Software Engineering Dictionary

for Beginners

Plain-English keywords for new developers and tech students

Quick reference · Clear language · Real-world examples

By Ana Ribeiro Fernandes

Introduction

When I first started learning software engineering, I was constantly surrounded by words I did not understand. People would say things like "deployment", "latency", or "microservice" as if everyone was born knowing what they meant. I often felt confused, intimidated and out of place, even when I was doing my best to keep up.

Learning these terms usually meant searching for each word one by one, digging through articles and documentation, and hoping the explanations made sense. It was slow and frustrating, and it sometimes made me feel like I did not belong in tech.

This dictionary was created to change that experience for other learners. I believe every student, beginner and career changer deserves quick, clear access to the language of software engineering. You should not feel shut out of conversations just because of jargon.

Inside this guide, you will find plain-English definitions of common terms, grouped by topic, with simple examples of how each word is used in real projects. The goal is to help you read job descriptions, follow tutorials, understand documentation and join technical conversations with more confidence.

Keep this dictionary open next to you while you study, code or attend calls. Any time a word makes you pause or feel unsure, look it up, highlight it and make it your own. You belong in these spaces, and you deserve to understand what is being talked about and what is being asked of you.

Contents

1. Absolute basics

Bug
A mistake in the code that makes the program behave incorrectly or crash.
Example: We found a bug in the login screen that stops users from signing in.

Code / Source code
The text you write in a programming language to tell the computer what to do.
Example: I pushed my latest code to GitHub so the team can review it.

Feature
A useful behaviour or capability the program is supposed to have.
Example: The new search feature lets users find products by name.

Program
A set of instructions that tells the computer how to perform a task.
Example: This image-editing program lets you crop and resize photos.

Programming language
A language such as Go, Python or JavaScript used to write programs.
Example: Our backend is written in the Go programming language.

Syntax
The grammar rules of a programming language (where brackets, commas and keywords go).
Example: The code failed to compile because of incorrect syntax.

Notes:

Acronyms on this page

PR – Pull Request
RAM – Random Access Memory

2. Variables, functions & types

Boolean

A value that can only be true or false.

Example: The isActive flag is a Boolean that tells us if the user is logged in.

Constant

A value that is stored and is not allowed to change while the program runs.

Example: We use a constant for the maximum number of login attempts.

Function

A reusable block of code that performs one specific job and can be called by name.

Example: We wrote a function to calculate a user's total order price.

Method

A function that is attached to something, such as a struct or object.

Example: The User object has a method that formats the full name.

Parameter / Argument

Information passed into a function when you call it.

Example: The sendEmail function takes the address as a parameter.

Return value

The result a function gives back when it finishes running.

Example: This function's return value is the total number of items in the basket.

String

A piece of text, such as 'Hello'.

Example: We store the user's name as a string in the database.

Type

The kind of value something is, such as a number, string or Boolean.

Example: Changing the variable's type from string to integer fixed the error.

Variable

A named box that stores a value in your program.

Example: We use a variable called count to track how many items were added.

Acronyms on this page

CI – Continuous Integration
PR – Pull Request
RAM – Random Access Memory

3. Structs, classes & objects

Class
A blueprint for creating objects, describing what data and behaviour they have.
Example: The Order class contains the list of items and a method to calculate the total.

Class hierarchy
The family tree of classes and how they inherit from each other.
Example: In the class hierarchy, Car and Bike both inherit from Vehicle.

Inheritance
When one class is based on another and automatically gets its data and behaviour.
Example: The AdminUser class uses inheritance to reuse code from the regular User class.

Instance
One specific object created from a class.
Example: We created an instance of the User class for the person who just signed up.

Interface
A list of functions that something must provide, without saying how they are written.
Example: Any type that implements the Storage interface can be plugged into this service.

Object
A concrete thing created from a class, such as a single User in your system.
Example: Each object in the users list represents one real account.

Property / Field
A piece of data stored inside a struct or object.
Example: The email property stores the user's email address.

Struct
A simple container that groups related data together, like a form with named fields.
Example: We defined a Product struct with name, price and description fields.

Notes:

Acronyms on this page
CI – Continuous Integration
PR – Pull Request

4. Memory, stack & heap

Garbage collection

Automatic clean-up of memory that the program is no longer using.

Example: Thanks to garbage collection, we do not have to manually free memory.

Heap

Flexible memory used for values that may live longer or have unknown size.

Example: Large data structures are usually stored on the heap.

Memory

The space in the computer where values are stored while the program runs.

Example: The process crashed because it ran out of memory.

Pointer

A value that stores the address of something in memory, not the thing itself.

Example: We passed a pointer to the function so it could modify the original value.

Reference

A way of pointing to a value instead of copying it.

Example: Passing a reference avoids copying the entire data structure.

Stack

Fast, organised memory used for short-lived things like function variables.

Example: Local variables are usually stored on the stack while the function runs.

Notes:

Acronyms on this page

PR – Pull Request
RAM – Random Access Memory
SPA – Single Page Application

5. Data structures

Array

A fixed-size list of values in order.

Example: We use an array of seven items to store sales for each day of the week.

Map / Dictionary / Hash map

A set of key–value pairs, like a mini phonebook (name to number).

Example: The config map lets us quickly look up settings by name.

Queue

A 'first in, first out' line of items, like people queuing at a shop.

Example: Incoming jobs are placed in a queue and processed in order.

Slice / List

A list of values that can grow or shrink.

Example: We keep all active sessions in a list that we can add to or remove from.

Stack (data structure)

A 'last in, first out' pile, like a stack of plates.

Example: The undo feature uses a stack of previous actions.

Notes:

Acronyms on this page

PR – Pull Request

6. Files, packages & libraries

File
A single document on disk that contains code or data.

Example: The main.go file contains the entry point for the service.

Folder / Directory
A container that holds files and other folders.

Example: All our test files live in the tests directory.

Framework
A bigger structure that gives you a standard way to build applications.

Example: We use a web framework to handle routing and middleware.

Library
Reusable code someone else wrote that you can use in your own program.

Example: We pulled in a logging library instead of writing our own.

Module / Package
A group of related code files bundled together.

Example: The auth package contains all the login and signup logic.

Notes:

Acronyms on this page

PR – Pull Request
RAM – Random Access Memory

7. Compile, build & run

Build

The process of turning source code into something that can be run or deployed.

Example: The build step generates a binary we can run on the server.

Compiler

A tool that translates source code into machine code before it runs.

Example: The compiler warned us about an unused variable.

Executable / Binary

The final file the computer can run directly.

Example: We deployed the latest binary to the production servers.

Interpreter

A tool that reads and runs code line by line without a separate build step.

Example: Python uses an interpreter rather than producing a binary.

Notes:

Acronyms on this page

CI – Continuous Integration
PR – Pull Request

8. Web basics: HTTP, APIs & JSON

API

A defined way for one program to talk to another, using agreed rules.

Example: Our mobile app calls the payments API to charge the customer.

Client

The side that makes requests, such as a browser or mobile app.

Example: The client sends a GET request to fetch the product list.

Endpoint

A specific API URL that does a job, such as /users or /login.

Example: The /orders endpoint returns all orders for the current user.

HTTP

The basic set of rules used on the web for sending requests and responses.

Example: The service exposes an HTTP API on port 8080.

JSON

A simple text format for sending data that looks like JavaScript objects.

Example: The server responds with user data encoded as JSON.

Request

A message sent from a client to a server asking for something.

Example: The browser made a POST request to submit the form.

Response

The message a server sends back, containing data or an error.

Example: The API response includes a status code and a JSON body.

REST API

A common style of web API using HTTP methods like GET, POST, PUT and DELETE.

Example: Our REST API lets clients create, read, update and delete tasks.

Server

The side that receives requests and sends back responses.

Example: The server processes the request and looks up data in the database.

Status code

A number in the HTTP response that shows how the request went, such as 200 or 404.

Example: A 404 status code means the resource was not found.

Acronyms on this page

API – Application Programming Interface
CI – Continuous Integration
HTTP – Hypertext Transfer Protocol
JSON – JavaScript Object Notation
PR – Pull Request
RAM – Random Access Memory
REST – Representational State Transfer
URL – Uniform Resource Locator

9. Databases & SQL

Column / Field

One piece of information in a row, such as an email address.

Example: The email column stores each user's email address.

Database

A system for storing and organising data so it can be found quickly.

Example: All our customer records live in a PostgreSQL database.

Index

Extra structure on a table that makes certain lookups faster.

Example: We added an index on the username column to speed up logins.

Query

A question you send to the database, written in SQL.

Example: The query selects all active users created this month.

Row / Record

One entry in a table, such as a single user.

Example: Each row in the users table represents one account.

SQL

A language used to ask questions to databases.

Example: We used SQL to join the orders and customers tables.

Table

A grid of data in a database, like a spreadsheet.

Example: The orders table stores one row per order.

Notes:

Acronyms on this page

PR	Pull Request
SQL	Structured Query Language

10. Concurrency & performance

Channel

A safe pipe that concurrent pieces of code use to send messages to each other.

Example: Each worker goroutine reads jobs from the same channel.

Concurrency

Having many tasks in progress at the same time.

Example: We use concurrency to handle hundreds of requests in parallel.

Goroutine

A very lightweight thread managed by Go, used to run code concurrently.

Example: Each incoming request is handled in its own goroutine.

Latency

How long it takes for something in the system to respond.

Example: Database latency increased when traffic spiked.

Lock / Mutex

A tool that makes sure only one piece of code uses some data at a time.

Example: We use a mutex when updating the shared map.

Parallelism

Running multiple tasks at exactly the same time on different cores.

Example: On a 4-core CPU we can get real parallelism for CPU-heavy tasks.

Process

One running program on your computer.

Example: The server runs as a background process on Linux.

Thread

A lightweight path of execution inside a program.

Example: The operating system schedules threads onto CPU cores.

Throughput

How much work the system can do per second, such as requests handled.

Example: After optimising the code we doubled the API's throughput.

Acronyms on this page

API	Application Programming Interface
CPU	Central Processing Unit
PR	Pull Request
RAM	Random Access Memory

11. Git, GitHub & version control

Branch

A separate line of development so you can work without affecting the main code.

Example: I created a new branch to work on the payment feature.

Commit

A saved snapshot of your code at a particular moment.

Example: Each commit message should explain what changed.

Git

The most common tool for tracking changes to code over time.

Example: We use Git to collaborate and review each other's changes.

Merge

Bringing changes from one branch into another.

Example: Once the PR is approved we merge it into main.

Pull request (PR)

A request to merge your changes into a main branch, usually reviewed by others.

Example: Please open a PR so the team can review your code.

Repository / Repo

The place where your project's code and history are stored.

Example: The backend repo lives in our organisation's GitHub account.

Version control

A system that tracks changes to code and lets you go back to earlier versions.

Example: Thanks to version control we could quickly revert the broken change.

Notes:

Acronyms on this page

PR – Pull Request

12. Environments & deployment

Deploy / Deployment

Putting new code into an environment, often production.

Example: We deploy a new version of the API every Wednesday.

Environment

A particular setup where the application runs, such as local or production.

Example: The bug only appears in the staging environment.

Local

Your own computer where you do development.

Example: I tested the change locally before pushing it.

Production / Prod

The real system that actual users interact with.

Example: We monitor production closely during busy periods.

Rollback

Reverting to a previous version if a new release causes problems.

Example: We rolled back the deployment after errors increased.

Staging / Test

A safe place that imitates real life so you can test before going live.

Example: We run all end-to-end tests in the staging environment.

Notes:

Acronyms on this page

API – Application Programming Interface
PR – Pull Request

13. Cloud & containers

Cloud
Renting other people's computers and services over the internet.
Example: Our app runs in the cloud rather than on our own hardware.

Container
A lightweight package that includes an app and everything it needs to run.
Example: Each service is shipped as a Docker container.

Docker
A popular tool for creating and running containers.
Example: We use Docker to ensure the app behaves the same on every machine.

Kubernetes (K8s)
A system that manages many containers, starting, stopping and scaling them.
Example: Kubernetes automatically restarts containers if they crash.

Server (machine)
A computer in a data centre that runs applications and services.
Example: The database server has more memory than the web servers.

Notes:

Acronyms on this page
REST – Representational State Transfer

14. Error handling

Catch / Handle
To deal with an error so the program does not just crash.
Example: We catch the error and show a friendly message to the user.

Error
A signal that something went wrong in the program.
Example: If the file is missing, the function returns an error.

Exception
An error that jumps out of normal code and must be caught or handled.
Example: The code throws an exception when the API call fails.

Fallback
A backup action used if the main action fails.
Example: If live prices are unavailable we use cached data as a fallback.

Graceful degradation
Letting the system still partly work instead of completely breaking.
Example: Without the recommendation service the shop still works but shows fewer suggestions.

Panic
A serious error that usually stops the program immediately.
Example: We should avoid panics in production code and return errors instead.

Retry
Attempting an operation again after it has failed.
Example: The client retries the request up to three times if the server is busy.

Stack trace
A list of which functions were running when an error happened.
Example: The stack trace showed exactly where the null pointer error occurred.

Throw / Raise
To create and send out an error or exception.
Example: If validation fails we throw an exception.

Timeout
An error that occurs when something takes too long to respond.
Example: The request hit a timeout after waiting 30 seconds for the database.

Validation
Checking that input is sensible and allowed before using it.
Example: We run validation on the form fields before saving anything.

Acronyms on this page

API	Application Programming Interface
PR	Pull Request
RAM	Random Access Memory

15. Testing basics

Assertion
A statement in a test that says something must be true.

Example: The test includes an assertion that the result equals 10.

Code coverage
A measure of how much of your code is executed by tests.

Example: Our goal is to keep code coverage above 80%.

Continuous Integration (CI)
A system that automatically runs tests whenever code is pushed.

Example: CI failed because one of the unit tests broke.

End-to-end (E2E) test
A test that exercises the whole system like a real user would.

Example: An E2E test signs in, adds an item to the basket and completes checkout.

Integration test
A test that checks how different parts of the system work together.

Example: Our integration tests hit the real database and API gateway.

Mock
A fake version of something, such as a database or API, used in tests.

Example: We mock the email service so tests do not send real emails.

Regression
A bug where something that used to work is broken by new changes.

Example: We added tests to prevent that regression from happening again.

Stub
A very simple fake that returns fixed answers in tests.

Example: The payment stub always reports a successful charge during tests.

Test
Code that checks another piece of code does what it is supposed to do.

Example: Each bug fix should come with at least one new test.

Test case
One specific scenario being tested.

Example: We added a test case for invalid email addresses.

Test runner
A tool that finds and runs your tests.

Example: The test runner prints a summary of passes and failures.

Test suite
A collection of tests that run together.

Example: The full test suite runs in about five minutes.

Unit test
A test for one small piece of code, usually a single function.

Example: Unit tests check that our helper functions behave correctly.

Acronyms on this page

API	Application Programming Interface
CI	Continuous Integration
E2E	End to End
PR	Pull Request

16. Logging & monitoring

Alert

A notification that something important has gone wrong or needs attention.

Example: An alert is sent to the on-call engineer if error rates spike.

Dashboard

A screen showing key metrics and graphs for a system.

Example: The ops team watches the monitoring dashboard during big launches.

Log

A message written by the program to describe what happened.

Example: We logged an error when the payment request failed.

Log level

How serious or detailed a log message is, such as debug or error.

Example: We changed the log level to debug to investigate the issue.

Metric

A number that is tracked over time, such as requests per second.

Example: Latency is a key metric for our API.

Monitoring

Watching metrics and logs to see if the system is healthy.

Example: Our monitoring showed a sudden drop in successful logins.

Tracing

Following a request as it moves through different parts of the system.

Example: Distributed tracing helped us find the slowest service in the chain.

Notes:

Acronyms on this page

API – Application Programming Interface
CI – Continuous Integration
PR – Pull Request
RAM – Random Access Memory

17. Security & authentication

Authentication (Auth)

Checking who you are, such as logging in with a password.

Example: The authentication service verifies the user's credentials.

Authorisation (AuthZ)

Checking what you are allowed to do once you are logged in.

Example: Authorisation rules prevent normal users from accessing admin pages.

Encryption

Scrambling data so only someone with the right key can read it.

Example: User passwords are sent over an encrypted HTTPS connection.

HTTPS

The secure version of HTTP that encrypts traffic between browser and server.

Example: Modern sites should always use HTTPS, not plain HTTP.

JWT

A common type of token containing user information, signed so it cannot easily be faked.

Example: The client sends a JWT in the header with each request.

Password hash

A one-way scrambled version of a password stored in the database.

Example: We never store raw passwords, only their hashes.

Salt

Extra random data added before hashing to make passwords harder to crack.

Example: Each user has a unique salt stored with their password hash.

Secret

A sensitive value, such as an API key, that must be kept private.

Example: API secrets are stored in a secure vault, not in the code.

Token

A small piece of data used as a pass to prove you are logged in.

Example: The app stores the login token and sends it with each request.

Acronyms on this page

API – Application Programming Interface
DOM – Document Object Model
HTTP – Hypertext Transfer Protocol
HTTPS – Hypertext Transfer Protocol Secure
JWT – JSON Web Token
PR – Pull Request
RAM – Random Access Memory

18. Architecture & design

API gateway

A front door that receives requests and forwards them to the right service.

Example: All external traffic first goes through the API gateway.

Cache

A short-term store that keeps frequently used data close by for faster access.

Example: We cache product details to reduce database load.

Event-driven

A style where actions happen in response to events being published.

Example: When an order is created, an event triggers the email service.

Horizontal scaling

Handling more load by adding more machines or instances.

Example: We scaled horizontally by adding two more web servers.

Message broker

A system that passes messages between services, such as Kafka or RabbitMQ.

Example: Events are sent through a message broker so services stay loosely coupled.

Microservice

A small service that does one main job and talks to other services.

Example: The payments microservice is responsible only for processing charges.

Monolith

One large application that does everything in a single codebase.

Example: The old system is a monolith that handles all features in one app.

Scalability

How well a system can handle more users or more data as it grows.

Example: We redesigned the database schema to improve scalability.

State

Information that the system remembers, such as whether a user is logged in.

Example: The shopping basket is part of the user's state.

Stateful

A system that remembers what happened before with a user or session.

Example: The chat server is stateful because it tracks open conversations.

Stateless

A system that handles each request on its own, without remembering previous ones.

Example: Our API is stateless so any server can handle any request.

Vertical scaling

Handling more load by giving one machine more power, such as CPU or memory.

Example: We vertically scaled the database by adding more RAM.

Acronyms on this page

API – Application Programming Interface
CPU – Central Processing Unit
PR – Pull Request
RAM – Random Access Memory

19. Frontend & UI basics

Component
A reusable bundle of structure, style and logic for one part of the interface.

Example: The button component is reused across the entire site.

CSS
The language that controls how a web page looks, including colours and layout.

Example: We updated the CSS to make the page mobile friendly.

DOM
The in-memory tree of all elements on a web page that JavaScript can change.

Example: JavaScript updates the DOM when the user clicks a button.

HTML
The markup language that defines the structure of a web page.

Example: The HTML contains headings, paragraphs and links.

JavaScript
The language that makes web pages interactive and dynamic.

Example: We use JavaScript to submit the form without reloading the page.

Responsive design
Designing pages so they look good on different screen sizes.

Example: Responsive design ensures the layout works on phones and laptops.

Single Page Application (SPA)
A web app that loads once and then updates the page without full reloads.

Example: The dashboard is a SPA built with a modern frontend framework.

Notes:

Acronyms on this page

CSS – Cascading Style Sheets
DOM – Document Object Model
HTML – HyperText Markup Language
RAM – Random Access Memory
SPA – Single Page Application

20. Workflow, agile & teamwork

Backlog

A list of tasks and features waiting to be done.

Example: We pulled the next few items from the backlog for this sprint.

Code review

When another developer checks your changes before they are merged.

Example: Please request a code review before merging your branch.

MVP

The simplest version of a product that is still useful to users.

Example: For our MVP we only built login and basic search.

Pair programming

Two developers working together at one computer on the same code.

Example: We used pair programming to tackle the tricky bug.

Refactor

Improving the structure of code without changing what it does.

Example: We refactored the service to make it easier to test.

Sprint

A fixed time window, often one or two weeks, where a team focuses on specific tasks.

Example: This sprint we are focusing on performance improvements.

Stand-up

A short daily meeting to share what you did, what you will do and any blockers.

Example: At stand-up I mentioned that I am waiting on the API changes.

Technical debt

Shortcuts in code that make future work harder, like mess you will need to tidy later.

Example: We scheduled time to reduce technical debt in the old module.

Ticket / Issue

One piece of work tracked in a tool like Jira or Trello.

Example: I created a ticket to track the bug the customer reported.

Acronyms on this page

API – Application Programming Interface
CI – Continuous Integration
MVP – Minimum Viable Product
PR – Pull Request
RAM – Random Access Memory

Notes