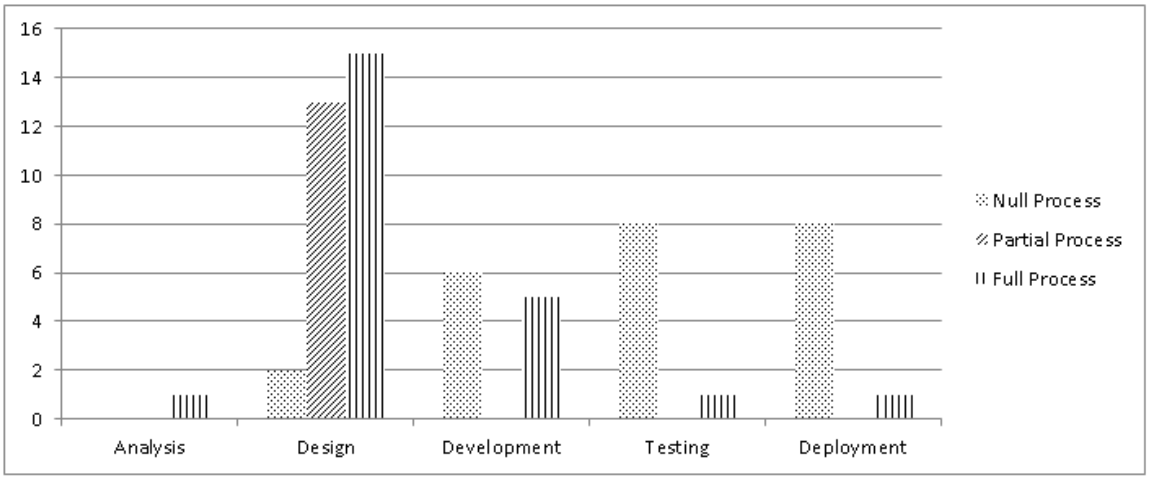

Figure 4: Usability problems with respect to process used

4.3.2 Changes in requirements

Table 4 and 5 respectively indicate the number of problems according to particular projects and with respect to the process used. Like usability problems, changes in requirements for the projects which used the proposed approach where uncovered earlier as compared to the projects which did not use the process.

Table 4. Number of changes in requirements in different case studies

Phase of development	PHR (NP)	PHR (FP)	HHA (FP)	sRx (PP)	RBA (NP)	RBA (PP)
Analysis	1	1	4	2	4	2
Design	0	8	4	2	1	6
Development	1	1	3	-	6	-
Testing	4	1	4	-	9	-
Deployment	3	0	1	-	3	-

Table 5. Number of changes in requirements with respect to process used

Phase of development	NP	PP	FP
Analysis	5	4	5
Design	1	8	12
Development	7	-	4
Testing	13	-	5
Deployment	6	-	1

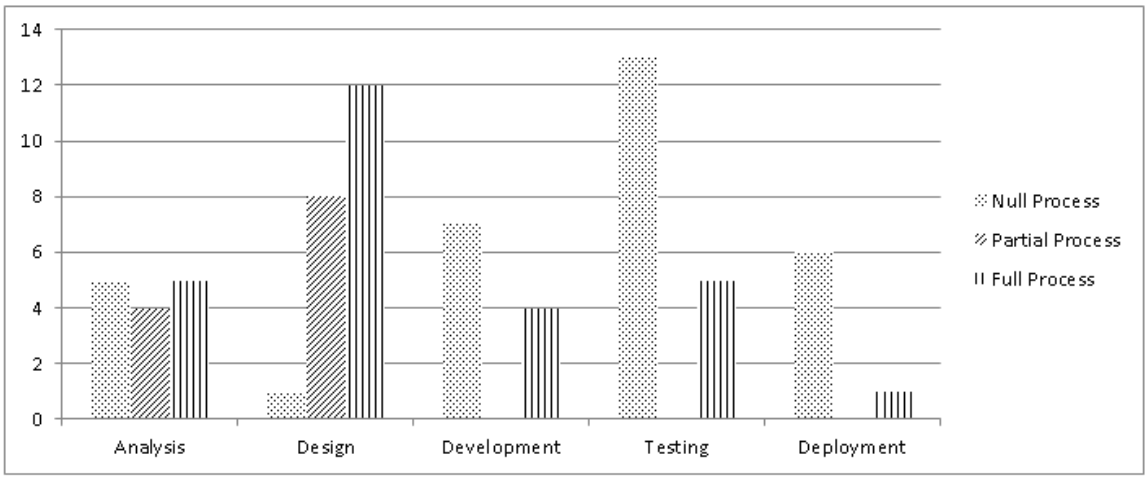

Figure 5: Changes in requirements with respect to process used

4.3.3 Development Time

Table 6 shows the expected and actual time spent in different case study projects. Expected time was calculated initially by project lead/manager. Although, additional time was spent in design phase for the projects which used the process, yet the projects completed in expected time. Significant time was saved in development as major problems and changes in requirements were already dealt with. Development teams did not face problems like revisiting already completed tasks, due to client feedback and criticism. In Table 6, only the projects with complete data are shown, as for the projects of PP category data was collected up till design phase only.

Table 6. Development Time *(in Man Hours)* of different projects

Project	Expected Time	Actual Time
PHR (NP)	64	62
PHR (FP)	64	58
Home Health Agency (FP)	233	217
Restaurant Booking (NP)	368	534

4.3.4 Complexity of proposed approach

Table 7 shows the average complexity of the overall process and different tasks performed by participants. The data is from rating of different individuals who participated in developing case study projects. Individuals were guided on performing certain tasks like field study, task analysis and heuristics evaluation etc. A five point scale was used for rating the complexity with 1 for 'Very Difficult' and 5 for 'Very Easy'. The results show that participants did not find the overall approach as a difficult one. Average complexity of tasks is 3.60 and average complexity of overall process got a rating of 3.70 from participant's feedback.

Table 7. Complexity of proposed approach
1 - 'Very Difficult', 5 - 'Very Easy'

Task	Average Complexity
Field study	3.10
Task analysis	3.00
Competitor analysis	3.10
Designing workflow	4.40
Evaluating workflow	4.10
Designing Prototype 1	4.40
Evaluating Prototype 1	3.70
Designing Prototype 2	4.10
Evaluating Prototype 2	3.10
Development according to usability guideline	3.70
Heuristics evaluation	3.60
Getting user feedback	3.40
Average complexity of tasks	3.64

4.3.5 Usability level

To evaluate usability of the projects included in evaluation, we used Single Ease Question (SEQ) for task level satisfaction, and System Usability Scale (SUS) for an overall usability score. Table 8 shows the average task level satisfaction calculated from SEQ for different projects. This data has also been represented in Figure 6. Table 8 shows that the projects for which our approach was used, task level satisfaction was higher as compared to other projects. Moreover, it was also up to an acceptable level of satisfaction as average SEQ score is around 4.8 to 5.1 [20].

Table 9 shows the average score of System Usability Scale obtained from user feedback of different projects. SUS scores of case study projects have also been represented in Figure 7. It is clear that projects using our approach scored better than others as well as better than the average score of SUS that is 68 [18].

Table 8. SEQ scores

Project	SEQ score
PHR (NP)	4.55
PHR (FP)	5.67
Home Health Agency (FP)	5.90
Restaurant Booking (NP)	4.10

Table 9. SUS scores

Project	SUS score
PHR (NP)	51.25
PHR (FP)	69.25
Home Health Agency (FP)	81.00
Restaurant Booking (NP)	59.50

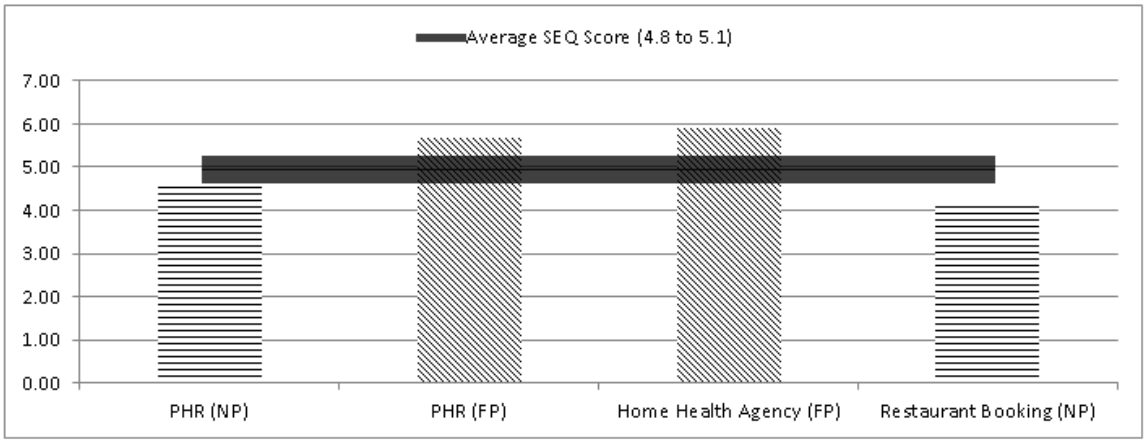

Figure 6: SEQ score of case studies

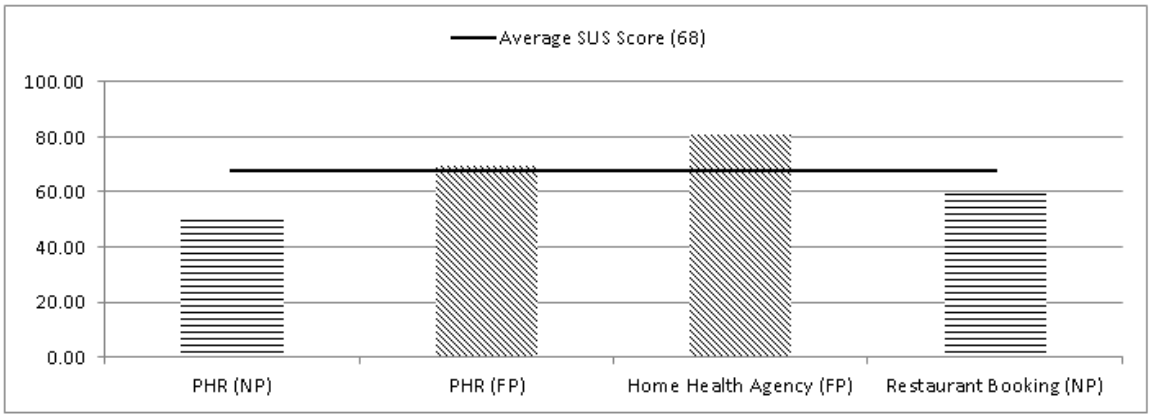

Figure 7: SUS score of case studies

5. DISCUSSION

In the previous section, data captured from different case studies has been presented. The data helped in analyzing the proposed hypothesis of this work. The analysis of usability problems shows that the adoption of proposed process helped in fixing usability problems in early design phase. For these projects, most of the usability problems were uncovered and fixed in design phase and a very few usability problems were left in later stages. However, usability problems were addressed very late in the projects which did not apply the proposed approach. Deployed software applications were also left with usability problems in cases where the process was not used.

The projects which followed the proposed process were not troubled by changes in requirements at later stages of development. Most of the changes in requirements were uncovered in the design phase for these projects. However, the projects which did not follow the proposed process had to deal with changes in requirements in later stages of software development as well as after deployment.

The participants of different case studies who used the proposed process have rated the complexity of the process and the tasks they performed. If we analyze the feedback of the participants it can be concluded that in general the participants did not find the process much difficult as the average rating is 3.70 (1- Very Difficult and 5 - Very Easy), which is clearly towards easy side. However, it can be analyzed from the data that performing some specific activities like task analysis, competitor analysis, field study and getting user feedback were considered as slightly complex. We can realize that certain professionals are required for performing such tasks, and only general software developers might face difficulties in doing such activities.

The relation of adoption of the proposed process with development time can also be analyzed from the presented data. Although, it seems that additional usability tasks involved in the process may impact the overall development time. However, results from case studies show that even if the participants spent additional time in performing usability activities earlier in the process, yet the overall development time was not affected. The major reason behind this is the time saved in development phase. The subject projects which did not apply the proposed process had to suffer from requirement changes and other issues in later stages of project development, which led to additional time spent.

If we analyze the results obtained from standard usability questionnaires, we can conclude that the case studies, for which the proposed process was used, got higher levels of usability and of user satisfaction at task level. The projects that did not apply the proposed process got significantly low levels for these metrics. The results advocate that using the proposed representation can help in achieving better usability.

Through software development case studies, we have proved that the presented approach helps in handling usability issues and changes in requirements in early stages of development. Moreover, it can be adopted to develop software applications with better levels of usability. Individuals, who used the proposed process in their software projects, have rated it as quite easy and have not overall spent additional time for completing their projects.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a complete software development process model that aims at increasing usability level of software applications. The approach has been applied on different case studies, and results prove that the presented development model is successful in handling usability problems and requirement changes in early development phases. In addition, the proposed approach also helps in attaining better usability levels.

The paper also opens different avenues for future research work. Few variables were selected for evaluation of this work; however, in future we intend to analyze relation of the proposed approach with more variables related to software development projects. Possible examples of such variables are size of teams, domain of projects, application platform and some others.

From the obtained results an indication has been received that certain tasks in the process might require specific professionals in order to perform them in a better way. A study on structure of teams required for this process may be of great assistance in practical implementation of the proposed process in software development industry.

Different usability evaluation methods were discussed in this paper, however, it was not specified that which method is most suitable for specific tasks in the process. Effect of different usability evaluation methods with respect to process tasks should be studied and it should be advised that which usability evaluation technique is most suitable for particular tasks.

Moreover, presenting ways to modify the proposed approach for software applications that cannot be anticipated in early design process is another future research line. One of the examples of such applications is the video games for which interactive user interface cannot be actually experienced through prototyping. One such approach has been presented in [21] to ensure usability, playability and effectiveness of video games.

REFERENCES

1. Abran, A., Khelifi, A., Suryn, W., & Seffah, A. (2003). Usability meanings and interpretations in ISO standards. *Software Quality Journal*, *11*(4), 325-338.
2. Quesenbery, W. (2001, October). What Does Usability Mean: Looking BeyondEase of Use'. In *Annual Conference-Society for Technical Communication* (Vol. 48, pp. 432-436).
3. ISO, W. (1998). 9241-11. Ergonomic requirements for office work with visual display terminals (VDTs). *The international organization for standardization*.
4. Geraci, A., Katki, F., McMonegal, L., Meyer, B., Lane, J., Wilson, P., & Springsteel, F. (1991). *IEEE standard computer dictionary: Compilation of IEEE standard computer glossaries*. IEEE Press.
5. Nielsen, J. (2003). Usability 101: Introduction to Usability. Jakob Nielsen's Alertbox.
6. Iso, I. S. O. (1999). 13407: Human-centred design processes for interactive systems. *Geneva: ISO*.
7. Göransson, B., Gulliksen, J., & Boivie, I. (2003). The usability design process–integrating user-centered systems design in the software development process. *Software Process: Improvement and Practice*, *8*(2), 111-131.

8. Whitefield, A., Wilson, F., & Dowell, J. (1991). A framework for human factors evaluation. *Behaviour & Information Technology*, *10*(1), 65-79.
9. Masip, L., Granollers, T., & Oliva, M. (2011, April). A heuristic evaluation experiment to validate the new set of usability heuristics. In *Information Technology: New Generations (ITNG), 2011 Eighth International Conference on*(pp. 429-434). IEEE.
10. Jakob Nielsen. 10 usability heuristics for user interface design, 2013. http://www.nngroup.com/articles/ten-usability-heuristics/
11. Belden, J. L., Grayson, R., & Barnes, J. (2009). *Defining and testing EMR usability: Principles and proposed methods of EMR usability evaluation and rating*. Healthcare Information and Management Systems Society (HIMSS).
12. Pinelle, D., Wong, N., & Stach, T. (2008, April). Heuristic evaluation for games: usability principles for video game design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1453-1462). ACM.
13. Ko, S. M., Chang, W. S., & Ji, Y. G. (2013). Usability principles for augmented reality applications in a smartphone environment. *International Journal of Human-Computer Interaction*, *29*(8), 501-515.
14. Rusu, C., Roncagliolo, S., Rusu, V., & Collazos, C. (2011, February). A methodology to establish usability heuristics. In *ACHI 2011, The Fourth International Conference on Advances in Computer-Human Interactions* (pp. 59-62).
15. Sauro, J., & Dumas, J. S. (2009, April). Comparison of three one-question, post-task usability questionnaires. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1599-1608). ACM.
16. Tedesco, D., & Tullis, T. (2006). A comparison of methods for eliciting post-task subjective ratings in usability testing. *Usability Professionals Association (UPA)*, *2006*, 1-9.
17. Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry*, *189*(194), 4-7.
18. Sauro, J. (2011). Measuring usability with the system usability scale (SUS).
19. Nielsen, J., & Landauer, T. K. (1993, May). A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems* (pp. 206-213). ACM.
20. Jeff Sauro. 10 things to know about the single ease question (seq), 2012. https://www.measuringu.com/blog/seq10.php
21. Olsen, T., Procci, K., & Bowers, C. (2011). Serious games usability testing: How to ensure proper usability, playability, and effectiveness. In *Design, user experience, and usability. Theory, methods, tools and practice* (pp. 625-634). Springer Berlin Heidelberg.