



UTT

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

GOBIERNO DE BAJA CALIFORNIA

TOPIC:

Planning of the process of development of software

STUDENT:

Maciel Leyva Ariana Lizeth

GROUP:

10A

SUBJECT:

Comprehensive Mobile Development

TEACHER:

Ray Brunett Parra Galaviz

Tijuana, Baja California, January 8th of 2025

“Planning of the process of development of software”

Software Development Process is an iterative logical process that aims to create programmed software to meet unique business or personal objectives, goals, or processes. The objective is achieved by a software developer writing computer code. However, it also involves multiple steps such as research, designing a data and process flow, writing technical documentation, comprehensively testing, debugging, and pushing it iteratively to live.

The Software Development Process is a series of steps followed to design, create, test, and maintain software applications. It typically includes stages like planning, designing, coding, testing, deployment, and ongoing maintenance to ensure the software meets user needs and functions as expected.

What is a software development plan?

Before we get into exactly how to draft a software development plan, let's align on exactly what it is. A software development plan outlines how the project requirements will be turned into working software. It covers planning, ideation, development, documentation, deployment, launch, and maintenance details.

Why you need a software development plan?

With the adoption of agile development, where more emphasis is put on discovery, experimentation, and iteration than on deadlines, you might wonder if a project plan is still necessary.

It may be true that in some situations dates and forward running plans may take a back seat to ongoing product development. But there are some situations where aligning to a project plan and specific deadlines is valuable. Examples of this are:

In a client/agency engagement: The agency will be expected to provide delivery dates to the client as part of the bidding process. This is especially necessary when the agency will be charging based on time and materials.

In a startup environment with phased funding: The startup will need to know that their funding runway provides sufficient support to achieve the MVP or the next phase of delivery.

In a complex project that is responding to external deadlines: An example of this would be a project that is subject to regulatory governmental requirements that have a fixed date. Failure to deliver before these external deadlines could result in excessive fines for the organization.

10 Stages of Software Development Process

Following are the steps involved in the software development process:

1. Requirement Gathering

The first step in the software development process involves understanding the project's goals and collecting detailed requirements from stakeholders. This includes identifying what the users need, the functionality the software should provide, and any constraints. This stage sets the foundation for the entire project by defining what the software will achieve.

2. Feasibility Analysis

Once the requirements are gathered, the next step is to conduct a feasibility study. This involves analyzing whether the project is achievable from a technical, financial, and operational standpoint. It helps determine if the software can be built within the available budget, time, and resources while meeting the user's needs.

3. System Design

In this stage, the technical architecture of the software is designed. This includes deciding on the software's overall structure, technologies to be used, user interface design, database models, and the flow of data within the system. Detailed system design helps in ensuring that all components work together seamlessly.

4. Prototyping

Prototyping involves creating a working model or a mock-up of the software to visualize and test key features before full-scale development. It allows stakeholders to review and provide feedback early in the process, helping to refine requirements and design elements. This stage helps mitigate risks by validating ideas before investing in full development.

5. Development

In this phase, the actual coding of the software begins. Developers write code based on the design specifications and implement the features required. This is usually the most intensive phase of the software development process and involves converting design documents into functional software components.

6. Testing

Once the code is written, the software undergoes rigorous testing. The goal is to identify and fix any bugs or issues, ensuring the software functions as intended and meets the quality standards. Testing can include unit tests, integration tests, system tests, and user acceptance tests, among others. This phase is critical to ensure reliability and performance.

7. System Integration

The integration phase focuses on ensuring that different software modules work together as a cohesive system. If the software is built in parts or has dependencies, those components need to be integrated and tested together to ensure smooth operation and proper communication between them.

8. Deployment

After successful development and testing, the software is ready to be deployed into a live environment. This stage involves releasing the software to the users or into

production systems. It can include activities like configuring servers, setting up databases, and ensuring the software runs in the target environment.

9. Maintenance

Once deployed, the software enters the maintenance phase. During this time, the software may require updates to fix bugs, patch security vulnerabilities, or add new features based on user feedback. Regular maintenance ensures the software continues to perform effectively over time and remains aligned with evolving user needs.

10. End-of-life

Eventually, every software system reaches the end of its life. This stage involves retiring the software when it is no longer useful or supported, often due to technological advancements, changing business requirements, or better alternatives. End-of-life management ensures a smooth transition for users to new systems or software replacements.

Frequent SDLC Models

The Software Development Life Cycle (SDLC) encompasses various models that outline the processes involved in software development and maintenance. Here are seven commonly used SDLC models:

Waterfall Model: This is a linear and sequential approach where each phase must be completed prior to moving on to the next step. The phases include requirements, design, implementation, testing, deployment, and maintenance.

Iterative Model: This model involves repetitive cycles of development, allowing for feedback and improvement in each iteration. Phases are often repeated until the final product is achieved with success.

Incremental Model: This is more of an incremental model that divides the system into small, manageable parts (also known as increments) with each increment representing a portion of the entire system's functionality. In this approach, each increment is developed and delivered separately.

Spiral Model: The spiral model incorporates elements of both iterative and incremental models. In this model, development progresses in a spiral fashion through repeating cycles of planning, risk analysis, engineering, and critical evaluation.

V-Model (Verification and Validation Model): Consider this an extension of the waterfall model that emphasizes the relationship between development stages and testing stages. In this model, each development stage has a corresponding testing phase.

Agile Model: The agile methodology is an iterative and incremental approach that emphasizes flexibility and collaboration between cross-functional teams. When implementing an agile model, requirements and solutions evolve through collaboration and adaptation to change.

RAD Model (Rapid Application Development): This is not about giving fellow surfers props after riding a killer wave. Alternatively, the RAD model focuses on rapid prototyping and quick feedback from end-users. It involves user feedback and iterations to rapidly refine and enhance the software. It's important to note that these models are not mutually exclusive, and development teams often use a combination of methodologies tailored to the project's specific needs. Factors such as project requirements, budget, timeline, and flexibility determine the choice of an SDLC model.

How to Plan a Software Project?

How to plan a software project? How to estimate a website project? Consider using the following guidelines when developing your next software project:

Step 1: Defining Your Project Goals

The first step in designing a software project is to establish a set of well-defined objectives. Clearly define the scope, purpose, and expected results of the project. This part of the planning process sets the stage for the rest of the process and ensures that the project stays on track and true to its original goal.

Step 2: Developing a Project Timeline

Make a comprehensive schedule for the software development project that includes all of the necessary steps. Estimate how long each stage will take after it has been broken down into more manageable chunks of work. When allocating resources, keep task dependencies in mind. Create a Gantt chart or use any other project management tool to draw out the schedule.

Step 3: Selecting The Right Software Development Methodology

Pick a software development approach that works for you. Whether you're considering an Agile, Waterfall, Scrum, or hybrid methodology, each has its advantages and disadvantages. Check that the chosen approach helps the team work together efficiently and contributes to the project's overall objectives.

Step 4: Budgeting for Your Software Project

Create an all-inclusive plan and software project estimation. Think about the money you'll need to spend on things like programming, hardware, software, infrastructure, licensing, staff training, and documentation. Keep an eye on spending and make adjustments as needed over the project's duration.

Step 5: Risk Management

Find the unknowns and hazards that might derail your software project. Classify hazards according to their frequency and severity, then create a strategy to deal with them. Maintain a close eye on potential dangers as the project progresses, and take

preventative actions as needed. Successful risk management helps prevent problems from becoming worse and keeps projects on pace.

Step 6: Measuring and Tracking Progress

You may monitor the development of your program by establishing metrics and key performance indicators (KPIs). Set objectives that may be achieved as project milestones. Make sure you check in on the project's progress against these criteria regularly and make any necessary modifications.

Step 7: Testing & Quality Control

To guarantee that the delivered software satisfies all of the given criteria and quality standards, a thorough testing and quality control procedure must be put into place. Create a full-fledged testing strategy that covers all the bases, from basic functionality to performance, security, and user experience.

Step 8: Transition to Service

The last step is to release the program to the public and make sure it is prepared for usage. Make a transition strategy that includes everything that needs to be done, from moving data to training users to integrating systems to creating documentation. Maintain company continuity by coordinating with key stakeholders including IT operations teams and end users.

Step 9: Support and Maintenance

A strategy for updating and supporting the software beyond its first release is essential. Maintain constant vigilance over the software's availability, safety, and performance, and address any issues as soon as they are discovered. The anticipated response times and service quality for customer assistance may be defined by creating service level agreements (SLAs).

Sources

- Understanding the Software Development Process* | BrowserStack. (2024, September 18th). BrowserStack. Retrieved on January 8th, 2025, of <https://www.browserstack.com/guide/learn-software-development-process>
- Pham, T. (2024, 30 dicembre). *6 Stages for Software Development Procedure You Need to Know*. Saigon Technology. Retrieved on January 8th, 2025, of <https://saigontechnology.com/blog/6-stages-for-software-development-procedure-you-need-to-know/>
- The Seven Phases of the Software Development Life Cycle. (2025, January 2nd). *Harness.io*. <https://www.harness.io/blog/software-development-life-cycle-phases>
- Kosmowski, B., SoftKraft, Kosmowski, B., & SoftKraft. (2024, September 13th). *Software Development Planning - Perfect Project Plan in 10 Steps*. SoftKraft. <https://www.softkraft.co/software-development-planning/>
- Tech, S. (2024, October 15th). *Breaking Down the Code: How to Plan a Software Project?* SupremeTech. Retrieved on January 8th, 2025, of <https://www.supremetech.vn/how-to-plan-a-software-project/>