4 Methodology

Methodology seems to be a buzzword in the academic context. However, it is not as well understood and known in undergraduate as it is in postgraduate studies. Besides, because the term and its concept has widely been customized and adapted in software engineering and development, it becomes somehow difficult for students, especially in undergraduate level, to properly understand and apply it. Usually, final year students in computer science, software engineering and other divisions of computing have studied the concept of software development methodology in modules/courses such as software engineering, IT project management, system analysis and design, etc. However, I found it useful to review the concept again and to focus on this issue from another perspective.

4.1 Overview

Methodology has different meanings. As a branch of logic, it is about understanding the way that human beings knowledge is formed. It is also the combinations of best practices, procedures, rules and guidelines, in one word the methods, of the specific field of science and art by which professionals, specialists, and researchers can conduct their projects, research, development and study activities. Again, from another perspective, it is the study of those methods, which were just mentioned. By giving different meanings of methodology, I did not want to confuse you. It was just to let you have wider perspective of what the methodology can be in the different contexts. However, for our purpose the second meaning plus a trivial part of the first one is sufficient more than enough. It means that we are taking the methodology as the combination of methods by which we can conduct our project, and control it, on one hand, and the methods that we can apply in software development in order to deliver a software product, on the other hand. Moreover, we consider different methods and we argue why we have chosen a particular method.

Clearly, if your project is not involving software development, such as a research project for instance, then the methodology should be chosen and discussed based on a generally established approach for the scientific research. However, we are not going to discuss this concept in this book, instead, we will discuss and explain some of the major adapted approaches in software development context will be discussed and explained.

4.2 Software Development Methodology

Fortunately, we have left this perception behind that software development is a synonym to programming, long time ago. Nowadays, most of software developers, with or without a formal educational background in the field, have a clear idea that this activity is far beyond sole programming. They understand that it has more science and engineering in it than art. Indeed, the term “software engineering” has widely been accepted despite some trivial arguments against it that can be heard, here and there, from time to time.

My personal experience has brought me to this belief that software development, as a production activity, should be considered "naturally" and "mainly" an engineering process. However, like many other engineering activities it includes and actually, it needs some artistic endeavors and flavors. Moreover, in many cases, when the activity is not a simple reproduction or mass production of software, it becomes much more difficult to manage. This is what I have tried to teach my students in academy or share with my colleagues in the industry over the years.

The problem is, and surprisingly most of the software developers, at least verbally, confirm it as a problem, that because of the nature of software, which mainly comes from its intangibility, this engineering process can sometimes be jeopardized. There is a large tendency towards the ignorance of the engineering process in software development as the process it is very difficult to follow and control, especially when novice developers are involved. However, the complete argument on this subject is beyond of the scope of this book. Indeed, I assume that, hopefully, most of the readers of the book are already familiar with the concept and, again, hopefully, in line with the idea of taking software development as an engineering process.

Having what was mentioned, you could select and follow one of well known and well-practiced (software engineering) methods to accomplish your projects. Perhaps you have heard about different methods such as waterfall, spiral, prototyping, agile, Xtreme, object oriented, structured, Model View Controller (MVC), Rapid Application Development (RAD) and different frameworks such as Unified Process, Rational Unified Process (RUP), Microsoft SolutionFramework, and many more. How can we survive within this jungle of methods? How can we choose among them? To give you a straightforward answer is not easy though, it is not impossible!

It would not be a wrong claim that the notion that software development is an iterative activity has almost unanimously been accepted by developers. Whether by following a spiral method or non-spiral, you would usually build software during an evolutionary process. The process,at least most of the time,is iterative, and the final result can be obtained through accumulative increments, each one of which is supposed to satisfy one or several parts, but not all, of the requirements.

There is another decision that should be made before the development process starts. Which approach are you going to use to analyze, design, and implement your software? Structured, or object oriented? Some people may have a single and simple answer to this question. They would say, "Certainly, it is object oriented." If you ask about the reasons for this decision, you may receive several arguments which most of them are legitimate but not complete. For example, an argument can be about the dominance of object-oriented paradigm and its related technologies in the market and industry. Another one can mention the technical and theoretical capabilities of this method. Although, as it was mentioned,these arguments and other similar reasons are quite legitimate and understandable, yet the answer to that question, especially in your case, is not that straightforward. We will discuss this issue in the following section.

4.3 OO vs. Structured

OO or Structured? Which one is better? The second question is not appropriate. When we are in the process of choosing a method, we are not talking about these methods in general. We have to consider them in a specific context, which here is your project. The mentioned questions should be changed to "OO or Structured, which one is the most suitable approach to do my project?" The answer depends on several parameters and conditions. Below you can find these conditions in the format of some questions that can guide you through your decision.

OO vs. Structured

- How familiar are you with the method?
- How complex is your project?
- Which method is more suitable to analyze and design your system?
- What kind of development tools are you going to use?
- Is there any technical specification in your project definition about the methodology?
- Is there any preference or evaluation bonus on choosing a method?

Your answers to the above questions can provide you with a general view of which method is preferable in your case. However, as an engineering method we try to specify the situation in a more precise way, which you can find in the Table 4-1. For each parameter, you can provide a value between zero and five to indicate the measure of the parameter. Zero means the lowest rate and five the highest. This table shows an example. Clearly, you better to seek some advice from your supervisor in some cases, for example, to help you on the third question above.

Table 4-1 gives you an example. As you can realize, the results are too close to each other, however, in this example the OO is preferable. You may find some situations that the results are even. What should we do in these cases? Well, it is up to you! Do not forget about the harmony between the heart and the brain. Object-Oriented paradigm is not a fashion only. It is a very powerful approach, which stays in the software development as long as there are no other stronger competitors that they can give developers more than it has given. Therefore, as a person that should look at the career as well, developing a project with this method is a benefit. However, do not forget the approach that we took during the project selection. The aim at this step is to deliver a successful project.

Methods / Parameters	OO	Structured
Familiarity with the method	3	2
Suitability for the project	3	4
Tools availability	5	3
Forced by project specification	0	0
Affects project evaluation	0	0
Result	11	9

Table 4-1 Choosing a method

Nevertheless, there are other concerns that should be taken into account. Are there only two options? Can we decide to take another approach? Can we get benefit of both object-oriented and structured methods? These questions would be addressed in the following section.

4.4 A Mixed-mode Approach

There are many undergraduate projects, which are not dealing with complex systems. They are not using complex objects and the functionality of the system is not difficult to understand. Most of these projects aim to equip the students to understand a general-purpose system and to utilize their knowledge in order to build a system from scratch. Many of these systems are using some kind of database, which users create and update them. In addition, these systems usually provide different kind of reports and outputs based on the expected functionality. Hence, the analysis and design of this type of projects would focus on the user interface and database design, none of which requires advanced object-oriented approach. However, one aspect of object-oriented, the Use Case Modeling, should be considered as a powerful tool that makes requirement management, analysis, and user interface design to be sound and flexible.

Using Use Case Modeling was introduced by Ivar Jacobson.Jacobson is one of the pioneers of object orientation, and a member of Gang of Three (including Grady Booch and JamesRumbaugh). Use Case Modeling has steadily become a powerful instrument in different steps of software development, since 1990s. This model has proven itself as a reliable, user-friendly, flexible, and powerful instrument to catch and model the software requirements. The independence of the model from other sections of the design gives the model the flexibility to be used with both object-oriented and structured approach in the software development. This is why when it is used with the traditional (structured) approach I call it "Mixed-mode " method, in order to show that this method has its roots from both main methods. Indeed, in this case, I have replaced the traditional Data Flow Diagrams (DFDs) with the Use Case Modeling.

In the rest of this book, my focus would be on this approach, because you can find extremely wide resources on each approach, separately. Hence, I am not repeating the material that you can find them in their origin formats. You can use the bibliography at the end of the book or simply search your university's library to find a huge pile of resources in this regard.

The mixed-mode approach uses the UseCase Modeling in four stages of your software development, which are Requirement Management, Analysis, User Interface Design and Testing. It also helps you during the Database Design and users guide compilation. As it is obvious, this method does not consider Class Modeling and Object Modeling so you can use development tools and programming environments in a traditional, structured method. But, it does not mean that you cannot use object-oriented aspects of your development environment.

Anyway, if you do not interested to bear with the complexity of object-oriented design and development, then this method eliminates this complexity and lets you implement your system in a more straightforward manner. Object oriented developers may not agree with me on this, but as a developer with more than 25 years of experience in both methods, I can tell you that the approach works properly and the development cycle can be shortened if the project falls in the categories, which are proper for the method. Among the projects that we discussed in section 2.1, only the first one, the Asset Management System is a good candidate for this approach. During the next chapters, you will find how this mix-mode approach can be implemented.

4.5 Summary

To do your project you need to follow amethod. Methodology discusses different methods, which are available in order to accomplish different projects. It is important to understand different methods that exist and applicable to software and information technology projects. Usually, this topic is addressed in different modules/courses such as software engineering, project management, and/or research methods. However, you have to have to make a trade-off between different methods and select the one that is more efficient regarding to your project.

Two main approaches to software development are structured (traditional) and object oriented. Both of these methods are supported by different tools and development environments. There are programming languages that support one or both methods. However, the processes of analysis and design are very different both in terms of their techniques and in terms of their outputs. Although object oriented method is a dominant method in the industry, however, structured method has still its own application in software development. Besides, there is a huge legacy of systems, which have been built, based on the structured approach. Therefore, this method is taught at universities and you should not ignore it as an option.

But, in this chapter, a mixed-mode approach was presented which utilizes the Use Case Modeling from the object-oriented method in different phases of software development and intertwines it with traditional Database Design in order to accomplish a project. In addition, a method was presented which helps you to choose a proper method. Moreover, eventhough several factors were presented in order to select a method, the success of the project was emphasized as the key decision factor in order to select the suitable method.