Extreme Programming Certification Process using Confidence Grade

Issam Al-Hadid
Computer Information Systems
Isra University
Amman, Jordan
Email: issamfe [AT] hotmail.com

Suha Afaneh
Computer Science
Isra University
Amman, Jordan

Heba Almalahmeh
Management Information System
Isra University
Amman, Jordan

Abstract— **Extreme Programming (XP) is one of the most popular Agile Software development methodologies that promote early and quick production of working code. This paper proposes a certification process based on Validation and Verification using Confidence Grade to evaluate reliability and accuracy of the Extreme Programming Process, which improves the quality of the Software. The researchers have examined the proposed certification process through the implementation of two projects; one of them applied the proposed Certification steps, while the other project has implemented using the standard XP process. The results of the proposed process show a greater customer satisfaction and software quality improvement which has not been achieved in the standard XP project.**

Keywords-- Extreme programming, Certification, Validation, Verification, Confidence Grade

I. INTRODUCTION

This template, modified in

In the age of the large scale business information systems, Software Development Life Cycle (SDLC) has emerged as a framework that is used to structure, plan, and control the process of developing the information systems [1, 2]. Several Software development approaches and frameworks have been created, such as; Linear Development approaches (Waterfall), Iterative framework (Prototype, Rapid Application Development), combined Linear- Iterative framework (Incremental, Spiral), and Agile Approaches (Extreme programming, Scrum, Adaptive Software Development).

Agile Software Development is a group of software development methodologies that combines a philosophy and a set of development guidelines based on iterative and incremental development that encourages individuals' self-organization, motivation and interactions, as well customer collaboration and satisfaction based on responding to change, and early working software delivery [3, 4, 5].

Many Agile Software development processes have been suggested such as Adaptive Software Development (ASD), Dynamic Systems Development Method (DSDM), Scrum, Crystal, Feature Driven Development (FDD) and Extreme Programming (XP). XP has emerged as preferred Agile methodology that focuses on short development cycles and close interaction with customers to adapt requirements' changes as a natural and required feature of software development and provides a real time problem solving.

Sommerville [6] claimed that "XP is perhaps the best known and most widely used of the agile methods". Also Appelo [7] stated that "XP is often seen as complementary to Scrum, filling most of the holes that Scrum leaves wide open".

The Certification process, which is used to ensure that the target behavior and the expected characteristics of the XP processes are achieved, should be integrated within the development process to make sure that the entities of each process fulfill the expected target. The process entities include Input Data, Process and Output Data, in addition to the overall certification for the developed system as a whole. The certification has been defined by the International Organization for Standardization (ISO) as follows [8]: "Certification is a procedure by which a third party gives written assurance that a product, process, or service conforms to specified characteristics".

This definition in general can be applied anywhere to check if the product quality satisfies the requested, expected and the unexpected requirements, characteristics and behavior. Certification should be applied to all the steps; it means that a concurrent process should be run during the system's development steps which includes Verification and Validation (V&V) to make sure the product quality possesses the desired set of characteristics.

The XP Certification process should be conducted by a third party that has a formal recognition from an accreditation authority. This will check that Certification agent maturity includes the process and the team who will conduct the process. It should also be fully independent (technically, managerial and financial) [8].

According to Pressman [3]; "Verification refers to the set of activities that ensure that software correctly implements a specific function", and the "Validation refers to a different set

of activities that ensure that the software that has been built is traceable to customer requirements". On the other hand, Boehm [9] stated that: "Verification: Are we building the product right?" and "Validation: Are we building the right product?"

Validation and Verification (V&V) processes are used to ensure that the functional requirements are fulfilled, behavioral characteristics are accomplished, performance requirements are satisfied, usability, transportability, compatibility, error recovery, maintainability are met. In addition, to guarantee that the whole process and data are complete, consistent, no missing steps, accurate and realistic [3, 9]. This paper proposes a certification process based on Validation and Verification using Confidence Grade to evaluate reliability and accuracy of the Extreme Programming Process, which improves the quality of the software.

II. Related Work

A lot of process models defined a distinct set of activities, tasks, actions, and work products have been suggested by researchers and professionals to produce high quality software. Agile software engineering combines philosophy and a set of development strategies that encourage to change, deliver, make software frequently, collaboration between business people and developers, and simplicity [3, 10, 15]. There are many agile methodologies, processes, models, and modeling methods proposed by researchers. Highsmith [11] proposed the Adaptive Software Development (ASD) which is presented as a technique for building complex software systems. Dynamic Systems Development Method (DSDM) which is introduced by Stapleton [12] is based on the incremental prototype, provides the ability for building systems that meet tight time constraints. Scrum [13] is another agile method that proves effective response to the changing requirements, tight timelines, and business criticality during the development activities. Crystal agile methods created by Cockburn [14] and Highsmith [15], define a set of methodologies based on common elements, roles, processes, patterns, working products, and practices, which guarantee the primary goal of delivering working useful software. Feature Driven Development (FDD) is another agile process proposed by Coad, Lefebvre and DeLuca [16]. This process, which has been extended by Palmer and Felsing [17], is suitable to moderated sized and large projects. Extreme Programming (XP) suggested by Kent Back's [14], coding activity is based on the pair programming [3, 18] which provides a real time problem solving.

Valkenhoef [19] suggested XP optimization model that generates a release plan, based on structuring the planning problem and providing a suitable release plan which can be used to enable more informed decision by the customer representative and in projects where Plan-Driven approaches have been used. Alshayeb [20] investigated the relationship between the three XP engineering activities: new design, Refactoring and error fix. The researcher claimed that the Refactoring helps the project managers and developers to control software development using XP. The researcher also stated that "the more the new design performed the less Refactoring and error fix the programmers do (or perform)". Shahzad [21] suggested a continuous review for the development process in order to find the best appropriate way to perform the practices which are needed for software development activity.

Amey [22] suggested a static verification used with some XP practices in Critical Projects to improve Pair-wise programming and Refactoring practices.

III. Confidence Grade

Before you begin

According to Alegre, et al. [23], the Confidence Grade (C.G.) is a coding technique that uses the alpha numeric code to describe the reliability and accuracy of the data or process. (A1; for the best input/output data or the best process's behavior and characteristics, D6 for the worst) where the letter refers to the reliability and the number refers to accuracy. The technique is based on two bands to specify and describe the quality:

A. Reliability Band

according to Winschiers and Paterson [24], the probability that process behavior and characteristics perform the required function without failure for a predefined period of time under satisfied conditions in addition to the dependable of the input/output data. According the Alegre, et al. [25], the Reliability Bands are represented in Table (1).

TABLE I. Reliability Bands Codes

Brand Code	Description
A	Highly Reliable: Data based on the best available methods
B	Reliable: as in band A, but with minor shortcoming
C	Unreliable: extrapolation from limited samples
D	Highly unreliable: Unconfirmed verbal reports or cursory inspection or analysis

B. Accuracy Band

according to Alegre, et al. [25] Accuracy is the "approximation between the results and the correct values". It represents the quality and degree of conformity of being close to the data actual value or process expected behavior. Alegre, et al. [25] suggested the Accuracy Bands based on the intervals, as in Table (2).

TABLE II. ACCURACY BANDS INTERVALS

Brand Intervals	**Description**
[0; 1]	Better than or equal +/- 1%.
]1 ; 5]	Not band 1, but better than or equal +/-5%.
] 5; 10]	Not band 1 or 2, but better than or equal +/-10%.
] 10; 25]	Not band 1, 2 or 3, but better than or equal +/-25%.
] 50; 100]	Not band 1,2,3,4 or 5, but better than or equal +/-100%.
Values are out of the valid range; such as >100%	

For example, A2 is the C.G. for the input data. It means the data based on the best available methods (high reliable band A) is estimated to be within +/- 5% (Accuracy band 2).

TABLE III. CONFIDENCE GRADE MATRIX

Confidence Grade			
	Reliability Band	***Accuracy Band***	***Result***
Case 1	✔	✔	Confidence Grade is Achieved
Case 2	✔	✘	Accuracy Band is not achieved
Case 3	✘	✔	Reliability Band is not achieved
Case4	✘	✘	Confidence Grade is not Achieved

Where:

✔ (Achieved); equal or more than the Target

✘ (Not Achieved); less than the target

Applying the C.G. schema to the XP Certification process as the target for the V&V processes would increase the degree of the reliability and accuracy for each step of the XP's study and the data that will be collected (Input / Output Data). Therefore, the C.G. target should be assigned by a third party.

IV. PROPOSED EXTREME PROGRAMMING CERTIFICATION USING CONFIDENCE GRADE

The Extreme programming process goes through four main activities; Planning, Design, Coding and Testing. This process is performed by the XP team which includes both the customers and the developers. The researchers applied the Certification process using the V&V supported by the C.G. The proposed technique provides the Software Engineers with the required plan and schedule that control the Software accuracy and reliability. Figure (1) shows all the proposed XP steps with the certification proposed methodology. As mentioned earlier, the certification is given to the process in case the process achieved the V&V Confidence Grade's acceptance level (target), so the researchers have to run he V&V steps in concurrent with each phase to check the certification for the process.

A. Planning

The planning activity goes through three steps: User Stories, Release Plan and Iteration Plan.

1) User Stories:

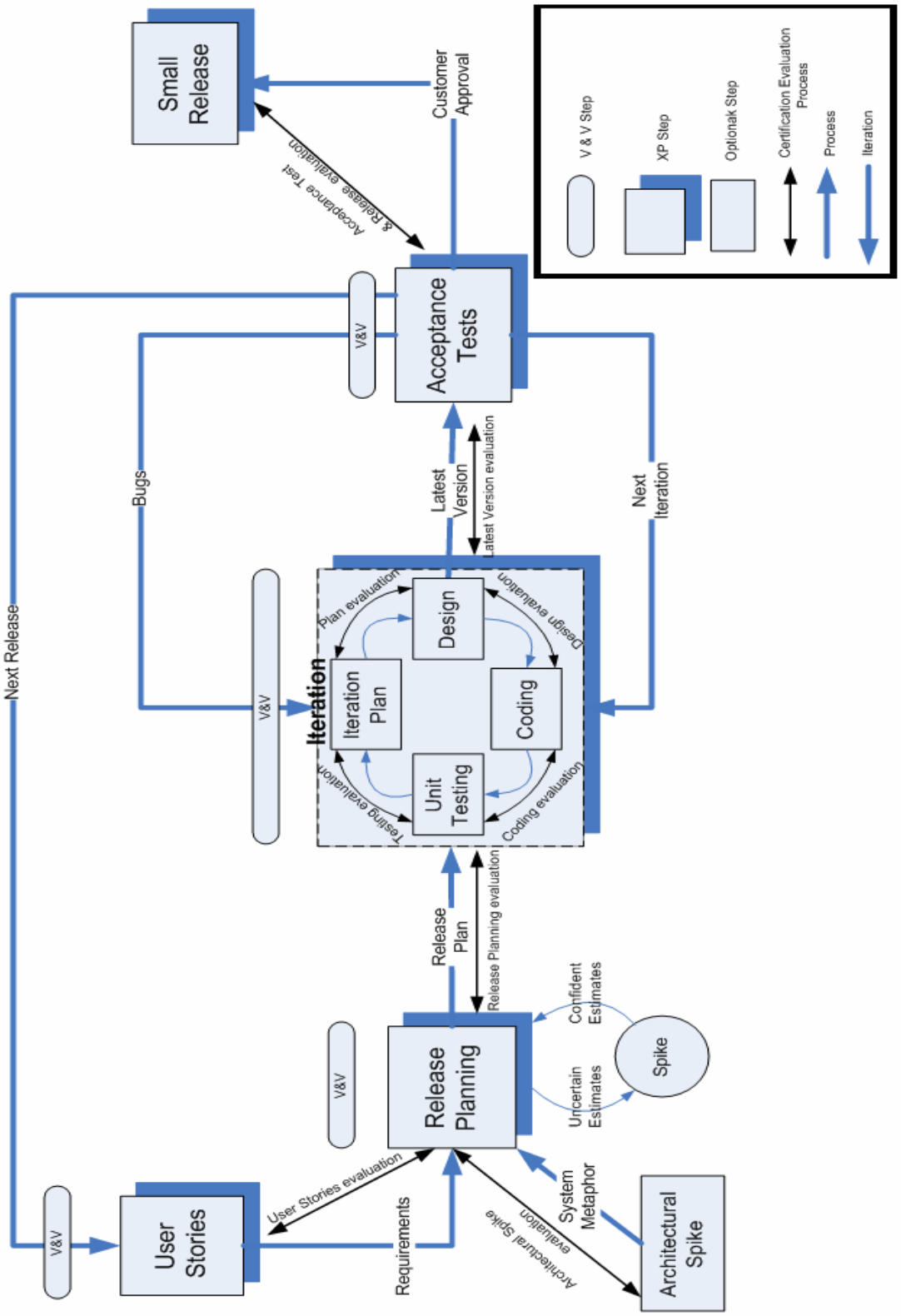
The customer writes the User Stories in cards, these stories describe the right system needs or functionality with right business priority without any techno-syntax terminology [3, 6, 26]. The XP team might use the spike architecture to develop the system metaphor which is used to simplify and explain the system design structure [19, 26]. The spike architecture includes the plan spike and the design spike. The plan spike aims to make more clear and detailed image about the user stories in order to get more accurate estimation of time, resources and budget. The design spike builds a prototype for the design in case of design problems to get a common view point for solution.

a) Verification: The Confidence Grade target is achieved if: the customers are writing the user stories right.

b) Validation: The Confidence Grade target is achieved if: the customers are writing the right user stories.

2) Release Plan

The customers and the developers decide together which User Stores are grouped into a Release. The dates are also specified in the Release Plan step. It is worth mentioning that the initial Release Plans are inaccurate in priorities or estimates, and these plans are regularly revised by the XP team. After delivering the first release, the project velocity is computed as the number of customers' stories is implemented in the first release [3, 26, 27].


Small Release
Customer Approval
Acceptance Test & Release evaluation
V & V Step
XP Step
Optionak Step
Certification Evaluation Process
Process
Iteration
Acceptance Tests
V&V
Bugs
Latest Version
Latest Version evaluation
Next Iteration
Next Release
Iteration
Plan evaluation
Design
Design evaluation
Iteration Plan
Coding
Testing evaluation
Unit Testing
Coding evaluation
Release Plan
Release Planning evaluation
Confident Estimates
Uncertain Estimates
Spike
Release Planning
System Metaphor
Architectural Spike
Architectural Spike evaluation
User Stories evaluation
Requirements
User Stories

a) Verification: The Confidence Grade target is achieved if: the XP team is planning the Release right.

b) Validation: The Confidence Grade target is achieved if: the XP team is planning the right Release.

3) Iteration Plan

The Iteration Plan starts at each iteration, where each iteration length is constant throughout the project (about 1 to 3 weeks), and this length is considered as the heart beat of the project. Each Release is divided into several iterations, where the customers select the desired User Stories for the next Iteration from the Release Plan according to the customer priority, and then the developers translate them into individual programming tasks. These tasks are written down on index cards in developers' terminology with schedule estimations for the current Iteration [3, 4, 26].

a) Verification: The Confidence Grade target is achieved if: the XP team is planning the Iteration right.

b) Validation: The Confidence Grade target is achieved if: the XP team is planning the right Iteration.

B. Design

The design activity is responsible for choosing the simple design, writing the Class- Responsibility- Collaborator cards (CRC), and using the Spikes and the Refactoring. The design is not a onetime thing but it is an all-the-time thing; there are design steps in Release planning, Iteration planning and Refactoring through the entire project.

1) Simple Design

Although it is hard to measure simplicity, and what is simple for one person, is considered hard for another, choosing a simple design that meets the current requirements is very critical and important, because it always takes less time to be finished than a complex one [6, 26, 27, 28].

a) Verification: The Confidence Grade target is achieved if: the Developers are choosing the Simple Design right.

b) Validation: The Confidence Grade target is achieved if: the Developers are choosing the right Simple Design.

2) Class- Responsibility- Collaborator cards (CRC)

CRC cards are written from the index cards created at the Iteration Plan step, each CRC card includes: the class name with the Super and Sub classes, the responsibilities of the class, and the names of other collaborating classes with this class to accomplish its responsibilities. The identification and organization of relevant classes of the current Iteration by using the CRC cards minimize the design's complexity, because it focuses on the essential details of the classes [26, 28].

a) Verification: The Confidence Grade target is achieved if: the Developers are writing the CRC cards right.

b) Validation: The Confidence Grade target is achieved if: the Developers are writing the right CRC cards.

3) Spike solutions

The spike solution is a very simple program or a design prototype to explore potential solutions of a different design problem, so it is not good enough to keep; it is just to reduce the technical problems [26].

a) Verification: The Confidence Grade target is achieved if: the XP team is designing the Prototype right.

b) Validation: The Confidence Grade target is achieved if: the XP team is designing the right Prototype.

4) Refactoring

The Refactoring process can be used to improve the internal structure of the code without modifying the external behavior [3]. it includes removing redundancy, eliminating unused functionality, and redefining obsolete designs to have a simple design, and to avoid needless clutter and complexity, and also to make sure everything is expressed once and only once. The importance of the Refactoring process is to keep the code clean and concise so that it is easier to understand, modify, and extend. this leads to save time and increase quality [26, 28].

a) Verification: The Confidence Grade target is achieved if: the Developers are Refactoring/enhancing the code right.

b) Validation: The Confidence Grade target is achieved if: the Developers are Refactoring/enhancing the right code.

C. Coding

The Coding activity starts with preparing the Unit Tests first, then the coding is performed in Pair Programming, and when it is finished, it should be integrated with consideration of Collective Ownership. The Coding Standards are agreed and followed by the developers, so all the code looks as if it is written by a single developer, and this makes it consistent and easy for others to read and Refactor [26, 27].

1) Coding the Unit Test first

Before starting the coding, a series of unit tests are developed to exercise each of the user's stories to be included in the current release [3]. Creating the unit tests first helps the developers to really consider what needs to be done (requirements) [26, 28].

a) Verification: The Confidence Grade target is achieved if: the Developers are preparing the Unit Test right.

b) Validation: The Confidence Grade target is achieved if: the Developers are preparing the right Unit Test.

2) Pair Programming

Pair Programming, where two programmers work together at one computer station, sitting side by side. This provides a mechanism for real-time code's problem solving [3]. Pair Programming increases software quality without impacting time to deliver [26]. It ensures that the production code is reviewed by at least another programmer, which leads to better

results in design, testing and coding. Knowledge sharing and skills improvement are also accomplished in Pair Programming [27, 28].

a) Verification: The Confidence Grade target is achieved if: the Developers are coding in Pair Programming right.

b) Validation: The Confidence Grade target is achieved if: the Developers are doing the coding right in Pair Programming.

3) Integration

All new code is released to the source code repository by taking turns, which means that only one development pair integrates, tests and commits changes at any given moment, so single threaded integration allows a latest version to be consistently identified [32]. As soon as work on a task is completed, it is integrated into the whole system [6]. Development proceeds in parallel while integration is sequential [26]. The continuous integration strategy helps to avoid compatibility and interfacing problems and helps to uncover errors early [3].

a) Verification: The Confidence Grade target is achieved if: the Developers are integrating the units right.

b) Validation: The Confidence Grade target is achieved if: the Developers are integrating the right units.

4) Collective Ownership

The pairs of developers work on all areas of the system based on the agreed Code Standards, so all the developers own all the code and anyone can change anything without affecting the code pattern [6]. Collective Ownership encourages everyone to contribute new ideas to all segments of the project, so any developer can change any line of code to add functionality, fix bugs, improve designs or refactoring, No one person becomes a bottle neck for changes. After changing the code, the unit tests are run successfully and passed before this code is released [26].

a) Verification: The Confidence Grade target is achieved if: the Developers are using the coding standards right.

b) Validation: The Confidence Grade target is achieved if: the Developers are using the right coding standards.

D. Testing

The Testing activity starts with implementing the Unit tests that are created in the previous activity, then the acceptance test which is the integration and validation test is implemented to make sure that all units are working together as expected [3, 28].

1) Unit Tests

First, the Unit Test Framework should be created or downloaded, to be able to create automated unit tests suites. Then, all classes in the system are tested. As mentioned before, Unit tests are released into the code repository along with the code they test, and if a code is discovered to be missing a Unit Test must be created at that time. Unit tests enable collective ownership, and re-factoring as well [26, 28].

a) Verification: The Confidence Grade target is achieved if: the Developers are testing the unit right.

b) Validation: The Confidence Grade target is achieved if: the Developers are testing the right unit.

2) Acceptance tests

Acceptance tests are specified by the customer and focus on overall system features and functionality that are visible and reviewed by the customer. These tests are derived from the user's stories [3]. During iteration, the user's stories selected during the iteration planning meeting, are translated into acceptance tests. The customer specifies scenarios to test when a user story is correctly implemented [26].

a) Verification: The Confidence Grade target is achieved if: the XP team is testing the Version right.

b) Validation: The Confidence Grade target is achieved if: the XP team is testing the right Version.

V. IMPLEMENTATION

This section describes how the proposed methodology is adapted by BelieveSoft Software Company: in developing new Point of Sale (POS) system. BelieveSoft dedicated two XP teams to develop the POS system; the first XP team (Group1) adapted the proposed certification process while the second XP team (Group2) applied the standard XP process.

A. POS System

The need for a stable and secured POS System that automates the sale process and helps the cashiers and managers initiate the POS project in BelieveSoft Software Company. Figure (2) illustrates the POS project involved parties using the proposed Certification process. The third party (Group of researchers) are responsible for conducting proposed certification process and making sure that Group1 adapts and applies the Certification process as required.

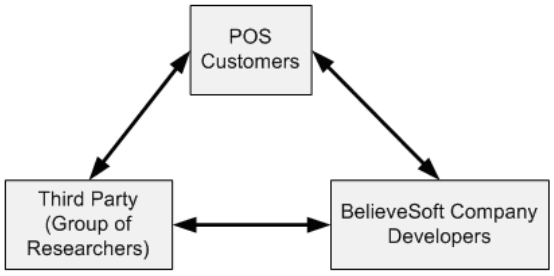


Figure 2. POS project parties for Group1

Table (4) shows the minimum planning phase Certification bands and intervals suggested by the third party to achieve the C.G which guarantee the quality of user's stories, in addition to implementation results achieved by the two groups, Release Plan and the Iteration Plan.

TABLE IV. MINIMUM PLANNING CERTIFICATION BANDS AND INTERVALS SUGGESTED BY THE THIRD PARTY AND THE IMPLEMENTATION RESULTS

	Min. Required Reliability Band	**Min. Required Accuracy Interval**	**G1 Results**	**G2 Results**
User Stories	B	]1, 5]	A2	B7
Release Plan	B	]5, 10]	B6	B15
Iteration Plan	B	]5, 10]	B3	B20

Figure (3) shows Group1 (G1) results which adapt the proposed Certification process and the results of Group2 (G2) using standard XP process.

The certification value for the user's stories developed by the G1 is A2, which means the stories based on the proposed method are high reliable and their accuracy is 95% or more. The G1 certification value is even better than the suggested value according to Table (4). On the other hand, the user's stories developed by G2 using the typical XP process achieved the certification value B7 which is less than the ranges suggested by the third party, which affects the customer's satisfaction after the system delivery.

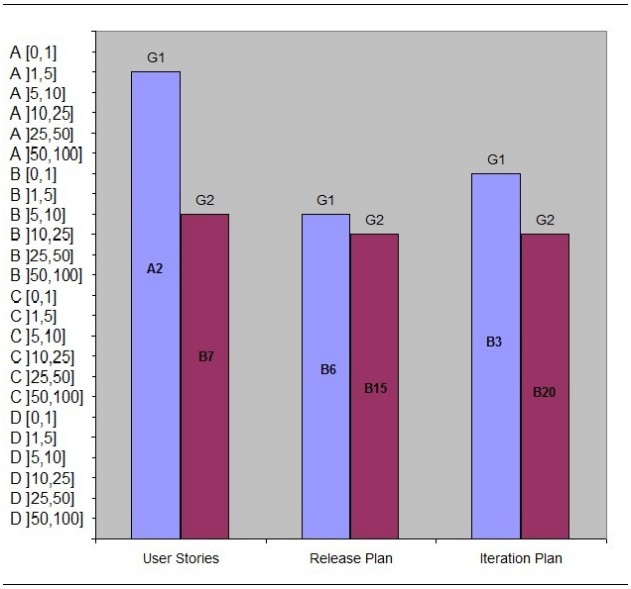


Figure 3. Planning Certification Results

Table (5) shows the minimum design phase certification bands and intervals suggested by the third party for the simple design, CRC, Spike and Refactoring and the results of the development groups (G1 and G2).

TABLE V. MINIMUM DESIGN CERTIFICATION BANDS AND INTERVALS SUGGESTED BY THE THIRD PARTY AND THE IMPLEMENTATION RESULTS

	Min. Required Reliability Band	**Min. Required Accuracy Interval**	**G1 Results**	**G2 Results**
Simple Design	B	]10, 25]	B12	B23
CRC	A	]1, 5]	A2	B9
Spike	B	]10, 25]	A5	B15
Refactoring	A	]5, 10]	A7	B11

The results after the design phase are shown in Figure (4). The results of G1 which applied the proposed process, achieved the suggested reliability bands and the accuracy intervals which are not achieved by the G2 which followed the typical XP process.

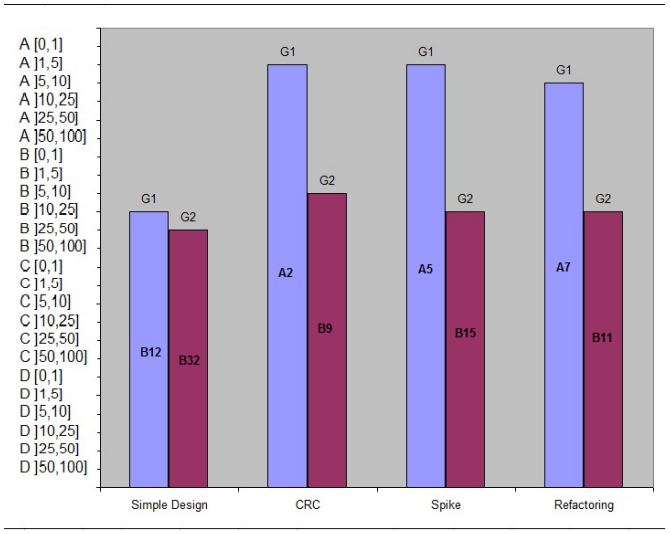


Figure 4. Design Certification Results

Table (6) shows the certification bands and intervals suggested by the researchers group to achieve the reliability and the accuracy for the coding phase. In addition, the table shows the implementation results for the two groups; G1 results for coding the unit tests first, Pair Programming, Collective Ownership and Integration results which are better than the results of G2.

TABLE VI. MINIMUM CODING CERTIFICATION BANDS AND INTERVALS SUGGESTED BY THE THIRD PARTY AND THE IMPLEMENTATION RESULTS

	Min. Required Reliability Band	**Min. Required Accuracy Interval**	**G1 Results**	**G2 Results**
Coding the Unit Tests First	A	]1, 5]	A2	B7
Pair Programming	B	]5, 10]	A5	B13
Integration	A	[0, 1]	A1	B3
Collective Ownership	A	[0, 1]	A1	B3

Figure (5) shows G1 results which adapted the proposed Certification process and the results of G2 that worked using standard XP process of the Coding phase.

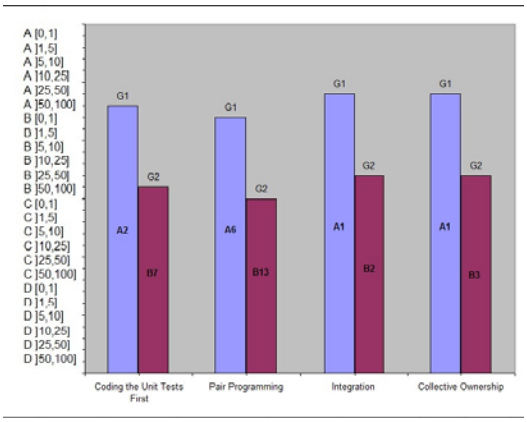


Figure 5. Coding Certification Results

Table (7) shows the minimum tests Certification bands and intervals suggested by the third party for the Unit tests and the acceptance tests as well the implementation results.

TABLE VII. MINIMUM TESTING CERTIFICATION BANDS AND INTERVALS SUGGESTED BY THE THIRD PARTY AND THE IMPLEMENTATION RESULTS

	Min. Required Reliability Band	**Min. Required Accuracy Interval**	**G1 Results**	**G2 Results**
Unit Tests	A	]1, 5]	A4	B6
Acceptance Test	B	]5, 10]	A8	B13

As illustrated in Figure (6), G1 Testing results using the proposed process are better than the results of G2 using the typical XP which are less than the minimum Testing Reliability bands and accuracy intervals.

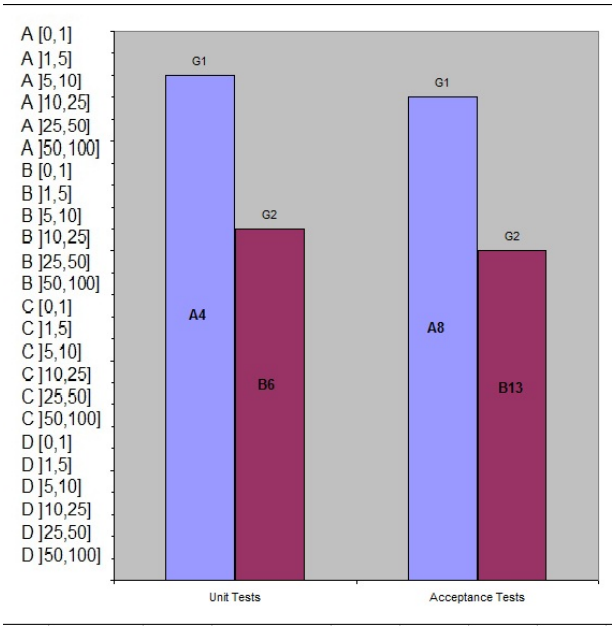


Figure 6. Testing Certification Results

Figure (7) shows the overall XP Certification phases, results for G1 and G2. The figure shows that G1 achieved the Certification targets for all the development phases which are not achieved by G2, where G2 achieved the Reliability band without the Accuracy band in the Planning and Design phases, while the bands are not achieved in the Coding and Testing phases.

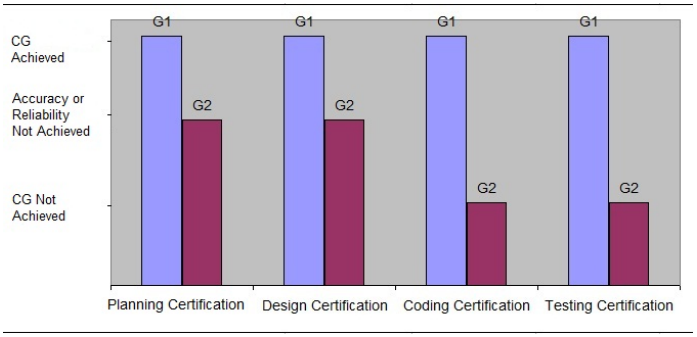


Figure 7. Certification Process Results

Figure (8) illustrates the projects development durations for both groups. It also shows that the time taken by G2 is less than the time taken by G1, which is a normal result of the extra conducting process used by the third party to certificate the processes.

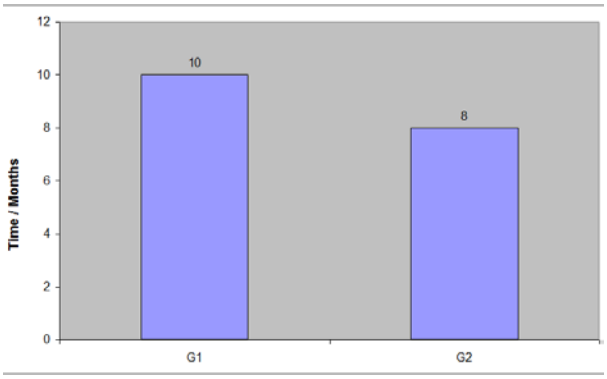


Figure 8. Projects Development Duration

Figure (9) shows the customer's satisfaction after the system delivery. The system delivered by G1 shows a greater satisfaction than the system delivered by G2, which means the customers' requirements have been fulfilled using the POS system developed by G1 using the proposed certification process.

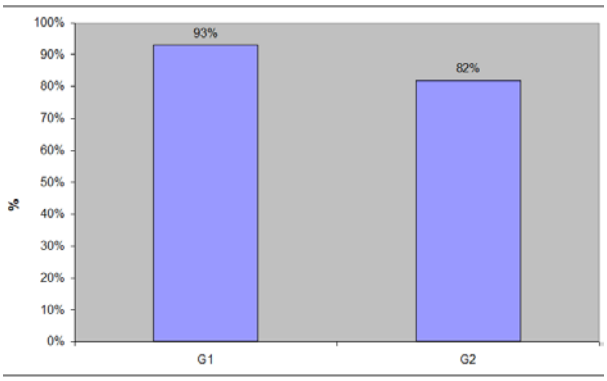


Figure 9. Customer's Satisfaction Percent

VI. CONCLUSION

The proposed Extreme Programming Certification Process using Confidence Grade approach improves the success of a software development project. As we can see, the certification process for the XP is not an easy process, and it should not be performed using the ad hoc manner; it should be planned, scheduled, and run concurrently within the software development life cycle. However, the researchers might face some barriers during the certification process such;

- A conflict between the Reliability and the Accuracy.

- Some processes are hard to estimate their reliability or accuracy.

- Some XP team members may not incorporate or provide the information to the Certification Agent (Third party).

The system developed by G1 showed a greater customer's satisfaction than the system developed by G2, it even took more time. Using the C.G. as a unified approach to evaluate the certification of the XP makes it easy to measure the Reliability and Accuracy of Software development life cycle steps and the System as a whole. These conclusions are limited to systems with similar scope, size, and context as the systems investigated. Future research is recommended to test more systems in similar and different contexts.

REFERENCES

[1] E. Geoffrey, Global Business Information Technology: An Integrated systems approach, Pearson Education, 2004, pp.87.

[2] P. Naur, B. Randall, Software Engineering: A Report on a Confernce Sponsored by the NATO Science Committee, NATO, January, 1969, pp.231.

[3] R. Pressman. *Software Engineering: A practitioner's Approach*, 6th Ed., McGraw-Hill, 2005, pp. 105-123,388.

[4] B. Kent. *Extreme Programming Explained: Embrace Change*, Addison-Wesley, 1999, pp. 32-41, 107-108.

[5] B. Kent, et al. Manifesto for Agile Software Development. Agile Alliance. Retrieved 1 Jan 2012 from http://agilemanifesto.org/.

[6] I. Sommerville, *Software Engineering*, 7th Edition, Addison-Wesley, 2006, pp.398-405.

[7] J. Appelo, "The Definitive List of Software Development Methodologies", Retrieved 4 Jan 2012 from http://www.noop.nl/2008/07/the-definitive-list-of-software-development-methodologies.html

[8] O. Balci, S. Saadi, Proposed Standard Processes For Certification Of Modeling And Simulation Applications, In Proc. of the Winter Simulation Conference, *IEEE*, December, 2002, Vol(2), pp.1621-1627.

[9] B. Barry. Software Engineering Economics, Prentic-Hall,1981, Pp 37.

[10] I. Jacobson, A Resounding 'Yes' to Agile Processes-But Also More, *Cutter IT Journal*, 2002,15(1), pp. 18-24.

[11] J. Highsmith, Adaptive Software Development, An Evolutionary Approach to Managing Complex Systems, Dorest House Publishing, 1998, pp.8-11, 70, 115-118.

[12] S. Jennifer, DSDM-Dynamic System Development Method, The Method in Practice, Addison-Wesley, 1997, pp. 65, 89.

[13] S. Ken, Beedle, Mike. *Agile Software Development with SCRUM*, Prentice-Hall, 2001, pp. 89-94.

[14] A. Cockburn, "Agile Software Development Joins the "Would-Be" Crowd," Cutter IT Journal, January 2002, Vol. 15 No. 1, pp. 6-12.

[15] J. Highsmith., "What Is Agile Software Development?," CROSSTALK The Journal of Defense Software Engineering, pp. 4-9, Oct.2002.

[16] C. Peter, L. Eric, D. Jeff, *Java Modeling in Color with UML*, Prentice-Hall, 1999, pp. 42-43.

[17] P. Steve, F. Mac, *A Practical Guide to Feature Driven Development*, Prentice-Hall, 2002, pp. 77-88.

[18] B. Kent, F. Martin. *Planning Extreme Programming, Addison-Wesley*, 2001, pp. 48-49, 63-69.

[19] G. Valkenhoef, T.Tervonen , B. Brock , D. Postmus, Quantitative release planning in extreme programming, *ELSEVIER, Information and Software Technology*,2011, 53: 1227-1235.

[20] M. Alshayeb , W. Li , An empirical study of relationships among extreme programming engineering activities, *ELSEVIER, Information and Software Technology*, 2006, 48: 1068-1072.

[21] S. Shahzad , Learning from Experience: the Analysis of an Extreme Programming Process, In Proc. 6th Int. Conference on information Technology, IEEE, 2009, pp. 1405 – 1410.

[22] P. Amey , R. Chapman , Static Verification and Extreme Programming, Praxis Critical Systems, uk, SIGAda'03, December 7–11, 2003, San Diego, California, USA. Copyright 2003 ACM 1-58113-476-2/03/0012

[23] H. Alegre, W. Hirner , JM. Baptista, R. Parena., Performance Indicators for Water Supply Services, 2000

[24] H. Winschiers, B. Paterson, Sustainable software development , SAICSIT '04: Proceedings of the 2004 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries, 2004.

[25] D. Wells, Retrieved 25 Dec 2011 from http://www.extremeprogramming.org/

[26] R., Jeffries, "What is Extreme Programming?", Retrieved 31 Dec 2011 from http://xprogramming.com/what-is-extreme-programming

[27] G. Succi, M. Marchesi, "Extreme Programming Examined", Addison-Wesley, 2001,, pp. 9-11, 85-110, 122-125.

[28] B., Kent., C. Ward, "A laboratory for teaching object oriented thinking", ACM SIGPLAN Notices (New York, NY, USA: ACM) 24 (10) 1989 , PP. 1–6.