

**Research Report**

**Agile methodologies for mobile projects**

Author: Bertocco Thomas-Killian - #13758322

Lecturer: Dr. Jeyaseelan Thirunavukarasu

CP5307 Advanced Mobile Technology | SP51 | 2020

30 April 2020

# Abstract

Agile methodologies are now used by a lot of companies, as they provide a viable solution for all types of projects, particularly those with varying requirements during development, such as mobile applications, as they need to meet a series of specific requirements and characteristics.

Nowadays, a lot of mobile applications satisfy users' needs to a certain degree because of their low development standards, since the use of software development methodologies is not considered necessary in this domain. As a consequence, developments for mobile devices have so far been carried out mainly in a haphazard manner and, in most cases, by individual developers who do not implement the software engineering methods that ensure their maintainability and thus their quality.

In this document, we will explain how to solve this problem using agile methods. We will cite several of them and point out their differences.

Table of Contents

[Abstract 2](#_Toc39260805)

[Introduction 4](#_Toc39260806)

[Fundamental knowledge 5](#_Toc39260807)

[Development methodology 5](#_Toc39260808)

[Agile methodologies 5](#_Toc39260809)

[Extreme Programing (XP) 6](#_Toc39260810)

[SCRUM 7](#_Toc39260811)

[Test Driven Development (TDD) 7](#_Toc39260812)

[Application development for mobile devices 9](#_Toc39260813)

[Mobile Application Development Overview 9](#_Toc39260814)

[Android 9](#_Toc39260815)

[iOS 10](#_Toc39260816)

[Agile methodologies in the development of applications for mobile devices 11](#_Toc39260817)

[Methodologies currently used for the development of mobile applications 11](#_Toc39260818)

[Mobile-D 12](#_Toc39260819)

[Hybrid Methodology Design 13](#_Toc39260820)

[Mobile Development Process Spiral 14](#_Toc39260821)

[Conclusion 16](#_Toc39260822)

# Introduction

Software engineering plays an important role in the development, portability, maintainability, functionality, reliability, and productivity of the software. Likewise, agile methodologies are processes to develop software quickly with great ease of adoption by teams (O. Salo & P. Abrahamsson, 2008). From February 11 to 13, 2001, a meeting of prominent members of the scientific community was held in Snowbird, Utah, where the term "agile methods" was born. This term implies being efficient and easy to handle (P. Letelier, J. H. Canós, & C. Penadés, 2003), which is very convenient for the size and agility of mobile application projects, which have been growing considerably in recent years, in the number of developments, but not in quality, this allows us to recognize that software methods must be adapted (K. Conboy y B. Fitzgerald, 2010) and involved in the development of applications for mobile devices if an optimal effect is to be achieved in this area (V. E. Jyothi & K. Nageswara Rao, 2012).

However, there is very little research so far on the use of agile methods in the development of applications for mobile devices, because of this, this research presents a synthesis of the development of applications, operating systems and development methodologies, for which, we have selected the three (3) most referenced agile methodologies (VersionOne, 2019), with a greater presence of documentation on the Internet and oriented to small developments of applications for mobile devices, such as Extreme Programming (XP), Scrum and Test Driven Development (TDD), also presents a summary of methodologies currently used for such developments. In the first part, some characteristics of the most used agile methodologies are exposed; in the second part, some generalities of the development of applications for mobile devices will be exposed as well as some characteristics of their operating systems, to conclude with a review of the methodologies currently used in the development of applications for mobile devices.

# Fundamental knowledge

This chapter shows a brief description of the software methodology and agile methodologies, focusing on the three (3) agile methodologies with the greatest presence of documentation, adaptable, with good acceptance within the development teams (O. Salo & P. Abrahamsson, 2008), (P. Abrahamsson, J. Warsta, M. T. Siponen, & J. Ronkainen, 2003), and adaptable to the specific characteristics of the development of applications for mobile devices.

## Development methodology

"A methodology is a collection of procedures, techniques, tools, and supporting documents that help software developers in their efforts to implement new information systems. A methodology is made up of phases, each of which can be divided into sub-phases, which will guide system developers in choosing the most appropriate techniques at each stage of the project and also in planning, managing, controlling and evaluating it" (D. E. Avison & G. Fitzgerald, 2006).

In this definition, Avison and Fitzgerald present a description of development methodologies and highlights its main components, phases, tools, and techniques. However, a methodology is more than a collection, since it is based on a philosophy, distinguishing itself from methods or simple recipes, which mark a few steps to follow, and that is it. Thus, the methodologies differ either by the number of phases, the techniques of each phase, the content of the phase or in its philosophical basis; all this applies, depending on the context of development, project size or teamwork, organizational culture, among other aspects, by which in the case of mobile developments, is of the vital importance of their selection, to guarantee a product of quality.

## Agile methodologies

The methodologies, in general, are classified according to their focus and essential characteristics. The most recent ones, which were developed at the end of the last century and have started to manifest themselves since the beginning of the current one, have been called "agile methodologies" and have emerged as an alternative to the traditional ones, These methodologies are derived from the list of principles found in the "Agile Manifesto" (Agilemanifesto, 2001), and are based on an iterative development that focuses more on better capturing changing requirements and risk management, breaking the project into iterations of different length, each generating a complete and deliverable product; and incremental where a product is built block by block during the whole life cycle of product development, the individual iterations must produce some fully functional or improved feature (V. Szalvay, 2004) its main objective is to reduce the development time, in the same way as with the cascade or waterfall model that was introduced by Royce in 1970 (W. Royce, 1970) and initially used for software development, but expanded by Boehm in 1981 (B. W. Boehm, 1981), where all requirements are analysed before starting to develop, however, requirements are divided in independently functional "increments". Many ideas that are raised in agile methodologies are not new, most of them were already reflected by Brooks in his book, The Mythical Man-Month (F. P. Brooks, 1995), and in great part, they respond to common sense. Some authors consider that a circle has been completed that began with a reaction provoked by multiple factors and pointed out temporarily by Dijkstra's manifesto, in which a call to discipline was made and which closes with the already famous Manifest for Agile Software development, a petition for the relaxation of processes for people (Agilemanifesto, 2001).

The emergence of agile methodologies cannot be associated with a single cause, but a whole set of They, however, most authors relate it to a reaction to traditional methodologies, what were the causes of this reaction, the factors that are commonly mentioned are the heaviness, slowness of reaction and over-documentation, in short, lack of agility of formal development models; another important point would be the explosion of the network, the Web and mobile applications, as well as the notorious growth of the open-source movement.

To all this can be added a fairly important change, in terms of market demand for software, which is increasingly oriented to the Web and mobile devices, with very volatile and constantly changing requirements, which require increasingly shorter development times, causing companies to look more to new developers, with new "amateur" methods that are combined with techniques of formal methodologies. The development models of the open-source communities could certainly determine the emergence of agile methodologies, but each author determines the emergence of agile methodologies in different ways.

## Extreme Programming (XP)

It focuses on best practices for the development of software. It consists of twelve practices: the planning game, small emissions, the metaphor, the design simple, testing, refactoring, scheduling in pairs, collective ownership, continuous integration, week 40-h, customers on-site, and standards of coding (K. Beck & J. Zapata Martínez, 2002). The revised version "XP2" consists of of the following "primary" practices: sitting together, equipment, informative workspace, the work of energy, paired programming, the stories, the cycling weekly, the quarterly cycle, workflows, 10-minute construction, continuous integration, test programming, and incremental design. There are also 11 corollary practices (K. Beck & C. Andres, 2004), this extended version of XP was introduced in 2004 by Beck and others; however, very few investigations have focused on the new version, these could be simply due to the fact that any method takes time to gain strength and popularity, and that can only be a matter of time before the use of the revised version reaches the same levels than the use of the original.

Extreme Programming is described by Beck as "...a lightweight methodology for small to medium-sized software development teams in the face of imprecise or rapidly changing requirements..." (K. Beck & J. Zapata Martínez, 2002), explicitly recognizes that XP is not a set of revolutionary new development techniques. Rather, it is a set of proven and reliable principles, well established as part of the conventional wisdom of software engineering, but taken to an extreme level hence the name "extreme programming."

Most of the existing literature on XP suggests that its methods can be easily adapted (J. Bowers, J. May, E. Melander, M. Baarman, & A. Ayoob, 2002) (J. Rasmusson, 2003). However, Conboy (K. Conboy & B. Fitzgerald, 2010) presents a very detailed study, which proposes XP as a series of puzzle pieces that fit together like a Swiss watch, despite the fact that XP is supposedly adaptable to a wide variety of projects, some of which are practical but do not add value, they are impossible to eliminate as they are necessary to keep the others in place.

## SCRUM

One cannot speak of Scrum without mentioning Takeuchi and Nonaka (H. Takeuchi & I. Nonaka, 1986) where they present an adaptive, rapid and self-organized process of product development and expose for the first time the term Scrum which is derived from the same term in rugby and refers to how a ball that has been thrown off the pitch is returned to the pitch in a collective manner. Scrum emerged as a practice in technological product development, and it was not until 1993 that Jeff Sutherland applied the model to software development at Easel Corporation, as shown by Avison and Fitzgerald (D. E. Avison & G. Fitzgerald, 2006). Scrum focuses project management on situations where it is difficult to plan for the future, with "empirical process" control mechanisms, where feedback loops are the central element. The software is developed by a self-organizing team in increments (called "sprints"), starting with planning and ending with a commentary. The features to be applied in the system are recorded in a backlog. The owner of the product then decides which elements of the backlog should be developed in the next sprint. The team members coordinate their work in a stand-up diary of the meeting. One team member, the "Scrum Master" (equivalent to the project manager), is in charge of solving the problems that prevent the team from working effectively (K. Schwaber & M. Beedle, 2002). This team is usually made up of ten or fewer members, although Schwaber and Beedle recommend teams of five (K. Schwaber & M. Beedle, 2002), dividing the main team into smaller teams if necessary.

Even with the recommendations presented by Schwaber and Beedle (K. Schwaber & M. Beedle, 2002), for current projects and more specific developments for mobile devices, the development team is facing problems of variable requirements and technologies with very different characteristics. In this situation, it is recommended that the first sprint has a functionality implemented with the technology that is giving problems, will continue with a product backlog, prioritizing the tasks to be executed, in order to boost morale for the developers and the whole team in general. Scrum is a very popular methodology, and we can find many companies who like to use it, among them we can find companies as important as Yahoo or Google. In 2000 Rising and Janoff published three projects successfully using Scrum (L. Rising & N. S. Janoff, 2000).

## Test-Driven Development (TDD)

Test-oriented development conditions the mentality of the developers by guiding them through the development and focusing on the quality of the final product, according to Astels (D. Astels, 2003) is a style of development where maintains an exhaustive programmer's test suite, no part of the code goes into production without unless you pass your associated tests, you write yourself first the tests and these determine the code that is needed writing and because of his radical approach to writing code, drastically changes the mentality of any development team, usually streamlining the results and increasing the quality of the system.

TDD is sometimes understood as a quality assurance procedure and originally was thought as a technique to improve productivity, the increase of quality was a side effect, that is why nowadays we can find many experiences in which TDD has been used as part of Extreme Programming, it is more difficult to find experiences in which the use of TDD is documented as an isolated methodology, usually appears always complementing another methodology (K. Schwaber & J. Sutherland, 2019) (P. Letelier, J. H. Canós, 2003).

Test-Driven Development is one of the most popular methodologies in the professional field and continues to expand due to its good results. The current trend is to integrate TDD independently in any methodology whether agile (K. Schwaber & J. Sutherland, 2019) or traditional (P. Letelier, J. H. Canós, 2003) and take advantage of the benefits of practicing a methodology that always allows for undoing mistakes, ensuring product quality and protect against errors both malicious as humans.

# Application development for mobile devices

## Mobile Application Development Overview

The market for mobile applications has expanded rapidly over the past ten years, mobile platforms continue to improve their performance, and users' need for a wide variety of mobile applications is increasing (K. Restivo). Software development for mobile platforms comes with unique features and constraints that apply to most stages of the life cycle. The most important distinguishing characteristics are identified in (P. Abrahamsson & Co.) (M. Satyanarayanan, 1998) (P. Abrahamsson, 2005) such as a high level of competitiveness, short delivery time, mobility, portability, specific and constantly changing capabilities of the terminals, different and incompatible operating systems, between others.

Abrahamsson (P. Abrahamsson, 2005), presents a series of characteristics to be considered when developing software for mobile devices, as they are, that software is released in an uncertain and dynamic environment with a high level of competence, the teams developing mobile applications are generally small and medium companies, the applications themselves are small in size, are delivered in fast versions in order to satisfy the demands of the market and address a large number of end-users. The author suggests that in the development of software applications for mobile devices the development teams must meet the challenge of a dynamic environment, with frequent changes in the needs and expectations of the customer (P. Abrahamsson, 2007), therefore, must be developed with cycle-oriented approaches relatively short development periods, typical of agile methodologies for software development.

## Android

It was acquired by Google in 2005 when it bought the firm Android Inc. in order to ensure that a mobile operating system (OS) could be created and maintained on an open platform (M. Burton & D. Felker, 2012), since then Google invests a great deal of time and effort each year resources in the Android project, which has already to be a very profitable business. In 2019, Android was present on 2.5 billion active devices, and at the same time, the world level achieved a market share of 70% during the fourth quarter of 2019, more than double the iOS from Apple, Inc. (R. Brandom, 2019) (Statcounter, 2019). But also for any application programmer for mobile devices, since Android users may not know who the programmer is, but they know what Google is, and they trust him (M. Burton & D. Felker, 2012). Because its application lies in the Android Market (which is controlled by Google) and/or Google Play, Google assumes responsibility for the quality of its application and, therefore, of its distribution. Android is a mobile operating system based on Linux, which made its formal debut in 2008 at HTC's G1 and is currently developed by the Open Handset Alliance, which is led by Google (Statcounter, 2019).

One of the unique features that Google has, to ensure the quality of the applications available on your Android Market, a feature that also shared with Apple's iOS, is a "kill switch" that allows remote and global deletion of applications that are deemed unsuitable for their platforms (Network World, 2010).

Google has used this service once, but it did in a transparent manner and for a good reason (S. Hollister, 2011).

## iOS

Apple's mobile operating system, designed for your mobile devices. Initially developed for the iPhone and presented to the public on January 9, 2007, it was launched on June 29, 2007. It had a closed platform that only allowed the installation of native applications developed by Apple (T. Ricker, 2011), but on July 10, 2007, a group of hackers called the iPhone Dev Team launched the PwnageTool application to do jailbreak, a process that eliminates the limitations imposed by Apple on devices using the iOS operating system through the use of modified kernels (C. Miller & Co., 2012); the IOS was later implemented for the iPod Touch, iPad and Apple TV (M. Antic, S. Jovanovic, & S. Cvetanovic, 2013).

During the Worldwide Developers Conference ( WWDC ) in June 2019, Apple gave the world the first glimpse of iOS 13, the latest operating system update company's mobile phone, this version has a lot of new features and security improvement, according to Apple (Apple, 2019), but the iOS 14 is already expected to be released in September 2020.

As for statistics, in the fourth quarter of 2019, the IOS operating system widened the gap with Android taking only 28% of the market, compared to 70% of Android (R. van der Meulen & Janessa Rivera, 2012).

Faced with the problem of security, says Barak Shrefler IT director and security manager at the Hospital Hadassah University in Jerusalem, "Evaluating Android and we think it's more vulnerable than the iOS." (E. Messmer, 2012) although other experts express their distrust of the IOS jailbreak.

# Agile methodologies in the development of applications for mobile devices

The use of agile software development methods has received both support and opposing arguments. The main argument against it is the lack of validation.

There is also a certain amount of uncertainty in distinguishing agile methods of ad-hoc programming. However, some authors defend the theory that agile methods provide an organized development approach (O. Salo & V. Teknillinen Tutkimuskeskus, 2006).

When it comes to comparing the characteristics of mobile applications to those of an agile method, the difficulty stems, in part, from the fact that the limits of agile methodologies are not clearly established. An overview of research in this field is presented in (T. Dyba & T. Dingsoyr, 2009), where the authors divide their research into four categories: introduction and adaptation, human and social factors, the perception of agile methods, and comparative studies. The results indicate that the introduction of agile methods to small software projects generates great benefits, especially if agile practices do not completely replace the traditional ones, but they work together.

Another point of view (P. Abrahamsson, 2005), Abrahamsson, makes a direct comparison between the characteristics of the agile method and characteristics of mobile applications, focusing on the amount of documentation produced, the planning involved, the size of the development team, client identification, and object orientation. Except for customer identification, all the characteristics of agile methodologies are suitable for the development of mobile applications. The customer can be identified as the software distributor; however, especially in the case of mobile applications, the problem of customer identification is much more complex.

## Methodologies currently used for the development of mobile applications

A new development methodology, specially designed for the development of mobile applications, is called Mobile-D and is proposed by Pekka Abrahamsson and his VTT team ( Valtion Teknillinen Tutkimuskeskus, Technical Research Centre of Finland) in Finland who lead a very important agile development (P. Abrahamsson, 2007) very focused on mobile platforms, and which is presented in more detail in(P. Abrahamsson & Co.). The method is based on agile practices such as Extreme Programming and crystal; the practices associated with Mobile-D include test-based development, programming in pairs, continuous integration and refactoring, as well such as software process improvement tasks; according to Abrahamsson (P. Abrahamsson & Co.) Mobile-D should be used by a team of no more than ten developers, working together to supply a ready-made product in a maximum period of ten weeks.

Another very important point of view at present is the Rahimian, and Ramsin proposal (V. Rahimian & R. Ramsin, 2008) HMD (Hybrid Methodology Design), which relies on a combination of adaptive software development (Adaptive Software Development, ASD) and the design of new products (New Product Development, NPD), part of the traditional life cycle (analysis, design, implementation, testing, and development) and also includes marketing.

The recent proposal of methodologies designed specifically for mobile applications is called Mobile Development Process Spiral (A. Nosseir & Co., 2012), which is a model-driven by the usability and takes the spiral model as a basis.

## Mobile-D

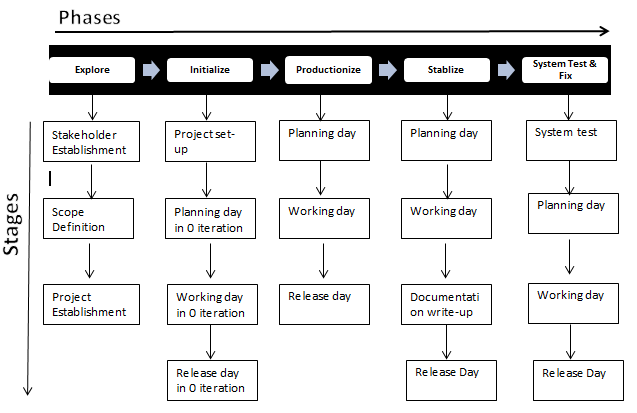


Figure 1: Mobile-D Methodology process

Mobile-D consists of five phases: exploration, initiation, production, stabilization, and testing of the system. Every one of these phases has a number of stages, tasks, and associated practices. The full specifications of the method are available in (VTT, 2012). In the first phase, Explore, the development team should generate a plan and establish the characteristics of the project. This is done in three stages: the establishment of actors, definition of the scope, and establishment of projects. The tasks associated with this phase include the establishment of the client (clients who take an active part in the process the initial planning of the project and the collection requirements, and the establishment of processes.

In the next phase, initiation, the developers prepare and identify all necessary resources.

Plans are prepared for the following phases and establish the technical environment as the physical resources, and communications (including training of the development team). This phase is divided into four stages: the implementation of the project, the initial planning, trial day, and departure day.

In the production phase, the scheduling of three days (planning, work, release) is repeated iteratively until all the functionalities are implemented.

First, the work iteration is planned in terms of requirements and tasks to be performed. Tests are prepared for the iteration beforehand. The tasks will be carried out during the working day, developing and integrating the code with existing repositories. During the last day, the system integration is carried out (in case of that, several teams were working independently) followed by acceptance tests.

In the stabilization phase, the last integration actions to ensure that the system complete works properly. This will be the most important in multi-team projects with different subsystems developed by different teams. In this phase, the developers will perform tasks similar to those that were to be deployed in the "production" phase, although in this case, all the effort is directed to the integration of the system. In addition, the following can be considered at this stage the production of documentation.

The last phase (system test and repair) is aimed at the availability of a stable and fully functional version of the system. The finished and integrated product is tested against customer requirements, and all defects found are eliminated.

## Hybrid Methodology Design

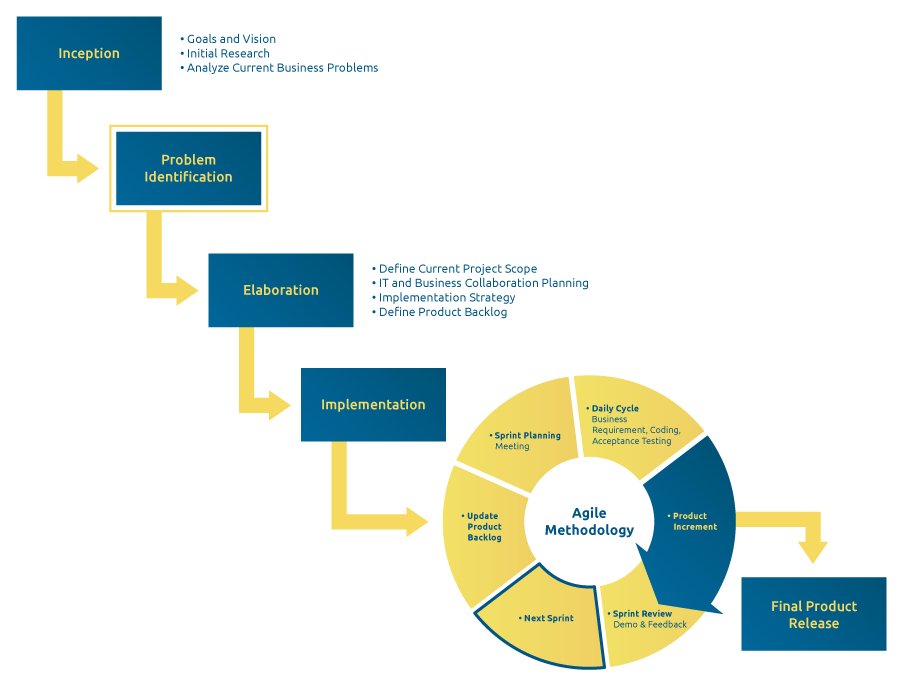


Figure : Example of Hybrid Methodology process

This methodology uses the iterative, incremental model for the development process to achieve rapid software delivery and improve risk management capabilities. Some of the agile features that are highlighted and also aligned with mobile application development needs are as follows (S. B. Kaleel & S. Harishankar, 2013):

* Evidence-based development.
* Ongoing customer involvement.
* Prioritization of requirements.
* Effective communication.
* Quality assurance.
* Expert developers.
* Review of the whole process and learning sessions.
* Adaptation process.

This methodology is part of the traditional life cycle, and its specifications are available in (V. Rahimian & R. Ramsin, 2008). The first iteration divides the analysis phase with the intention of mitigating development risks; likewise, the design is also segmented to introduce some architecture-based design. Implementation and testing, however, are merged by introducing test-driven development (TDD) concepts.

There is also a commercialization phase, towards the development of products that are imposed in the scenario of application development for mobile platforms. From a methodological point of view, the authors claim to have relied on metamodels such as SPEM (Software Processes Engineering Metamodel, supported by the Eclipse development environment) and OPF, (Open Processes Framework), as well as generic object-oriented life cycle concepts such as OOSP (Object-Oriented Software Processes).

The second iteration performs integration of certain parts of the NPD (New Product Development) models, adding the generation of ideas at the beginning of the cycle and a market test before launching the marketing phase. The third iteration directly integrates the "engine of development" of adaptive development methods (ASD) very oriented to quality assurance in development processes with the idea of having the physical architecture early in the process.

In the fourth iteration, prototyping elements are added; the project initiation phase is also refined, based on the same element of the adaptive processes.

## Mobile Development Process Spiral

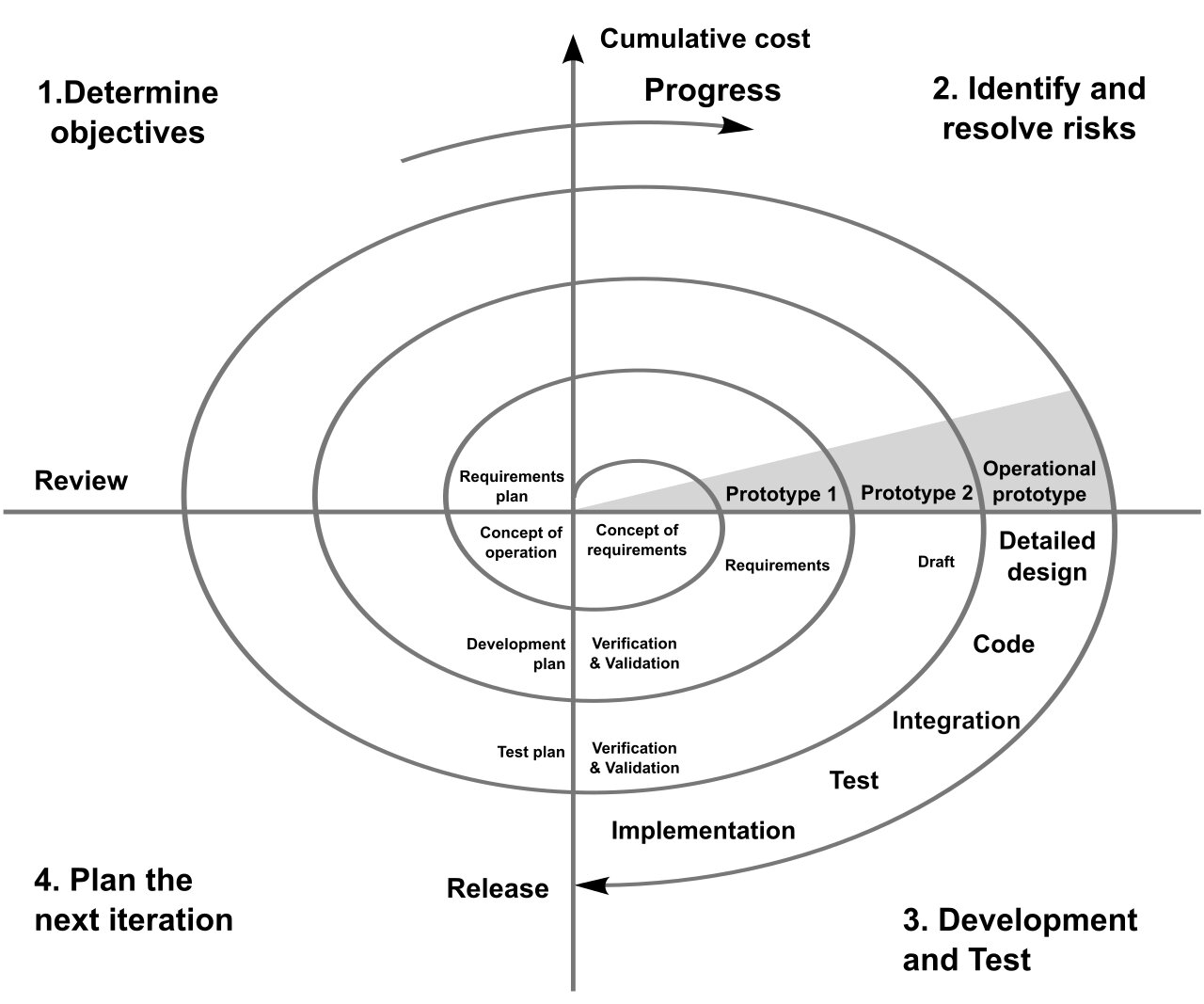


Figure : Spiral Methodology process

This methodological proposal uses the spiral development model as a basis and incorporates processes of evaluation of usability, prioritizing participation of the user in all processes of the life cycle of design, in order to ensure a design centered on the user, even if it is a process model oriented to large and expensive projects, as it is intended to be a model for risk reduction (A. Nosseir & Co., 2012).

The process allows application developers to the usability criteria of the application; the first step is to identify the users, the tasks and the contexts in which the mobile application will be used, the next step is to prioritize the usability attributes, identify which attributes are the most important for the application, and for each define a set of metrics to verify the degree to which they are met in the end application.

The spiral mobile application development process contemplates five (5) iterations, for each of the three tasks (requirements determination, design, and testing) and ends each iteration with the planning of the next one; in the first iteration the system requirements are determined, and users, tasks and contexts in which the application will be used are identified. Then, ease-of-use attributes are defined and prioritized, and metrics are identified for each attribute; a prototype of the application interface is drawn, and the prototype is tested; developers can use different usability techniques to measure the value of each attribute.

In the second iteration, the development team will collect more data and requirements; it will explore whether there are more potentials, tasks, and contexts in which the application. The usability attributes are then redefined and prioritized; as a result, developers will alter the metrics to accommodate added requirements; a prototype is made in the design interface hi-fi and tests are performed, using usability techniques for each attribute, the rating is compared with the results of the iteration previous.

In the third iteration, developers can identify and prioritize usability attributes with greater clarity using the results of the previous iteration; the design of the whole system is developed, and the alpha version with its respective tests, the development compares the results with the rating of the previous iteration.

In the fourth iteration, the results of the previous iteration are used to identify and prioritize the ease of use attributes; the beta version developed and is released for evaluation by the client.

In the fifth iteration, the final product is developed; it performs an evaluation of user-friendliness, the rating of each attribute is calculated and compared with the score from the previous phase. An alteration in the final product is made on the basis of the results, and the product is released.

# Conclusion

In my opinion, agile methodologies are an excellent alternative to guide small software development projects, such as applications for mobile devices, because of the great adaptability they have; but they must be adapted to the particular characteristics of these devices in order to obtain quality products.

Among the characteristics is the great speed with which versions of the mobile operating system (OS) are released, the emergence of new hardware features, the fierce competition of companies and development communities to capture the market, which leads to the tendency to develop in short periods and in most cases where product quality is sacrificed precisely for not having followed an appropriate development methodology or techniques and therefore, in most cases, it is their efforts end up not bearing the expected fruits.

Although there is a wide variety of methodologies (agile and traditional), in the field of development for mobile devices, most of the efforts have been oriented towards the development of new methodologies, based on traditional practices, with some of the additions of modern techniques such as usability among others, that is why this proposal has been guided in the agile methodologies.

I think that all teams wishing to meet their customers' expectations perfectly should use agile methods.

O. Salo & P. Abrahamsson, «Agile methods in European embedded software development organizations: a survey on the actual use and usefulness of Extreme Programming and Scrum», IET Software, vol. 2, n.o 1, p. 58, 2008.

P. Letelier, J. H. Canós, & C. Penadés, «Metodologías Ágiles en el Desarrollo de Software», presentado en VIII Jornadas de Ingeniería del Software y Bases de Datos JISBD, Alicante - España, 2003, pp. 1-8.

K. Conboy & B. Fitzgerald, «Method and Developer Characteristics for Effective Agile Method Tailoring: A Study of XP Expert Opinion.», ACM Transactions on Software Engineering & Methodology, vol. 20, n.o 1, pp. 2:1-2:30, jun. 2010.

V. E. Jyothi & K. Nageswara Rao, «Effective Implementation of Agile Practices Incoordination with Lean Kanban.», International Journal on Computer Science & Engineering, vol. 4, n.o 1, pp. 87-91, ene. 2012.

«Annual State of Agile Development Survey Results | VersionOne». Retrieved from : https://www.stateofagile.com/#ufh-i-521251909-13th-annual-state-of-agile-report/473508

P. Abrahamsson, J. Warsta, M. T. Siponen, & J. Ronkainen, «New directions on agile methods: a comparative analysis», 2003, pp. 244-254.

D. E. Avison & G. Fitzgerald, Information system development. Maidenhead: McGraw-Hill Education, 2006.

«Manifiesto por el Desarrollo Ágil de Software». Retrieved from : https://agilemanifesto.org/principles.html

V. Szalvay, «An introduction to agile software development», Danube Technologies, 2004.

W. Royce, «Managing the Development of Large Software Systems: Concepts and Techniques», WESCON Western Electronic Show and Convention, 1970.

B. W. Boehm, Software engineering economics. Englewood Cliffs, N.J.: Prentice-Hall, 1981.

F. P. Brooks, The mythical man-month : essays on software engineering. Reading, Mass.: AddisonWesley Pub. Co., 1995.

K. Beck & J. Zapata Martínez, Una Explicación de la programación extrema. Aceptar el cambio. Madrid [etc.]: Addison Wesley, 2002.

K. Beck & C. Andres, Extreme programming explained : embrace change. Boston, MA: AddisonWesley, 2004.

J. Bowers, J. May, E. Melander, M. Baarman, & A. Ayoob, «Tailoring XP for Large System Mission Critical Software Development», vol. 2418, D. Wells & L. Williams, Eds. Springer Berlin / Heidelberg, 2002, pp. 269-301.

J. Rasmusson, «Introducing xp into greenfield projects: lessons learned», IEEE Software, vol. 20, n.o 3, pp. 21-28, 2003.

H. Takeuchi & I. Nonaka, «The new new product development game», Harvard Business Review, 1986.

K. Schwaber & M. Beedle, Agile software development with Scrum. Upper Saddle River, NJ: Prentice Hall, 2002.

L. Rising & N. S. Janoff, «The Scrum software development process for small teams», IEEE Software, vol. 17, n.o 4, pp. 26-32, ago. 2000.

D. Astels, Test-driven development : a practical guide. Upper Saddle River, N.J.; London: Prentice Hall PTR, 2003.

K. Schwaber & J. Sutherland, «The Scrum Guide», Scrum Alliance, 2019 Retrieved from : https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf

P. Letelier, J. H. Canós, & E. A. Sánchez, «An Experiment Working with RUP and XP», en Extreme Programming and Agile Processes in Software Engineering, vol. 2675, M. Marchesi & G. Succi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 41-46. 2003.

K. Restivo, «Worldwide Quarterly Mobile Phone Tracker». Retrieved from : http://www.idc.com/tracker/showproductinfo.jsp?prod\_id=37

P. Abrahamsson, A. Hanhineva, H. Hulkko, T. Ihme, J. Jäälinoja, M. Korkala, J. Koskela, P. Kyllönen, & O. Salo, «Mobile-D: an agile approach for mobile application development», en Companion to the 19th annual ACM SIGPLAN conference on Objectoriented programming systems, languages, and applications, 2004, pp. 174-175.

M. Satyanarayanan, «Fundamental Challenges in Mobile Computing,», oct. 1998.

P. Abrahamsson, «Keynote: Mobile software development–the business opportunity of today», Proceedings of the International Conference on Software Development, pp. 20-23, 2005.

P. Abrahamsson, «Agile software development of mobile information systems», en Proceedings of the 19th international conference on Advanced information systems engineering, 2007, pp. 1-4.

M. Burton & D. Felker, Android Application Development For Dummies. Wiley, 2012.

R. Brandom, «There are now 2.5 billion active Android devices», 2019. Retrieved from : https://www.theverge.com/2019/5/7/18528297/google-io-2019-android-devices-play-store-total-number-statistic-keynote

Statcounter, «Mobile Operating System Market Share Worldwide», 2019. Retrieved from : https://gs.statcounter.com/os-market-share/mobile/worldwide

«Android and apps: Some favorites.», Network World, vol. 27, n.o 14, pp. 16-17, 2010.

S. Hollister, «Google flips Android kill switch, destroys a batch of malicious apps (update)», 2011. Retrieved from : http://www.engadget.com/2011/03/06/google-flips-android-kill-switchdestroys-a-batch-of-malicious/

T. Ricker, «iOS 5 jailbroken», Engadget. Retrieved October, vol. 26, 2011.

C. Miller, D. Blazakis, D. DaiZovi, S. Esser, V. Iozzo, & R.-P. Weinmann, iOS hacker’s handbook. Wiley, 2012.

M. Antic, S. Jovanovic, & S. Cvetanovic, «Development of eStudent iOS Mobile Application.», International Journal of Interactive Mobile Technologies, vol. 7, n.o 1, pp. 35 40, ene. 2013.

Apple, «iOS 13», Apple, 2019. Retrieved from : https://www.apple.com/ios/ios-13/

R. van der Meulen & Janessa Rivera, «Gartner Says Worldwide Mobile Phone Sales Declined 1.7 Percent in 2012». Retrieved from : http://www.gartner.com/newsroom/id/2335616

E. Messmer, «Apple iOS vs. Google Android: It comes down to security», Network World (Online), p. 22, 2012.

O. Salo & V. teknillinen tutkimuskeskus, «Enabling software process improvement in agile software development teams and organisations», VTT Technical Research Centre of Finland, [Espoo, Finland], 2006.

T. Dyba & T. Dingsoyr, «What Do We Know about Agile Software Development?», IEEE Software, vol. 26, n.o 5, pp. 6-9, 2009.

V. Rahimian & R. Ramsin, «Designing an agile methodology for mobile software development: A hybrid method engineering approach», Research Challenges in Information Science, 2008. RCIS 2008. Second International Conference on, pp. 337-342, 3.

A. Nosseir, D. Flood, R. Harrison, & O. Ibrahim, «Mobile Development Process Spiral», 2012, pp. 281-286.

«Electronics -AGILE - Agile Software Technologies», 2012. Retrieved from : http://virtual.vtt.fi/virtual/agile/mobiled.html

S. B. Kaleel & S. Harishankar, «Applying Agile Methodology in Mobile Software Engineering: Android Application Development and its Challenges», 2013.