Introduction

Introduction

Fresher to Expert : A beginner's guide for software skill development is a book written for the beginner's in the software field. You may be a fresher in an engineering college or new employee in a corporate world, this book is going to help you.

Why Book?

Over the years, I have used different resources to become a good developer. These resources really helped me to transition from a clueless computer graduate student to competent developer.

So I have put together this book to help any student who out there wondering how they can improve their skills. This book also tries to show how different things students study in college come together to create something useful.

Opinionated

This book is highly opinionated. Resources and advices shared here are what I think is the best. Not all going to agree on my point of view and that should be ok. There will be better resources or different ways to do the things.

Audience

This book is primarily targeted towards the engineering students and software developers in India. But as many resources mentioned in book are globally available it should help any new software developer out there.

At same time, **this book is not how to**

- Get good marks in Engineering exams
- Get job in MNC
- 24 hour guide to become successful

This book is written for the people who are passionate about software development and want to broaden their horizon.

Acknowledgment

This book is a effort of collaboration. This book is developed in open source spirit. Lot's of people helped right from the idea to the completion of book.

A big thanks to Shashidhar and Shashi Sagwan for their help in refining language. Thanks to Hemalatha, Rose, Swathi V for their contribution. Thanks to all my friends at Zinnia Systems for sharing their perspectives on the book.

Read - Keep up with world

Read - Keep up with world

Textbooks become obsolete faster than you think. In a fast moving industry like us, technologies come and go in few years. So keeping up with latest trends and technologies is very important for a developer.

People coming from the engineering background hate to read. The part of the problem is with the kind of books recommended for reading. The resources shared here are nowhere near the boring or useless.

In following chapter, you are going to find variety of resources to read,watch and enjoy. These resources will help you to improve your thinking about software development.

Feedly

Feedly - News paper of the Developer

Everyone read news paper to get updated with world news. Same way developer can be up to date with technology news using a RSS reader or digital newspaper.

Many developer share their experiences with programming,tools or practices through their blogs.You can write your own too. We will discuss more about that in next section. First let's read other's blog and understand their opinions and skills.

RSS

RSS stands for *Rich Site Summary*. It's a standard which is usually used by blogs to publish their new articles. You can subscribe to these feeds and get updates when there is new articles.

Feedly

Feedly is free service which allows you to subscribe to many websites and read from one place. It's available on Web,Android and iOS. Use this as magazine/newspaper to get up to date with technology.

What to read?

There are wide variety blogs on different things. The following are my favorite ones which I spend most time reading. You can also start with this list, as you become more comfortable with tool you can start exploring others.

The following is the list

- Verge

  Great blog to get latest information on product announcement, device reviews. You should read this in order to get update to date with latest products in the market place.

- Hackernews

  It's a new aggregator. Here the best articles of all over Internet are posted. Read the comment section of each blog to understand more about the article. It's not the prettiest looking site, but don't be fooled by that. The site has best developer community and it's right place ask questions and get your doubts resolved.

- Techcrunch

  Want to build a company? It's right blog to read. It's has good articles about how different startups are built from scratch.

Now the following are the optional one's. If you are interested in those specific areas, you can add it to the feedly.

- Smashing Magzine

  Great blog on how to build nice user interfaces. If you are interested in user interface design on web or mobile this blog is must read.

- Harvard Buissness

  It's a blog you should read if you are interested in business. This blog publishes great articles on entrepreneurship and other aspects of the business.

Read it Later - You don't need to rush it

Sometimes you may find a great article that you really want to read in depth but don't have time. Don't worry . You can use tool called Pocket . Just sign up there and in feedly setup the pocket. From now on you can save the articles to pocket and use from web,android or iOS devices.

Read everyday

Make a habit of reading feedly everyday. It's like morning news paper. As it's available in mobile, you can read wherever you want. I have been reading from last 6 years and it has served me very well.

Happy reading :)

Books

Books

Coming from the engineer background, I always hated reading textbooks. Most of the textbooks were dry and presented topics in not interesting way. But over the course my career I have found very interesting books which make you think in completely new direction.

As you now are getting up to date with world technology news its very important to read books. Books distill the best practices of craft and gives very depth the understanding of the topics.

The books presented are here like starting points to the interesting and fun programming books. After reading few of these, I hope you get interested in read more books which again will help you to read more.

What kind of books you should read?

Try to avoid the books which teaches you just syntax and semantics. Though syntax and semantics are important for the programming they may not be helpful in the longer run. So I have avoided those kind of book and have given set of books that goes deeper into the programming.

You can get hard copy and e-books of these from amazon. If you try hard you should be able to get downloadable pdf too.

Programming Books to start with

- **Coders at Work**

In every department of engineering, people learn about great people contributed to the field. In music, every one will learn about Beethoven, Tyagaraj. Sadly in our field, we hardly know about the those people. So let's start our reading list by knowing the great people in our field and understanding their contribution the computer science.

Coder's at Work is a collection of interviews with Great programmers from various sections of computer science. Going through these interviews make you familiar with how people learn programming and how they contributed overtime. It's gives you inspiration to become just more than a programmer.

- **The C Programming language**

Everyone starts learning programming with C programming language. This book not only teaches you the C programming language but it's also teaches you the art of programming. Even if you have read any other C programming books, I recommend you to go through this book and do the exercises . C will be never same again.

- **Unix programming environment**

Learning about the operating system is cool. But what about learning the operating system from it's creator? Unix programming environment is written by creators of the UNIX operating system.

To read this book correctly, grab a Linux machine and practice it each and every example. You will learn more about operating system in one month than your six month operating system course.

- **Operating Systems Design and Implementation**

This book teaches you operating system concepts with real source code. If you ever wanted to learn how to build your own operating system, this book is must read.

- **The practice of programming**

Ever wondered how different softwares are written from scratch? . This is the book you should read to get answer. The book is 40 years old but still relevant today. Book gives in depth view of what it takes to build different kind of programs from scratch. Book includes the entire examples source code in C. So if you know C you should be able to pickup the book and learn the best software techniques.

Additional programming books

The following are additional reads. If you are familiar with specific technologies pickup those books and run with it.

- **Thinking in Java**

This should be the book you should read if you want to be fluent in Java. It not only goes in depth into the Java programming language instead it also teaches how to do Object Oriented programming right.

- **JavaScript - The good parts**

JavaScript is one of the most famous programming language in the world. It's also most misunderstood language in the world. If you want to learn the language right, this book is highly recommended.

- **JavaScript Patterns**

People think JavaScript is just a scripting language. But there is more to it. Ever wondered how the libraries like jQuery work? If you have, then this book is must read. This book teaches how different aspect of the language gel to create powerful libraries. It's little bit advanced book. So have patience.

Entrepreneurial / Inspirational books

The books which I am going to list now are the ones which doesn't teach how to code. But they teach you one of the most important factor of software development - design. These books talk about how to build great designed products and also how to build long lasting companies.

Read these books to get inspired. These books show how people achieve greatness in their field.

- **Outliers : The story of success**

How people like Bill Gates, Steve Jobs achieve great things in life? Is there any commonalities in successful people across different field. This book explore these ideas in depth which makes it a very interesting read.

- **Build to Last**

How great companies built? This book takes a research oriented approach to understand how great companies built overtime. Great read if you are into entrepreneurship.

- **Innovator's Dilemma**

Opposite to the previous book, this book explores how great companies fail. This book is fascinating read as it gives in depth look into the companies repeating failures.

Reading books is very important to improve the skills and horizons of thinking. So make a habit of reading books.

MOOC - Online courses

MOOC - Massive open online course

Want to learn from the best lectures in the world? Then you should start using online courses.

From last few years, there is a new trend in education. In earlier times, people were travelling great distance in order to get quality education. But with Interenet that's no more required. With Interent you can learn anything sitting in home. Most of the time these courses will be free or will be charging very less money. These kind of courses are called as MOOC- Massive open online course.

Whenever people hear about the online courses, there will be some questions arise. Particularly around the quality of the teaching and content. The following are few of FAQ asked by students

- **Is quality of MOOC courses are good?**

  The courses provided by the MOOC companies like Coursera and Udacity are top notch.They are usually taught by the best in field. Most of the instructors hold Phd in the field and respected lecturers in the top colleges.

- **Are they going to charge a lot?**

  One of the advantages of online courses is they don't cost too much for the universities or companies to offer them to large amount of students. There are many great free courses in Coursera and very less costly courses in Udacity. So you should get your money worth.

- **Do this courses offer hands on?**

  Almost every computer related MOOC course will have extensive hands on. You will have sample examples and exercises for every week. These exercises should be submitted and will be evaluated by instructor for their correctness.

- **Do I get a Certificate?**

  Yes. If you follow the course and complete the exercises you will get a certificate with marks by the time of completion.

- **Do instructors help students like in real classes?**

  Yes. These courses not just videos recorded by the lecturer. Every week there will be quizzes and exercises which will be evaluated by the instructor. There will be also group where you can ask instructor or fellow students any doubt about specific topic.

- **Can I learn with my own pace?**

  Coursera keeps the videos and exercises even after the course is completed. So if you don't care about the certificates you can watch videos and follow the course in your own pace.

- **Are these courses are limited to computers?**

  No. There are variety of courses which cover business,fashion,psychology etc.

- **Which sites offer the MOOC courses?**

  Most well known sites are coursera, udacity and khan academy

MOOC are the new way of learning. These courses offer high quality content with very cost effective way.

Courses to start with

The following are the few courses you should start with. Once you become comfortable, you can explore more courses in future.

- Algorithms: Design and Analysis, Part 1
- Automata
- An Introduction to Interactive Programming in Python
- Machine Learning
- Functional Programming Principles in Scala

Youtube

Youtube

You may have used Youtube for watching songs, movies or television shows. But do you know youtube hosts great educational content? You can use youtube for learning many things without paying any money.

Youtube is popular site to share the videos across the world. People use it's popularity to share their content to the world. In same way many universities also share their content through youtube.

University courses

There are many universities in the world, who share their recorded course contents on youtube. So if you want to learn a specific topic most probably youtube has you covered. The following are few of the courses availble in youtube in it's entierity.

- **Statistics 101**

  This course is offered by Harvard university. If you want to learn the probability theory and other statistics courses this course is best one to start with.

- **Introduction to Psychology**

  This course is offered by Yale university. It's a good starting point to start understanding the wonders of brain.

- **MIT open courseware**

  This is a huge collection of courses offered by MIT which can be accessed freely. Courses cover different disciplines like Physics,Chemistry,Computer science etc.

These are few courses available in the Youtube. Use youtube's search capability to explore more.

Youtube course vs MOOC

Unlike MOOC courses, youtube courses do not have any hands on or instructor feedback. They are just recording of the sessions inside the college. Though MOOC courses are superior than the Yotube one's, not every course is offered in MOOC format yet. So stick with Youtube courses till those courses come in MOOC format.

Conference talks

Every year in different parts of the world many computer science conferences are held. It's almost impossible to attend all those conferences but don't worry most of the videos of these conferences are available on Youtube.

Conference talks are best way to learn the state of art technology from the best in field. Over the years many great videos have changed the way I think about the computing and programming. These talks are highly useful to improve the skills.

Following are the few talks to start with

- Inventing on Principle
- The process of Innovation
- How to design a good API and why it matters
- Legacy - Creating software that lasts long

I have collected the conference videos that I enjoyed watching here. They will be next ones to explore after finishing the above list.

Youtube is your friend. Use it wisely to get most of out of it.

Github - Reading other's code

Github - Reading other's code

If you ask any professional in the industry, what is the best way to improve programming skills, they will give one common advice "Read other's code". Reading other's code is as important as reading your own. Reading master's code is much more desired as it will show you best practices.

Why to read other's code?

Most of people think to get better at programming, they have to code a lot. Those people are half right. As much you write code, you have to read also. Often people wonder why they should read code written by some one else?

In any field people get better by observing and learning from best people in that field. Great writers are also great readers. In one of the interview, George Martin, author of Game of Thrones told **"It is very important that you read lot of books before you write one"**. Reading books teaches writer how to construct different pieces of story and put together in a nice manner. It's also true in field of music. Many people learn to play different instruments by watching others play.

Same way you can learn great things of building software by reading other's code. By reading, you can understand how different abstractions in the programming is achieved. If you read code of masters in the field, you will be surprised to find out how straight forward their code is and how they compose things nicely. As Erik Meijer, inventor of Reactive programming, says **"Great programmers write baby code"** .

Reading other's code is not easy

Like it is not easy to read new author's book, it's not easy to read other's code. To understand other's code, you should understand their style of expressing things. Every author/programmer has their own way of expressing things. Also different people have different priorities. Some programmers are obsessed by the performance whereas some are obsessed by the readability of code. So it will take time for you to get used to the reading. It will be hard in the beginning but don't give up. Over a time you will become master at reading other's code.

Everyone has their own way of reading. If you want to know how the best programmers read other's code read "Coders at Work" book, referenced in earlier chapter.

Open source

Now you have convinced that you should read other's code. But you may be thinking why people will give access to their code? Even if they give the are going to charge right?. Don't worry. There are so many good people in world, who put out their great creations in public to read and use it for free. This way of doing things is called OpenSource.

Lot of people think open source is getting things for free. But the actual underlaying principal of open source is to learn from others knowledge. The great projects like Linux,Android,Chrome etc are open source which means you can look at their code and learn from it.

Github - Repository of open source code

Github is a website which hosts large repositories of open source. You can freely sign up to this website and browse through all public repositories. You can also download the code and many times you can extend also!.

Repositories to start with

The following are few of the repositories you can start your journey of reading code

- jQuery - Complete source of famous javascript library JQuery.
- underscore.js - Small javascript library which teaches you advanced javascript
- JSON- Parsing JSON in javascript with less than 500 lines
- Junit - Unit testing framework in Java.
- Java collections - Java collections source code is one of the best way to learn Java.
- Linux Kernel - Complete source code of Linux Kernel. Only for adventures people!!!

Reading other's code is an art. It takes long time to learn. So have patience and start reading code

Conclusion

Conclusion

In last few chapters, we have learnt about lot of resources available to improve our knowledge. Now it's your time to use them and get great benefit out of it.

Programmer setup

Programmer setup

In this section, we are going to look at the setup of a developer computer and other devices. Having right set of devices will make you more productive and efficient.

As paint brush is for an artist, laptop/desktop is for a developer, So do the setup with care.

Operating system

Operating system (OS)

An operating system is the program that, after being initially loaded into the computer by a boot program, manages all the other programs in a computer. So it's an essential component of the system software in a computer system. There are many families of operating systems available in the market. Examples of operating systems are Unix,Linux,Windows,MacOS etc.Windows is the most widely used OS in the world as it's good for most of casual users of the computers. But as a developer, you can make better choice by not using Windows for developmental work because you are more than a casual user, consider switching to Linux based operating system. Well ,You can make your own choice, that's all upto you!.But as I am a Linux user,I would like to highlight its usage over other operating systems.

Why Linux?

Because it's assembled under the model of free and open-source software development.The underlying source code may be used, modified, and distributed by anyone under licenses such as the GNU General Public License. The following are other few reasons why you should adopt linux.

- **It's Unix based**

  Unix is famous for its simplicity and robustness. As Linux is based on Unix, you will get similar robustness and simplicity out of the box. No more random crashes or hangup. Linux also has awesome security which means no worries of viruses, which is everyone's dream!!.

- **Operating system course teaches you Linux**

  All the concepts in operating system course are based on linux operating system. So if you want to understand shell, process management, file system , inode number etc its better to use Linux. In linux you can see all these concepts and make hands on which makes your understanding much better.

- **Most servers run Linux**

  It is a leading operating system on servers .Most of the server programming , like Web programming, network programming is done on Linux. Most of the websites in the world uses linux to host and run their sites. So if you want to learn all about network programming or server side development,linux is the ideal option.

- **Free**

  Linux is free to use. So if you use linux on your machine you don't have to pay any money. If you are using pirated OS you should stop using it. Switching to Linux will get you an operating system for free without stealing!!

- **Great development support**

  Linux supports C development out of the box. Similarly, you can develop Java,python or any other language programs very easily on the linux. So as a developer you get an excellent support from the operating system to the programming. Linux developer toolset contains compilers,debugging tools,the LAMP application stack, and git and lot more.

But Linux is command line right?

This is one of misconception about Linux. Linux,in its early age, used to come only with command line interfaces. But with newer distributions of linux like Ubuntu you get nice GUI like you find on windows.

Ubuntu - Linux distribution for beginners

There are many distributions of linux available. You might have heard of Redhat, Fedora,Ubuntu etc. To make a switch to Linux, you can start with Ubuntu. Ubuntu is very easy to use and well supported operating system. So you can start with ubuntu and as you become more comfortable with linux you can switch to other distributions.

I have windows, can I tryout linux without partitioning?

Yes of course!. Before you make up mind to use Linux as your regular operating system you can try out using virtualbox. Find more about how to install Ubuntu on virtualbox refer this article.

Power of command line

Ubuntu comes with nice GUI. You can use ubuntu like Windows just by using GUI only. But the real power of linux lies in command line, and it can be quite easy to learn and use, and its value soon becomes apparent after a little practice. Many developers fear the command line when they use first time. But please don't give up. Even a basic familiarity with it can make computers easier to use and facilitate performing tasks that might be difficult or impossible with a GUI. Such familiarity can also lead to an improved understanding of how computers actually work. So I highly recommend you to spend a lot of time on Ubuntu command line. You

will find it highly useful after sometime.

If you are more interested to know the history of Linux and how it has changed people's view, you can watch below video.

So switch to Linux and become smart and power user.

Code editor

Source code editor

Source code editors are the most fundamental programming tool, as the fundamental job of programmers/developers is to write and edit source code. Many source code editors and IDEs have been evolved over a time.

Here are some well-known source code editors :

Vi and emacs

Vi and emacs are two most famous keyboard-only code editors. These editors are 40 years old, still they are used by millions of developers everyday. What makes so old software to be so useful and why a modern developer like you want to learn?

Following are few reasons which will enforce you to give vi and emacs a shot

- **Ubiquitos**

  You can find vi and emacs on almost every distribution of Linux ,Unix and Solaris. These come pre-installed in OS, which means you don't have to install anything to use them . This ubiquitous nature becomes important, when you are accessing remote machines where you may want to update scripts or some other files. And most of the remote servers operate on Linux. In remote machine, you may not have ability to install new software. So if you are comfortable using vi or emacs you don't need to worry as most probably they are going to be there everywhere and anywhere!.

  vi and emacs provide keyboard macros for performing all sort of operations like editing ,saving a file, moving between lines, selecting text etc. Beginners are uncomfortable to use it and they think that it's overwhelmingly hard to master it. But that's not true!. So if you've never used it before Just give it a shot!.

Most of the computer users are dependent on GUI to interact with computer, they perform actions through graphical icons. As a developer you have to do rapid text editing and code refactoring for which GUI becomes cumbersome to use. Each time when you lift your hand from keyboard, it makes you think where to click now?! So using keyboard macros not only makes you a better user, it also saves lot of your time. .

- **Extensible**

  Both vi and emacs allows you to extend the editor with plugins. There are thousands of plugins available which allows you to do syntax highlighting, code compilation, debugging and lot more from the editor itself without leaving your keyboard. Now that's the power!.If you are really an adventurer you can even read your emails in these editors.Isn't that cool!.

Both, vi and emacs, have a steep learning curve and looks too old fashioned for getting started. Don't judge an editor by it's look. For developers like us performance matters more than beauty.

SublimeText

What if you want all power of vi and emacs but with nice GUI and project management. Then you should give SublimeText a try.

SublimeText is clone of Textmate which is a superb editor of Mac. SublimeText has a slick user interface and is very lightweight with extraordinary features. Sublime is available for Windows,Linux and Mac.SublimeText allows you to edit many languages like Java, JS,HTML.

Sublime Text can be downloaded and evaluated for free, however a license must be purchased for continued use.

This book is written using sublime text!.

The other editors you can try are atom,Notepad++ etc.

IDE

All the above mentioned tools are focused mostly on code editing and reading. What if you want an integrated environment where you can

- Edit
- Compile
- Build
- Debug
- Project Management
- Source code management
- Refactor

then you can use an IDE.

IDE stands for Integrated developer environment. It's a software which allows you to do all the above tasks in one place. Though

IDE are more powerful than normal code editors, they consume lot of resources and make you slow as they come with lots of options.

The following are few options for IDE

- **Eclipse**

Eclipse is an open source IDE developed by IBM and now maintained by Eclipse.org. Eclipse supports multiple languages like Java,C, C++ etc. You can extend eclipse by installing plugins which allows you to build, source code management etc.

- **IntelliJIDEA**

The free open-source version of IntelliJ IDEA, a premier IDE for Java, Groovy, Scala and other programming languages.Though you can use eclipse for development, IDEA has few advantages over eclipse. One of the advantages is memory usage. Eclipse is a memory hog. But IDEA is very light on memory. This makes IDEA extremely fast to use.

Mastering a Code editor or IDE is extremely important for a developer. So try all the above mentioned tools and settle with the one you like. Once you settle you are going to spend rest of your life to master it in the process of becoming highly productive at work.

Smartphone

Smartphone

These days everyone owns a phone. Most of the those phones are touch screen based phones with advanced capbilities, called Smart phones. You may own or might have used one of those smartphone. Though most of the time, people use their phones to make calls, messaging or play games you can do more as developer. In this section we are going to discuss how owning a smartphone can improve you as a developer and add a new dimensions to you knowledge and productivity.

What is a smartphone?

Smartphone is a hand held computer which is normally operated using touchscreen interface. They are essentially limited version of desktop or laptop. They run operating system that are created from basis of computer operating system.Various mobile operating systems are android,iOS,blackberry,windows etc.

Which smartphone is better?

Android and iOS are the most popular smartphone operating system in market today. You may be wondering which phone you should buy, whether it is an iPhone running iOS or phones with Android OS. Both phones have similar user facing characteristics but they have different level of control for developer.

iOS is based on Apple's Mac operating system and Android is based on Linux operating system. As we have discussed earlier, linux is a viable operating system for a developer in Desktop and its same for Phone too. So if you want to explore the phone like you do with your computer then go with Android!

Why Android ?

The following are the few reasons which supports my decision for choosing Android smartphone.

- **Linux based**

  Android is a mobile operating system based on linux kernel. It follows similar conventions for file system, process management etc like Ubuntu or any other linux based computer operating system. So its like carrying a linux system with you.

- **Open source**

  Android is an open source project. This means you can look at the code and try to learn how a mobile operating system works. Not only you can play around code, if you feel adventurous you can create your custom operating system from this source code like thousands of other developer does. One of the most famous custom ROM is Cynogenmod. Here thousands of developers collaborate and create their own version of Android flavor. You can join them too!.

- **Apps**

One of the best things about Smartphones are, they come with lot of great applications. It's true in case of Android too. You can download apps from Google Play store, which hosts more than millions of apps. Some of the must have apps on developers phones are:

- Feedly - Allows you to read news anywhere
- Pocket -Provides read it later service when you dont have enough time for bigger articles
- Calendar - Keeps track of all your appointments
- Google Keep - Todo app

These apps are highly useful to keep track of news and work right on the phone. They sync seamlessly on web so you don't have to put info in multiple places.

- **Java based Application development**

Using apps developed by others are good. But how about writing your own android App? Well, if yes, then that's going to be awesome!. One of the good thing about Android is that it uses standard language like Java and IDE tools like Eclipse, Idea for mobile application development. Most of these tools you already know from previous sections, which means you are ready to build apps for your phone. Learn how to do android development here

- **Reading device**

As we discussed earlier in section, reading books is important part of improving knowledge. E-books are great way to read books and your Android device is a good place to read your books. We will go more about reading device in next chapter.

Which Android phone to Buy?

Now you know the value of owning a Android smartphone, you may be wondering what phone you should buy?. People consider lot of things when they buy phone right from price to camera quality. So the recommendation here is based only on the developer friendliness. Normally I want to buy an Android phone which

- Runs stock Android
- Should be upgradable to newer OS
- Should support custom ROM like Cynogenmod
- Easy to customize both in OS and apps
- Good developer community support

As you see I have no say on hardware features and price range. That's all up to you to decide . But if you want to own a great developer friendly phone I recommend you Google Nexus series phones.

As you know smartphone can be more than just phone or gaming device. Put your phone in use to improve your skills and have fun exploring it.

Reading device

Reading devices

From last chapter we know that why reading books are important and it's very clear that reading is integral part of the developer life. So, In this section we are going to focus on dedicated devices which help you to read better and more.

Ebook Revolution

Ebooks have revolutionized the way we read. E-reading has lot of advantages over traditional reading in terms of delivery and pricing. So most of your reading will be happening in computer/phone rather than on a traditional book. Being an e-reader has few advantages over traditional way of reading.

- You can read anywhere as and when you want!
- Easy to store thousands of books
- Cheaper than physical copies

But e-reading also comes with additional challenge as laptops or Phones are not well suited for reading books for long time. On phone, the screen size is too small to do any kind of serious reading and on laptop, its difficult to hold it for long time and having keyboard makes it cumbersome.

So having a dedicated device for reading is always great. You may not have to invest immediately as you can do most of the reading through computer or phone. Incase, you are a serious reader give a consideration to reading devices.

The following are few of the devices which gives you a superior reading experience when compared to phone or computer.

Amazon Kindle

Amazon is a biggest retailer of e-books in the world. Amazon kindle is a dedicated ebook reader which allows you to store and read thousands of books or magazines and it's light and portable. The basic version of Kindle comes with E-ink display which has battery life that lasts over 30 days.Other advanatges are that it is easy on eyes and allows you to read in any position when you are reading for longer period of time.

Every thing comes with its own pros and cons!.The con of the kindle is that, its only for dedicated reading. You cannot do much with kindle other than reading. It's ok for an avid reader. But for majority of developers, other than just reading they also want to access the web, apps like feedly which makes reading complete.

Tablet

For most of the casual readers, a tablet is much more useful than a kindle device. Tablet normally sports a screen from 7-10 inch touch screen display. Having a bigger screen gives a pleasant reading experience and also gives you better battery life.

In market, there are tablets running on differnt platforms like iOS,android,windows etc from various manufacturers. The tablets running on iOS are called as iPad .

iPad

I have recommended an Android phone in Smart phone section as it allows developer to be much more productive. But in tablet section I recommend iPad over any android tablet. The following are few reasons

- iPad has better apps for reading and learning
- Extremely good battery life and durability
- Very light weight

The app ecosystem of iPad is in much more improved state compared to android tablets. So if you are planning to invest money on dedicated reading devices for serious reading I recommend iPad.

Apps

There are great apps available for reading on android and iOS platforms . The following are few apps which integrate well with web, and makes reading seamless across devices.

- iBooks - Book reader from Apple
- Amazon kindle - Kindle book reader
- Google play books - Sync reading between Android,Web and iPad
- Feedly
- Google apps like Gmail,Google Search, Google maps etc

Reading on iPad or kindle is much more pleasant and satisfying compared to Laptop or phone. So if you want to take your reading to next level, own a dedicated reading device!.

Conclusion

Conclusion

Devices are integral part of a developer's life. Having a right Laptop, Smartphone and Reading device with good software makes his life much more productive and easier.

In this section, we discussed essential devices and softwares to begin with. Use it as a starting point and explore more essential tools by yourself.

Happy reading, coding and have fun.

Learn the craft

Learn the craft

In last two sections we concentrated on improving skills through reading other's code, books and blogs. Reading is great but not enough. Until you code and see the things yourself, it will not be completely understood.

In this section, I am going to answer one of the most common question asked by many freshers in software industry. "How the things I learnt in college relate to the real world problems?".

This section contains resource to build things from scratch. There will be books,courses and other resources which will make you build thigns in step by step manner.

These resources assume you know at least one computer programming language, know how to use cmd line, data structures and some basic knowledge of basics of algorithm. These resources take long time to master so take time and use them. If you use all of the resources in the chapter you will be more comfortable with your theoretical knowledge.

Nand to Tetris

Nand to Tetris

Computer is a complex beast. It is made up of lots of moving parts. Right from basic logic gates to complex games like Counterstrike. You may have learnt gates, operating system , compiler etc etc in pieces. But have you ever wondered how all these pieces are put together to create an actual computer? If yes, then you should take this course where you build a computer from scratch

Nand to Tetris

Nand to Tetris is a free course to build computer from first principles. The course comes with a book, software and source code to build things on your computer.Watch this video to get complete understanding of the course.

You will be starting to build computer from basic unit of any device, Nand gate. Moving forward you will build memory, processor and other hardware components.

Once hardware components are ready, then you will be starting to put together the software part of the equation like operating system, compiler and then designing a complete game on top of your own machine.

Why Nand to tetris

Most of us are good in one aspect of development aka programming. But we lack knowledge of complete system. Having complete knowledge of the system makes our understanding of individual pieces more sound.

Course contents

The following are the things you are going to learn

Hardware

- Logic gates
- ALU/CPU
- Memory
- Computer Architecture
- Machine code

Software

- Machine code
- Assembler
- Compiler
- Virtual machine
- Operating system
- Your own programming language
- Apps written in your language

If you see the topics, you may feel you already learn most of the topics. Yes you have!. The beauty of the course will teach how you can use all those topics put together to build a real machine.

Have Patience

It takes time to build a computer. The course teachers says its 12 week course. It will take time to understand each and every concept. So dedicate some time to the course every day and build the machine over time. At the end of course you will be clear of most of the concepts that you learnt in college.

Happy time building :)

Building your own

Building your own

Last chapter talks about building complete computer system. Though its a powerful enough course to teach you most of the pieces, you cannot use the software you built there in real world.

So in this section I am going to discuss about the softwares you can use for your daily usage. Here you are going to create few of the softwares that you use everyday like Operating system, compiler etc to get a feel how these things are built from scratch.

Why build when it already exist

Most of the students want to build new shiny things. It's easy to understand the urge to create something that no one has ever created. Before creating a new software it is equally important to understand things that we use every day. Building your own version of tools makes you understand these tools intimately .

Use what you create

People ask me how to improve their code quality. I say use what you create. It's the best way to improve. So the idea of creating our own language,compiler is not only for learning, instead it is to use those tools as your daily driver. That makes you understand how much hard it is to create a quality software and why you should make your code more readable.

Operating system

Operating system

Writing my own operating system was always my dream. Operating system is the first level of software that talks to computer hardware. So its like the most important piece of software running on your laptop.

We all have studied operating system as course in our curriculum. Though it introduced the concepts like memory management, disk management etc we hardly can relate that knowledge in reality.

It will be great if you can see or build the pieces of operating system as you learn the concepts. It will be more awesome if the operating system is powerful enough like real operating system like Linux.

Minix

Minix is an operating system created by Andrew S Tanenbaum , to teach operating system course. The lecturer created entire operating system just to teach his students about operating system. Since the operating system is inspired by Unix, most of the architecture and concepts are similar to Unix.

Minix and Linux

Linus tornvald, the creator of Linux, was one of the student for this course in university. He learnt the basics of operating system from this course and applied the same when he createed Linux operating system. So you will be learning from the book which Linus himself used to create Linux operating system

Book

The book Operating Systems Design and Implementation contains explanation on every topic of operating system with complete source code in C. It's a big book but if you have enough patience you can go through book with working examples in order to build your own operating system

So, if you want to learn how a real operating system is built, go through this book and practice the code. By the end of the book you should have thorough understanding of the operating system.

Language

Language

You may have studied different computer languages like C, C++ in curriculum. You have used IDE, compiler, assembler to compile the source code to the machine code and run the program. Have you ever wanted to create your own computer language? If yes, then here is your opportunity.

Why our own computer language

Computer languages are primarily the way we communicate with computer. Different languages have different characteristics. When we study them, initially they feel difficult to grasp. So we want to understand what it takes to create a good enough language from scratch. Creating our own language gives us the better understanding of existing computer languages and also adds powerful tool to our toolkit when we solve problems.

Learning from the masters

The book you are going to use here is created by masters of the field. The book is written by Brian W. Kernighan - creator of Unix, B language and Rob pike - one of the creator of Go language. Both have created the software that had endured many decades. So you are in good hands.

Book

Unix programming environment is the one you should refer. If you have read earlier chapters you know that the book is already refrenced in the Read section. But for learning to create langugage you have to concentrate on last chapter of the book name "Programming environment"

In this long chapter, authors show how to create a C like programming language in under 3000 lines of C code. It's shows with right approach creating a programming language is not a big deal.

So go through this chapter from this book, by end of the chapter you will know how to create your own language.

Conclusion

Conclusion

Creating is the best way to learn. If you want to understand a machine, best way is to build one. In this chapter we have discussed about different resources that will help you build your own computer, operating system , computer language.

To master all the content specified in this chapter will take years. So have patience and go through books and courses with peace.

Share

Share

In last few chapters you have consumed the resources shared by others. Now you are ready to share your own. You can share your knowledge through many ways. You can write about it, talk about it or even contribute in terms of the code.

In this chapter we are going to discuss about the different resources you can use to share your knowledge.

Blog

Blog

Writing blog is like keeping a dairy. Blog allows you to keep track of things that you do and share it with world. A good blog not only helps author, it also helps readers to improve their knowledge. Writing blogs seems to be easy in first sight, but when you do its not that easy.

Writing is hard

Writing is a different art compared to talking. Talking comes natural to humans but not writing. Writing is something we have cultivated as a culture. So when I ask some developer to talk about what they are doing, they can happily talk for hours. But if I ask developer to write down what they did, they will have hard time doing it. This is one of the main reason for not having documenation for most of software tools.

Writing needs a focused manner of expressing your understanding to fellow person. A good blog always wins its audience by its simplicity and precise explanation to the point. So writing not only keeps track of your work, but also teaches you how to express your understanding in a better way.

Writing as communication tool at workplace

In any job, writing is a primary communication skill. You have to write documents to express your ideas to your clients. You have to write mails to express your ideas to higher management. So if anything helps you to improve writing skills better you should start doing it. Blogging is one of the best way to do that.

What is a blog?

Blog is collection of articles shared by a person or collection of people on web. You have already read many blogs like Verge,Techcrunch in the read section. As you can read others blog, you can write your own too.

Why Blog?

Eventhough blogging seems to be a lot work which involves thinking, writing and publishing, following reasons should get you start blogging.

- Online dairy

  Blog is the best way to share your work with outside world. If you put your understanding of a given framework or a language out there in web, people find you to be authoritative on that subject. Not only it gives you more legitimacy, it also helps in job interviews. Blog is one the way people can understand your passion and skill set on a given subject.

- Improved communication

  As mentioned above writing is basic communication in companies. You have to communicate effectively in order to win over your fellow mates. So writing blog helps you immensely improve your communication.

- Share your point of view

  As we learn new things, we develop our own view of the world. Sometimes your view may be radically different from majority of others. Having your own blog allows you to share your point of view on things. Sometimes this may help others to look at things differently.

- Fun

Finally writing blog and sharing is a lot of fun. It may be challenging in the beginning, but as more people read your blog and appreciate your hardwork it will be a fulfilling exercise.

Different ways to create blog

There are many ways one can create blog. You can write and share blogs that are hosted by others. You can host your own too.

As a beginner, its better to get started with hosted blogs. Once you become comfortable, then you can host your own blog.

Hosted blogs

There are many companies which gives you free blogging software to create blogs. Some of them are

- **Wordpress**

Wordpress is one of the most famous blogging software out there. In wordpress.com you can create blog for free. Wordpress comes with nice tools for content creation and publishing. One of the most famous blog "Techcrunch" uses wordpress.

- **Tumblr**

Tumblr is another blogging platform where you can create your blogs. In Tumblr you can follow other blogs too. Tumblr is well known for its great UI, ease of use. Tumblr is also popular for creating photo blogs.

- **Blogger**

Blogger is a blogging platform by Google. If you have gmail id, its the fastest way to get started. Blogger integrates well with other google services like search, ads etc. I created my first blog on blogger.

There are many more other blogging sites, but above are the top three. You can try out those and settle with one you like.

Self hosting

You can host your blog yourself using using Github and Jekyll. For more info read through jekyll documentation

Having a blog is the best way to communicate with the world. What are you waiting for? Go create your blog today.

Knowledge Sharing

Knowledge sharing

Today people across the globe collaborate through internet. You can share your idea or queries in different forums and people from different parts of the world. In this chapter we are going to discuss about few of the resources you can use for sharing the knowledge with others.

Mailing lists

Every open source project normally comes with a mailing list. Mailing list is just a common email address that you can send mails or receive mails from the group of people.

Mailing list is a group mail. You can subscribe to this group mail. The following are few of things you can do with them

- You can ask questions about the project
- You can view other's questions
- You can answer to other's questions
- You can put forward new development ideas for the project

So if you have doubts on specific feature of a project, it's mailing list is the best place to ask. By doing so you will become more familiar with the project, you can also help other's by answering their questions.

Mailing lists are not just limited for Q&A. They are also used for proposing new changes and voting for the releases. So if you have a great idea to improve project, you can propose your idea in mailing list. If other members of the community are keen on the feature you propose, it will be accepted.

There will be different mailing lists for a given project. There will be dedicated mailing list for users, developers and committers of the project.

The following are few of mailing lists you can start with. Just google "project mailing list" to get mailing list of any specific project you are interested in.

- Linux kernel mailing list
- Apache Http server mailing list

Groups

Mailing lists are based on email. Though email is great for communication, it poses few of below challenges

- Too many emails from the project clutter the inbox
- It's hard to search for previous queries
- Difficult to follow a long running conversation
- Difficult to link to the conversation

To overcome these challenges, few open source projects use something called Group.

Group is a website where you can create/join a group for specific project or topic. Groups are almost same as mailing list with a difference of dedicated website which solves the above mentioned challenges. Few of examples for Group sites are Google groups, Yahoo Groups

The following are few of the interesting groups to start with. Again google will be your best friend to find more groups.

- Android Developer Group
- JavaScript Developer Group

IRC

Mailing lists and groups are great for passive communication. But what if you want to discuss about project like a chat. Then IRC channels are the right place to go.

IRC stands for Internet Relay Channel. It's a group chat service where people who are interested in a given project, can virtually hangout and communicate.

One of the advantage of IRC over mailing lists and groups is, it doesn't need any login. You can choose any pseudonym and use the irc channels.

If you want to know more about irc and how to use it refer this , an excellent guide.

The following

- Android IRC channel
- Google Chrome IRC

Stackoverflow

All above mentioned tools are generic communication tools which are specific to a project. But if you are looking for Q&A website which covers all different kinds of project, then stackoverflow [http://www.stackoverflow.com] is your friend.

You may already know stackoverflow. Whenever you search for specific programming question on google you may have ended up in stackoverflow. But what you may not know is that each project has dedicated stack overflow section where you can get answer to your question faster than Google.

In stackoverflow, you can ask a question or answer for a question also. So from now start searching in stackoverflow.

Social networks

You may have used different social network like Facebook,Orkut to share photos and be in touch with friends. But there are other social networks out there where you can follow the rock stars of the programming. Twitter and Google plus are two of such social networks. Using these networks you can get in touch with the people whose work you admire.

The following are few of the people you can start with.

- Linus torvalds - creator of Linux
- John Resig - creator of Jquery

So now you know different avenues to ask questions and answers other questions. Be on top of those resources. They will be extremely helpful to share ideas and queries.

Meetup

Meetup

Doing everything with interenet is great. That allows you to work from anywhere you want. But sometimes its not enough. Sometimes it's better to meet real people to understand things in deep manner. So that's where meetups come along.

Meetup is a non formal event where like minded people meet to discuss on a specific theme. Meetups have become quite famous these days as they allow people to share their experience on specific tools or technology with their fellow developers.

If you want to improve your skill set, attending meetup is one of the best way to do it. Few of the things you can do at meetup are

Listen to the expert

Meetup's are normally the places where people talk about their experience with tool in real world. This information is extremely important for a developer, as they teach you do's and dont's. Once you are physically present in meetup, you can ask any doubts directly to the speaker. This allows you to clarify the doubts on spot.

Give a talk

Want to share your experience and get feedback from the community? Meetups are great place to do that. You can share your experiences through a talk and when people ask questions they will challenge your understanding. Giving a talk not only allows you to share your idea, but also makes you visible in the community.

Network with community

Networking with fellow developers in the community is extremely important. This allows you to share ideas which will enrich you as a developer.

Now you understand the importance of meetup. You may be asking yourself, how I know about the meetups and join them.

The following are few websites you can use

- **Meetup.com**

  Meetup.com is one of the global meetup site. It allows you to create,follow and subscribe to the meetups across the world. You can search meetups specific to your interest,location and time. Head over to meetup.com and start rolling.

- **Eventbrite**

  Eventbrite is another website which hosts wide variety of different meetup events. You can search and subscribe any event you like to attend.

Meetups are great way to involve in the community. So start meeting your community pals and you will benefit a lot from it.

Conclusion

Conclusion

Sharing always improves your knowledge. Sharing is also one of the best way to get noticed in the community. So build a habit of writing a blog quite often as possible. Hangout in irc,mailing list and meetups to understand what other people are trying to do. Being in the community is the best way to grow.

Build

Build

The last four sections were concentrated on learning things from different sources. We have also discussed about how to learn by building things for ourselves. But isn't it fun to develop something useful for others too?. In this section we are going to discuss about different avenues and opportunities which you can harness to build something useful for others.

Hackathon

Hackathon

hackathons are the marathons to hack. Hack stands for building something useful or cool over given short period of time. hackathons are great fun as they challenge you to have continuous focus over long period of time. There will be added fun being up for whole night and hacking on things you like.

What is a Hackathon?

hackathon is an event where many developers get together to build something interesting in a short period of time like 24 hours. These event goes on for 24 hours without any break , many times stretching overnight. In this event team of people will be expected to build something around a given theme/topic. Most of the these events are free to attend and some even have perks like prizes or hiring for best hacks.

Benefits of hackathon

Hackathons are great for prototyping

As you have limited timeline, hackathons are not meant for full fledged product/library development. It can be a great place to try out the ideas that you are interested in and prototype them fastly. This gives a great benefit of having feedback in very early stage of idea. You can build a prototype in 24 hours and ask people to give feedback.

Hackathons are great for focusing

The time in hackathon is very valuable, as you will be completely focused on the thing that you have to build. You will not be distracted by anything outside of the one you are building. It's great to have this kind of focus because often lot of nice products are the outcome out of such hackathons.

Hackathons teach team work

Normally all hackathons encourage people to work in teams. One obvious reason is to save time. In adiition, being in team forces you to be a team player.Each team member has a responsibility to contribute equally. If you don't have a team don't worry many hackathons even allow you to form team on spot.

Hackathons brings you recognition

At the end of hackathons, there is usually a series of demonstrations in which each group presents their results.It may not be the greatest thing in the world, but even if it's little good people will appreciate. It builds a good recognition around developer community which will be invaluable for your career.

Hackathons may have perks

Some hackathons may give you some perks like latest gadgets or some other coupons etc. Some may even give you job offers!. So participation in these events opens a great opportunity to win these perks.

Now,as you know the importance of hackathons, you may be interested in knowing different places where hackathons are conducted.

The following are few of resources you can checkout to get to know about hackathons in India.

- HasGeek
- Eventbrite hackathons
- Inmobi hackathon
- Hackathon watch
- Startup digest

Hackathons are the best place to build useful things in a short peroid of time. Attend whenever you get an opportunity.

Google summer of Code

Google summer of Code

Google summer of code is a program conducted by Google for encouraging students to contribute towards open source. This section is only applicable to students. If you are interested to know , how to contribute towards open source in general refer to next chapter.

- **What is Google summer of Code?**

Google Summer of Code is a global program offering, post-secondary student developer aging 18 and older, stipends to write code for various open source software projects. Google works with many open source, free software and technology-related groups to identify projects and funds projects over a three month period. Under this program, you can contribute to your favorite open source project in your free time.

- **Why Google is doing it?**

Google is built on top of open source projects. So google wants to contribute to the community. Google also wants to encourage students to contribute towards open source from very early age.

- **Eligibility to participate it?**

You should be a student in a recognizable college. You should be older than 18.

- **Choosing a project**

There are more than 50 open source projects to participate in GSC every year. You can go through these projects, talk to concerned mentors and decide which project you want to contribute.

There will be some criteria to get selected. But most of the time its your passion and time which matters. If you have more questions about the program, go through official GSC FAQ

You may be wondering why you have to spend 3 months of time just contributing to the open source.

The following are few of benefits of Google summer of code

- **Great mentorship**

You will be mentored by the best people from community. It's a great opportunity to work under someone who is 10+ years experienced. Most of the time you will be working with the people who have contributed a lot to the project. This not only makes you a better programmer, but you also get a thorough understanding of software development.

- **Chance to contribute to your favorite project**

You may have favorite open source project that you often use. Isn't it great if you can contribute and make a mark in the project?. So if you want to improve your favorite project, GSC is the best chance to do.

- **Money**

If you put enough work, google will pay you handsomely for all the hardwork you put in. So you will be not spending all your time without making any money.

- **Fun**

Working on open source projects is great fun and GSC gives you opportunity of having fun and making money too!!.

GSC is the best program for a student to make a mark in open source world. Grab this opportunity in both hands and contribute to open source.

Contribute to Open source

Contribute to Open source

In the read section, we discussed about why it's so important to read other's code. In last chapter, we discussed how you can contribute to open source through google summer of code . But GSC is available only for students. But what if you are not a student and want to contribute to open source in free time?

This chapter we talk about how you can contribute to open source through various ways.

Choosing a project

Most of the people who talk to me, ask how to choose a project to start contributing. I normally answer "whichever you use the most". It is always good to choose a project which you normally use in your work. It may be a Javascript library, Java library or a framework. The reason being that you are highly familiar with the project,so it will be easy for you to start thinking on how you can improve the project which in turn improves your work.

Code is not only way to contribute

There is a misconception within people that contributing to open source means just contributing code. Though contributing code may be great, but it is not the only way to contribute to open source. There are many ways you can contribute besides coding!

The following are few ways you can contribute to the project:

- **Answer in mailing lists**

As you become more familiar with the project, you will be more experienced in using the tool or library. In mailing lists, lot of people will be in the need of that expertise. So start with answering questions in the mailing list. This improves people's understanding on the project and in turn more people will be willing to use.

So answering in the mailing lists will be one of the good place to start with.

- **File bugs**

Every software written by humans are buggy in nature. So if you come across some bug in the library don't keep quiet. Raise a bug in appropriate bug database so that developers of the project knows about it. By filing the bug you are making project better.

- **Improve the documentation**

Not every project comes with great documentation. After using it for sometime you may feel the documentation is not adequate or it's out of date.Documentation is another way to contribute to open source projects.On improving the documentation, you are helping people in understanding the project .

- **Write test cases**

Test cases are the best place to start contributing the code to a project. Writing test cases improves your understanding of the project and also it will help developers of the project to find out the corner cases that they might have not considered.

- **Fix bugs**

After getting experienced on the project, you are sure to know a lot about how the given library works and file few bugs yourself. Now if you think you can fix the bugs go ahead and fix it.

- **Contribute new features**

This is the ultimate contribution you want to do. Putting a new feature in a well established project is not easy. But after going through all the above contributions you should be able to do it.

There are many ways you can contribute to your favorite project. Go ahead and make a mark.

Conclusion

Conclusion

Building things gives immense joy. The Art of Building allows you to use your knowledge in a useful manner and it allows you to execute your ideas. So don't just concentrate on acquiring knowledge,put equal effort in building things too!.So Participate in hackathons and make a mark in the open source.

Profession

Profession

In last five sections we talked extensively about various ways by which you can improve your knowledge and contribute to the world. This section we are concentrating on how you can take up software development as profession.

In this section, I have discussed about various opportunities, which software developer can pursue. These opportunities are specific to India, as I am more familiar with the place. You may find similar opportunities in your country too.

There are many criteria people use to choose a job. There is no single way to say one company is better than other company. So use this section information as a guidance rather than rule.

This section is written with aim of introducing readers to the different opportunities. I won't endorse any specific company or type of industry. You go through the opportunities and make up your own mind.

Startups

Startups

Startups are small companies which are normally newly started and will have very limited number of people. In India, there is a new trend of entrepreneurship where people want to create new companies and want to build there ideas into product.

As a software developer, you can work for these newly started companies rather than working for a well established companies. In this chapter we are going to look at pro's and con's of working for a startup company.

Pro's in working for a startup

- **Lot's of learning**

Working in a startup makes you learn lot of things. As startup will have very limited resources you have to do lot of things which you may not do in a big company. So you will be learning new things almost every day which will contribute immensely to your career.

- **Full stack development**

Startups will have only few developers. They cannot hire separate people for backend and frontend. So if you work for a startup you end up working in all different levels of software development stack. You will be doing backend server coding, database and front web and mobile development simultaneously. Surely it will be a challenging task, but if you are really passionate about work it will be really rewarding experience.

- **Entrepreneurial experience**

In a startup you will not do just development, as company grows you will be given more and more responsibilities like managing people,interacting with customers and going to on site etc. So being in startup teaches you how you can build a company from scratch. If you have entrepreneur fire inside you, working in a startup is best choice.

- **Working with great people**

Normally startup attracts best people. If you look into silicon valley, every great company was started as a startup and many of these founders will have a great product ideas which are built in these startups. So if you work for startup you will get a chance to work with them. It's a very good experience as you can learn lot of things from them in very short time.

- **High career growth**

As you learn in very fast pace, you will be improving your capabilities in very short time. You will be more confident on things than your peers with same experience. It gives you edge over others in job market. You can easily go from startup to bigger company but reverse is not so easy.

Now you know pro's of working for a startup. But you also have to be aware of con's. The following are few con's of working in a startup.

Con's of working for a startup

- **Economic instability**

Startups in their initial phases are funded by their founders. They may not have deep pockets like big companies do. So if company doesn't do good in the longer run they may run out of money and they may have to close it. Though it's a worst case scenario , but according to the numbers only 20% of startups survive more that two years. So before getting into a startup make sure that you are informed about economic status of the company.

- **Not all startups are created equal**

You have to be very careful when you choose a startup. Not every startup created with same passion and interest. There are many reasons for a person to start a company. So before you signup for any job, make a point of talking to it's founders and understand what's the rationale behind having a new company. Unless you are convinced about their idea and passion it's not good to work for them.

- **Startup takes all of your time**

If you work for a startup, it becomes your life. You will not really have much time for your personal life. It's manageable in the beginning of your career as you may not have many commitments. But as you become more experienced it may become challenging. So join startup only if you can dedicate lot of time. It's not a typical 9-4 job.

Resources to find a startup job

Now you understand the pro's and con's of the startup life. After knowing all these, you may ask how you can look for startup jobs. There are many resources you can use to get into a startup

- **Friends**

Most of the startups encourage recruiting through referrals. Normally good candidates try to bring more good candidates with whom they want to work with. So find out any of your friends working in the startup. If you like their startup, ask them to refer you. It's one best way to find startup jobs.

- **Job boards**

There are many job boards dedicated for startup recruitment. You can go through these job boards to find out which are the startup jobs available in your city. The following are few job boards which you can refer

- Angel List
- Github jobs
- HasJob

Working for startup is a rewarding experience with a little bit of risk. Go through more about the startups online and make up your own mind.

MNC

MNC Companies

MNC stands for Multi national companies. India is a country where almost every software company has it's branch. The tech giants like Google,Microsoft, Facebook, Amazon etc have offices here in India. As a developer you can work for these big mnc companies also.

This chapter discusses about pro's and con's of working for a MNC.

Pro's of working for a MNC

- **Economically more rewarding and stable.**

Most of the MNC companies will have deep pockets. Most of them will pay you well compared to the Indian counter parts. They will be more stable compared to the startups. In MNC, you don't have to worry about the paycheck most of the time.

- **Chance to work in global scale**

Many MNC companies employ people from different countries in a given team. This gives you the opportunity to work with different kind of people and learn from their expertise. Sometimes you may be even able to meet your hero's from other countries.

- **3 You contribute to products used by millions**

Most of the product/services offered by an MNC is consumed by millions of users. So whenever you are working in MNC, whatever you contribute to the product it's going to effect life of millions of people. So it will be very rewarding experience when you know even the smallest of your contribution has a huge impact on others.

The above are the few of pro's. But MNC companies also comes with few con's too.

Con's of working for a MNC

- **Many MNC's only do support in India**

Many MNC companies have their product team sitting in US/UK whereas support teams sitting in India. So even though you are part of big well known MNC, your work may not be that great. So don't just go by name. Make sure that you understand the job profile before you get in.

- **Limited growth in technology**

When you work for a MNC, you will be not working on full stack. Your work will be more focused on specific piece in the complete stack. It will be great to be expert in that piece, but if you prefer to learn the end to end working of a product, MNC may not be a right place for you.

MNC's are great for global collaboration. But choose carefully. Don't go with the just the name. Go with proper job profile.

Service Based Companies

Service based companies

Service based companies are companies offering their resources as their product. These resources can be man power, intellectual property or development skills. In India, Infosys, Wipro,TCS comes under the service based companies.

I think these companies are the one you may be most familiar with. In this chapter we are going to discuss about pro's and con's of working for a service based company.

Pro's of working for a Service based company

- **Easy to get in as fresher**

Most of the Service based companies recruit in masses. As many of these companies also do outsourcing they need as many people as possible. So these companies comes to college campusses and recruit in 100's. As a fresher you may not need much experience to crack in these companies.

- **Work life balance**

Most of these companies will not make you work as hard as startups. So you can balance both your personal and professional life in a way which does not overwhelm you.

- **On site opportunities**

Most of the service based companies work for companies across the globe. So there will be always chances that employees have to travel and be in client place. So if you work for these companies you will get chance to travel different places at the cost of company.

Service based companies are attractive to a fresher. But working for these companies have few con's in longer run.

Con's of working for a Service based company

- **Limited growth**

You rarely will be working on full stack or cutting edge technologies in these companies. Most of the time for most of work you does not need intensive learning. So in longer run you may not be exposed to many technologies and product stack which may hinder your growth.

- **Shifting to a startup and product based company is very hard**

As you become more experienced, you may want to switch to other kind of companies like startups. But in MNCs and startups there is stigma against people from service based companies. As most of the people will have limited knowledge in companies, their experience may not mean much to other section of companies. So be careful how you spend time in these companies.

- **Limited focus on individuals**

These companies normally have lakhs of people working for them at a time. So they may not put much effort in making an individual to grow in more manner. Also as most of the work may not need high level skills, may be these skills are not respected much.

Service based companies are easy to get in for a fresher. But in longer run they may become hindrance for your career. So before getting into these companies explore other opportunities also.

Conclusion

Conclusion

Choosing a job is not easy. Choosing a right company to work for is hard. There is no thumb rule where you should work. It's highly subjective topic. But this chapter has given overview of all different opportunities you have as a software developer. Be contentious of all these opportunities and make a good decision.

Table of Contents