Applying Agile Methods to Embedded Software Development: A Systematic Review

Mengjiao Shen, Wenrong Yang
Software Institute
Nanjing University
Nanjing, China
{tracy860703, ryangivy}@gmail.com

Guoping Rong, Dong Shao
Software Institute
Nanjing University
Nanjing, China
{ ronggp, dongshao}@gmail.com

***Abstract*—Since the 21st century, embedded software products have been pervasive in the world we live in, however, this emerging industry is facing many challenges due to its own unique characteristics which distinguish it from non-embedded software development. The other hand, agile software development methods are adopted by organizations worldwide at an increasing speed. It seems meaningful to explore to apply agile methods to embedded software development and in this paper a systematic review was conducted to supply an initial guide to the researchers and industrial readership.**

***Keywords*-embedded software development; agile method; agile practice**

I. INTRODUCTION

Since the twenty-first century, embedded systems have frequently emerged in peoples' daily work and life. To illustrate, the average home today has over 50 embedded systems [1]. Due to the large market demand and economic benefits, an increasing number of corporations and even researchers show great interest in how software development should be organized in order to deliver faster, better, and cheaper solutions and one of the answers is agile software development method. In fact, agile method has had a huge impact on traditional non-embedded software worldwide. Nonetheless, embedded software development has a number of unique characteristics which distinguish it from traditional non-embedded development and these characteristics bring new challenges.

Our group set up to explore the status quo about research of applying agile software development methods to embedded software development and we use the "systematic review" to guide our research. This systematic review seeks to evaluate, synthesize, and present the existing findings and we believe it will gives the latest state of research on applying agile software development method to embedded software development and is important to researchers who try to identify topic areas that have been researched or where research is lacking. Besides, we also believe that it will attract the concern of the embedded software industry.

The article is organized as follows: Section II, Introduces the systematic review protocol and shows some related research progresses in detail; Section Ⅲ, Reports the findings of the review; Section Ⅳ, Discuss some issues of concern; Section V, Conclusion and future work.

II. SYSTEMATIC REVIEWS

In this section, we describe the systematic review protocol [4] which guides and restricts our team members throughout the research. Meanwhile, we'd like to show the entire study progress to the readers. The systematic review protocol includes the following parts: identification of research questions, Search Strategy, inclusion and exclusion criteria, quality assessment and data extraction.

A. Research Question

The review aims to answer the following research questions.

RQ 1.What is the general information of embedded software development which is related to Agile Methods?
RQ 2.What is the current status of applying agile software development methods to embedded development?
RQ 3.What is the implications of these studies for the industry and the research community?

B. Search Straregy

We chose thirteen databases which we have the permission to use and the databases are listed in the Table I

TABLE I. DATABASES

ID	Database Name
1	IEEEXplore,
2	ISI Web of Science,
3	ProQuest,
4	ScienceDirect
5	SpringerLink,
6	Kluwer Online，
7	CiteseerX Library,
8	Wiley Inter Science Journal Finder
9	google
10	ACM Digital library,,
11	EBSCOhost
12	Inspec
13	Kluwer Online

TABLE II. EVALUATION OF SEARCH ITEM

Venue	Publisher	#Included	sensitivity	Precision
Agile 2003-2011	IEEE	4	4/4=100%	4/8=50%
Sigsoft	ACM	2	100%	100%

Final search item: (ASD OR DSDM OR (FDD AND software) OR (XP AND software) OR (Lean AND software) OR Xbreed OR Scrum OR RUP OR agile OR "Adaptive Software Development" OR (Crystal AND software) OR "Dynamic Systems Development Method" OR "Feature Driven Development" OR "Extreme Programming" OR "Rational Unified Process" OR EVO OR "Evolutionary Project Management") AND (Embedded OR mobile OR automotive). "RUP" is involved according to the [5]. We appraise the search item according to the [6] and Table II shows the result.

C. Study Selection Criteria and Proceures

Inclusion criteria:

- If the concrete agile method or even some agile practice is applied to embedded software development in the paper.
- If the paper generally states that applying agile method to embedded software development.
- If the paper expresses the attitude of suspicion or opposition to the opinion of applying some concrete agile process or agile practice to embedded industry.

Exclusion criteria:

- Exclude Non-English literature..
- Exclude the papers are absolute unrelated.

After the two steps of database search and study selection, we show the situation of accessed papers in the Table III ("ID" in the Table III is mapping to the "ID" in the Table I;"Presult" means the Preliminary results of the database search; "Rpaper" equals the relevant papers we selected from preliminary results according to the inclusion and exclusion criteria.).Removing the duplicate papers,we finally got 40 papers listed in the Table X.

D. Quality Criteria

we defined quality criteria to assess the quality of papers in Table V and the conclusion is show in Table IV.

TABLE III. SEARCH AND SELECTION RESULTS

ID	Rpaper	Presult	Remark
1	20	259	No keywords search
2	13	41	Just do title search
3	3	1714	No keywords search
4	1	53	Title, keyword and summary search
5	7	88	No keywords search
6	0	3	Title, keyword and summary search
7	4	113	Title, keyword and summary search
8	0	88	Title, keyword and summary search
9	20	149	Just do title search
10	25	212	No keywords search
11	1	119	No keywords search
12	12	43	Just do title search
13	0	3	Title, keyword and summary search

E. Data Extraction

Data Extraction is the most crucial step in the research, thus, we produced a data extraction *T*able XI to record several relevant information.

TABLE IV. CONCLUSION OF QUALITY ASSESSMENT

	1	2	3	4	5	6	7	8	9	10	Total
S1	1	1	1	1	1	1	1	1	1	1	10
S2	1	1	0	1	0	0	0	1	1	1	6
S3	1	1	1	1	1	1	1	1	1	1	10
S4	1	1	1	1	1	1	1	1	1	1	10
S5	1	1	1	1	0	0	0	1	1	1	7
S6	1	1	1	1	1	1	1	1	1	1	10
S7	1	1	1	1	0	0	0	1	1	1	7
S8	1	1	1	1	0	0	0	1	1	1	7
S9	1	1	0	1	0	0	0	1	1	1	6
S10	1	1	1	1	0	0	0	1	1	1	7
S11	1	1	1	1	0	0	0	1	1	1	7
S12	1	1	1	1	1	1	1	1	1	1	10
S13	0	0	0	0	0	0	0	1	1	1	3
S14	1	1	1	1	0	0	0	1	1	1	7
S15	1	1	1	1	0	0	0	1	1	1	7
S16	0	0	0	0	0	0	0	1	1	1	3
S17	1	1	1	1	1	1	1	1	1	1	10
S18	1	1	1	1	0	0	0	1	1	1	7
S19	1	1	1	1	1	1	1	1	1	1	10
S20	1	1	1	1	0	0	0	1	1	1	7
S21	1	1	1	1	0	0	0	1	1	1	7
S22	1	1	1	1	0	0	0	1	1	1	7
S23	1	1	1	1	0	0	0	1	1	1	7
S24	0	0	0	0	0	0	0	1	1	1	3
S25	1	1	1	1	1	1	1	1	1	1	10
S26	1	1	1	1	1	1	1	1	1	1	10
S27	1	1	1	1	0	0	0	1	1	1	7
S28	1	1	1	1	1	1	1	1	1	1	10
S29	1	1	1	1	1	1	1	1	1	1	10
S30	1	1	1	1	0	0	0	1	1	1	7
S31	1	1	1	1	1	1	1	1	1	1	10
S32	1	1	1	1	1	1	1	1	1	1	10
S33	1	1	0	1	0	0	0	1	1	1	6
S34	1	1	1	1	0	0	0	1	1	1	7
S35	1	1	1	1	1	1	1	1	1	1	10
S36	1	1	1	1	0	0	0	1	1	1	7
S37	1	1	1	1	0	0	0	1	1	1	7
S38	1	1	0	1	0	0	0	1	1	1	6
S39	1	1	1	1	0	0	0	1	1	1	7
S40	1	1	1	1	0	0	0	1	1	1	7

TABLE V. QUALITY CRITERIA:

1. Is the paper based on research or is it merely a "lessons learned" report based on expert opinion?
2. Is the aim of the research clearly stated?
3. Is there an adequate description of the context in which the research was carried out?
4. Is the research design appropriate to address the aims of the study?
5. Is there an adequate description of the sample used and the methods for identifying and recruiting the sample?
6. Is appropriate data collection method used and described?
7. Is the data analysis sufficiently rigorous?
8. Is there a clear statement of findings?
9. Are there no any limitations on results of the study reported?
10. Is the study of value for research or practice?

III. RESULTS

A. Characteristics of Embedded

TABLE VI. CHARACTERISTICS OF EMBEDDED

Characteristics	literatures
team members' skill	S40,S15
hardware dependencies	S3,S5,S9,S15,S28,S32,S40
competitive pressures	S10,S27,S31,S28,S38
limited resources	S40,S2,S3,S31,S9,S28,S5
change	S9,S12,S15
performance requirements	S3,S9,S15,S28,S31

The unique characteristics of embedded software development bring new challenges of applying agile methods to embedded industry, thus, it is necessary to know these characteristics to explore a great solution. Table VI summarizes six main characteristics.

- Team members' skill: Usually, embedded software developers are asked higher technical level not only software development abilities but also special domain knowledge than traditional software developers. However, one situation existed in [S15], [S3], [S28], [S31] and [S40] is that many embedded software engineers are not well trained software engineers and pick up programming as necessary to solve the task at hand.
- Hardware dependencies: The hardware change also brings change to software [S15]. In most cases, the hardware for the target machine is often developed concurrently with the software, and therefore not available until late in the project [S9]. This limits the progress of software development.
- Competitive pressures: Gary Mueller et al. (S27) said "The world has changed, with shorter and shorter market windows and increasingly sophisticated customers demanding more and more capabilities at lower and lower cost. Embedded system development must also adapt to these market imperatives to stay competitive". Fierce competition in the embedded industry inevitably brings some challenges such as time-to-market, budgetary, etc.
- Limited resources: Embedded systems can be termed as resource constraint systems [S2]. To illustrate, limited memory space, processing power, execution time and so on.
- Change: Change in the software industry can be caused by many various factors such as the changing requirement, hardware dependencies, inventions of competitors, customers' needs and so on. Specially, hardware change which is a typical representative of the embedded characteristics makes the change in the embedded software industry more frequent.
- Performance requirements: Parts of the embedded software demand high performance requirements. For example, real-time constraints (S3, S9, S15, S31) and the safety issues (S3, S9, S28, S31).

B. Overview of Studies

With respect to the concrete agile methods, we see from Table VII that 28 (70%) of the literatures were done on XP. Literatures on Scrum come next, with 12 (30%) of the literatures. Besides, we can see that 5 (12.5%) of the literatures studied only one practice. What's more, 9 (22.5%) of literatures (S3, S12, S15, S38, S16, S28, S31, S32, S40) involve both scrum and XP. [S38] is a mix of FDD, scrum and XP and [S13] supplies a hybrid method called Mobile-D which is a mix of XP, Crystal and RUP. Note: "General" in Table VII refers to studies on applying agile method to embedded software development in general.

In Table VIII, we concluded some famous techniques which are non-agile method or practice related. It helps us to better understand our concern about applying agile methods or practices to embedded software development.

- UCD: User-Centered Design is an approach to user interface design focusing on end-users throughout the whole process of producing a product [S22].
- Platform-based design approach: In [S3], [S28], [S31], [S40], a platform-based design approach is used to balance costs and time-to-market in relation to performance and functionality constraints. Simply to say, a platform is instantiated for a given product and then the product will be built in this platform.
- HW/SW co-design: An approach which emphasizes the sameness of the development process of hardware and software through the concept of collaborative design and mutual verification [S35].
- Testing techniques: It seems that nearly every literature we reviewed has a content of testing and many testing techniques are used such as the famous automated software testing tools.
- Others: stage-gate managed product development, requirements engineering, agile Simulation, etc.

TABLE VII. TYPE OF AGILE METHOD IN THE LITERATURE

Agile method	Literature. NO	Total
XP	S2, S3, S5, S6, S9, S10, S12, S13, S14, S15, S16, S19, S20, S21, S22, S23, S24, S25, S27, S28, S29, S31, S32, S34, S35, S36, S38, S40	28
Scrum	S1, S3, S7, S11, S12, S15, S16, S28, S31, S32, S38, S40	12
FDD	S38, S18	2
ASD	S30	1
RUP	S26, S13	1
Crystal	S13	1
TDD	S4, S17, S33, S39	4
Refactorng	S33	1
General	S7, S8	2

TABLE VIII. TECHNIQUES

NO	Technique	Literature.NO
1	User-Centered Design	S14,S21,S22,S34
2	platform-based design approach	S3,S28,S31,S40
3	HW/SW co-design	S35,S40
4	Testing techniques	Many

C. Extreme Programming

Some general information on literatures of XP:

a) S14, S21, S22 and S34 are written by the same author and cooperatively describe adopting one mix method in a mobile application.

b) S3, S28, S31, S40 separately applies the same proposed methodology to four different embedded project.

c) The number of literatures using each research method is listed in Table IX. 33.3% of total literatures are about mobile application which is a famous embedded field.

There is a survey [S32] involved 13 industrial organizations in eight European countries and 35 individual software development projects and it aims at the level of adopting of XP and how useful the XP is perceived to be. The level of adopting of XP is relatively good and nearly 90% of the responses where practices of XP had been applied (at least to some extent) could be considered positive. This survey result inspired us a lot.

There are many literatures present their views that XP can fit the challenges due to characteristics of the embedded software industry to a certain extent and some real successful cases prove it. Facing a situation that a majority of members in Intel firmware development team are not professional software engineers, Bill Greene (S15) decided to choose XP because that it is effective at conveying software development best practices to engineers with hardware backgrounds; Another challenge is that in embedded development, the hardware is usually developed concurrently with the related software and therefore not available until late in the project. XP has the ability to make meaningful progress prior to hardware availability and lower the change [S9]. Besides, [S35] shows how to apply XP to complement HW/SW co-design to manage the cost, development time and risk; [S15] said XP is perfect to accommodate the attitude of "embracing change". Both [S12] and [S20] express a similar idea that the team iteratively delivers product according to a customer's requirements who itself is obviously a part of the development team as the contact person to reduce communication's obstacles and manages the changing requirements into the development process. Last, in [S9], XP can be used to lower the cost of change for much of the embedded software development cycle.

The other hand, some literatures suggest that a slight adjustment of XP practices, tools, combining XP with some traditional methodology and other technical support are necessary. [S27], [S20], [S10] and [S25] emphasize that the XP should be adjusted slightly according to the organization and the project context in which it is applied; In [S25], "StoryManager" is a tool transfer manual XP requirement

TABLE IX. STUDIES BY RESEARCH METHOD

Research method	Literature.NO	Total
Single-case	S3, S10, S12, S14, S15, S19, S20, S21,S22, S27, S28, S31, S34, S35,S36S40	16
Multiple-case	S5, S29,	2
Survey	S32	1
Experiment	S6,	1
Mix	S25,	1
Other	S2, S9, S16, S23, S24, S38	6

management practices into an electronic form. Although the conclusion of the project showed the usefulness of " StoryManager" is not ideal, the idea that XP needs some tools support is advisable. [S9] also said lack of tools support could be the biggest obstacle of using XP to the embedded software development; [S29] successfully integrate XP with project management models of the stage-gate type which is a traditional complex project management approach; [S2], [S5] also show other technical support for XP.

D. Scrum

To start with, I have to give some general information of literatures on the scrum: there are a total of seven embedded software projects applied with Scrum and two of them are in the mobile application domain.

A survey in [S32] shows that 77% of the responses could be strictly positive, whereas nearly 11% of the responses referred to negative experiences of Scrum (not useful). Besides, the ones having actually applied the practices reported more positive experiences (77%), than were the positive expectations of the group that had no experience of the practices (28%).

It is worth mentioning that [S3], [S28], [S31], [S40], [S12], [S15] offer a mixed approach both involves scrum and XP methods and [S38] is based mainly on Feature Driven Development with practices and parts from Agile Modeling, Extreme Programming and Scrum. The cause of the existed phenomenon may be due to that the scrum is a project management framework whereas XP focuses more on project level activities of implementing software (S32, S15, S16), that means these two can complement each other.

Finally, both [S11] and [S15] support process customization according to the real embedded project and focus on the continuously improvement of the process.

E. Other Agile Methods

There are five literatures cover the rest of agile methods including FDD, ASD, RUP and Crystal and two of them ([S13] and [S30]) are mobile application development.

[S13] introduces a hybrid method named Mobile-D for mobile application development. Mobile-D is based on agile practices, drawing elements from XP, Crystal Methodologies and RUP (Although RUP is considered as non-agile method in [S13]). In [S26], a real time embedded software prototype is successfully developed by RUP methodology together with IBM-Rational I-CASE-E Suite and this case shows the possibility of applying RUP to develop embedded system.

FDD (Feature Driven Development) is an incremental, short-iterative and model-driven process consisting of five basic activities including Develop an overall model, Build a Feature List, Plan by Feature, Design by Feature and Build by Feature [S38]. [S38] presents a new process which is based on FDD and by suitable parts to XP, Scrum and Agile Modeling. The motivation for selecting FDD as a base process is that FDD has good features to offer to the embedded software development. Besides, Tomi Juhola, the author of [S38] stresses that FDD needs to be modified and enhanced. Furthermore, [S18] said that FDD may be suitable for embedded systems.

ASD (Adaptive Software Development) is only appeared in [S30] which provides a hybrid methodology design approach for designing a new agile methodology for mobile software development. The new agile methodology is highly influenced by the ASD method and NPD (New Product Development). Three elements of ASD are integrated into this new methodology: the Speculate-Collaborate-Learn cycle which is good at deal with uncertainties of software development; Quality review for quality assurance; component-based development practices for reusability.

F. TDD

Test-driven development has become popular with the introduction of the XP [S17] and the concept named "Test first" of extreme programming is equivalent to the TDD. Simply to say, TDD requires developers to create tests before writing the code itself [S38]. We can see a TDD cycle in [S17] which can be summarized into six steps: Add a test, Run all tests and see if the new one pass, Modify the code to make the test pass, Run tests again, Refactoring code, Repeat.

In the review, many literatures show that the TDD clearly has some difficulties to be adopted by the embedded development and the reasons are various:

a) Schedual Pressure: The practices of TDD were dropped because that schedule was paramount and product delivery was a top priority [S27][S17].

b) Negative Impression about TDD: TDD requires extra code and many developers believe that it is a waste of time [S17]. It is also difficult to convince a developer that unit testing before developing code is beneficial and necessary[S.27][S39].

c) Mindset Change: [S15], [S4] and [S17]think that it is still very unusual for the developers to turn from the sequence implement-document-test to test-implement-document.

d) The Complexity of the TDD itself: TDD is not learned in a one-day course and we suspect that mastery of TDD needs several months of intense use [S17].

e) Immature Development Environment: TDD relies on using an extensive set of tests that are constantly executed during development and in [S17], the poor result of applying TDD is partly due to the immature tool for implementing the TDD.

Although so many difficulties we faced in applying TDD to embedded software development, the advantages derived from TDD still attracts developers and researchers to try.

a) Increased Code Quality: Test-driven Development provides higher-quality code because the code is constantly tested [S10]. [S5], [S23], [S27] and [S39] also express a similar view.

b) Design and Coding Software: TDD aids design and coding software. Design test before coding may become a tool to support the coding itself [S15]. In [S39], all of the code was developed test-first and the tests helped us define precisely what each unit was responsible for and what other units we were dependent upon. [S4], [S5] and [S27] said test first helps developers understand the real requirements and [S4] also identifies the differences between work products and specifications as soon as possible. TDD [S38] makes the software incrementally developed and guarantees that every part of the software has been tested at least to some extent. This resulst in more instant feedback from the system too. Finally, [S17] said that TDD is good at logic testing and test set could be run to verify if the application broke or not.

c) Save Time: When adding functionality to the application, the developer could use the tests to see fit broke the existing functionality. Thus, TDD could save time at later development phases [S17]. TDD ensures much faster testing by automating tests, removing the time spent on repeatedly doing thorough tests of the system [S10].

d) Change: Both [S9]and [S10]expresses the idea that test-driven development joint with other design-centric practices of XP could support for constantly ongoing change during a project.

At last, some suggestions of better applying TDD to embedded software development are proposed as follows:

a) Make the developers believe that TDD is useful and worth to try [S20].

b) Make the TDD process easier to developers such as some technical support [S5], [S17], [S4].

c) Training of the TDD and a mentor are needed [S17].

d) Perhaps at the start, we may consider carrying out TDD mandatorily according to [S17], [S39].

IV. DISCUSSION

Our team believes that this review answered the research questions mentioned in II SYSTEMATIC REVIEWS to some extent. For RQ 1,we found that it is characteristics of embedded software development that bring new challenges of applying agile methods to embedded industry and we concluded this general information in A of III RESULTS. Besides, in the part B of III RESULTS, some digits imply the current status of applying agile software development methods to embedded development (RQ 2). XP seems to be most widely applied in embedded software development and Scrum comes next. Many literatures show that some agile method or practice brings power to embedded software development, however, it is inevitable to face some challenges and efforts. The information from C, D, E, F of III RESULTS is consistent with this view and some famous agile methods and practices in embedded software development are explained separately in detail. Last, the answer of RQ 3 is described in V CONCLUSION.

What's more there are three issues need a further explanation: Search item, literatures selection and second papers.

a) Search Item: We can see in B Search Strategy of II that the search item across two areas. The agile area is clearly consisted of concrete agile methods and their corresponding substitutions because that the concept of agile is too broad. Besides, we added the "agile" into the search item because the concrete agile method we know is not comprehensive and limited. The difficulty is another area of embedded and the process was complex, iterative and time-consuming. During the process, the most significant decision was to define this part as (embedded AND some specific domains of embedded applications) and an initial set of seven domains was produced by searching the hot topics concerned by several famous embedded conferences. Then, our team discussed to exclude five domains weighing the advantages and disadvantages. "Real-time", "Medical devices", "Multimedia" and "Middleware" were excluded mainly because we thought that "Embedded" is not their natural attribute and "Sensor" was excluded principally due to the "precision" [6]. After some cycles of search, the ultimate search item was decided and the appraisal of search item showed in Table II seems not bad.

b) Paper Selection: Part of the forty literatures reviewed do not mainly focus on Agile method, however, the project observed in the literature chose at least one agile method as the embedded software development process. Our team believed that such kind of papers is still valuable and decided to involve them.

c) Second Paper: We identified just two second studies ([2] and [3]).Although five literatures used in [2] belong to the set of forty literatures reviewed, our review is still worth doing. The focus of these two second literatures is not similar to our paper to a certain extent and both [2] and [3] don't supply a nice approach to guide the research. In contrast to the previous reviews, this review is guided by a systematic review.

V. CONCLUSION

Due to the limitation of time and page, we had to first report the most important items including characteristics of embedded, overview of studies, concrete agile methods involving XP, Scrum, FDD, ASD, RUP and Crystal in embedded and TDD practice.

This systematic review gives some implications to research and practice. Although faced with many challenges, most of the researchers and practitioners hold a positive opinion. The review also shows that the current state of theory and research on applying agile method to embedded software development is distinctly not mature and a more flexible, rigorous and effective research design is needed. Besides, the agile methods studied concentrates on XP and Scrum, other agile methods which are popular in embedded industry could be priority in the future researches. Specially, integration of XP and Scrum seems to be a trend. What' more, some real successful projects prove that it is necessary to actively collaborate with the embedded industry.

Some future works need to be done as follows:

a) Mobile Application: 14 (35%) of 40 literatures is applying agile method to mobile application development, thus, a conclusion of this specific domain is needed.

b) The rest of XP's Practices: We have tried to summary the practice of TDD and the other eleven practices may also deserve further study.

REFERENCES

[1] D. J. Eggermont, "Embedded systems roadmap 2002," STW Technology Foundation, Utrecht (The Netherlands), pp. 90-73461, 2002.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2] Jayakanth Srinivasan, Radu Dobrin, Kristina Lundqvist, "'State of the Art' in Using Agile Methods for Embedded Systems Development", COMPSAC '09: Proceedings of the 2009 33rd Annual IEEE International Computer Software and Applications Conference - Volume 02

[3] Oisín Cawley, Xiaofeng Wang and Ita Richardson, "Lean/Agile Software Development Methodologies in Regulated Environments - State of the Art ", Lean Enterprise Software and Systems Lecture Notes in Business Information Processing, 2010, Volume 65, Part 1, 31-36, DOI: 10.1007/978-3-642-16416-3_4

[4] Barbara Kitchenham ,"Procedures for Performing Systematic Reviews", Empirical Software Engineering National ICT Australia Ltd. Bay 15 Locomotive Workshop Australian Technology Park

[5] Scott Ambler. Overcoming the Myths of IBM Rational Unified Process (RUP) and Agile Development. http://www-01.ibm.com/software/info/television/html/M649306B47502P68.html

[6] Zhang, He; Babar, Muhammad Ali; Bai, Xu; Li, Juan; Huang, Liguo "An Empirical Assessment of A Systematic Search Process for Systematic Reviews", Evaluation & Assessment in Software Engineering (EASE 2011), 15th Annual Conference on

TABLE X. REVIEWED LITERATURES LIST

ID	Title	Author
S1	Guiding Global Software Development Projects using Scrum and Agile with Quality Assurance	Christelle Scharff
S2	Development Process of Mobile Application SW Based on Agile Methodology	Yang-Jae Jeong, Ji-Hyeon Lee, Gyu-Sang Shin
S3	TXM : An Agile HW / SW Development Methodology for Building Medical Devices	Lucas Cordeiro, Raimundo Barreto, ect.
S4	Breaking the Ice for Agile Development of Embedded Software: An Industry Experience Report	Peter Manhart, Kurt Schneider
S5	A Test-oriented Embedded System Production Methodology	Michael Smith, James Miller, Steve Daeninck
S6	Empirical Studies for the Application of Agile Methods to Embedded Systems	Dirk Wilking
S7	Transition to Agile Development Rediscovery of Important Requirements Engineering Practices	Juha Savolainen
S8	Systematic Piloting of Agile Methods in the Large: Two Cases in Embedded Systems Development	Jeanette Heidenberg, Mari Matinlassi, ect
S9	Extreme Programming and Embedded Software Development	James Grenning
S10	Mobile Systems Development: Exploring the Fit of XP	Ole Pedersen, Martin Lund Kristiansen,ect
S11	Scrum to Support Mobile Application Development Projects in a Just-in-time Learning Context	Christelle Scharff, Ravi Verma
S12	Supporting Agile Change Management by Scenario-Based Regression Simulation.	Christian Berger, Bernhard Rumpe
S13	Agile Development Methods for Mobile Applications	Andrei Cristian Spataru
S14	User Interface Design for a Mobile Multimedia Application: An Iterative Approach	Zahid Hussain, Martin Lechner, Harald, ect.
S15	Agile Methods Applied to Embedded Firmware Development	Bill Greene
S16	Embedded Agile	Timo Punkka
S17	Improving Business Agility Through Technical Solutions A Case Study on Test-Driven Development in Mobile Software Development	Pekka Abrahamsson, Antti Hanhineva, Juho Jaalinoja
S18	How to steer an embedded software project: tactics for selecting the software process model	Petri Kettunen, Maarit Laanti
S19	Embedded Agile Project by the Numbers With Newbies	Nancy Van Schooenderwoert
S20	Learning From Experience : The Analysis of an Extreme Programming Process	Sara Shahzad
S21	Agile User-Centered Design Applied to a Mobile Multimedia Streaming Application	Zahid Hussain, Martin Lechner, ect
S22	Integration of Extreme Programming and User-Centered Design: Lessons Learned	Zahid Hussain, Harald Milchrahm,, ect.
S23	A More Agile Approach to Embedded System development.	Michael Smith, James Miller, Lily Huang, ect.
S24	Extreme Programming for Mobile Applications	Pankaj Kamthan
S25	Improving Requirements Management in Extreme Programming with Tool Support – an Improvement Attempt that Failed	Jukka Kääriäinen
S26	Testing Critical Software: A Casestudy for an Aerospace Applicatin	Denis S Loubach, ect.
S27	Extreme Embedded A Report From the Front Line	Gary Mueller, Janet Borzuchowski
S28	A Platform-Based Software Design Methodology for Embedded Control Systems: An Agile Toolkit	Lucas Cordeiro, CarlosMar, ect
S29	Integrating agile software development into stage-gate managed product development	Daniel Karlstro¨m & Per Runeson
S30	Designing an Agile Methodology for Mobile Software Development: A Hybrid Method Engineering Approach	Vahid Rahimian, Raman Ramsin
S31	An Agile Development Methodology Applied to Embedded Control Software under Stringent Hardware Constraints.	Lucas Cordeiro, ect.
S32	Agile methods in European embedded software development organisations : a survey on the actual use and usefulness of Extreme	O. Salo,etc.
S33	Vision: Testing of Mechatronics Software using Agile Simulation	Steve Masticola, Michael Gall
S34	Concept and Design of a Contextual Mobile Multimedia Content Usability Study	Zahid Hussain, Martin Lechner, ect.
S35	The Partitioning Methodology in Hardware/Software Co-design Using Extreme Programming: Evaluation through the Lego Robot Project	Heeseo Chae, Dong-hyun Lee, Jiyong Park Hoh Peter
S36	Software Development Under Stringent Hardware Constraints: Do Agile Methods Have a Chance?	Jussi Ronkainen, Pekka Abrahamsson
S37	Taming the Embedded Tiger – Agile Test Techniques for Embedded Software	Nancy Van Schooenderwoert, Ron Morsicato
S38	Customized agile development process for embedded software development	Tomi Juhola
S39	Evolving into Embedded Development	Matt Fletcher, William Bereza, ect
S40	Agile DevelopmentMethodology for Embedded Systems: A Platform-Based Design Approach	Lucas Cordeiro, Raimundo Barreto, ect

TABLE XI. DATA EXTRACTION FORM

General Information		
Identifier		Unique identifier for the study
Data extractor		The reviewer that perform the data extraction
Bibliographic		Author, year, tile, source
characteristics		If there is any description of characteristics of embedded software development, record it here.
Methodology		Agile practice? Some concrete agile method? Or a mix new method ?
Real project		Record the real cases which have adopted the agile methodology above.
Challenge		Is there any challenge metioned in the literature?
Advantage/Disadvantage		Advantage/Disadvantage of applying the proposed agile mehtod to embedded software development.
1	***Agile***	description of agile element
	Detail	About the description of agile element appled in embedded software project in the paper .