**Job Opportunities and Skillsets in Software Engineering**

1. **What Does a Software Engineer Do?**

If you’re an analytical thinker who enjoys problem solving, software engineering might be the career for you! Software engineers develop a plethora of different types of software, from desktop to web applications, to mobile apps, games, operating systems, and network controllers. They use many technologies to do this, including programming languages, development environments, frameworks and libraries, databases, and servers. There are two categories of software engineer:

Back-end engineers, also known as system developers, who build the computer systems and networks that front-end applications use.

And front-end engineers, or application developers, who are more client focused. They create the software that users will interact with, such as Android, iOS, and Windows applications, and platform agnostic websites.

Software engineers work in a wide variety of settings. You could work in a team developing: Off-the-shelf software, Bespoke software to meet a specific client’s requirements, or Internal software for users in your organization. And within your team, you might work on: Data integration layers which access and load data from a variety of sources into your solution, Business logic which applies real-world business rules to the data in your solution, or User interfaces which enable users to interact with your solution.

On a day-to-day basis you might undertake tasks such as: Taking user specifications and designing new software systems to meet their requirements. Writing code and testing that it works as expected. Evaluating and testing new software programs. Optimizing software programs for maximum efficiency Maintaining and updating existing software systems. Documenting code so that other developers can understand it or, Presenting new systems to users and customers. And some software engineers, such as DevOps practitioners, also integrate and deploy their code onto the underlying infrastructure. As well as building your own systems you'll also be testing, improving, and maintaining software built by your colleagues. The responsibilities of a software engineer are as many and varied as the tasks which they undertake. In a junior position you are likely to start out with a limited set of responsibilities focused on writing, testing, deploying and documenting code, but as your career progresses this will widen. And, in a senior role, you will likely have primary responsibility for multiple areas of the software solution including the planning and designing phases.

Software engineers design and develop a range of software solutions Back-end engineers build the computer systems and networks And front-end engineers build the user interfaces Software engineers undertake a range of tasks, from designing and writing new software to maintaining and updating existing software And as your career develops, you will move from being responsible for one small section of code to many areas of one or more products

1. **Skills Required for Software Engineering**

Hard skills are the practical skills needed to perform a particular role, so for a software engineer these will be the technical skills they need to design, build, maintain, and repair software solutions. Hard skills are learned skills. In the case of software engineering, they are usually learned in a school, college, or university environment, or by studying online courses, diplomas, or certificates. Alternatively, they can be gained from years of experience in the field. Hard skills are quantifiable, so it’s easy to measure whether an individual can demonstrate a particular skill and to certify them in that skill.

Commonly required hard skills in the software engineering sphere include programming languages, version control, cloud computing, testing and debugging, monitoring, troubleshooting, Agile development, and database architecture. Now, Soft skills on the other hand are less tangible. They are your personal characteristics and interpersonal skills. They’re the non-technical skills that are part of your personality and as such, are harder to define, quantify, or certify than hard skills. Because they’re not linked to a particular business though, they are easily transferable between roles and across industries. The job requirements for any role will be a combination of hard skills and soft skills. So, let’s look at some of the hard skills that are relevant to a software engineer.

Software analysis and design skills are essential for a software engineer. You need to be able to analyze your users’ needs through a variety of methods and then design effective solutions that meet those needs. You also need to be able to develop those solutions. Computer programming and coding are essential development skills. While some job roles require a specific language and/or toolset, employers will sometimes welcome you with experience in any language but expect you to cross-train into their preferred language. The coding bootcamp website, Coding Dojo, states that some of the most in-demand languages are currently Java, Python, C#, and Ruby. An understanding of a variety of frameworks and object-oriented principles are also key skills for a software engineer.

While you’d hope to always create flawless solutions, it is likely that at times your code will either not work or not work in the way that you intended. You need testing skills to determine whether your code meets the functional specification of the solution and if it’s easy to use. And when your code isn’t working as expected, you need debugging skills to work out why. And when your solution is complete, you need deployment skills to distribute it to your users. These could include shell scripting, containers, and continuous integration and continuous delivery (or CI/CD) tools. You’ll also need monitoring skills so that you can review the performance of your solutions and troubleshooting skills to resolve any issues that may occur. So now let’s look at the soft skills of a software engineer.

Teamwork is a key soft skill. Software engineers work in a variety of teams, some based on the project they’re working on and some based on their specific role. If you’re practicing Agile development, you may also work in small teams known as squads. And you might also work closely alongside another developer in pair programming. Working in teams enables you to take advantage of each individual’s strengths as well as providing opportunities for you to gain new knowledge and skills. You’ll need to be able to communicate with a wide variety of stakeholders in your project – from technical colleagues to non-technical personnel. For example, you may need to ask: your peers for support and ideas, your manager for guidance and direction, your client for clarification of their needs, and your users how they’d best interact with your solution.

Software solutions are often time-sensitive projects and as such, your manager will be keen that you meet their deadlines. Managing your own time is imperative to ensure that you don’t cause delays to others waiting for your work. And with increasing numbers of teams working across time-zones, what could previously only seem like a small local delay can create a whole day of lost time for someone located elsewhere in the world. Software engineers need great problem-solving skills to succeed. You need them: in the design phase to work out how to create an appropriate software solution, in the development phase to work out the code required to perform the task required, in the testing and debugging phase to locate and resolve any bugs, and throughout the lifecycle of the software to manage any issues that may arise. And when those issues do arise, you need to be adaptable to meet the changing needs of the project. Examples of changes include: Your client requesting a change or addition to the functionality, Your manager requesting you to move onto a different area to meet a looming deadline, Or your user requesting an alternative method of achieving a specific task.

And finally, you need to be open to feedback on your work. Most software teams use some type of peer review system where peers review each other’s code. This helps enforce any corporate standards and improves the code. In a junior role, you’ll also likely have a mentor who’ll provide feedback and pointers where you can improve. And your stakeholders will provide feedback on pre-release and final versions of your solution. Accepting all of this feedback in an open and welcoming way ensures that your solution becomes the best that it can be and that you progress in your role.

А combination of hard and soft skills are essential to a software engineer. Hard skills are measurable, learned skills such as: Programming, testing, and troubleshooting, And, soft skills are your personality and characteristics, such as communication and problem solving.

1. **Job Outlook for Software Engineers**

There are various reasons behind the current high demand for software engineers. Almost all industries need software to compete and grow. This results in a continuous demand for software engineers. Most organizations require applications and websites for their company to function. Some apps and websites are for internal use, while others are used to interact with customers.

The Internet of Things is also driving the need for software that interacts with products. This is a need that will continue to expand for the foreseeable future. All types of software programs, from messaging applications to commerce websites to office software, are available in different flavors from different vendors. Whenever one vendor releases a new version of their particular program, a competing company is likely to also update their program, requiring more software engineers to create the new releases, enhance functionality, and add new capabilities. So, the outlook is bright for software engineers. The US Bureau of Labor Statistics predicts a higher than average 22% job growth rate for software developers, analysts, and testers from 2020 to 2030. That’s an average of almost 200,000 openings each year – in the United States alone!

These roles are likely to be across many industries, from mobile application development to health and insurance. There is also predicted to be a large increase in security software due to the increase in computer security threats and cyber-crime. Many software engineers are graduates with degrees in software engineering or computer science. However, increasingly employers like IBM, Google, and Tesla are starting to hire non-graduates who demonstrate the required skills for the role. This opens software engineering up to a wider range of society ensuring that the cost of obtaining a degree doesn’t preclude candidates from the industry. Salaries for software engineers vary significantly with years of experience in the field. As you acquire more experience and become harder to replace, employers typically provide higher compensation.

In the USA, salaries range from $90,000 for a junior role through to $120,000 or higher for an experienced position. The average of $110,000 per annum for a software engineer is more than 2.5 times the average base salary across the USA. This, alongside bonuses and benefits such as medical insurance, gym membership, profit shares, and retirement plans make a software engineering role an attractive proposition. So, what should you expect from a software engineering role? Dress codes are unlikely to be formal unless you’re in a customer facing situation. Hours are often flexible, sometimes around core hours set in the middle of the day. You may occasionally be required to attend meetings or calls which can dictate a physical or online presence at a particular time of the day.

The number of hours are likely to increase as you get near to product release dates, often without any matching increase in pay. The flexibility to work from home varies across employers, but there is no technological reasons why this cannot be done. In fact, some software teams are spread across the globe, working across all hours of the day. Software engineer employers include almost any type of business that you can imagine. From large technology companies such as Facebook, Amazon, Apple, Netflix, and Google (also known as FAANG), to medium-sized software companies that develop specific software and solutions, to small software houses and start-ups. And almost all large non-technology companies like banks, retailers, and pharmaceuticals need software engineers for their own internal or external systems. And lastly, most employers will encourage continual learning, whether that be expanding your technical skills or enhancing your soft skills.

They’re usually keen to invest in keeping your skills up to date so that you can develop the best software solutions for them. Now in the software engineering world, regardless of your job title, there are different employment options to suit your lifestyle and situation. Employed roles in a company or organization provide stability and a regular income. These can range from an apprenticeship or internship, to a part-time role, through to a full-time role. In software engineering, more so than a typical job, there is a huge independent contracting market and opportunities in the gig economy. These can range from simple website development to super-specialized skills, languages, stacks, and products. In this type of employment, you can provide contract or consultancy services to organizations on a time-based or project-based model or you can work on a freelance basis taking on short-term contracts to work on projects that interest you.

And many software engineers also code for free as volunteers contributing to open source projects. This can help you gain experience, enhance your skills, improve your employment opportunities, and enhance your technical eminence. It also makes you feel good by doing something for a cause or community or social initiative that’s important to you.

Тhe high demand for software engineers is due to increasing needs for software, increasing complexity of applications, and a continuing growth of technology the outlook for software engineers is promising the role of a software engineer can be flexible and satisfying and that employment options for software engineers are flexible and varied, from full-time employment through to voluntary activities

1. **Career Paths in Software Engineering**

A career path in software engineering opens up as you gain experience and attain new skills. Often it heads in one of two directions: technical or management. If you thrive on working with code and solving problems, then the technical path enables you to continue working closely with technology as your career progresses. Or, if you exhibit strong leadership qualities alongside the soft skills of a good software engineer, you might progress onto managing a team of software engineers and the work that they do. So, let’s take a look at a common career path for a software engineer progressing on the technical or managerial path.

Remember that different organizations will use different job titles, but the roles will be similar across the board. You’re likely to start out in an entry-level position as a Junior Software Engineer or Associate Software Engineer. At this level you’ll be developing small chunks of software to meet agreed client specifications. You’ll be assigned a team leader or mentor who will guide you along the way and provide support when you need it. During this phase of your career, you’ll be on a steep learning curve, gaining new skills and experience with every block of code that you write. You could then move on to a Software Engineer position. In this role you’ll be expected to be more independent and able to able to break larger tasks down into smaller achievable sub-tasks.

You may be learning new programming languages and demonstrating an understanding of the software development lifecycle. At this stage, you might be asked to mentor a junior software engineer or take responsibility for a larger part of a project. The next stage of your career could be a Senior Software Engineer position. In this role you’re likely to have involvement across the whole of a project and with an entire codebase. You may be asked to mentor software engineers and to conduct code reviews across the team. And from the experience you’ve gained in earlier roles, you’ll be expected to be able to solve a wide range of problems in an efficient way. If you choose to follow the technical path, then your next role might be as a Staff Software Engineer.

In this role you’ll work as part of the technical team developing, maintaining, and extending software. You’ll be responsible for ensuring that the software meets customer and user expectations and that it uses resources efficiently. If you decide to follow a managerial career path, you could become a Technical Lead. In this role you’ll manage the team of developers and engineers developing the software in your organization. You could be responsible for the entire development lifecycle and report to the project stakeholders. Continuing along the technical path, you might progress to be a Principal Engineer or Technical Architect. At this stage you’ll be responsible for the architecture and design of a software solution, as well as the development of it.

You’ll be expected to create processes and procedures for your team and provide technical direction. On the managerial path, you could become an Engineering Manager. In this role, you’ll ensure that the entire team is appropriately supported and encouraged to progress in their careers. Becoming a Director of Engineering could be the next stage in the progression of your career. It is a strategic and technical role. On the strategic side, you’ll determine priorities for the projects within the company, identify hiring needs, and define long term goals. On the technical side of the role, you’ll be involved in defining new projects, specifying requirements, and overseeing the project.

The lead of an organization’s technology arm is likely to be the Chief Technology Officer (or CTO). In this role, you oversee all of the research and development in the company. You’ll also monitor the company’s systems and infrastructure to ensure that it meets your needs and budget. Even at this level, you’ll continue learning. As technology evolves and new products appear on the market, you’ll be responsible for deciding which ones meet your business needs and could give your organization a competitive advantage. Of course, not everyone who starts out as a Junior Software Engineer follows these traditional software engineering paths, but you’ll find that the skills you gain will benefit you in a wide variety of other roles. If after working on client projects for a while you decide that you prefer interacting with the clients to writing the code, you could transition into a technical sales or customer support role.

If you enjoy coding, but find working with numbers and data is more appealing, you could transition into a data engineer or data scientist role. And, if you find the data side of your new role is your passion, you could move on to become a database administrator or database developer. And if you enjoy finding and fixing bugs, you might transition to a software tester role. IT is a vast field with an abundance of opportunities across many skill sets, so starting your career in software engineering is a great choice even if you use it as a stepping stone to move to other technical or leadership roles in the future.

A career in software engineering is likely to follow a technical or managerial path. At each stage on that path, you will take on more responsibility and a wider range of tasks. And, starting out as a software engineer enables you to take an array of other options at any stage in your career.

1. **Software Engineering Job Titles**

Job titles such as software engineer and software developer are very broad and can be used to describe a spectrum of roles. As well as these titles, you may come across other titles related to specific sub-domains in software engineering. While individual organizations might use company-specific job titles, you’ll find that most fit into one of the more generic titles. Types of software engineer roles include:

* Front-end engineer
* Back-end engineer
* Full-stack engineer
* DevOps engineer
* Software quality assurance engineer
* Software integration engineer
* Software security engineer
* Mobile app developer and Games developer

**Front-end engineers** focus on developing the user interface, or UI, of a software solution. They’re also sometimes referred to as UI developers or web developers.In this type of role, you’re responsible for the visual design of the software, including the layout of UI elements and the overall aesthetics of the application or website. It’s important for a front-end developer to understand how users interact with software and the principles of user experience design. You’ll also need to understand how objects and code run differently on different operating systems, devices, and browsers to ensure that your solution works on the user’s chosen system. Key skills of a front-end engineer include: web development languages and UX and UI frameworks.

**Back-end engineers** focus on the business logic of a software solution. In this type of role, you’re responsible for the core logic of the software implementing functionality to perform tasks such as accessing data and databases, logging information, and caching systems by using application programming interfaces, or APIs. You’ll also be responsible for ensuring the scalability and performance of the solution. Key skills of a back-end engineer include programming languages, application frameworks, web servers, app servers, and load balancers, databases and deployment and containerization tools and technologies.

**Full-stack engineers** work across the whole software solution. In this type of role, you’re able to create both the user interface and the back-end functionality of an application or website using the skills of both front-end and back-end engineers. Key skills of a full-stack engineer include web development languages, programming languages, UX, UI, and backend frameworks, web and application servers, databases, APIs and web services, and deployment and containerization tools.

**DevOps,** or development and operations, aims to deliver software in an agile manner by combining software development and IT operations. DevOps engineers apply agile processes and methodologies to streamline their product development, improvement, and maintenance as well as transcend the boundaries of the traditionally distinct development and operations teams. In a DevOps role you need familiarity with both front-end and back-end technologies. The skills, technologies, and products that a DevOps engineer is likely to use include source code management tools, programming languages and frameworks, scripts, and deployment, containerization, and monitoring tools. Depending on the actual job posting, you might use alternative products and services, but they will be of a similar nature to those shown here.

**Software quality assurance engineers**, also referred to as software QA engineers or software test engineers, test, review, assess, and write software to validate the quality of an application. In a QA role, you’ll develop automated tests, tools, and procedures to test the functionality of software solutions. You’ll use bug tracking software to log any errors you discover and report them to the software development team. The key skills for software QA engineers include programming languages, shell scripting, bug tracking and issue management tools, testing automation tools, And specific software stacks depending on how the application is deployed.

**Software integration engineers** write code to integrate software into hardware products enabling smart devices and the internet of things products. In this role, you’ll use programming languages and frameworks to program hardware such as consumer devices, home security systems, electronics, and other interfaces. The key skills for a software integration engineer include programming languages and proprietary technologies, frameworks, and toolkits.

**Software security engineers,** sometimes referred to as white hat or ethical hackers, work to find security flaws and vulnerabilities in software. As a security engineer, you’ll create systems, methods, and procedures to test software solutions and exploit their security weaknesses so that they can be fixed before the solution ships. The key skills for a security engineer include programming languages, reverse engineering, shell scripting, tools for vulnerability and penetration testing, and network security and encryption tools.

**Mobile app developers** design, develop, and implement software solutions for mobile devices such as smartphones and tablets. You’ll use web skills to create front-end apps as well as learn the various different platforms to write the back-end code interacting with the specific type of device. The key skills of a mobile app developer include mobile operating systems, web development languages, programming languages, as well as Web services and technologies. Games engineers write gaming software for a wide variety of devices, from PCs, to smartphones, to web browsers, to games consoles. As a games developer, you’ll work alongside graphic artists, sound technicians, and game designers to create the code used in games.

1. **Code Ethics**

The Joint Task Force on Software Engineering Ethics and Professional Practices developed a Code of Ethics outlining their goals and standards for software engineers engaged in the design and creation of software. The task force was formed by the Institute of Electrical and Electronics Engineers Computer Society or IEEE-CS and the Association for Computing Machinery or ACM in order to recognize the prominence of computing in global commerce, government, and society. This IEEE-CS ACM joint task force championed the need to hold software engineers accountable so that the present and future status of the field as a beneficial and respected profession is maintained. The Code of Ethics consists of eight principles that pertain to the specification analysis, design, development, testing, and maintenance of software, and are dedicated to serving the public interest. These principles are intended for anyone who is in or related to the profession, and includes engineers, mentors, instructors, managers, students, and policymakers.

Here, we will summarize each principle. However, the full text of each principle and its additional clarifying clauses can be found on the IEEE-CS and ACM websites. The principles in the software engineering code of ethics focus on the following topics:

1. Public Client or employer Product Judgement Management Profession Colleagues Self. The public principle comes first because it is expected that software engineers should act primarily in accordance to positively affect the public good. This includes accepting responsibility for their work with regard to safety, fairness, accessibility, and integrity.
2. Next, software engineers should act in the best interests of both the clients and their employers. They should act honestly and be forthright when it comes to unethical actions such as plagiarism or illegal activities. They should seek consent where necessary and appropriate and honor confidentiality.
3. The third principle relates to the product under production. Software engineers should seek quality while keeping in mind cost and timelines.
4. This next principle outlines how software engineers act with integrity and independence in their professional judgment. They are expected to maintain objectivity and honesty when working with the software and relevant documents they are involved with. Software engineers should not engage in any inappropriate financial activities such as bribery and double billing, nor may they accept duties that create a conflict of interest.
5. Software engineering managers and leaders should also behave in a manner consistent with these principles where it applies to them and those they manage. They should work to minimize risk and employ security procedures. Managers and leaders should also work to ensure realistic expectations of their staff, provide just compensation, and procure intellectual property rights whenever appropriate.
6. The profession is principle describes the duty of software engineers to protect the reputation of the profession by acting with integrity and not elevating themselves at the expense of others. Software engineers should let managers, employers, and clients know they intend to act in compliance with this code of ethics and express concern over violations of the code.
7. Similar to the profession principle, software engineers shall treat their colleagues with respect and fairness. They should encourage their peers to comply with this code of ethics and not take credit for the work of others.
8. The final principle, focused on the software engineer themselves reminds software engineers about the importance of lifelong learning and professional development. They should endeavor to create quality software and strive to conduct themselves in a professional manner.

It should be noted that this code of ethics is not a replacement for conscientious decision-making and common sense, but it can be used as a supplemental guide. Knowing when and where to apply these principles will always be up to the discretion and wisdom of the individual.

Summary & Highlights

* Software engineers design and develop software solutions and maintain and update existing software.
* Learning on the job is a key part of a software engineering role.
* A combination of hard and soft skills is essential for the role of a software engineer.
* There is high demand for software engineers in flexible and satisfying roles.
* A career in software engineering often takes a technical or managerial path, but software engineering skills can also apply to a wide variety of other roles.
* There are many different job titles under the umbrella term of software engineer, each of which has a specific set of skills and responsibilities.
* The software engineering code of ethics contains eight principles: public, client/employer, product, judgment, management, profession, colleagues, and self.