

What is software? What is software engineering?

Software is a program or set of programs containing instructions that provide desired functionality. And Engineering is the process of designing and building something that serves a particular purpose and finds a cost-effective solution to problems.

Software engineering is the process of designing, developing, testing, and maintaining software. It is a systematic and disciplined approach to software development that aims to create high-quality, reliable, and maintainable software. Software engineering includes a variety of techniques, tools, and methodologies, including requirements analysis, design, testing, and maintenance.

Explain types of software.

Application software:- The most common type of software, application software is a computer software package that performs a specific function for a user, or in some cases, for another application. An application can be self-contained, or it can be a group of programs that run the application for the user. Examples of modern applications include office suites, graphics software, databases and database management programs, web browsers, word processors, software development tools, image editors and communication platforms.

System software:- These software programs are designed to run a computer's application programs and hardware. System software coordinates the activities and functions of the hardware and software. In addition, it controls the operations of the computer hardware and provides an environment or platform for all the other types of software to work in. The OS is the best example of system software; it manages all the other computer programs. Other examples of system software include the firmware, computer language translators and system utilities.

Driver software:- Also known as device drivers, this software is often considered a type of system software. Device drivers control the devices and peripherals connected to a computer, enabling them to perform their specific tasks. Every device that is connected to a computer needs at least one device driver to function. Examples include software that comes with any nonstandard hardware, including special game controllers, as well as the software that enables standard hardware, such as USB storage devices, keyboards, headphones and printers.

Middleware:- The term middleware describes software that mediates between application and system software or between two different kinds of application software. For example, middleware enables

Microsoft Windows to talk to Excel and Word. It is also used to send a remote work request from an application in a computer that has one kind of OS, to an application in a computer with a different OS. It also enables newer applications to work with legacy ones.

Programming software:- Computer programmers use programming software to write code. Programming software and programming tools enable developers to develop, write, test and debug other software programs. Examples of programming software include assemblers, compilers, debuggers and interpreters.

What is SDLC? Explain each phase of SDLC

An SDLC (software development life cycle) is a big-picture breakdown of all the steps involved in software creation (planning, coding, testing, deploying, etc.). Companies define custom SDLCs to create a predictable, iterative framework that guides the team through all major stages of development.

The character and the exact number of phases in an SDLC vary between businesses and projects. The most common models are variations on the following steps:

Requirement analysis

In-depth planning

Product design

Coding

Testing

Deployment

Post-production maintenance

An SDLC strategy enables a business to set a tried-and-tested foundation for every software-related project. Teams develop high-quality products with more speed and consistency while the company maximizes its ROI by boosting its ability to:

Meet deadlines and keep projects within the assigned IT budget.

Maintain high code quality standards.

Keep bugs and vulnerabilities out of production.

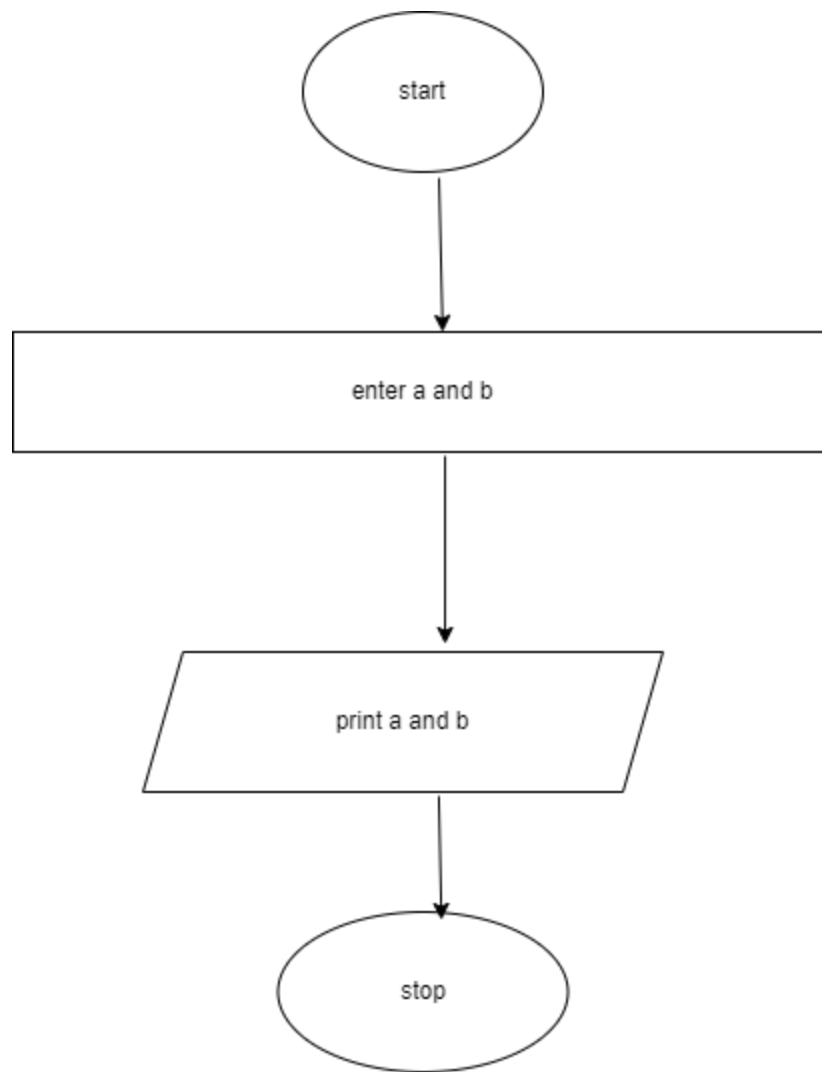
Align product features with business goals.

Prioritize tasks correctly.

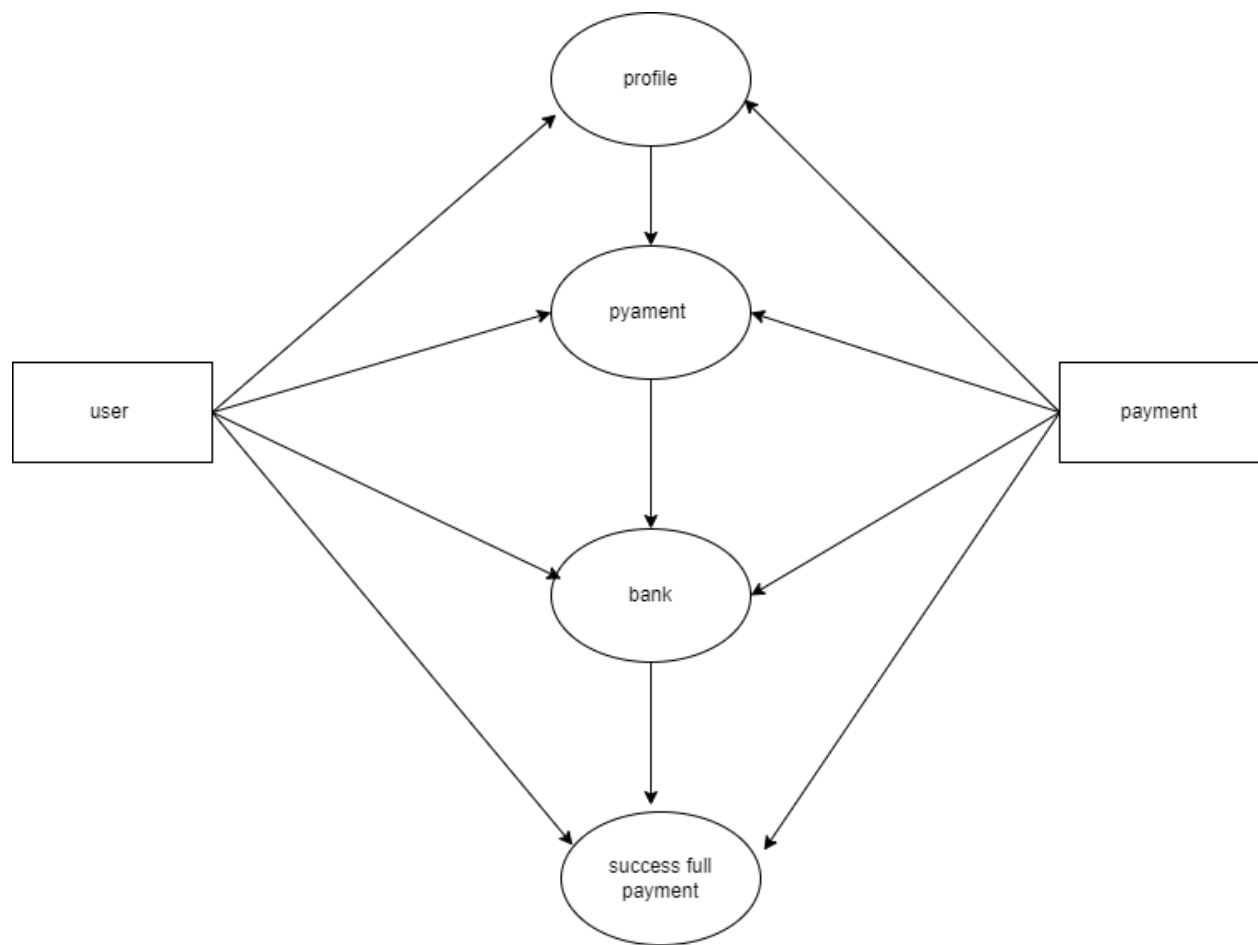
Avoid scenarios in which team members work on the same, conflicting, or low-value tasks.

Lower the number of after-the-fact fixes that impact the UX.

What is Flow chart? Create a flowchart to make addition of two numbers



What is Use case Diagram? Create a use-case on bill payment on paytm.



What is DFD? Create a DFD diagram on Flipkart

