International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2019): 7.583

Embedded Applications Agile Software Approach

N. Michael Franklin

Assistant Professor, Stella Mary's College of Engineering, Vellamodi, India

Abstract: ***Through this article, it has been discovered that there is a framework called the Adaptive Agile Model that blends the concepts of agile development with organizational patterns and can be applied to create embedded real-time systems based on the system's constraints. The method promotes hardware partitioning and the platform configuration design through the initial approach of covering and barely minimizing the unit boundaries and thereafter after a while, and finally, achieving the final design. By using this design strategy, one is expected to be able to provide quicker customization so that less time is spent in the design stage of the project. So in order to demonstrate how necessary it is to keep the oximeter in its proper position, my colleague will clarify the importance of holding an oximeter in its proper position.***

Keywords: Agile, Embedded systems, oximeter

1. Introduction

The more affordable, less costly and more powerful micro controllers can be used for a wide variety of systems that could make it economically enticing for them to be used in combination with these systems. As the Personal Insightful Chips take the market and add more and more value to the applications they are used for. As the applications themselves advance, the chips themselves will advance as well, adding in more value and utility. Where a 32-Bit micro-controller is needed, the smaller 24-Bit micro-controllers are used as opposed to the 32-Bit micro-controllers as a whole. Those of you who own a car must not only take the advice of several safety experts seriously safety on the road, but must also trust the systems that are used to protect you and your loved ones from traffic accidents. When we continue to build more functionality into the systems, the time we have to make the system executes get longer. Whoever is on the project team and whoever is being produced the product then need to have a base for calculating the team size in the community, clients and customers should be included in the planning phase, specifications need to be considered before the project is initiated and once the project is finished then the end result can be achieved at the right time. Software appears to operate well on a personal computer (PC) because of the isolation of the software within the computer from other processes on the PC. However, on embedded systems, the software must interact with other processes on the system; otherwise the design does not work as it was intended to. When you shut the doors of your fridge, there are many components that are different from the things you would find in your desktop or laptop computer. It also gives a special look to the viewer so it will be able to distinguish one particular ending (e.g. energy consumption, execution time, memory foot-print). Considering they do not have any of the requisite expertise in software engineering, the embedded systems engineering department does not have a worthwhile department. We may build a hardware that could use certain algorithms of data storage and retrieval in order to withstand the difficulties of storage and retrieval of high-dimensional data sets. In the future, a significant thing is to remember that the number of embedded real-time systems is high, and there may be some at risk when they go down (mission criticality). With this in mind, when the cloud is absent and errors are frequent, it is a prudent decision to treat them differently than when the data is distributed or few users and data are present.

Since the context of embedded systems is very different from the ordinary, we feel that it is important to create an adaptable architecture which is just like agile, and to constantly change the system and release it on the market to keep pace with the industry. The hardest thing to remember when designing embedded systems are the uncertain nature of specifications, which is why this is looked at as a project by a team of software engineers. The goal of this project is that it will enable the utilizing of agile techniques, such as the extreme programming (XP) style of coding, as well as applying a feedback loop of prepping, implementing, and reviewing proposed features before deploying them to the production environment. (i.e. platform-based design [1]). The strategy you can use for tackling this is first, to set targets (goals), second, to set goals that can be met (ability), third, to set goals that can be met by one, or more of you (skill), fourth, to set goals that can be met by you by being able to see the best data possible while still being able to perform the job (or work) that you have to do (data), fifth, to set goals that you all agree upon to make sure to achieve everyone's target (job, or work) and goals that are achievable to the best of your ability.

2. Literature Survey

Embedded teams typically use only software development approaches [2]. This is why the product's functionality was at that stage really weak. Because of technological difficulties that were not handled correctly at the time. In order to have a successful convergence test, we had to commence a literary analysis. In order to start our convergence test, this review revealed three separate applications.

A paper on the implementation of agile firmware generation methods for the Intel Itanium family [2] has been included in this study. Greene outlines in this paper the agile techniques his team is actually using every day, but does not detail the use of the agile principles in a project that uses a software application to create or modify hardware. Greene just notes that another Intel team has begun to embrace agile

principles and to make substantial progress. Nevertheless, our methodology was very beneficial when the input got from that paper on the implementation of agile principles, as it supported the advantages of using agile concepts outside object-orientated software development.

Chiranjeevi Aradhya [3], [4] says applying agile practices can help produce working applications on a daily basis and be able to adapt to changes, thereby lowering the risk of budget and schedule overruns.Also heexplains the evaluation of the robustness testing practice of embedded software systems.

The Gajski's methodology [5], which attempts, through a methodical explanation of the functions of the computer, to construct embedded systems rather than natural language. The system design requires the executable specification of assignment, partitioning and refinement to be configured. Genetic designs and innovative processes are used to discuss design alternatives. Once the components have been established, they are simply compiled into a program using an implementation scheme. Our approach is effective on other systems, we have already built and demonstrated, and we plan to use the same methods for our project. This approach will, however, include assumptions that all specifications are documented before applying partitioning algorithms.

Finally, Manhart and Schneider [6] gained fascinating experience in designing integrated system software for agile waterfall development. A well-developed automotive cad software framework integrating agile components has been enhanced some little to meet their current needs as capabilities and high-speed software for production. Studies in this paper show that e-cigarettes can be useful in many other areas, although their findings have not yet been produced by the researchers. They are talked to at a conference, maybe just around the corner.

The method of estimating the number of homeless people in Amritsar is somewhat different from the others. By programming a large number of architectural hardware constraints, I establish our methodology. This can lead to a complicated, cost effective machinery framework, ii) electric stimulation based software technology built through the use of intelligent methods and the design of our software/hardware assessment, and iii) iterative and graduated approaches enable us to create and enforce our methodology gradually. (ii) The proposed method or methodology clearly provides a process by which the designer can validate the algorithm-generated system specification partition or the proposed methodology clearly provides an iterative procedure through which an algorithm-split system specification can be validated. (ii) It functions in several forms, not only by taking into account possibilities that could arise at a late stage of the process, but also by acknowledging those adjustments.

3. Agile Methodologies

3.1 Extreme Programming

Extreme programming (XP), a collaborative team-oriented method, is the most recognizable agile approach[7]. XP contains the twelve main functional approaches, including: a technology that changes the software to prevent the external behavior of the code from changing and to enhance its internal structure.

(i)The Code will be automatically compiled and analysed for continuous integration. (ii)The test-led development (iii) means that developers write the tests before they are coded. These unit tests are automated tests testing the characteristics of code components. All project participants shall follow the same code style in the practical Coding Standard (iv). The format for the source code in the programming language selected is consistent.

By using the first three approaches discussed above, XP encourages an evolutionary approach to system creation. The great advantage is that the method progresses quickly and tries to reduce the risk and uncertainty of the project as soon as possible (risk management). Section 4 discusses how the proposed approach applies these XP procedures.

3.2 Scrum

The development of Software is a simple and easy approach to the Human age, on the premise that the method would possibly alter both the variables of the world (i.e. people) and of technology [8]. The Scrum consists of 14 operations and incorporates the key features, in addition to the suggested solution: The sprinting exercise is the 30 calendar day version. Two sessions are held in Sprint Preparation Operation.

At the first meeting the software backlog is refined and the owner, players and priorities for the subsequent iteration are prioritized. The features, usage events, updates, and system failures are identified. The Scrum team will learn to satisfy the demands during the second meeting, and will build a sprint backlog of comprehensive tasks for the current iteration to complete. At the end of the iterations for the product owner, consumers and those in the area of Sprint Testing, the team from Scrum presents the findings. Scrum team meetings are held everyday in the Daily Scrum practice, addressing special questions (iv).
Scrum uses the empirical process management model to evaluate the state of the job and empirically decide what to do next (product). The knowledge and enthusiasm of the participants is heavily dependent on efficiency and performance.

4. Proposed Method

The approach is to identify roles and tasks and to include procedures, life cycles, activities and tools to incorporate embedded real-time systems projects. It consists of three separate process classes used for the implementation of the system: system platform, product production and management.

The goal of the community application processing platform is a particular product platform. The system designer must choose system components that are part of a platform library's architecture and API platforms. The system engineer may also set up the architecture and API platforms to comply with the application limitations. The architecture and runtime-sets logic of the platform are used for the customization feature. The process of adaptation is applied in an iterative and gradual manner by successive refinements of the proposed approach.

The software development community provides software development activities and integrates framework elements. Either hardware or software elements of the platform constitute the functionality of the device. The algorithms to this task take into account the energy consumption, running time and memory of the components of the software. However, beyond the scope of this paper, mechanical design also forms part of this type of process.

The Department of Product Management follows and manages product scope, time, quality and cost parameters. The platform system and the product creation categories also are affected. If the project begins with an inviolable project plan requiring corrective action, the group will restart the project in order to ensure the project requirements have been met. The product management department comprises both Scrum agile framework-supported operations and agile patterns.

System Process

The system platform category comprises product specifications, system platform, product line and system optimization. The manufacturer's needs process attempts to satisfy the system specifications that are critical for deciding the platform for the product building system (functional and non-functional). The system instance process allows the team to define the platform across a set of design instruments and benchmarks.

The product line step helps the development team to configure a registry, where system components are available, once the system platform is developed. Device features are introduced and incorporated and new versions of software are released in the development team. The system optimisation process ensures the implementation constraints of system variables including runtime, energy usage, program size and data memory are met once the functionality has been implemented, incorporated into the software development system.

Product Development

Community product creation processes include: usability, integration of projects, refactoring and system optimisation. The feature method ensures that test cases are produced for each product's functions. It improves product consistency and prevents complicated functions.New implemented functionalities can be implemented through the roles integration phase into the product development line without having to collaborate with team members.

The device reconstruction process helps the development team identify ways to improve and adjust the code, but without altering it. After code refactoring, the development team optimizes small sections of the code by using profiles that track the software and figure out for example where space, time and energy is spent. This approach guarantees that program metrics meet the device requirements.

Product Management

The Product Management includes: product requirements, project management, bug tracking, sprinting criteria, product line and deadlines for implementation. The method of product specification (also part of the group system process platform) is to fulfill the system requirements that must be part of the product (functional and non-functional). Inventory management, backpack sprints, mission planning, system design and product bug monitoring support the production community in project management.

The framework for monitoring bugs helps the product manager to control the project life cycle via release notes for an end user (bug, mission and enhancement). The Sprint method helps the team to check and test the functionality of the system before beginning a new project. This knowledge is found in the sprint backlog, which enables the development team to divide into software or hardware in advance of sprint.

The process for product line ensures that device features are implemented during sprints into the product development line. Furthermore, the development team supports the arrival of new product types on the market. The framework for implementation of requirements helps the manager to fix any interruptions that could affect the goals of the project. This means that the project's tasks can be done 100%.

5. Pulse Oximeter Architecture

This section explains the use of the technique proposed in producing a pulse oximeter device. The pulse oximeter was chosen as a case study as it was previously developed in an ad hoc development phase by another research group [9]. We therefore offer an example of how to apply the proposed technology to embedded systems, only the pulse oximeter in this section. The pulse oximeter typically tests oxygen saturation through a non-invasive process of the blood system. Figure 1 demonstrates the architecture of this unit. The microcontroller traces the synchronization and amplification of the non-infrared and red pulse driver at the same time. Both leads emit infrared pulses and red pulses that move through a patient's finger. A photodiode absorbs the sum of radiation as it passes through its finger. There are a number of operations before microcontroller data are reached. On the basis of data collected, the microcontroller checks for oxygen saturation and displays display performance. The final product comprises approximately five thousand lines of the C code. Hardware components were used to stimulate and condition timing and energy limiting sensing signals while the microcontroller used the control algorithm and the signal conversion method in a program.

The key features of the unit are as follows: Each second, if saturation and frequency occur, (ii) the user must have the option to change Alarm Settings, (iii) pulsation device

oximeter and graphical display interface, (iv) the architecture of a machine must be highly optimized to provide cost-effectiveness and life cycle reliability, (v) as many software faults as possible.

The development of this product would involve about three persons, two developers and one product leader with the proposed approach. The commodity was assembled, checked and distributed over three weeks in roughly four sprints.

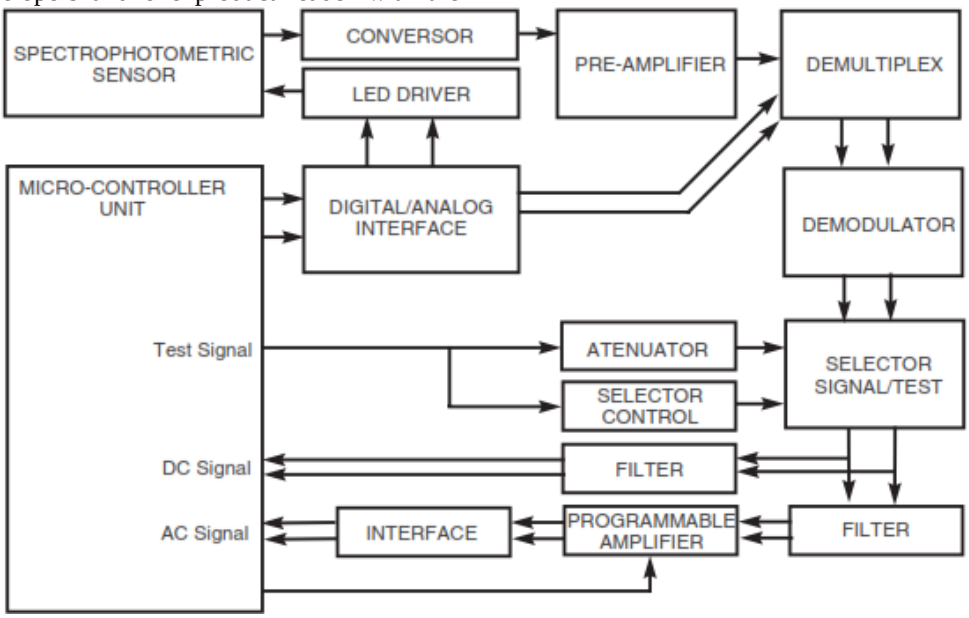

Figure1: PulseOximeter Architecture

6. Conclusion

This paper outlined an agile development approach for its application and pulse oximeter development. In this paper two agile approaches, XP and Scrum, have been selected as agile and corporate patterns. For agile software development projects Scrum is primarily developed. In comparison, XP is a collection of common methods of development. Agile framework frameworks provide ways to coordinate software processes.

XP, Scrum and Agile patterns are coupled with other aspects of the life cycle of the device. The mixture of Scrum, XP and agile patterns, however, does not permit the direct installation of embedded systems. Some changes were required: (ii) promoting the development of software and hardware through a full implementation flow; I should adopt methods and instruments to optimize product design rather than designs that have no chance of meeting limitations; We have shown how pulse oximeter can be built to demonstrate that methods and instruments are used in this regard. For the purposes of estimating and partitioning, we used UML notation in this case study to define applications and to translate them in program-languages. We are searching for models that can provide a broad understanding of end-of-life at a high abstraction level and plan experimental methodological studies. With a conventional measurement system, we intend to introduce the technology in the industry gradually.

References

[1] A. S. Vicentelli and G. Martin. Platform-based design andsoftware design methodology for embedded systems. IEEE Design and Test of Computers, 18(6):23–33, 2001.
[2] B. Greene. Agile methods applied to embedded softwaredevelopment. Proceeding of the Agile Development Conference (ADC'04)., 2004.
[3] Chiranjeevi Aradhya, "Safety Critical Systems and Agile Methodology For Avionics Software Development", International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.8, Issue 12, pp.2461-2468, December 2020
[4] ChiranjeeviAradhya. "A detailed survey and evaluation of various aspects of embedded software robustness testing." International Journal of Advance Research, Ideas and Innovations in Technology 7.1 (2021).
[5] D. Gajski, F. Vahid, and S. Narayan. A system-designmethodology: Executable-specification refinement. Eu- ropean Conference on Design Automation, Paris, France,1994.
[6] P. Manhart and K. Schneider. Breaking the ice for agile development of embedded software: An industry experience report.Proceedings of the 26th International Conference on Software Engineering (ICSE04), page 3647, 2004.
[7] K. Beck and C. Andres. Extreme Programming Explained - Embrace Change. Second Edition, Addison-Wesley, 1999.
[8] K. Schwaber and M. Beedle. Agile Software Developmentwith Scrum. First Edition, Series in Agile Software Devel- opment, Prentice Hall, 2002.
[9] M. Oliveira. Desenvolvimento de Um Proto´ tipopara a Medida Na˜ o Invasiva da Saturac¸ a˜ o Arterial de Oxigeˆnioem Humanos - Ox´ımetro de Pulso.Master thesis, Center for Informatics at Federal University of Pernambuco, 2002.