

Research Article

A Case of Engineering Quality for Mobile Healthcare Applications Using Augmented Personal Software Process Improvement

Shahbaz Ahmed Khan Ghayyur, Daud Awan, and Malik Sikander Hayat Khiyal

Faculty of Computer Sciences, Preston University, Islamabad 44000, Pakistan

Correspondence should be addressed to Shahbaz Ahmed Khan Ghayyur; shahbaz.ahmed@iiu.edu.pk

Received 9 January 2016; Accepted 21 February 2016

Academic Editor: Basit Shahzad

Copyright © 2016 Shahbaz Ahmed Khan Ghayyur et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mobile healthcare systems are currently considered as key research areas in the domain of software engineering. The adoption of modern technologies, for mobile healthcare systems, is a quick option for industry professionals. Software architecture is a key feature that contributes towards a software product, solution, or services. Software architecture helps in better communication, documentation of design decisions, risks identification, basis for reusability, scalability, scheduling, and reduced maintenance cost and lastly it helps to avoid software failures. Hence, in order to solve the abovementioned issues in mobile healthcare, the software architecture is integrated with personal software process. Personal software process has been applied successfully but it is unable to address the issues related to architectural design and evaluation capabilities. Hence, a new technique architecture augmented personal process is presented in order to enhance the quality of the mobile healthcare systems through the use of architectural design with integration of personal software process. The proposed process was validated by case studies. It was found that the proposed process helped in reducing the overall costs and effort. Moreover, an improved architectural design helped in development of high quality mobile healthcare system.

1. Introduction

Mobile health technology aids in practice of medicine and is aided by mobile devices. Google Play and App Store currently host nearly 47,000 mobile applications related to healthcare systems and the hits per day exceed 4 million. Hence, mobile health technology is leaping forward and patients and healthcare professionals have started to reap its benefits. This means that the demand of infrastructure and technology developers is rising day by day. This is only possible when the information is stored and retrieved from the management information systems maintained by hospitals and professionals. For better and effective management, it is necessary to control these heterogeneous working groups and the use of technology may help in better coordination of different activities of these groups.

Information Technology and mobile services are considered as a key feature to improve efficiency of the mobile

healthcare systems due to adequate support of the systems. Information systems may help in the improvement of the availability [1] and completeness [2], reduce failures [3], and enhance the orientation of comprehensive documents [4–6]. The modern technology has hypnotized the healthcare providers and its adoption has become a necessity in order to equip hospitals with modern trends [7] in order to incorporate those procedures that may help in effective practices related to the medication. These advancements are the result of wireless communications and network technologies in the previous years [6] and have a significant impact on mobile healthcare (M-health) systems and these systems are also termed healthcare systems for mobile computing or medical sensor and communications technologies [8]. M-health is one of the “biggest technology breakthroughs of our time” [9]. Moreover, “M-health technologies have the potential to change every aspect of the healthcare environment, while delivering better outcomes and substantially lowering costs

and at the same time collecting data about healthcare consumers' health status" [10]. The technologies that are involved in M-health are smart and mobile phones, WiFi, and Bluetooth and are used to transmit data in order to facilitate the health services [11].

The use of mobile devices was also successful in the United States in order to deliver the healthcare services and the situation was the same in Europe [12] and in Asia [13]. In developing countries for primary healthcare the key technologies that are used in the development of M-health systems are based on mobile/wireless ICT [14]. Thus, there is an ever growing demand for healthcare related software and system application development professionals. This research paper focuses on quality of work for individual software developers working on mobile healthcare management systems and how they can improve their own productivity and product quality by utilizing an enhanced personal process. PSP has been successfully utilized and implemented in the leading organizations like DEC, AIS, and HP Corporation [15], Motorola Paging Products Group, Signal Inc., Union Switch, and Advanced Information Services Inc. [16]. Seven competency areas are described in the current version of PSPBOK [17] including (1) Fundamental Knowledge, (2) Basic PSP Concepts, (3) Size Measuring and Estimating, (4) Making and Tracking Project Plans, (5) Planning and Tracking Software Quality, (6) Software Design, and (7) Process Extensions and Customization. However, it lacks in addressing architecture design and evaluation capability.

Software architecture has significant implications for mobile healthcare application development. Hence, in order to manage the complexity related to the development, maintenance, and evolution of a critical software-intensive system, the architectural details must be accurate and traceable during implementation [18]. Software architecture plays an important role in contributing towards a software product, solution, or services. It is like a blueprint or skeleton of a software system that is to be built [19]. Software architecture benefits include a tool for stakeholders communication [20], documentation of design decisions, identification of risks as a result of design decisions, and basis for reuse [20], promotes scalability [21], enables scheduling which saves time, cost of correction, or rework, and most importantly helps avoid software disasters [21]. Therefore, the productivity, performance, and quality of product for individual engineers working on mobile e-health applications can be enhanced by incorporating software architecture support into the personal software process for software development.

This research paper first explores the effects of adopting software architecture methodologies into the personal software process and later proving the effectiveness of modified process named Architectural Augmented Personal Process (AAPP). An AAPP is defined and executed with the help of a mobile healthcare system case study for demonstration of personal improvement which in turn results in better quality software systems. The purpose of this research is to find the impact of architecture with respect to the risk, time, cost, and product quality and explore the PSP based software process improvement (SPI) in light of local industry

constraints when developing specialized healthcare related software applications.

This paper investigates an augmented process for personal maturity of developers by implementing a mobile application in the domain of healthcare related to the hospital management and patient interaction system for a local hospital in Pakistan, so better quality systems with better risk management and with low cost and tighter schedule, while promoting simplicity feedback efficiency and adaptability, can be produced by incorporating and adapting software architectural practices. Four distinct case studies on two identical systems have been performed and data is recorded for the mobile e-health applications with the help of two teams both having equal skill set and experience from the same organizations. Both are trained in PSP and then the modified AAPP under study and then the results and yields are analysed for both teams to come up with a clear conclusion backed by solid data. The results are then analysed with experience of current system developers who had created the earlier application for the said system with the help of different tools.

2. The Proposed Solution

In order to answer the research questions, an architecture augmented personal software improvement process is proposed which not only does focus on yield of the programmer but also adds the additional benefits of software architecture incorporation which are early risk identification and management, quality and better communication, and rational management of the design and modification decisions while keeping stakeholders in collaboration. This shall in turn result in better time, cost, and scope management and shall also reduce time to market for e-health systems. This section details the activities that resulted in a PSP based architecture centric methodology AAPP. The method, how to customize the personal software process to meet the new challenges, is clearly stated as an 8-step strategy in the SEICMU PSPBOK. The following is the stepwise explanation of the customized PSP named AAPP in this text.

2.1. Steps for Architecture Augmentation. There are eight steps in architecture augmentation in AAPP and their detailed description is as follows.

Step 1 (determine personal needs and priorities). A process incorporating the SEBOK software architecture characteristics is required to be incorporated in the current personal software process which currently as per knowledge area 6 of PSPBOK supports only the detailed design and does not provide any measures to incorporate the architecture design for software product development. This incorporating is required in order to overcome the identified demotivators.

Step 2 (define process objectives, goals, and quality criteria). The incorporated architecture should be lightweight and in alignment with PSP principles and practices and should also encourage simplicity, communication, and feedback for efficiency of the proposed process. Furthermore, the quality

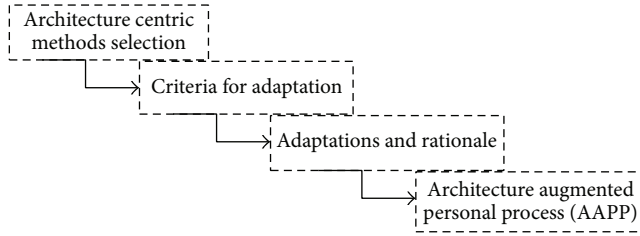


FIGURE 1: High level activities performed during the proposed methodology development.

and effort along with cost and time to market are not too high when compared to PSP. There are effects on basic measures of time, size, quality (defects), and schedule. Overall impact should also be positive as a result of this augmentation of software architecture into PSP. The proposed process adaptations shall improve the process quality by means of early risk identification and improving overall product quality focusing on quality attributes of interest. The process is also easily adaptable and efficient.

Step 3 (characterize the current process). The current PSP in its present form should be characterized by the PSPBOK.

Step 4 (characterize the target process). Target process should augment the classical PSP with process objectives defined in Step 2. It should also improve quality attributes which are nonfunctional requirements (NFRs). Performance, modifiability, security, and usability are few quality attributes a system may have.

Step 5 (establish a process development strategy). The activities include (1) architecture centric method selection, (2) criteria for adaptation to PP context, (3) adaptations and rationale, and finally (4) Architecture Augmented Personal Process (AAPP). Figure 1 shows key steps involved in the proposed architecture centric method (AAPP) development.

Step 6 (define the initial process). (a) *Architecture Centric Methods Selection*. In our research, SEI QAW [22] and ADD [23] methods have been adapted to form our architecture centric methodology (AAPP) since they are in direct alignment with our research goal for creating an architecture augmented personal process. Software Engineering Institute's methods of quality attribute workshop and attribute driven design are chosen due to their architecture centric nature and origin from the Software Engineering Institute and also for their respective maturity in the area (QAW 3rd Edition, ADD version 2.0) [24]. Moreover, research literature explicitly recommends the integration of architecture centric methods [25] with lightweight software process methodology.

(b) *Criteria for Adaptation to Context*. To adapt and integrate any process into personal process context, we must acknowledge the ground on which the personal process is based which is taken from capability maturity. Any proposed adaptations must stimulate, have strong grounding in, and be mapped directly with the principles. The criteria have been

TABLE 1: Quality attribute workshop (QAW) adaptations.

| Sr. number | Adaptation to quality attribute workshop |
|------------|--|
| (1) | Presentation and introductions (removed) |
| (2) | Business/mission presentation (removed) |

TABLE 2: Mapping ADD adaptations to PP values, principles, and practices.

| Sr. number | Adaptation to ADD |
|------------|--|
| (1) | More iterative than recursive attribute driven design |
| (2) | Confirm there is sufficient requirements information (amalgamated) |
| (3) | Define interfaces for instantiated elements (removed) |
| (4) | Verify and refine requirements and make them constraints for instantiated elements (amalgamated) |
| (5) | Repeat from Step 2 to Step 4 (amalgamated with iterative adaptation) |

used for the proposed adaptations and mapped values that stimulate such adaptations, practices that achieve those values, and finally the principles that are key for achieving such values.

(c) *Architecture Centric Methods*. SEI quality attribute workshop (QAW) [22] and attribute driven design (ADD) [23] methods were chosen for adaptations according to personal process context. The following subsections map adaptations with personal process values, principles, and practices.

(d) *Context to SEI QAW*. Table 1 shows the adaptations made to the quality attribute workshop (QAW) [22].

Adapted quality attribute workshop (QAW) consists of six steps. Figure 2 shows the adapted SEI QAW according to AAPP context.

(e) *Context Adaptations to SEI ADD*. Table 2 shows the attribute driven design (ADD) adaptations mapped to personal process (PP).

Adapted attribute driven design (ADD) according to personal process context consists of four steps that are shown in Figure 3.

(f) *Architecture Augmented Personal Process (AAPP) Stage*. Figure 4 shows architecture augmented personal process (AAPP) and its activities.

Architecture augmented personal process (AAPP) methodology includes merged activities from adapted quality attribute workshop (QAW) and attribute driven design (ADD) methods as described in [23], respectively. During the second and later iterations for modular development within AAPP, it is made sure that requirements are still consistent with current understanding of the system (Step 1: system stakeholders collaboration) as system is further explored or as a result of communication with stakeholders. Section 3 is about performed case studies in order to verify and validate our proposed AAP method.

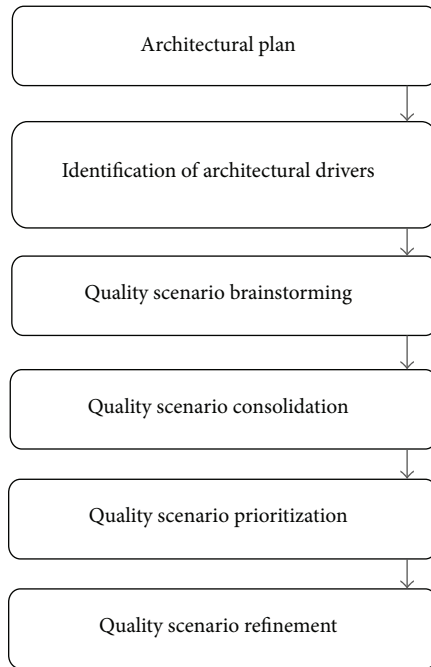


FIGURE 2: Adapted QAW according to AAPP.

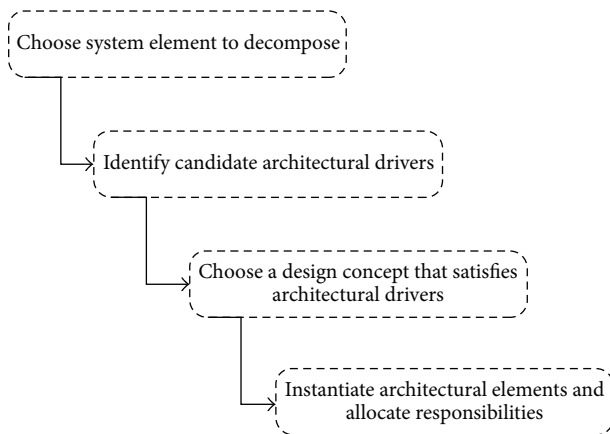


FIGURE 3: Adapted attribute driven design steps.

3. Case Study

Case study has been chosen as a method to carry out our research since it is a scientific or empirical method used when we want to test whether our theory holds any weight in the real world in a specific context while having no control over variables or context. It is defined as “a technique for detailed exploratory investigations, to understand and explain phenomenon or test theories, using primarily qualitative analysis.” The goal of the case study is to provide accurate and comprehensive description of the case. In software engineering, case studies are used for validating research, for example, evaluation of new tools, processes, or methods [25]. Case study is described as a research strategy that comprises design logic, data collection techniques, and specific approaches to data analysis. The following are the strengths and weaknesses of case studies described in [26].

3.1. Case Studies Components. Case studies have the following essential five important components according to Yin [26].

3.1.1. Research Questions. Research questions are usually stated as “who,” “what,” “where,” “how,” and “why.” Case study is most likely to be used with “how” and “why” questions; however, exceptions and overlaps are common. This research will address the following research questions.

RQ1: How can the software architecture centric methods be effectively incorporated in personal software process?

3.1.2. RQ2. What is the influence of software architecture knowledge support in context to risk, time, cost, and product quality for a process engineer developing mobile applications?

Propositions or Hypothesis. Hypothesis is an educated guess that keeps the research in the right direction [26]. Hypothesis for our research is as follows.

H: Adapted architecture centric method will result in a lightweight quality enhanced approach in terms of effort, cost, risk, and time and quality of product.

3.1.3. Unit of Analysis. There are two methodologies that are compared, that is, personal software process (PSP) and architecture augmented personal process (AAPP) for mobile and e-health application development.

3.1.4. Determination of How Data Are Linked to Propositions. Data collected during case study should be a reflection of the proposition and mapped to it [26].

Validity measures of risk, effort, cost, and quality are key success factors for mobile application software project [27] and they have been used along with time to market and other values’ achievement to evaluate the effectiveness of the two approaches. Table 3 shows the validity measures used in this research which reflects our proposition and research questions.

3.1.5. Criteria to Interpret Findings. Any findings and conclusions will be made on the basis of data collected during case studies keeping in view the research questions and propositions along with the statistical analysis.

4. Criteria For Judging Quality of Research Design

Four tests are described in [26] to establish quality of any empirical social research (case study being one of them). These steps are as follows.

4.1. Construct Validity. Construct validity ensures correct operational measures chosen for the concepts being studied. Effort, cost, time to market, and quality attributes achievement levels were measured in this research which reflects

TABLE 3: Validity measures.

| Sr. # | Validity measure | Measured in |
|-------|--------------------------------------|---|
| (1) | Effort (architecture) | Man-hours |
| (2) | Effort (overall) | Man-hours for development activity |
| (3) | Cost (architecture) | Man-hours × wage/hour |
| (4) | Cost (overall) | Man-hours for development activity × pay/hour |
| (5) | Time to market (TTM) | TTM = time product delivered – time product conceived |
| (6) | Quality attributes attainment levels | Percentage (%) of total score achieved |

TABLE 4: Architecture and overall effort measuring matrix.

| Day | 0-1 hour | 1-2 hours | 2-3 hours | 3-4 hours | 4-5 hours | 5-6 hours | 6-7 hours | 7-8 hours |
|-----|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| (1) | A/D | A/D | A/D | A/D | A/D | A/D | A/D | A/D |
| (2) | A/D | A/D | A/D | A/D | A/D | A/D | A/D | A/D |
| N | A/D | A/D | A/D | A/D | A/D | A/D | A/D | A/D |

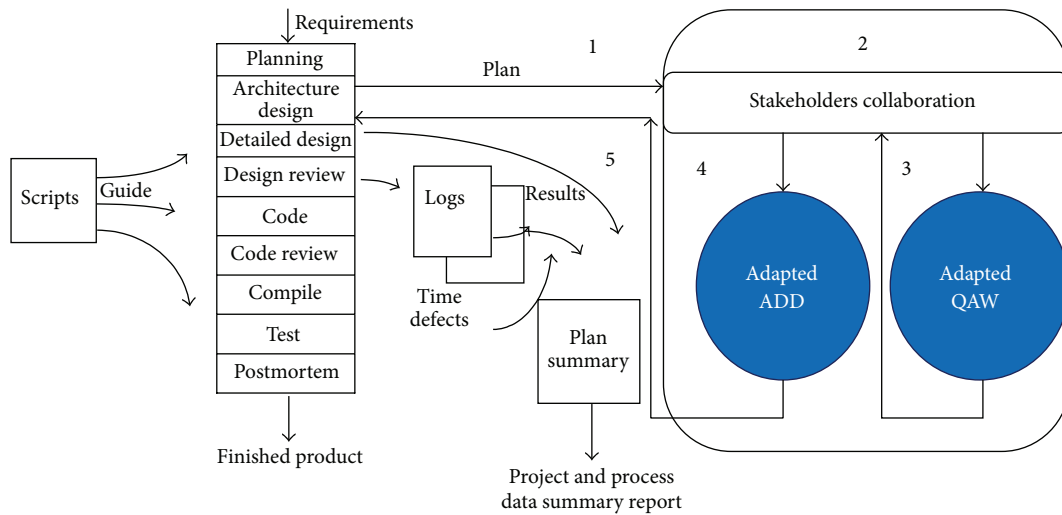


FIGURE 4: Architecture augmented personal process.

our research questions as well as proposition. These measures have strong architecture literature grounding.

4.2. Internal Validity. Internal validity is inapplicable to case studies that are not concerned with causal situation. In our research, each inference is given its due consideration and rationale during the research design.

4.3. External Validity. Within case studies, it means that the results can be generalized to similar cases to those that were studied. In our research, two separate instances of the same systems were developed to ensure that our results were repeatable and generalizable.

4.4. Reliability. Reliability means that if the same procedures were employed on the same case study again (perhaps by another researcher), the researcher should arrive at the same results/findings earlier recorded. In our research, these

steps were documented and executed and are described in Section 7.

5. Measurement Procedures

A day implies 8 working hours while a week is made up of 5 working days. The following were the data measurement procedures for each validity measure.

5.1. Effort. During the day, how much time (man-hours) was spent on architecture activity (marked by an A) and development activity (marked by D) is shown in Table 4.

Effort (architecture) was calculated counting all As and effort (overall) was calculated counting both A and D as shown in Table 4.

5.2. Cost. Architecture cost and overall cost were calculated by multiplying architecture effort and overall effort with wage in \$/man-hour, respectively.

5.3. *Time to Market.* Time to market is the total time taken from project conception to its completion as shown below:

$$\text{Time to Market} = \text{Time Product Delivered} - \text{Time Product Conceived.} \quad (1)$$

5.4. *Quality Attributes Achievement Levels.* The quality attributes are described along with their subcategories. Each software project has its own software quality attributes of importance in the view of stakeholders. Interviews (client and developers) were used to calculate quality attribute achievement score for each quality attribute; these scores were later aggregated and a holistic percentage (%) score for all quality attributes of importance is calculated.

6. Case Study Execution

The case study was executed in the university lab where two sets of graduate software engineers were selected as participants out of a group of volunteers. Both had 3 years of Management System Development Experience and were also well versed in system design and implementation. Both also have experience in maintaining and documentation of software process data for process improvement. A week of training was also provided to both of the participants of the case study that were provided with adequate training in personal software process and architecture augmented personal process and small examples were executed before moving on to the system under investigation. Identical System Vision documents for alternatively developing the same systems keeping the same lab environment and variables were provided to both sets of engineers along with details of software hardware interfaces available and constraints on the system along with the focused quality attributes of maintainability, usability performance while promoting better communication, simplicity, and feedback for the system.

For comparative analysis and calculation of significance of the effectiveness of both approaches, a small case of mobile application was executed first as a pilot study. The results of the mobile application were analysed to come up with a better understanding of the effectiveness of the proposed process. Time Lapse Assistant redeveloped for this study is an Android OS (or framework) based application and is implemented as part of case studies 1 and 2. Time Lapse Assistant is a tool for professional photographers to calculate time lapse photography parameters and save calculated parameters as part of different projects for later reviewing and other useful tools to aid time lapse photography such as synchronized timer, project map, and sun times.

The main health related mobile application system development undertaken is the hospital management system. This is a term that is mostly referred to where management activities take place residing inside the hospital. The particular system helps the entire hospital management including doctors and clerks. The most important entity that is being focused is the patient. The system is designed to be used on tablets and mobile phones and runs in parallel with the web based system.

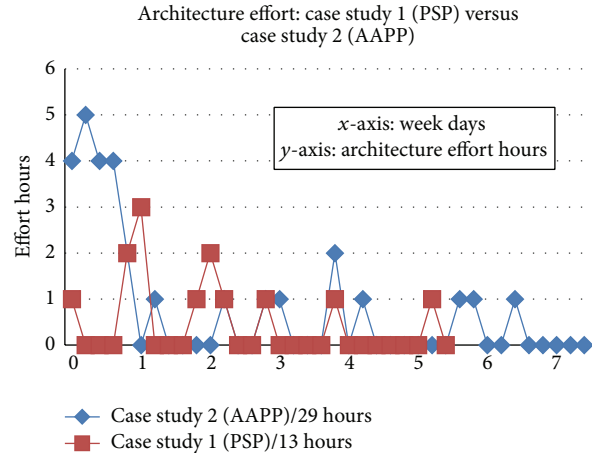


FIGURE 5: Case study 1 versus case study 2, architecture effort, project size ~2400 LOC.

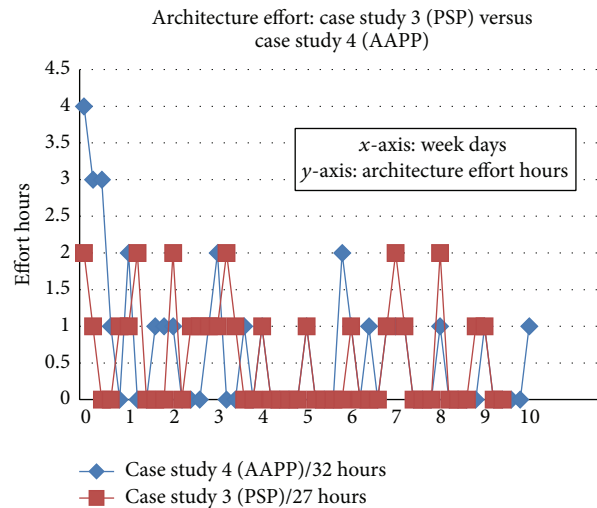


FIGURE 6: Case study 3 versus case study 4, architecture effort, project size ~4700 LOC.

7. Case Study Results

7.1. *Findings: Architecture Effort.* Figure 5 presents comparative architecture effort for case study 1, that is, by applying personal software process (PSP), and case study 2, that is, by applying Architecture Augmented Personal Process (AAPP) methodology per week as shown in the figure.

Total architectural effort calculated in *man-hours* for personal software process (PSP) was found to be 13 hours; on the other hand, AAPP took 29 hours (2.23 times greater when compared with PSP) while the project size was approximately 2400 LOC. It was noted that architecture effort for AAPP was considerably higher during the first week due to comprehensive focus on detailed architecture and stakeholder collaboration.

Figure 6 presents comparative architectural effort for case study 3, that is, by applying PSP, and case study 4, by

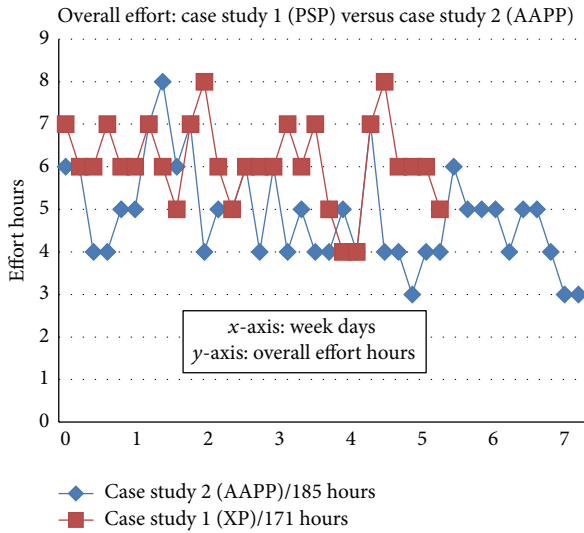


FIGURE 7: Case study 1 versus case study 2, overall effort.

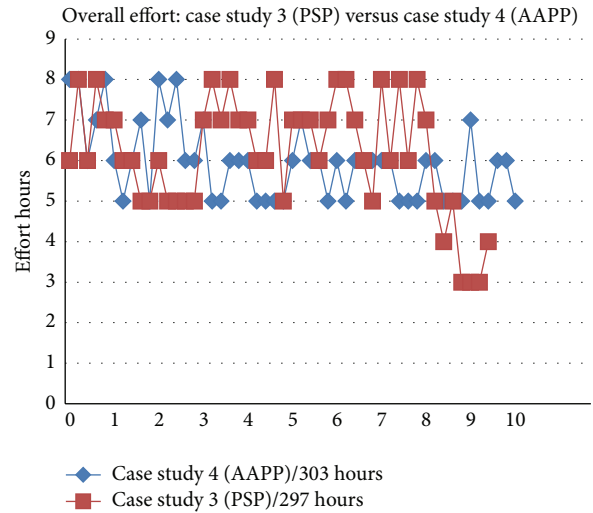


FIGURE 8: Case study 3 versus case study 4, overall effort.

applying Architecture Augmented Personal Process (AAPP) methodology per week as shown in the figure.

Total architectural effort calculated in *man-hours* for PSP was found to be 27 hours; on the other hand, AAPP took 32 hours (1.18 times greater when compared with PSP design) while the project size was approximately 4700 LOC. Once again, it was noted that design effort for AAPP was higher, but this time not as much significant as with case study 1 and case study 2 comparison, that is, 2.23 times. Furthermore, during case study 3 and case study 4, the project size is greater (4700 LOC) as compared to case study 1 and case study 2; the architecture effort for PSP versus AAPP became less significant, that is, from 16-hour (or 2.23 times greater during AAPP) difference during case study 1 and case study 2 to 5 hours during case study 3/case study 4 (or 1.18 times greater during AAPP).

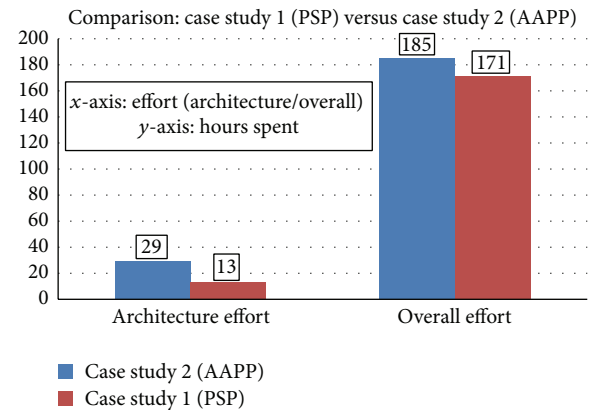


FIGURE 9: Case study 1 versus case study 2, architecture and overall effort spent.

7.2. Findings: Overall Effort. Figure 7 presents overall effort spent (*man-hours*) for the implementation of system during case study 1, that is, by applying PSP, and case study 2, that is, by applying Architecture Augmented Personal Process (AAPP).

During case study 1 (PSP), it took 171 hours, while during case study 2 (AAPP), it took 185 *man-hours* of overall effort. Case study 2 with AAPP took 14 hours more as compared to case study 1. This suggests that investing in architecture effort 2.23 times greater during AAPP saved considerable time during later stages, that is, code, test, and refactoring, and resulted in nonsignificant difference in overall time. Furthermore, there were 4 refactoring activities performed during case study 1 (PSP) suggesting little system understandability and more rework.

Figure 8 presents overall effort spent (*man-hours*) for the implementation of system during case study 3, that is, by applying PSP, and case study 4, that is, by applying architecture augmented personal process (AAPP) as shown in the figure.

During case study 3 (PSP), it took 297 *man-hours* of overall effort while case study 4 (AAPP) took 303 hours. The project sizes for case study 1/case study 2 and case study 3/case study 4 were approximately 2400 LOC and 4700 LOC, respectively. It was found that overall effort or hours got even less significant for AAPP with increased project size, that is, from 14 hours during case study 1/case study 2 to 6 hours during case study 3/case study 4.

7.3. Effort Comparisons for Case Studies. Figure 9 shows architecture as well as overall effort difference for case study 1 and case study 2 between PSP and architecture augmented personal process (AAPP).

Architectural effort in case of AAPP in case study is more than twice but the difference in overall effort is insignificant.

Figure 10 shows architecture as well as overall effort difference for case study 3 and case study 4 between PSP and architecture augmented personal process (AAPP).

With increased project size during case study 3 and case study 4, the difference between both architectural effort and

TABLE 5: Average architecture and overall effort per day for case studies.

| Case study # | Mean architecture effort/day (man-hours/day) | Mean overall effort/day (man-hours/day) |
|----------------|--|---|
| (1) Using PSP | 13/28 = 0.4642 hours/day | 171/28 = 6.1071 hours/day |
| (2) Using AAPP | 29/38 = 0.7631 hours/day | 185/38 = 4.8684 hours/day |
| (3) Using PSP | 27/48 = 0.5625 hours/day | 297/49 = 6.0612 hours/day |
| (4) Using AAPP | 32/51 = 0.6274 hours/day | 303/51 = 5.9411 hours/day |

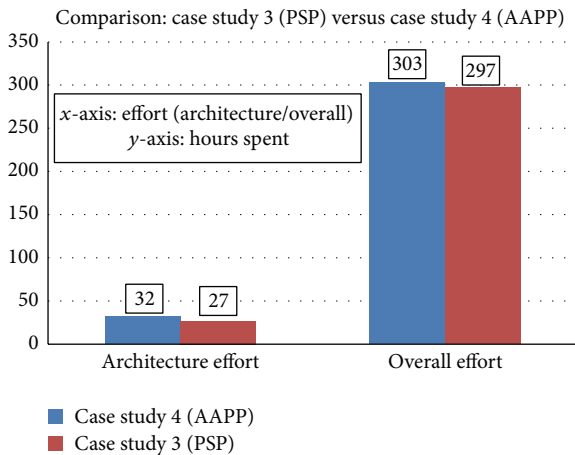


FIGURE 10: Case study 3 versus case study 4, architecture and overall effort spent.

overall effort gets insignificant. This suggests that design or architecture activity for smaller projects may provide its benefits but at doubled architecture effort when architecture augmented personal process (AAPP) was applied compared to traditional architecture activities in personal software process (PSP). However, with increased project size (2400 LOC to 4700 LOC in our case), we notice the effort difference getting less significant between PSP design activities and AAPP. This is because during PSP case studies 1 and 3 more refactoring and rework efforts were made as compared to AAPP case studies 2 and 4 and least planning or architectural effort. Such approach provided good results in terms of less effort spent for PSP architecture for smaller project but with larger project size PSP resulted in even more refactoring and rework effort where the difference between both approaches got even less significant. In other words, the least architectural effort for larger projects resulted in more rework as compared to the smaller projects canceling out any effort saved in the first place.

If we take average for architecture and overall effort, we get the results as shown in Table 5 and Figure 11.

Case study 1 (PSP) resulted in average daily architecture effort of 0.4642 man-hours a day while with case study 2 (AAPP) we get average daily architecture effort of 0.7631 man-hours a day. AAPP requires significantly higher effort as compared to PSP with project of small size. Case study 3 (PSP) resulted in average daily architecture effort of 0.5625 man-hours per day while case study 4 (AAPP) resulted in average daily architecture effort of 0.6274. With an increased project

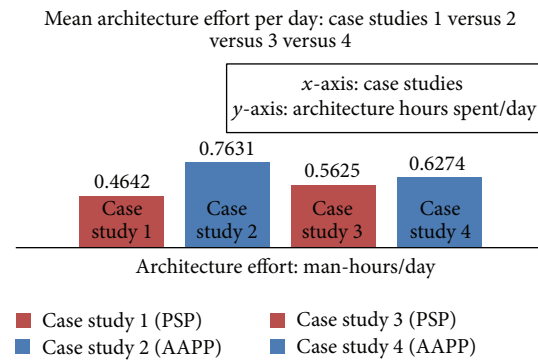


FIGURE 11: Average architecture effort per day during all case studies.

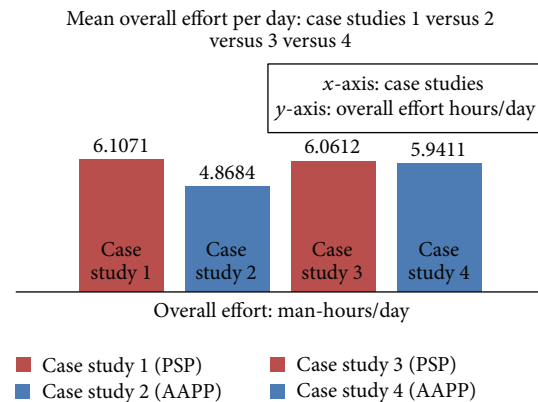


FIGURE 12: Mean overall effort per day during all case studies.

size, we see that average daily architecture effort became insignificant between PSP and AAPP, which is due to higher cost of replanning in case of larger project (case study 3) due to lack of extensive system understanding.

During case studies 1 (PSP) and 2 (AAPP) which represent a small sized project, we get overall daily average effort of 6.1071 and 4.8684 man-hours a day, respectively. This means that with AAPP methodology we saved 1.2387 man-hours on average per day; as more time on architecture was spent during AAPP, this resulted in a better understanding of the system and less rework. However, case studies 3 (PSP) and 4 (AAPP) were conducted on significantly larger project size and the results show an average daily effort of 6.0612 and 5.9411 man-hours, respectively. This difference is insignificant but with AAPP we saved 0.1201 man-hours of overall effort per day. See Figure 12 for further details.

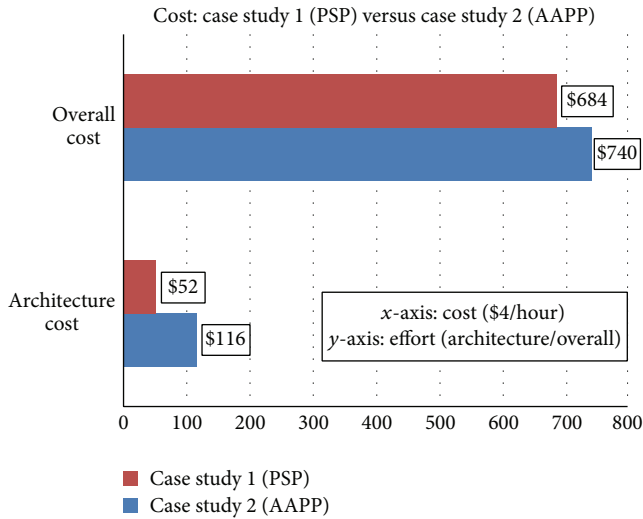


FIGURE 13: Case study 1 versus case study 2, architecture and overall cost.

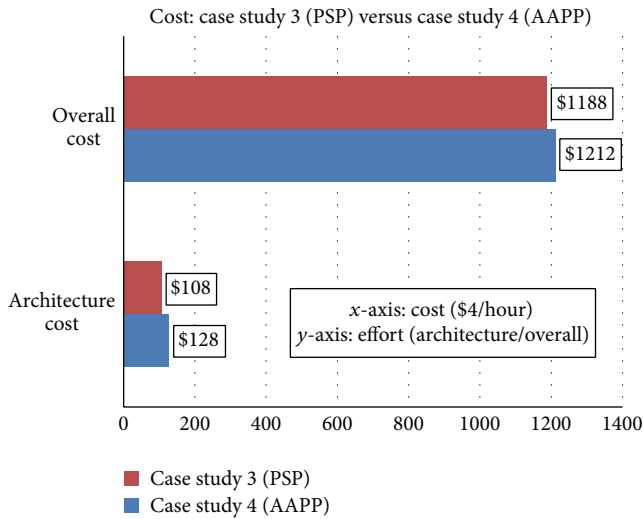


FIGURE 14: Case study 3 versus case study 4, architecture and overall cost.

8. Findings and Discussions

8.1. Architecture and Overall Costs. Figure 13 shows architecture and overall cost for PSP and architecture augmented personal process (AAPP) approach applied during case study 1 and case study 2, respectively.

During case study 1 and case study 2 with smaller project size (~2400 LOC) it costs more than twice for architecture with AAPP while overall cost is slightly higher as compared to PSP.

Figure 14 shows architecture and overall cost for PSP and architecture augmented personal process (AAPP) approach employed during case study 3 and case study 4, respectively.

As cost is directly proportional to the effort, that is, the more the hours invested in an activity, the greater the cost, AAPP resulted in twice the cost as compared to PSP design

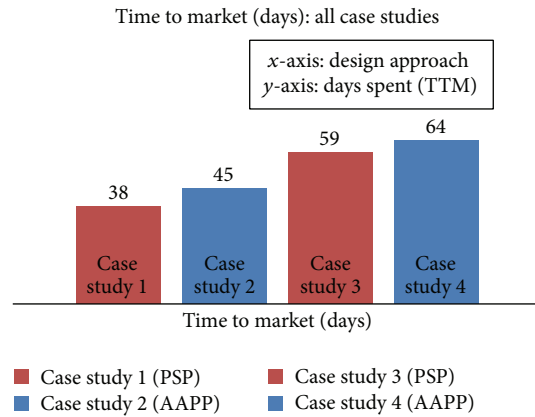


FIGURE 15: Time to market (TTM).

during case study 1 and case study 2 but the overall cost difference was not much significant (\$66 greater in case of AAPP) as it reduced the amount of rework due to better planning and system understanding that saved time during the later stages (coding, testing, and refactoring). As project size was increased with case study 3 and case study 4, the architecture and overall cost difference between PSP and AAPP became even less significant as compared to case study 1 and case study 2.

8.2. Time to Market. Time to market (TTM) is the total time taken for a product as a concept up till when the product is delivered. TTM for case study 1 (PSP Design) was 38 days while for case study 2 (AAPP) it was 45 days. Figure 15 shows TTM for both case studies.

TTM for case study 3 (PSP) was found to be 59 days while TTM for case study 4 (AAPP) was 64 days. The difference between the first and second case study in terms of TTM is 7 additional days in the case of AAPP, that is, case study 2. When compared with overall effort in hours as explained earlier which was 171 and 185 hours for PSP and AAPP, respectively, the difference was found to be 14 hours or less than 2 days of work. Difference between both approaches in terms of overall effort may not be very significant but TTM difference between the two approaches is more significant as compared to overall effort as it also includes the whole week including nonworking days and not just effort spent during working hours. During case study 3 (PSP) and case study 4 (AAPP), TTM difference was found to be 5 additional days for AAPP. Hence, TTM for larger project size (case study 3 and case study 4) was less significant when compared to smaller project size (case study 1 and case study 2) when AAPP was employed.

8.3. Quality Attributes Achievement Levels. Performance/efficiency, modifiability, and usability were found to be common quality attributes for our case studies. Stakeholders were interviewed and data from code was analysed. Each quality attribute achievement score is converted to percentage for ease of understanding.

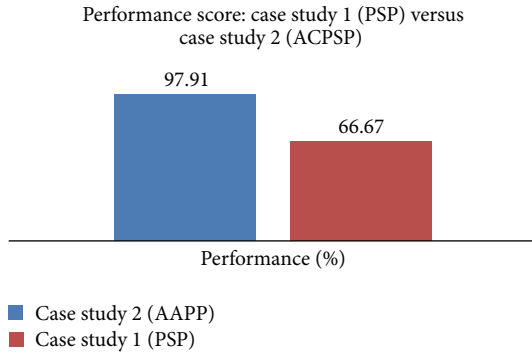


FIGURE 16: Case study 1 versus case study 2, performance score.

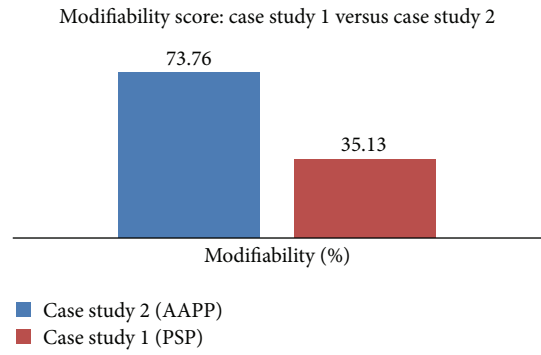


FIGURE 18: Case study 1 versus case study 2, modifiability score.

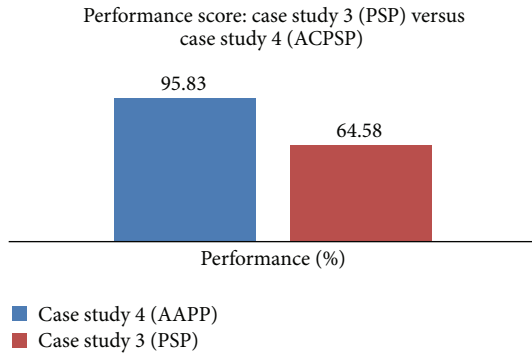


FIGURE 17: Case study 3 versus case study 4, performance score.

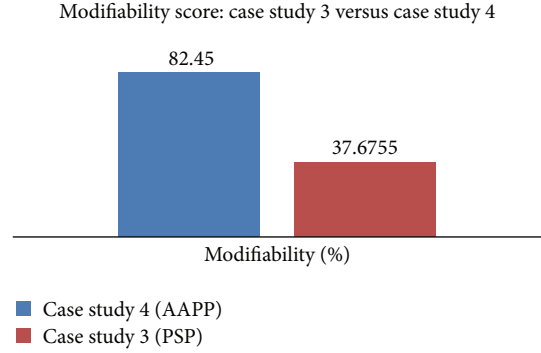


FIGURE 19: Case study 3 versus case study 4, modifiability score.

8.3.1. *Quality Attribute: Performance Score (%)*. Performance is the degree to which system or its component meets its required functions within defined constraints, that is, speed, accuracy, or memory usage. *Time economy* is one of the subfactors of performance and is used to measure performance in our case studies as *response time(s)*. Final score is converted into percentage for better understanding. Figure 16 showed the results found for performance quality attribute for case study 1 and case study 2.

Figure 17 showed the results found for performance quality attribute for case study 3 and case study 4.

8.3.2. *Quality Attribute: Modifiability Score (%)*. Modifiability is the ease with which system or component can be modified for corrections of faults, improved performance, adaptations to new environment, or any other attribute. *Cohesiveness*, *correctability*, and *extensibility* subfactors are used to measure modifiability. Final score has been converted into percentage for better understanding. Figure 18 showed the results found for quality attribute modifiability for case study 1 and case study 2.

Figure 19 showed the results found for quality attribute modifiability for case study 3 and case study 4.

8.3.3. *Quality Attribute: Usability Score (%)*. Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs from the system. Subfactors of *learnability* and *satisfaction* are used to measure usability. Final score for usability is converted into percentage for better

Usability score: case study 1 (PSP) versus case study 2 (ACPSP)

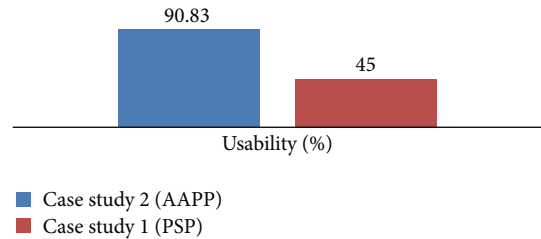


FIGURE 20: Case study 1 versus case study 2, usability score.

understanding. Figure 20 showed the results found for quality attribute usability for case study 1 and case study 2.

Figure 21 showed the results found for quality attribute usability for case study 3 and case study 4.

8.3.4. *Mean Quality Attribute Achievement Score for All Case Studies*. Overall results were found for quality attribute achievement levels for case studies 1, 2, 3, and 4 by taking mean of the earlier quality attribute scores in percentages as shown by the following formula:

$$\begin{aligned} \text{Mean QA score for Case Study } X &= \text{QA\% Score 1} + \text{QA\% Score 2} \dots \\ &+ \frac{\text{QA\% Score } n}{\text{Total QA } (N)}, \end{aligned} \tag{2}$$

where QA is the quality attribute.

TABLE 6: Mean quality attribute scores for all case studies.

| Case study # | Mean QA score (%) |
|--------------|-------------------|
| (1) PSP | 48.93% |
| (2) AAPP | 87.50% |
| (3) PSP | 51.30% |
| (4) AAPP | 91.92% |

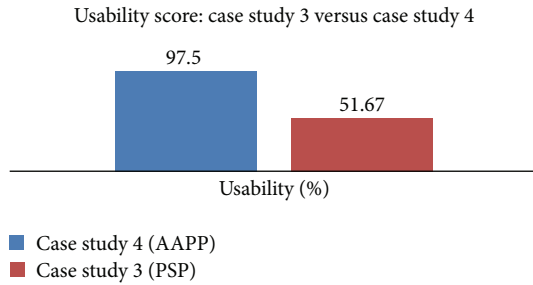


FIGURE 21: Case study 3 versus case study 4, usability score.

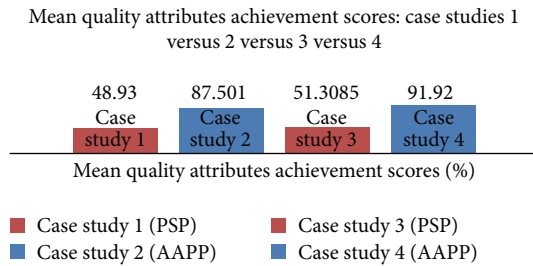


FIGURE 22: Case studies 1, 2, 3, and 4, mean quality attributes achievement scores.

Simply by adding all quality attributes score within a case study in % and dividing by total number of quality attributes, we get mean quality attribute achievement score for a case study. The mean quality attribute scores are shown in Table 6.

Figure 22 shows mean quality attribute scores for case studies 1, 2, 3, and 4 as shown in the figure.

As shown in Figure 22, the mean quality attribute achievement scores for case studies 2 and 4 by applying AAPP were found to be significantly higher as compared to case studies 1 and 3 by applying PSP.

8.4. *Project Values Achievement Score.* Project values of simplicity, feedback, and communication were measured by interviewing project stakeholders. First, we presented individual PSP value results and finally looked at the mean PSP values scores achieved during each case study.

8.4.1. *Value of Communication Score.* Figure 23 shows the results found for the impact on PSP value of *communication* for case study 1 and case study 2 when PSP and AAPP were applied, respectively.

Figure 24 shows the results for the impact on PSP value of *communication* for case study 3 and case study 4 when PSP and AAPP were applied, respectively.

Communication score: case study 1 versus case study 2

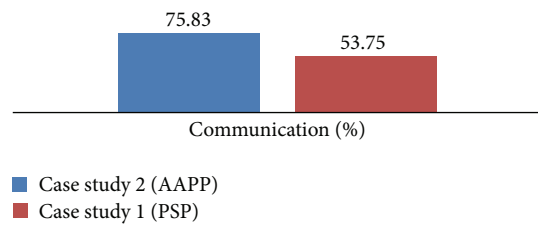


FIGURE 23: Case study 1 versus case study 2, communication value score.

Communication score: case study 3 versus case study 4

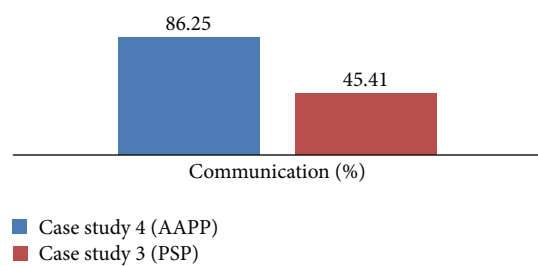


FIGURE 24: Case study 3 versus case study 4, communication value score.

Simplicity score: case study 1 versus case study 2

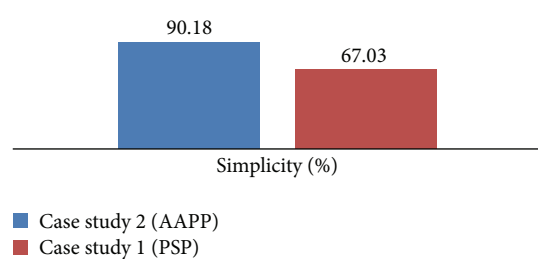


FIGURE 25: Case study 1 versus case study 2, simplicity value score.

8.4.2. *Value of Simplicity Score.* These were the results found for the impact on PSP value of *simplicity* for case study 1 and case study 2 when PSP and AAPP were employed, respectively, as shown in Figure 25.

Figure 26 showed the results found for the impact on PSP value of *simplicity* for case study 3 and case study 4 when PSP and AAPP were employed, respectively.

8.4.3. *Value of Feedback Score.* These were the results found for the impact on PSP value of *feedback* for case study 1 and case study 2 when PSP and AAPP were employed, respectively, as shown in Figure 27.

Figure 28 showed the results found for the impact on PSP value of *feedback* for case study 3 and case study 4 when PSP and AAPP were employed, respectively.

8.5. *Architecture Benefits Score.* Architecture benefits were measured using interview for personal software process

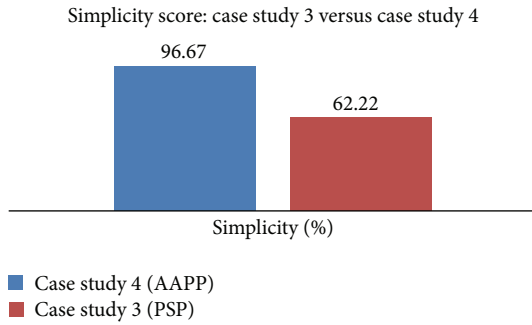


FIGURE 26: Case study 3 versus case study 4, simplicity value score.

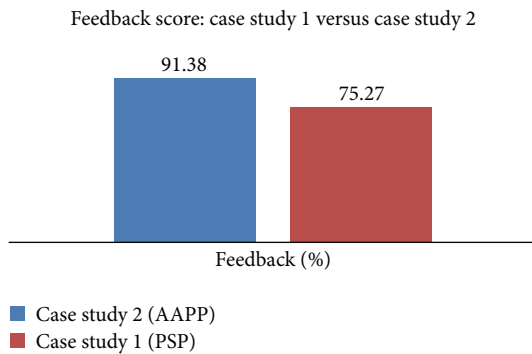


FIGURE 27: Case study 1 versus case study 2, feedback value score.

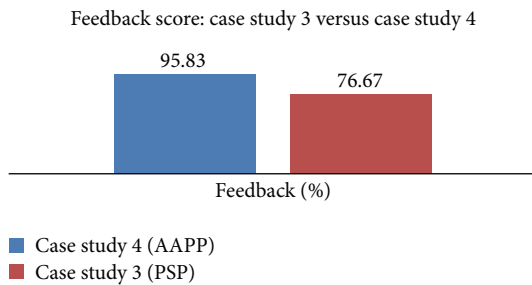


FIGURE 28: Case study 3 versus case study 4, feedback value score.

(PSP) and architecture augmented personal process (AAPP). Figure 29 shows *stakeholder communication* architecture benefit as percentage score for all case studies.

Figure 30 shows *documentation of design decisions* architecture benefit as percentage score for all case studies.

Figure 31 shows *identification of risks* architecture benefit as percentage score for all case studies.

Figure 32 shows *scalable solutions* architecture benefit as percentage score for all case studies.

Figure 33 shows *scheduling* architecture benefit as percentage score for all case studies.

Figure 34 shows mean architecture benefits score as percentage score for all case studies.

Case studies with architecture augmented personal process (AAPP) showed significantly greater mean achievement of architecture benefits score individually as well as mean, that is, over 20% for project with small size and over 30% for

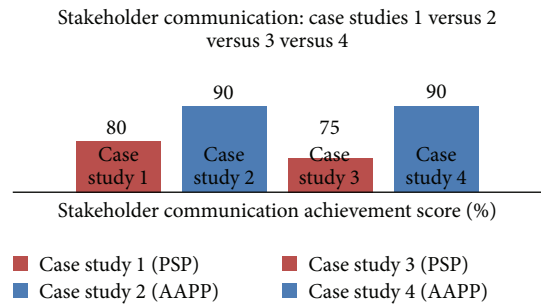


FIGURE 29: Stakeholder communication score for all case studies.

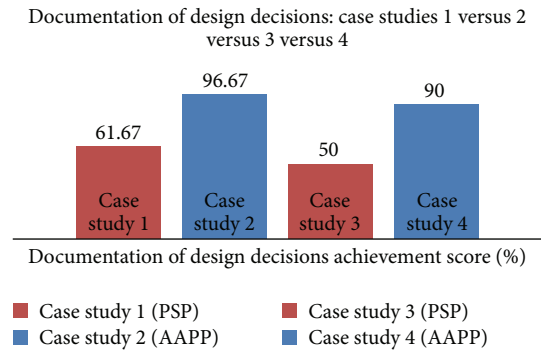


FIGURE 30: Documentation of design decisions score for all case studies.

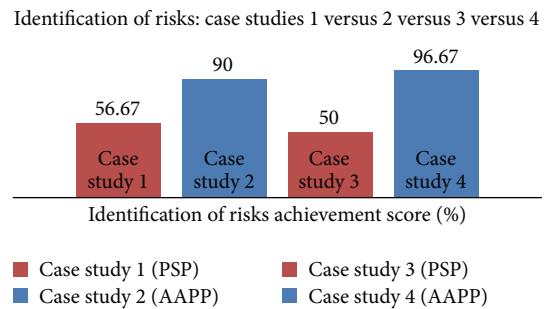


FIGURE 31: Identification of risks score for all case studies.

larger sized project as compared to personal software process case studies. It was also noted that team with AAPP had better understanding of the system and as a result created simple solutions that took less time with little or no rework in some cases. This also helped team schedule their tasks better. Table 7 showed the scores recorded for values and quality attributes during case studies 1, 2, 3, and 4; scores are out of 100 and represent percentage (%) as shown in the table.

From the data shown in Table 7, we get a correlation of 0.9923 which means a highly positive correlation and that for its set of data if we increase goal compliance scores, quality attributes scores also increase. In other words, if we stress goal compliance during a project, it has positive impact on quality attributes of that project.

Although the addition of software architecture has induced some additional cost time and effort, the overall advantages outweigh this burden. The results clearly

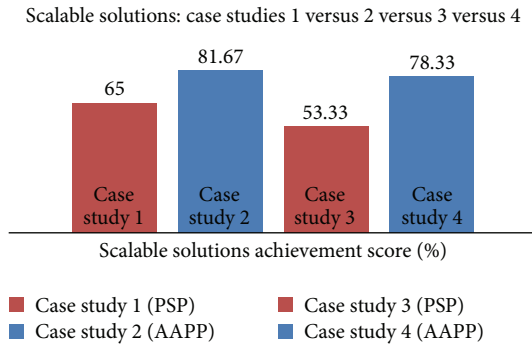


FIGURE 32: Scalable solutions score for all case studies.

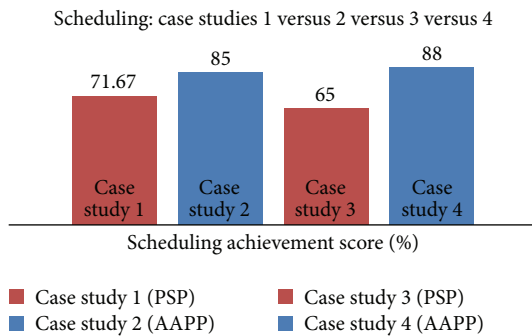


FIGURE 33: Scheduling score for all case studies.

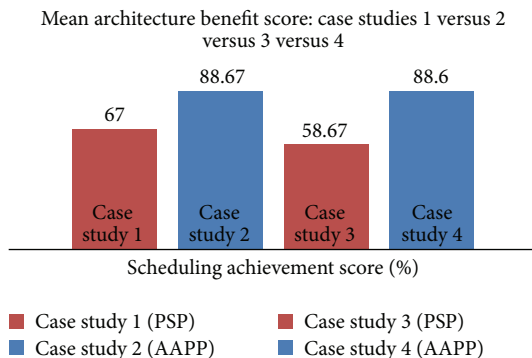


FIGURE 34: Mean architecture benefit score for all case studies.

show that the augmentation of architecture knowledge has enhanced the overall process output and also impacted the overall product quality. The early risk identification and mitigation also proved to be a handful which reduced the rework time during M-health system development process. Similarly as the size of the project and complexity increases to a certain limit the cost, effort, and time values effect would get reduced for architecture work.

9. Conclusion

The newly proposed process AAPP enhances the overall quality of the mobile healthcare systems in terms of consideration of the architecture with the software process. The case studies were performed in order to compare the proposed process AAPP. The research results show an improvement in terms

TABLE 7: Defined process goals and quality attribute scores for all case studies.

| Sr. # | Case study | Process goals score | Quality attributes score |
|-------|---------------------|---------------------|--------------------------|
| (1) | Case study 1 (PSP) | 61.5125 | 48.93 |
| (2) | Case study 2 (AAPP) | 86.325 | 87.501 |
| (3) | Case study 3 (PSP) | 59.5125 | 51.3085 |
| (4) | Case study 4 (AAPP) | 93.125 | 91.92 |

of architectural benefits that are achieved using AAPP as described in the findings. The application of AAPP is initially a bit costly in terms of time and effort but the performance of the personal process is improved by augmenting the architecture with personal software process. The proposed AAPP is applied in the domain of healthcare in order to get an utmost benefit of the proposed process. Initially, AAPP is applied on a simple healthcare project. In future research, we will apply AAPP on more domains and complex projects in order to better generalize and validate the proposed process.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

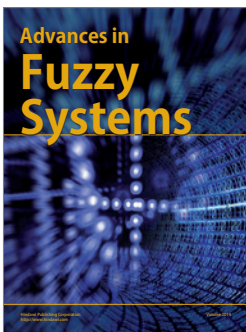
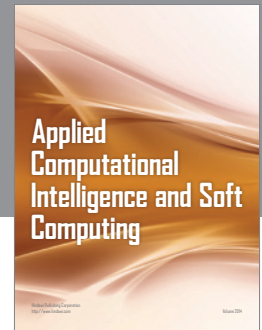
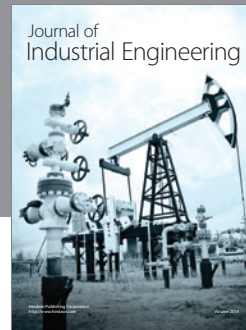
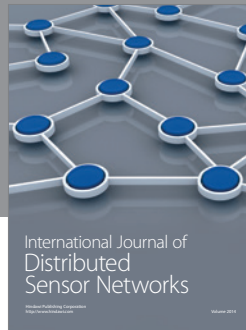
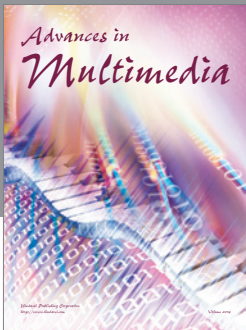
Acknowledgments

Special thanks are due to Preston University Pakistan, Participating Engineers, and Higher Education Commission of Pakistan for providing resources in order to complete this research.

References

- [1] R. B. Elson, J. G. Faughnan, and D. P. Connelly, "An industrial process view of information delivery to support clinical decision making," *Journal of the American Medical Informatics Association*, vol. 4, no. 4, pp. 266–278, 1997.
- [2] P. C. Tang, M. P. Larosa, and S. M. Gorden, "Use of computer-based records, completeness of documentation, and appropriateness of documented clinical decisions," *Journal of the American Medical Informatics Association*, vol. 6, no. 3, pp. 245–251, 1999.
- [3] D. W. Bates, A. C. O'neil, D. Boyle et al., "Potential identifiability and preventability of adverse events using information systems," *Journal of the American Medical Informatics Association*, vol. 1, no. 5, pp. 404–411, 1994.
- [4] C. J. McDonald and W. M. Tierney, "Computer-stored medical records. Their future role in medical practice," *Journal of the American Medical Association*, vol. 259, no. 23, pp. 3433–3440, 1988.
- [5] L. E. Garrett Jr., W. E. Hammond, and W. W. Stead, "The effects of computerized medical records on provider efficiency and quality of care," *Methods of Information in Medicine*, vol. 25, no. 3, pp. 151–157, 1986.
- [6] A. Sunyaev, J. M. Leimeister, A. Schweiger, and H. Krcmar, "Integrationsarchitekturen fur das Krankenhaus—Status quo

- und Zukunftsperspektiven,” *IM-MUNCHEN*, vol. 21, no. 1, pp. 28–35, 2006.
- [7] A. Kasbo and R. McLaughlin, *Mobile Health Applications: 2012 Study*, 2012.
- [8] R. S. H. Istepanian, E. Jovanov, and Y. T. Zhang, “Introduction to the special section on m-Health: Beyond seamless mobility and global wireless health-care connectivity,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 8, no. 4, pp. 405–414, 2004.
- [9] S. R. Steinhubl, E. D. Muse, and E. J. Topol, “Can mobile health technologies transform health care?” *The Journal of the American Medical Association*, vol. 310, no. 22, pp. 2395–2396, 2013.
- [10] C. E. Kuziemsky, H. Monkman, C. Petersen et al., “Big data in healthcare-defining the digital persona through user contexts from the micro to the macro,” *IMIA Yearbook*, vol. 9, no. 1, pp. 82–89, 2014.
- [11] F. Imouokhome and V. Osubor, “Mobile-device-based telemedicine for improved health-wealth,” *African Journal of Computing & ICT*, vol. 5, 2012.
- [12] E. Ammenwerth, A. Buchauer, B. Bludau, and R. Haux, “Mobile information and communication tools in the hospital,” *International Journal of Medical Informatics*, vol. 57, no. 1, pp. 21–40, 2000.
- [13] S. Hyun, J. Choi, J. Chun et al., “Implementation of mobile computing system in clinical environment: MobileNurse™,” in *Proceedings of the AMIA Symposium*, p. 1036, 2000.
- [14] R. Wootton, *Telehealth in the Developing World*, IDRC, 2009.
- [15] M. Frappier and M. Richard, “SMP: a process-driven approach to project management,” in *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, p. 9, January 2004.
- [16] S. B. de Oliveira, R. Valle, and C. F. Mahler, “A comparative analysis of CMMI software project management by Brazilian, Indian and Chinese companies,” *Software Quality Journal*, vol. 18, no. 2, pp. 177–194, 2010.
- [17] M. Umarji and C. Seaman, “Predicting acceptance of software process improvement,” *ACM SIGSOFT Software Engineering Notes*, vol. 30, no. 4, pp. 1–6, 2005.
- [18] T. Mens, J. Magee, and B. Rumpe, “Evolving software architecture descriptions of critical systems,” *Computer*, vol. 43, no. 5, Article ID 5472890, pp. 42–48, 2010.
- [19] N. Rozanski and E. Woods, *Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives*, Addison-Wesley, Boston, Mass, USA, 2011.
- [20] A. Rauf, S. Anwar, M. Ramzan, and A. A. Shahid, “Analysis of software process improvement efforts in Pakistan,” in *Proceedings of the 2nd International Conference on Computer and Automation Engineering (ICCAE '10)*, pp. 375–379, Singapore, February 2010.
- [21] D. Garlan and M. Shaw, “An introduction to software architecture,” in *Advances in Software Engineering and Knowledge Engineering*, vol. 1, World Scientific Publishing, 1993.
- [22] R. Barbacci Mario, “Quality attribute workshops (QAW),” *Reporte Técnico del CMU-SEI*, 2003.
- [23] R. Wojcik, F. Bachmann, L. Bass et al., “Attribute-Driven Design (ADD), Version 2.0,” *DTIC Document*, 2006.
- [24] P. Bourque and R. Dupuis, *Software Engineering Body of Knowledge (SWEBOK)*, IEEE Computer Society, EUA, 2004.
- [25] S. Easterbrook and J. Aranda, “Case studies for software engineers,” in *Proceedings of the International Conference on Software Engineering (ICSE '06)*, Shanghai, China, May 2006.
- [26] R. K. Yin, *Case Study Research: Design and Methods*, Sage, Thousand Oaks, Calif, USA, 2013.
- [27] R. Atkinson, “Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria,” *International Journal of Project Management*, vol. 17, no. 6, pp. 337–342, 1999.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

