ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 5, Issue 5, May 2016

Elderly Assistive Device : Testing, Integration And Bug Fixing

Dipika J. Patel
Electronics and Communication Dept.
Ganpat University
Kherava-Gujarat, India

Saurabh Soni
Software Head,
eInfochips Training and Research Academy
Ahmedabad-Gujarat, India

***Abstract*— Currently elderly peoples live alone because of some social reason. The crime rate also increases with elderly people because they do not have enough strength to take care about him or her or their property like house. They want to connect with their children, friends, relatives so that they can easily connect with social networking like Skype video call. Elderly people do not have enough strength to turn on/off electronics appliance and doors lock/unlock by theme selves. Sometimes Elderly people have to call to doctor or their relative in case of emergency. Elderly Assistive Device system which assists the old age person for their daily routine to be performed at the ease of touching Android device screen. As prototype of any project is ready, it must be tested with all possible test cases for both hardware and software.**

Keywords— testing; integration; bugs; bug fixing; LogCat; EAD

I. INTRODUCTION

Before many years, there was big Indian joint family, where all family members under one roof. But now a days, amount of elderly people living alone and long life is increasing across the world. Elderly people living alone need to be,

- Stay Independent
- Stay Connected
- Stay Secured
- Stay Healthy

Problems of elderly people living alone are,

- Less active
- Lack of memory
- Lack of confidence
- Loneliness
- Boredom
- Weak eyesight
- Medical emergency

In the most of all the development of the gadget industry the section of society, which was left out or very little had been done for, was the elderlies (age>60). So here we are up with a system which elderlies could cheer about. We developed prototype for "Saathi – An Elderly Assistive Device (EAD)" system to cater needs of elderly person in their day to day life at the ease of touch with low cost and ease of use.

In this paper methods for testing, integration, identify bug and bug fixing is described. This paper describes techniques for testing and bug fixing.

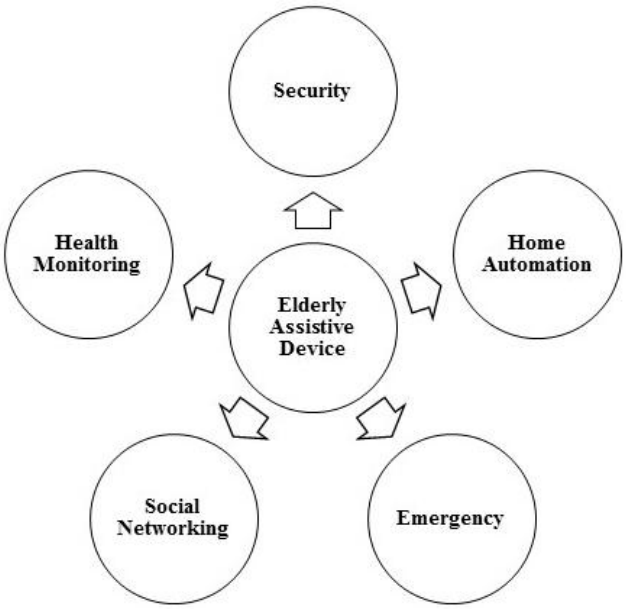

Figure 1Block diagram of EAD

II. INTEGRATION

In order to function any embedded project/product, it must be integrated properly. As the market of technologies is fragmented, so are the technologies themselves. Technologies in general are stand-alone solutions for specific problems. As embedded software development is collection of complex and technical problems, several technologies are applied in parallel. It does not take very much time or experience to observe that this lack of integration is a cause for problems too. Technologies are used separately but depend on each other, interfaces are not defined, inconsistencies occur, etc. Despite this loss of quality due to lack of integration, there is a loss of time, effort and money due to duplication, redundancy and cost of non-quality [1]. Here in EAD project various technologies are integrated like Wi-Fi technology, Bluetooth connectivity, Bluetooth low energy, heart rate wrist band, IP camera, electrical door lock, smart phone with android OS and microcontrollers. All this hardware components are integrated with suitable protocols like 801.11E, UART, SPI and I2C according to their pin and circuit diagrams. Hardware components are communicated via one Wi-Fi router using Bluetooth and Wi-Fi connectivity. To operate hardware remotely, android application called "SAATHI" is developed like measure heart rate

and display it to smartphone screen, turn ON/OFF home appliances, monitor person outside home, open and close the door, automatic calling, email and sms sending to doctor and screencast smartphone's screen to TV at ease of one touch.

III. TESTING

At the point when any application is created, it must be tested. After breezing through application through different experiments, testing engineer came to think about existence of any action or life of that application. How application will look like and how it will act can be resolved. Equipment testing is likewise expected to decide after the amount of time equipment will destroy. By testing, voltage and force prerequisites can be decided. Additionally conduct and operation of parts can be seen at different natural and electrical conditions.

- Equipment testing: As the span of both equipment circuits and programming framework expanded quickly, the comprehensive testing of both equipment and programming has ended up infeasible in the down to earth work. Some deterministic testing techniques have demonstrated their need of capacity to guarantee that all shortcomings in a vast size of framework under test can be identified [2].Different experiments for all equipment are rattled off and tried as needs be for different electrical and ecological conditions. On the off chance that any part comes up short, its interface done again and made it work appropriately.
- Software testing is a movement which is gone for assessing a property or ability of a system and guarantees that it meets the required result. There are different programming testing methods like white box testing, discovery testing and dim box testing [3]. Here all given testing systems used to enhance quality and life time of android application. In the first place period of testing was discovery trying. In this usefulness of utilization considered i.e. how application reacts on client cooperation's. Second period of testing was white box testing i.e. inside structure of coding is tried like punctuation mistake, legitimate blunders and so on. As this kind of testing worried with inside structure of use, all linguistic and legitimate mistakes recorded for bug settling reason. This will diminish mistakes furthermore lessens undesirable smashing of utilization.

IV. BUGS

Changes are unavoidable in programming improvement and each change made to a product framework's code base displays a danger that it may present a bug. At the point when such a bug happens it regularly goes unnoticed until somebody presents a bug report depicting the wrong conduct. At the point when the engineers get such a report, they assess it to guarantee that it is a genuine issue, and after that address the issue. As part of the determination procedure, designers may incorporate pointers about the bug they settle in the change log. Connecting a bug report to the update in which it was altered can help us rough which changes brought about bug. In the wake of leading tests of equipment and programming of Elderly Assistive Device, some bugs and changes are found. Here two sorts of bug found in this "Saathi" App. One is linguistic bug and other is sensible bug

A. Linguistic bug: In numerous java class records, variable names given are not proper. One can't comprehend the extent of that variable by name. In numerous XML documents, name of perspectives and widgets are not given appropriate. Some other individual other than designer can't decide the utilization of any that specific widget. Documentation of Saathi application is not there. Techniques, capacities and classes are not depicted. Extent of all these is definitely not given. In this way, documentation is required for intelligible design.

B. Sensible bug: Saathi application contains application particular and legitimate bugs. Health monitoring, Security and home Automation piece goes under this area. This all squares are deficient in necessity or sensibly not created legitimately. Health monitoring piece is inadequate in the application particular usefulness. It ought to contain programmed calling and programmed SMS sending highlight other than manual operations. As client's wellbeing is basic, he/she is not ready to call or SMS. For this, there must be programmed calling and SMS sending highlight. Since as heart point of confinement of client goes past impediments client cannot call or message to specialist. In security obstruct, there is linguistic and consistent bugs. In the wake of utilizing security obstruct the application halted working or get crashed. While leaving from security action, this application get crashed i.e. at the point when client does a reversal from security, when client straightforwardly goes to whatever other square like client profile, apparatus, security and health monitoring. In home automation, there is syntactical or legitimate bug. Utilizing this square client can work apparatuses legitimately however there is bug in this piece. Utilizing switches at PCB, apparatuses can be worked however ON/OFF status of apparatus/switch is not reflected at smart phone. Switch status must be upgraded/reflected when changed. Additionally switch status is not spared while leaving from home automation.

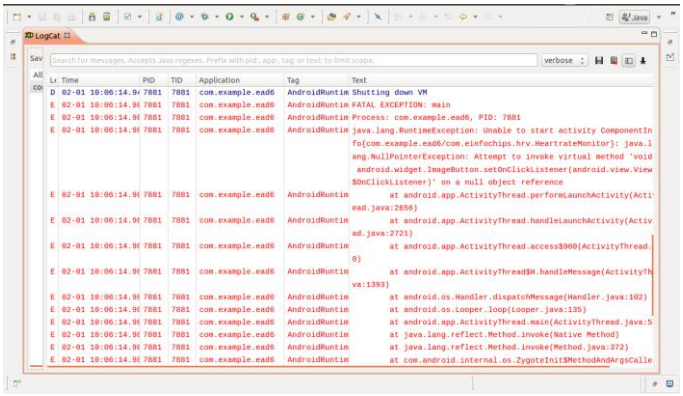

Figure 2security block bug

V. BUG FIXING

Bug fixing phase comes after testing and identifying bug and determining its exact place in code. Here in testing stage of "Saathi" app, bugs identified. All the bugs listed above are fixed in this bug fixing phase. To fix bugs, algorithms designed according to requirement for functionality of app.

Algorithm for Health monitoring bug fixing:

- Remove existing code of functionality to be updated or changed.
- Debug app and monitor its LogCat
- Determine at which line heart rate value is being measured
- Apply condition of high and low limit. If value goes beyond limitation, call the function for phone_call().

Phone_call() function,

- Using shared preferences, save the values of low limit and high limit which are set previously.
- Fetch data of contact and user profile from database.
- To send SMS use SMS manager API and for Call use CALL intent.
- Fetch data like, contact number of doctor and user, address of user, low limit, high limit, user name to form text message to be sent to doctor.
- Use setText API and add above details as message body and also add doctor's number to send message.
- When phone_call() function is called, SMS will be sent automatically.
- Set doctor's number in Call intent.
- After starting intent of call, call to doctor can be made.

Algorithm for Security:

- Debug Saathi App and monitor LogCat.
- Determine at which point the application is crashed.
- LogCat will show the error or exception message.
- Resolving this bug can be fixed.

Algorithm for Home Automation

- Debug application and monitor LogCat
- Go to home automation block of app
- Check at which place current status is
- Use shared preferences and save switch status

VI. CONCLUSION

To build up any application, engineer must take after coding rules. Legitimate depiction for all strategies, variables, classes, capacities ought to be given so that individual other than engineer can effectively comprehend coding. Names of variables, capacities, strategies, classes ought to be given according to standard coding rules. Sometime recently dispatching any application it must be tried for all viewpoint and all purpose of perspectives of clients. On the off chance that there is any bugs or blunders seen in the wake of testing, it must be determined and altered. Investigating is the strategy by which correct line at which bug dwells can be known. Bug can be settled by applying appropriate rationale by utilizing experimentation strategy.

REFERENCES

[1] Rini van Solingen Project manager of the MOOSE project CMG, The Netherlands, "Integrating Software Engineering Technologies for Embedded Systems Development" [Online]

[2] Shiyi Xu Shanghai University, Shanghai, 200072 China, "Orderly Random Testing for Both Hardware and Software", 2008 14th IEEE Pacific Rim International Symposium on Dependable Computing

[3] Muhammad Asaduzzaman Michael C. Bullock Chanchal K. Roy Kevin A. Schneider Department of Computer Science, University of Saskatchewan, Canada, "Bug Introducing Changes: A Case Study with Android", In Proc. IEEE International Conference on Android bugs. 2014.