Read the text and answer the questions.

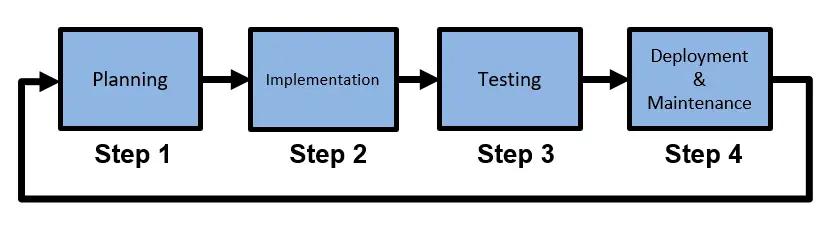
Purpose of a Software Development Process

A solid software development process ensures that high-quality software products are made quickly and well. A well-thought-out and well-executed method has several advantages:

1. Quality assurance: A solid method puts quality first at every step, from figuring out what needs to be done to testing and putting it into action. It includes things like code reviews, testing methods, and quality assurance techniques, which help find mistakes, bugs, and security holes early in the development process so they can be fixed.
2. Consistency and repeatability: A well-defined method gives software development a consistent framework. It spells out the steps, roles, and tasks, ensuring each project uses the same method. This consistency lets teams repeat good practices, reuse parts, and build on what they’ve already done, which increases output and reduces risks.
3. Collaboration and coordination: A robust process makes it easier for team members to work together by giving clear instructions and routines. It makes it easier for people with different jobs and responsibilities, like developers, designers, testers, and project managers, to talk to each other, share information, and work together. When people work together, there are fewer misunderstandings, and development cycles go more smoothly.
4. Risk management: A set method helps find and deal with risks throughout the software development lifecycle. It has ways to figure out the risks, how to deal with them, and what to do if something goes wrong. By dealing with possible problems early on, the process reduces the chances that the project will be late, cost more than expected, or fail in a major way.
5. Scalability and efficiency: A robust process makes scaling possible by letting teams work on bigger and more complicated projects. It helps with allocating resources, sharing work, and putting tasks in order of importance. With well-defined processes in place, organizations can get the most out of their development efforts, shorten the time it takes to get a product on the market, and adapt to changes in project needs.
6. Continuous improvement: One of the most important parts of a strong software development process is constantly focusing on improving things. It supports looking at past projects, lessons learned, and feedback loops to find places where things could be better. By reviewing and improving the process on a regular basis, companies can improve their software development skills, encourage new ideas, and stay competitive in a field that is changing quickly.

**Software Development Process Steps**

The software development process consists of four major steps. Each of these steps is detailed below.



**Taller**

1. Make a conclusion of the text (it may content 150 words)

R/ In summary, having a solid plan for developing software is like having a roadmap that leads to success. This plan brings a lot of benefits. It helps catch mistakes early on, making the final product better. It also makes sure that everyone is doing things the same way each time, which makes it easier to manage and saves time. People on the team can work together better because they know what to expect. This plan also helps to deal with problems before they become big issues. When projects get bigger, the plan helps manage everything and use resources wisely. And by always looking for ways to do things better, the team can keep getting better at making software that people love to use. So, a good plan is like a secret weapon for making great software!

1. Change the word in bold, according to the meaning in the text.

A solid software development process **ensures(**guarantees**)** that high-quality software products are made quickly and well. A **well-thought-out (**carefully planned**)** and well-executed method has several **advantages(**Benefits**)**:

1. Write the definition (English definition) of the words in bold

Consistency and **repeatability**: A well-defined method gives software development a consistent **framework**. It spells out the steps, roles, and tasks, ensuring each project uses the same method. This consistency lets teams repeat good practices, reuse parts, and build on what they’ve already done, which increases **output** and reduces risks.

1. Repeatability: Doing something again and getting the same results.
2. Framework: A basic structure that helps organize things.

3. Output: What you get as a result of your work or process.

1. Look for the information about phrasal verbs and its use.

R/

Figure out, Put into action, Spell out, Build on, Talk to, Deal with, Go wrong, Get out of, Shorten the time, Adapt to, Stay competitive.

1. According to the text, find the meaning of the following phrasal verbs.

* Quality assurance: A solid method puts quality first at every step, from **figuring out** what needs to be done to testing and **putting** it **into** action.
* It **spells out** the steps, roles, and tasks, ensuring each project uses the same method.
* This consistency lets teams repeat good practices, reuse parts, and **build on** what they’ve already done, which increases output and reduces risks.
* It has ways to **figure out** the risks, how to deal with them, and what to do if something goes wrong.
* One of the most important parts of a strong software development process is constantly **focusing on** improving things.
* It supports looking at past projects, lessons learned, and feedback loops to find places where things could be better.

R/

figuring out: Understand or solve through investigation.

Putting it into…: Implement or execute a plan.

Spells out: Explain something clearly and in detail.

Build on: Develop by adding new ideas or experiences.

Figure out: To understand something through investigation, analysis, or careful thought.

Focusing on: Giving special attention or concentration to a particular subject, task, or aspect.

1. According to the Software Development Process Steps

Structure the process of your project, just the steps.

R/ **Quality Assurance and Planning:**

Focus on identifying user requirements and desired features.

Determine the scope of the webpage, including routes, schedules, locations, and employee registration.

Plan a strategy for maintaining high-quality development throughout.

**Design and Framework Creation:**

Develop a well-thought-out visual design that is user-friendly and intuitive.

Create a consistent framework that outlines the layout, sections, and navigation.

**Front-End and Back-End Development:**

Implement the user interface using HTML, CSS, and JavaScript.

Build the back-end to manage data, including routes, schedules, locations, and employee information.

Integration and Testing:

Connect the front-end and back-end to ensure seamless functionality.

Conduct thorough testing to identify and fix bugs, mistakes, or issues.

Collaboration and Coordination:

Foster effective teamwork among developers, designers, testers, and project managers.

Maintain open communication to share progress, feedback, and updates.

**Risk Management and Security:**

Address potential risks and challenges in the development process.

Implement security measures to safeguard user data and sensitive information.

Scalability and Efficiency:

Ensure the webpage is scalable to accommodate future growth and enhancements.

Optimize resource allocation, including allocating resources for different tasks.

Continuous Improvement:

Regularly review past projects and lessons learned to identify areas for improvement.

Implement feedback loops to gather user input and suggestions.

Final Testing and Launch:

Conduct final testing to ensure all features and functions work flawlessly.

Prepare for the webpage's launch by addressing any remaining issues.

Deployment and Support:

Deploy the webpage to a live server for public access.

Provide support channels for users and address any inquiries or problems.