



HOW CAN YOU BE A PROFESSIONAL SOFTWARE DEVELOPER?

Abdulkawi Yahya Radman Al-Shamiri

Master student in Department of Computer and Information
Hefei University of Technology (HFUT), China

Abstract: There are many papers that speak about the most popular of software development models with their disadvantages and advantages that used in software development; So the main target of this paper is explanation of steps for to (be a professional software developer) , and not prefer a model on another model, because each model has disadvantages and advantages. We can benefit from advantages of all these models for software development if we follow the steps that I will mention in this paper.

Keywords: Software Development Models, Professional Software Developer, Project, Software Development Life Cycle (SDLC), Development Team, Risk Analysis

INTRODUCTION

Software are became that controlling and organizing wide areas of our life activities, and this space expands every day, so it is not logical to leave the development of such important things for coincidence to manipulating us, and our activities. For this reason, there was a need for a way to manage and develop the software (from the birth of the idea, then the stages of development, and even the production of projects that meet the requirements of users in the actual work environment).

I know there are many authors who have talked about software development models; so here I will not explain these models with their disadvantages and advantages, but I will explain steps for (to be professional software developer) Whether you are alone, a development team, or an organization interested in developing and designing systems and software. So that you do not need to daily updates for the software that you have designed or developed; but you will need new versions of annual, because people annually are waiting for the new versions. For the daily updates, I see it is not good because they express repair for errors, because of the rush in the software version.

The professional software developer is person who releases a software or system with reasonable long time, so that this program or system does not need to fix errors daily; because the user will be bored of the daily updates, and will be happy and impressed by the annual versions which are free of errors and defects.

After reading this paper, you will be as professional software developer and will try to think of finding a way includes all advantages for all SDLC models simultaneously.

[1] Has explained SDLC, its stages, and these models (Waterfall, Iterative, V-Shaped, Spiral, Incremental, Prototype, and Big Bang) approaches. [2] Has explained

these models (Waterfall, Iterative, Spiral, Big Bang, V-Model, Agile, Rapid Application Development (RAD), and Software Prototyping) approaches, with the disadvantages and advantages of each model. [3] Has also explained these models (Waterfall, Iterative, V-Model, Spiral, Extreme programming, Adaptive Software Development (ASD), Incremental, Rational Unified Process (RUP), Rapid Application Development (RAD), Chaos, Agile Software Process (ASP), Dynamic System Development Method (DSDM), Crystal, Feature Driven Development (FDD), Scrum, Wisdom, Big Bang, and Code & Fix) approaches. [4] Has explained (Waterfall, Iterative, Agile, Scrum, Lean, Spiral, V-Model, Big Bang, Rapid Application Development (RAD), and Prototype Model) approaches. [5] Has explained (Waterfall, Iterative, V-Model, Spiral, Big Bang, Rapid Application Development (RAD), V-Agile, and Prototype Model) approaches. [6] Has explained (Waterfall, Iterative, V-Model, Spiral, Extreme Programming, Incremental, Evolutionary, Rapid Application Development (RAD), Prototyping, and Big Bang Model) approaches. [7] Has discuss comparison the Waterfall problems in literature with the results of a case study at Ericsson AB in Sweden, investigating issues in the waterfall model. [8] Has discuss comparison between Prototype Model with System Development Life Cycle. [9] Has discuss experimental prototyping, exploratory prototyping and evolutionary development. [10] Has discuss seven case studies of RAD projects and compares each to issues relating to a number of RAD principles as represented in methodologies such as the recent open standard known as dynamic systems development method. [11] Has discuss, the Crystal Clear Model and compare it with Scrum Model. [12] Has proved that the most important and first factors to choose the appropriate model is the level of understanding of the requirements of the project, the second factor is the level of complexity of the project, the third factor is the level of interaction between the user and the project. [13] Has proved by their study that

traditional models are still in use to-now despite the spread of the agile model more than a decade ago. This proves that the professional developer does not adhere to a particular model but benefits from all models.

At this paper, I will focus on putting steps that will help those interested in software development become professional software developers.

RELATED WORK

I. Software Development Life-Cycle (SDLC):-

It is sets of stages that give a common grasp of the software- development process; and how will be the software realized and developed from the business grasp and its requirements stage to transform these ideas and requirements to functions and characteristics until its usage and running to achieve the business needs. A good software development team must have sufficient knowledge in how can they choose the appropriate SDLC model based on the project context and requirements. Therefore, it maybe wanted to choose the correct SDLC model basing to the project's specific requirements to make sure its success.^[16]

The Software Development Life Cycle (SDLC) is a process or action plan that defines due tasks with each stage of the development stages of any project, whatever its type, size or environment. It designed to produce high-quality software, meet the actual customer's requirements, and to be ready at the planned time and at the planned cost. The international criterion for software life-cycle processes is ISO/IEC 12207. Its goal is to be the measuring that defines each the tasks required for developing and maintaining software.^{[14][15]}

1) System life cycle:-

1. The planning or requirements stage, it includes :-
 - 1.1. Identify and define the problem (Requirements).
 - 1.2. Feasibility study.
2. Analysis stage.
3. Design stage.
4. The programming stage, it includes :-
 - 4.1. Writing the program.
 - 4.2. Program Test.
5. Implementation stage, it includes :-
 - 5.1. Setup and installation of the system.
 - 5.2. Training of users.
6. Delivery (acceptance) and maintenance stage.

2) Elements of the software development :-

1. System (Project) :-

A system (Project) is a set of interrelated components that achieve the desired goals.

2. System Components :-

Not all systems have the same elements, but they have main elements to build the system, a set of tools, machines, procedures and users.

3. Customer :-

A company (or person) which requests developing for its software (its project).

4. User :-

The person (or group of persons) who will actually use the project, and deal with it directly.

5. Development team :-

A company (or group of persons) which will develop the project according to the customer's requirements.

It includes the following members:-

5.1. Project Manager :-

The project manager (Project Master) leads the team, participate in all stages of the project, coordination among them, find solutions for the problems and constraints facing the system, and the development of the implementation strategy.

5.2. Systems Analyst :-

The system analyst does contacts with the manager, provide the necessary consultation at the planning stage to identify and define the problem, prepare the feasibility study (technical and material), identify the system's limitations, discuss the users, collect the necessary information, determine the requirements, prepare the preliminary models for the proposed new system in analysis stage, preparation of the time plan for implementation, and delivery to the system designer.

5.3. System Designer :-

The system designer designs the new system, make the necessary drawings and tables, write the system work scenario, identify the special tools used in the system, and propose new ideas that improve the performance of the new system and delivery to the programmer and developer.

5.4. Programmer :-

The programmer writes the program code for basic processes, builds the system interfaces using by standard tools available in the programming language, and creating the database through the designs.

5.5. Software Developer :-

The software developer creates and develops the special tools, and application of new ideas to improve system performance through designs.

5.6. Software Engineer :-

The software engineer makes the necessary adjustments to the new system and repairs and expands the system.

5.7. Project Engineer :-

The project engineer integrates the devices used in the system programmatically.

5.8. Quality Engineer :-

The quality engineer checks and tests system quality, detects errors, and documentation the system.

5.9. Operator for system :-

The operator for system prepares and runs the system for users, and trains them to use the system.

5.10. Technical Development Manager :-

The technical development manager plans to develop and integrate modern and advanced devices for system.

5.11. Business Development Manager :-

The business development manager plans to develop system processes and integration with other systems.

Figure (1) shows the participants in the software development process where the customer asks the developer to develop the project and give him the required requirements, the developer develops the project according to the requirements, and the users give their project needs to the developer.

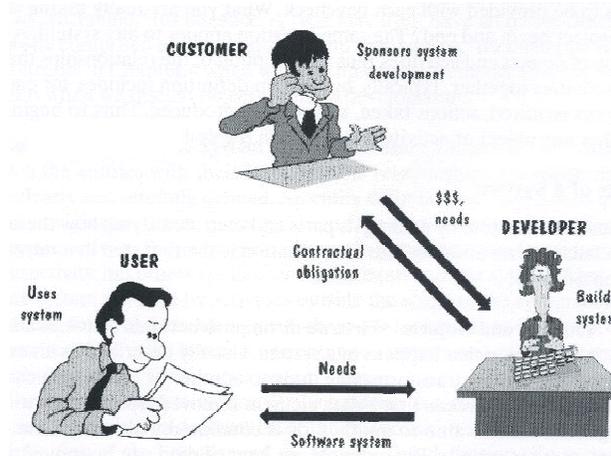


Figure (1) :- Participants in Software development

3) SDLC Stages :-

SDLC Stages are the same as the life cycle stages of the system, but SDLC Stages are more practical.

[1][2][14][15][18][19][20]

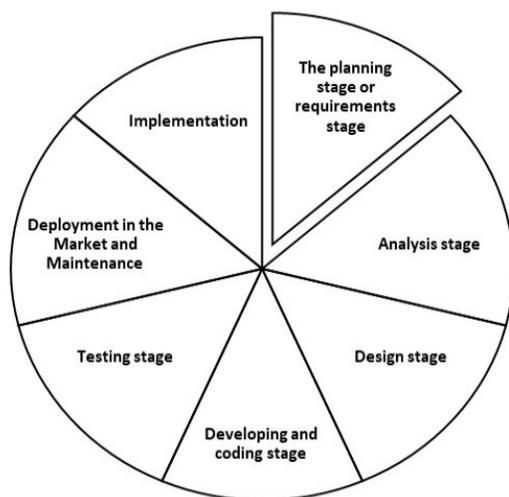


Figure (2) :- SDLC Stages

1. Stage 1 (Requirements and Planning) :-

At this stage, the requirements which must be met for the project and running are identified, that is done by answering this question (what will this project do?) In

addition, all that relates to this question. Where the information which we will get it at this stage, will help us to create a full plan for project to quality ensure and determination the risks that could accompany the project, and to conduct the feasibility study of the project in its economic, practical and technical aspects. The purpose of this stage is to "Fulfillment of requirements" or "Gathering of requirements" which means collecting project requirements from users, customers and other persons related to the project, so that we can create the plan and feasibility study. These requirements will be collected, and everything that is done at this stage together in a document called SRS (Software Requirements Specification). This stage is very important; since most projects fail because the requirements are not met, not accurate identification, and not aggregated, which leads to be the requirements are unclear, unqualified, incomplete, and unattainable.

2. Stage 2 (Analysis) :-

After completing the requirements identification and project-planning phase, these requirements are analyzed in a comprehensive, detailed and accurate analysis, understanding the problems and risks that may be encountered during development and put solutions. At this stage, the analysis team tries to discuss all the questions and provide all the answers they need to build the project according to the requirements. All this is added in the SRS document (Software Requirement Specification (SRS) document). In some models; this stage is called the risk analysis stage, since it is concerned with risk identification and put solutions. This stage is the backbone of the development process.

3. Stage 3 (Design) :-

It means to design the project structure, where this design is inspired by the SRS document (Software Requirement Specification (SRS) document) obtained from the previous two stages. At this stage, the accessories of design of the project (Hardware and Software) are specified, also the definition of structure and main parts of the project. The results of this stage are documented in a document specifying the elements of this design and its details (Design Specification Document (DSD)).

Designing stage usually includes problem solving and planning a software development. This includes design of a low-level objects and algorithm, design of a high-level structure. The design stage has many design fields, as data design, User interaction and experience design, process design, and others.

4. Stage 4 (Developing and coding) :-

It is depending on the design of the project structure which we got from the previous stage, the project and its code are developing, that led to translate these designs to actual objects, using any of the programming languages

that are often identified by the nature of the project that will be developed.

5. Stage 5 (Testing the Project) :-

In this stage, the project is tested and checked to identify errors and correct. Often, this stage is not separate from the other stages, where testing process is at the end of each stage and before the next stage.

In this stage, we make all types of testing, for example, unit testing, integration testing, quality testing, and others.

6. Stage 6 (Deployment in the Market and Maintenance) :-

The final project which has been completed, it is distributed to the actual work environment. After you distribute the project to the actual work environment, it may appear some the problems; to resolve these problems; some updates are released to overcome these problems. It is also possible to issue successive editions of the project that include significant additions and modifications required by the actual work environment.

7. Stage 7 (Implementation) :-

This stage has a lot supporting activities that include training end-users and administration. The project will need supervision and intelligent detection of problems and risk which we could not detect during the previous stages.

4) SDLC Models :-

There are many types of software development projects, and each project has its characteristics and environment that may not be the same as any other project. For this reason, it was necessary to have more than one model, to be each model appropriate for a type from the different types of software development projects.
[3][4][5][6][8][14][15][16][19][20]

The following is the most important and famous of these models:-

1. Waterfall Model.
2. Iterative Model.
3. Incremental Model.
4. Iterative and Incremental Model (IID).
5. The Prototyping Model.
6. Rapid Application Development (RAD) Model.
7. The Spiral Model.
8. The V-Model.
9. The Big Bang Model.
10. The code and fix Model.
11. The Agile Model :-
 - Rational Unified Process (RUP).
 - Agile Unified Process (AUP).
 - Scrum.
 - Dynamic Software Development Method (DSDM).
 - Extreme programming (XP).
 - Feature Driven Development (FDD).
 - Crystal Family.
 - Adaptive Software Development (ASD).

II. The Software Testing :-

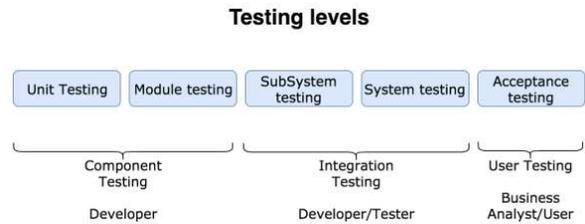


Figure (3) :- The testing approaches can go deep to different levels from unit testing to acceptance testing

Software Testing is important stage for any software development life cycle model, where making software testing ensure the software quality and getting a workable actual software at the end of the project. The main goal for the testing is to make sure that all functionalities are working correctly according to the requirements, to test how to break the project, and to test how to simulate the Misuse users of the software by the quality team before delivery the project. The software quality team write the testing code before the development process. The testing code is called Test-driven development (TDD) which famous as a development technique, where firstly you must write a test that fails before you write new functional code. The software developers adopt TDD for development of software, where there are many frameworks are supporting TDD. [17]

1) Importance of software testing :- [17]

Software Testing is important for:-

1. Making sure that the project performs the required tasks correctly, and making sure that every function meets the requirements, and delivered project to the customer according to requirements.
2. Making sure that the project does not do any task that it is not assumed to do.
3. Continuous learning for the team, when the team discloses an error in task or requirements, the team will learn from these errors, also they will learn how to parry that in the future. This will add ripeness for the team.
4. Minimize project failure.
5. High quality Projects and software will ensure the lasting reputation of software development companies.

2) The common activities in software testing process :- [17]

The software testing process varies from model to another, but will have main common activities in the process frequently.

Some common activities for testing process:-

1. Test of Requirements :-

It is meaning how making sure that each requirement is testable.

2. Test of planning :-

It is meaning how can plan the testing activities, evaluation of the effort, the wanted team, Etc.

3. Writing the Test Cases :-

In this activity, the testing team starts to write the testing scenarios, these scenarios must include unity, integration, project testing, Etc.

4. Execution of Test :-

It is meaning preparing the testing environment and starting testing execution.

5. Testing results :-

After the execution, the testing team must reported the development team about the testing results and defects to start fixing.

6. Defect Retesting :-

When the development team inform that the defects have been fixed, the testing team must do retesting.

7. Users Acceptance Test :-

It is meaning make sure the end users who will use the project that the project works correctly.

8. Testing Closing :-

After making sure that the test done correctly, there are no defects; in this case, this test should to be closed.

3) Approaches of Software testing :-^[17]

1. Blackbox (Functional) :-

It is used generally when the testing team have limited knowledge about the project which under test, or when the source code is difficult to access. It is mainly done by defined interfaces.

2. Whitebox (Structural) :-

It known as clear box testing, glass box testing, transparent box testing, and structural testing. It is a method of software testing which tests inner structures or processes of an application, as opposed to its functionality; it is mainly done on the source code itself.

3. Graybox :-

It depends on the merge of the two previous approaches, where the testing team have knowledge about the code and improve the testing cases in Blackbox approach.

How can you be a professional software developer?

In order to be a professional developer, you must have the following criteria:-

1) You have sufficient knowledge of all fields of computer science, especially field of programming for most programming languages, field of analysis and design of algorithms, field of analysis and design of systems:-

To you can follow the development process on your own, and there is no doubt that everyone has a specific skill is distinct from others, though he has more than a skill. If you have many skills, you will find that you distinct with

one of skill from those skills, and you tend to that skill a lot, and you will find that you are very eligible of it.

2) Formation a development team with sufficient knowledge in all fields of computer science :-

If you are working alone or want to create an organization that is interested in building, designing and developing software, you need to create one or more development teams according to the nature of your business. The only criterion in team composition should be competence away from kinship and nepotism to be you or your organization famous and successful.

The members of the team must be proficient in full knowledge of all fields of computer science, especially programming for most popular programming languages, algorithms analysis and design, system analysis and design. As I have already said, each person has a specific skill is distinct from others, though he has more than a skill. This also applies to the development team; the development team must works as one body, taking into consideration the skill of each member of the skill that characterizes him. In other words, when you are creating the team, you have to know that each member has all of these skills, which a skill that characterizes him?

For example, Of course all your team's member have all the skills required. But you will note that: the member(A) is distinct and perfected in the skill of analysis, the member(B) is distinct and perfected in the skill of programming, the member(C) is distinct and proficient in the skill of the professional design of the interfaces, and the member(D) is distinct and proficient in software testing skill and writing potential scenarios for program, and so on.

In this case, all members work as one body without being separated, and to benefit from the person who distinct and perfected in the skill of analysis in the analysis process without neglecting others, also to benefit from the person who distinct and perfected in the skill of programming in the programming process without neglecting others, and benefiting from the person who distinct and perfected in the skill of design of the interfaces in the process of design beautiful and attractive interfaces of the project without neglecting others, and benefiting from the person who distinct and perfected in software testing skill and writing potential scenarios for program in the testing process without neglecting others, and so on. It is best to each team has master, who follow the development process direct and step-by-step, and that there is a master for all teams (you will be master for all teams).

As for the number of team members, it is best not to specify a specific number; but, the number of team members depends on the nature and type of the project; there are projects that need few members, and there are projects that need a lot members; But the number of members of the team should be determined in a reasonable and appropriate manner.

3) You have leadership ability for management :-

In order to be able to manage the organization or team or teams; You should have to deal with members of the team as family; This is a very important point, because the psychological comfort at work, and the feeling of happiness and comfort in the place where you work, all these make the person Perfects the work and show all the energies, possibilities and creations; this is the secret and criterion of success and development for each organization.

4) You and the team should have full knowledge of the software development models :-

You and all members of the development team should have full knowledge of software development models; as knowledge that these models are not only used in software development, but also used in the case of building and designing a new system, a new project or a new application. As well as knowledge the best use of the model, the disadvantages and advantages of each model, when should we use this model and when should we use that model?, and so on.

How can you choose the appropriate model of Software development life cycle models:-

Choosing the Software Development Life Cycle (SDLC) model is a difficult task for a lot of software development institutions and software developers. To choose the appropriate model, you must have overall knowledge of all software development models, and knowledge of the advantages and disadvantages of each model.

When you want to choose a SDLC model for development of project, you need to have perception about:-^[15]

- Your Organization status.
- Your Technology Context.
- Your Team Composition.
- Your Customer status.

There are some steps to can choose the appropriate model:-

1. Knowledge of the about SDLC Models :-

You must have enough knowledge and to be have knowledge of the SDLC models which will be chosen.

2. Estimate the needs of customers :-

You must study the working field, worries and requirements of customers, working primacies, your efficiency technical, and technology constraints to become able choose the appropriate model for SDLC according to standards.

3. Define the standards :-

Some of the choosing standards that you may use to choose any SDLC model:-

- Is the SDLC appropriate for the size of the team and its skills?
- Is the SDLC appropriate for the chose technology that we will use for realization the development?
- Is the SDLC appropriate for worries and requirements of customers?
- Is the SDLC appropriate for the geographical area (distributed team)?
- Is the SDLC appropriate for the complexity and size of software?
- Is the SDLC appropriate for the kind of project that we will develop it?
- Is the SDLC appropriate for software developer's capability?

4. The decision :-

When you define the standards and the arguments, you will need to discuss that with the team, and you must have set of decision. After analyzing the results of discuss, you must document this decision in the project artifacts and share it with the customer.

5) Do not adhere to a specific development model on the basis that it is the best :-

You should not impose a specific model for adoption in your organization to develop or build all systems and software, because all models are suitable for the development and building of software, and each model has disadvantages and advantages. The choice of the model depends on some of criteria relating to the nature of the project basically and everything related to this project which will be built or developed.

Impose a specific model for adoption, this means that you will abide with its disadvantages and advantages; that will lead to need to issue updates that may be daily or weekly. As we note that in most applications we are dealing with, either be computer applications or mobile applications that require constantly updated almost daily, this makes me feel weary. For me, I would like that development is a yearly, because I am at the beginning of each year waiting for every new, and this method is a kind of suspense and excitement better than continuous updates.

Of course, updates are to fix errors that have not been discovered during the development process because the focus was only on the delivery time according to the model you imposed. Some software needs to integrate all the techniques and mechanism of work of some models together.

For some models that depend on that the customer or his representative be an indispensable member of the development team, and dependence on continuous and direct communication with the customer and periodic meeting with him; this will inevitably be a major obstacle to the development process, for the following reasons:-

1. The customer may not be available at the time that the team needs him, (He may be traveling, very busy, sick, or any other circumstance).
2. If the customer is available, and attend the meeting, he maybe stilling each time add a new idea or remove a previous idea, thus prolonging the development process, and the development team will remain a long time hovering around this project.
3. In case of what is mentioned in the previous paragraph, this leads to confusion of the customer and the team and out of the project requirements.

6) The programs and applications that you design should be flexible and easy to use :-

You should focus on making your designs suitable and easy to handle with all types of users, not all users are proficient in using of software, even if you train them on the application. For example, when you design a system for a particular organization, try to make the system interfaces easy and anyone who deals with them understands them easily, meaning you do not use complex software tools that are understood only by specialists; not all employees are specialized in this field, so this will be an obstacle for them. Try to make software interfaces easy to understand and use.

7) You must make area for users of the programs which you have designed or developed, to express about their views and suggestions for the next issue, this point is considered as the backbone of your success and success of your team :-

You should provide a specific mechanism through which people can express your organization and express their opinions and suggestions. When you succeed in achieving the goal of your organization, people see your software and systems, and are impressed with the way display and content for your software, they will ask you to design software or techniques that you and your team might not even think about. You should also take user feedback about your software seriously, and use it to develop it in the next release.

8) You must have at least one person in the team who has knowledge of the financial, accounting and managerial field, if you want to design or develop accounting, financial or managerial software :-

You should provide at least one person in your organization who is expert in financial, accounting and managerial, because you and your team members are expert in computer science. When you want to design financial, accounting or managerial systems, you will have difficulties in this field, so you will benefit from the person who is expert in financial, accounting and managerial fields in the process of designing and developing those systems in proper competent method.

9) You must have at least one person in the team who has knowledge of more than one important language, if

you want to design or develop programs for different nationalities :-

You should provide at least one person in your organization who speaks more than one language. For example, you and your team members speak only Chinese; your designs for software will be limited to those who speak Chinese, this is an imperfect ambition for your organization, you must expand your range of your programs worldwide. So how can you make those who speak English, Arabic, Russian or other language use of your software? You can do this by using this person who we assume that he can speak in Chinese, English and Arabic, for example. This will expand your business on a wide range and will not be limited to a specific language.

10) You should be continuous aware of and follow-up of the technical and technological developments, the requirements of the business market, and its needs, and discuss all this with the team to come up with new ideas for new designs or development of the software which you designed it :-

You and all members of the team and everyone working in your organization should be continuous aware of and follow-up of all the technological and technological developments taking place in the world, and follow up local and global business markets and their needs, to provide systems and software that keep abreast of these developments. For example, you are the owner of a software development company in China; you see China's development and prosperity, its openness to the world, and the increasing demand for Chinese products worldwide. Why do not you discuss this with all members of your organization to innovation an application that you can display and publish worldwide? The task of this application is to facilitate the purchase and payment process with ease and easiness around the world, and under the supervision of the government, so as not to happen fraud, and to preserve the rights of all seller, buyer and agent. In addition, other ideas that will emerge in the event of continuous follow-up of developments in the world and discuss it in your organization with all members of the organization.

CONCLUSIONS

The field of software development is considered one of the most important fields of different computer science, because of the rapid technological development in the world.

Therefore, software became controlling and organizing large areas of our life activities, and this space expands every day, so it is not logical to leave the development of such important things for coincidence to manipulating our activities and us.

Software is designed according to specific conditions and environment, but when these conditions and environment

change because of the quick development in the world and the challenges the world is undergoing, this change makes the software not live up to these developments and challenges.

What is the solution then?

The solution is to develop that software to live up to those developments and challenges.

For develop that software to live up to those developments and challenges, we need to professional software developer.

Through these steps, you will be able to become a professional software developer.

Through these steps, everyone interested in software development will benefit, and also anyone who wants to become a software developer but has difficulty identifying (What will be point of beginning?, and how to start?); he will benefit from these steps.

Through these steps, we will be able to create a way includes all advantages for all SDLC models simultaneously.

REFERENCES

- [1] Nancy Bindal and Aanchal Mehta, "Survey On Software Development Processing Models", International Journal of Exploring Emerging Trends in Engineering (IJEETE), ISSN. 2394-0573, vol. 02, pp. 100-109, 2015.
- [2] Kazim Ali, "A Study of Software Development Life Cycle Process Models", International Journal of Advanced Research in Computer Science, ISSN. 0976-5697, vol. 8, 2017.
- [3] T. Bhuvaneshwari and S. Prabakaran, "A Survey on Software Development Life Cycle Models", International Journal of Computer Science and Mobile Computing, ISSN. 2320-088X, vol. 2, pp. 262 – 267, 2013.
- [4] Ritika Arora and Neha Arora, "Analysis of SDLC Models", International Journal of Current Engineering and Technology, E-ISSN. 2277 – 4106, P-ISSN 2347 – 5161, vol.6, 2016.
- [5] Nalkar Sanjivani Baban, "Processing Models of SDLC".
- [6] Sukhdev Singh Ghuman, "SDLC Models- A Survey", International Journal of Computer Science and Mobile Computing (IJCSMC), ISSN. 2320-088X, vol. 2, pp. 33 – 38, 2013.
- [7] Kai Petersen & Dejan Baca and Claes Wohlin, "The Waterfall Model in Large-Scale Development".
- [8] Rajendra Ganpatrao Sabale, Dr. A.R. Dani, "Comparative Study of Prototype Model for Software Engineering with System Development Life Cycle", IOSR Journal of Engineering (IOSRJEN), ISSN. 2250-3021, vol 2, pp. 21-24, 2012.
- [9] Mahil Carr and June Verner, "Prototyping and Software Development Approaches".
- [10] P Beynon-Davies & C Carne & H Mackay and D Tudhope "Rapid application development (RAD): an empirical review", European Journal of Information Systems, (1999) eight, 211–223).
- [11] Mekan Taghavi Dilamani, "A short review on Crystal Clear methodology and its advantages over scrum, the popular software process model", International Conference on Advanced and Agile Manufacturing (ICAM), Copyright © 2014, ISPE and ISAM USA.
- [12] Rupa Mahanti & M.S. Neogi and Vandana Bhattacharjee, "Factors Affecting the Choice of Software Life Cycle Models in the Software Industry-An Empirical Study", Journal of Computer Science 8 (8): 1253-1262, ISSN. 1549-3636, 2012.
- [13] Leo R. Vijayasathya and Charles W. Butler, "Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter?", IEEE Software, vol. 33, 2016.
- [14] <https://www.tutorialspoint.com/sdlc/index.htm>
- [15] https://www.tutorialspoint.com/adaptive_software_development/index.htm
- [16] <https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/>
- [17] <https://melsatar.blog/2018/04/02/software-testing-fundamentals/>
- [18] <https://melsatar.blog/2017/06/13/what-do-you-need-to-know-about-the-eight-software-development-phases/>
- [19] <https://www.wisdomjobs.com/e-university/software-development-lifecycle-sdlc-tutorial-1513.html>
- [20] <http://tryqa.com/what-are-the-software-development-life-cycle-sdlc-phases/>
- [21] <https://www.subjectcoach.com/tutorials/detail/contents/introduction-to-software-development-life-cycle-sdlc/chapter/what-is-software-development-life-cycle-sdlc>