Applause from Michelé Clarke, Julien Missial, and 5 others

Adrian D. Finlay
@thewiprogrammer. Writer @hackernoon. Code, LOTS of it. Mangos, LOVE THEM! Barbering. Health. Travel. Business. & more! Network w/ me @ adriandavid.me/network
Oct 1, 2017 · 14 min read

How To Become a Java Web Developer (2017)—Part III: Frameworks, Real World Tools, Better Code & Beyond

Java Banner [10]

Enterprise Framework—The Engine of Contemporary Development

> *"While Java EE does a great job of standardizing the enterprise infrastructure, providing an application model, and providing components adequate to develop web applications, two major problems are associated with it."*

> *- Interacting directly with the Java EE Components often results in massive boilerplate code and even code redundancy.*

> *- Creating an Enterprise Application using the Java EE infrastructure is a nontrivial task that requires a great deal of expertise.*

> *Frameworks address these two major problems (among others)." [40]*

The Spring Framework, much like Hibernate, is unique in this area not only due to it's commanding mindshare among Java developers as an Enterprise Solution, but also because much like Java EE it is a suite of various different tools for building Enterprise Solutions. It is not merely an MVC Framework. Spring MVC is only one aspect of the behemoth that is the Spring Framework.

An Enterprise Framework is a collection of APIs and tools used to develop Enterprise Software. While the term "Enterprise" is a finicky one in that it is not very well defined, when contrasted to Web Development it usually is used to indicate that it is software beyond merely a web framework. Enterprise Frameworks include solutions for networking, persistence, web services, and much, much more.

Spring Framework is unique in that it is one of the only frameworks that are as broad and comprehensive in the amount of APIs and tools available. There are other Frameworks, however, such as the ZK Enterprise Framework, JBoss Seam (now deprecated), and a combination of various Apache Enterprise tools, but in my estimation, **Spring & JEE are the most cumulative Enterprise Solutions**.

OK, I lied. The Spring Framework is actually an application framework and inversion of control container for the Java platform. It can be used in several different applications. It merely contains components for building Enterprise Applications atop the JEE (mostly the Servlet API). Those components "happen" to be extremely popular for building Enterprise Applications.

It is beyond the scope of this article to discuss the Spring Framework in it's totality. You may never even use the Spring Framework, but I'm willing to bet that you will certainly encounter it throughout the course of a Java Enterprise Career. Because it's influence and widespread use in Java Enterprise Development, it must be discussed. The specific components that you may need to create Java Enterprise Applications will of course depend on need. **Much like the JEE, there are components for several different things**. Often enough, you will find JEE development done in tandem with the Spring Framework. The Current Stable version of the Spring Framework is Spring 4.3.10 released in mid July 2017. Spring 5 is expected to be released any day now, in September 2017; It is Java 9 compliant. You can see what's new in Spring 5, here.

You will want to take learning Spring Framework very seriously, as this is the predominant Enterprise Framework of choice for many Java developers.

There are various other pieces of Java Enterprise Software. For more, see here.

Where to learn, Recommended Resources:

Tutorial's Point Spring Framework 4.1.6 Tutorial

Pivotal's Official Spring Documentation, Tutorials, & Guides
You will visit this web-page. Trust me. Feel free to bookmark it ahead of time.

Free Spring Framework Tutorials—Spring Framework Guru (John Thompson)

Spring & Hibernate for Beginners, Chad Darby—Updated 08/2017
The most popular course on uDemy on the topic of Spring/Hibernate, Chad's course is relied upon by thousands. This is how I started.

Learning Spring 5.0

Mastering Spring 5.0

Pro Spring 5.0 : An In-Depth Guide to the Spring Framework and Its Tools, 5th Ed.

Building Web Apps with Spring 5 and Angular

What's New in Spring Framework 5, John Thompson (DZone/Java Zone)

Web Frameworks—Building Web Applications

Because we have already discussed the Java EE Web Tier (Servlets, JSP/JSTL/EL, JSF) as well as Enterprise Frameworks (Spring, Apache), this portion will be relatively brief.

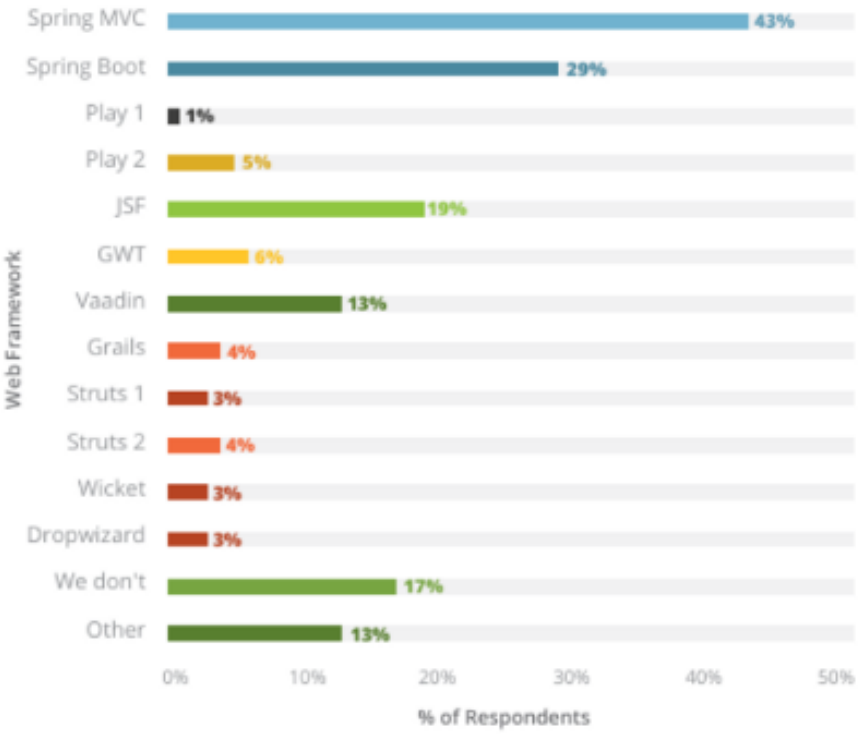
A **Web Framework** is an Application Framework that is designed to ease in the development of Web Applications. They are a common staple of Enterprise Development.

According to Rebel Lab's latest data (Sep 19 2017), Spring-Technology based Web Framework (Spring MVC, Spring Boot, Grails) has 47.81% of the popularity index among developers with JSF a distant second at 15.19% [41].

Rank	Framework	Popularity
1	Spring mvc	29.39
2	JSF	15.19
3	Spring Boot	11.69
4	GWT	7.6
5	Grails	6.73
6	Struts	7.47
7	Play framework	4.16
8	Seam	1.88
9	jax-rs	3.1
10	Vaadin	2.45
11	Wicket	1.92
12	Tapestry	1.83
13	JHipster	0.73
14	Dropwizard	1.05
15	Sparkjava	0.76
16	Lagom	0.71
17	Vert.x	0.72
18	Ratpack	0.15
19	Rapidoid	0

According to Rebel Lab's 2016 Java Tools & Technologies Landscape Report [27], 76% of developers used at least one Spring-Technology based Web Framework (Spring MVC, Spring Boot, Grails), with stock JSF again holding a distant second with 19%. It is important to note

that participants of this survey could choose more than one web framework. This makes sense as the use of multiple frameworks is common.



John Thompson claims that some estimates report that Spring is used in over 60% of Java Web Based Applications [5]. In my opinion, the unalienable truth is that Spring technology has made an indelible mark on contemporary Java Enterprise Development.

> *What about the Jobs?*

At the time of posting this article,

An Indeed US search for “Java Spring” yields 11,133 results
An Indeed US search for “Java Play” yields 4,067 results
An Indeed US search for “Java Struts” yields 1,847 results
An Indeed US search for “JSF” yields 1,035 results

This should not be taken as proof, but on the surface, it does appear that Spring is in much more demand than any other web framework.

> *Which Framework is the best?*

Are apples better than oranges? I will leave it to the user to decided which Framework **best meets their needs.** Many feel that the Spring MVC Web Framework is the golden child of Java Enterprise Development and following that guise, is the *de facto* best web framework available. The mere asking of this question presumes a defined notion of what qualifies something for being "the best". I am generally dismissive of these kinds of conversations as I generally do not believe there is a best language, tool, or framework for all tasks in a specific domain. I generally believe that some tools do some things better than others and that the quality of a tool is relative to need as well as both the community and engineering team behind the development of the tool. Try a framework or three. Let me know which you like best, and why, in the comments.

Where to learn, Recommended Resources:

For Spring, refer to the links in the Enterprise Frameworks segment above.

Play 2 Official Documentation & Tutorial

Tutorials Point Struts 2 Tutorial

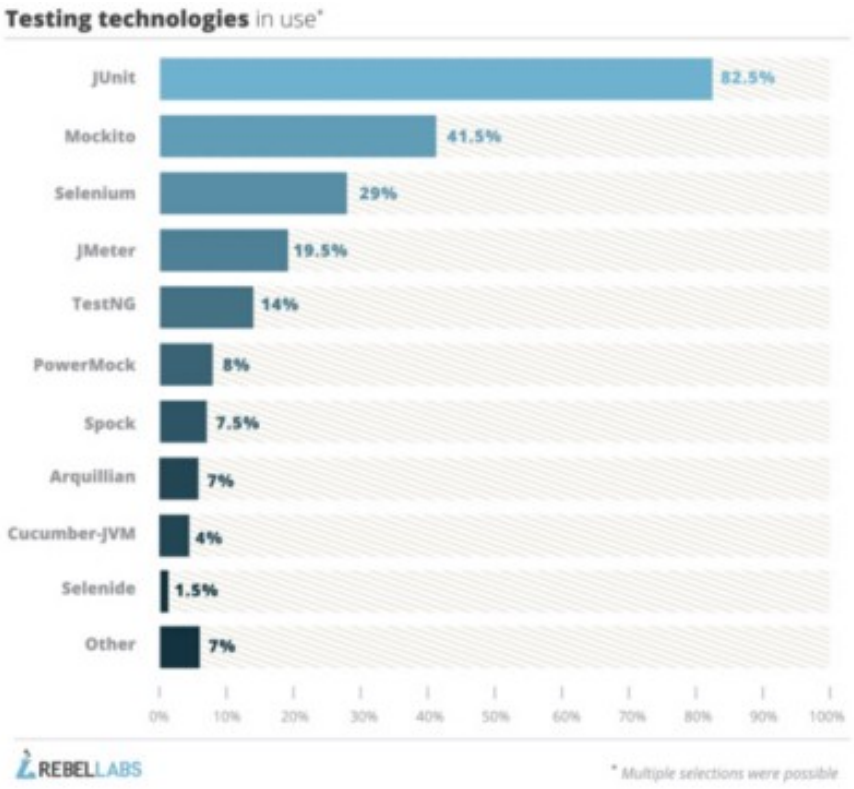
Unit Testing—Does it work the way we say it should?

While there is more to testing than unit testing, I will leave the task of discussing Testing to another author. Techopedia puts it best:

"A unit test is a quality measurement and evaluation procedure applied in most enterprise software development activities. Generally, a unit test evaluates how software code complies with the overall objective of the software/application/program and how its fitness affects other smaller units."[29]

Unit Testing is important because it provides us away to examine if individual application components are meeting end-user requirements. Unit Testing is useful in building components for large systems where other components rely on the confidence that the components it depends on work the way they are supposed to. Some community discussion on the topic is available here.

Of the Unit Testing Frameworks, JUnit holds a commanding market share, with 82.5% of Java developers making use of the technology, according to Rebel Labs [23]. While JUnit may not be all for your Applications needs, it is a very valuable tool and as such it is a very familiar among Java Enterprise developers.



Where to learn, Recommended Resources:

JUnit Tutorial for Beginners—Learn Java Unit Testing (Ranga Karanam, in28Minutes on uDemy)—This short video tutorial will equip you with the basics to start using unit testing in your Java Applications.

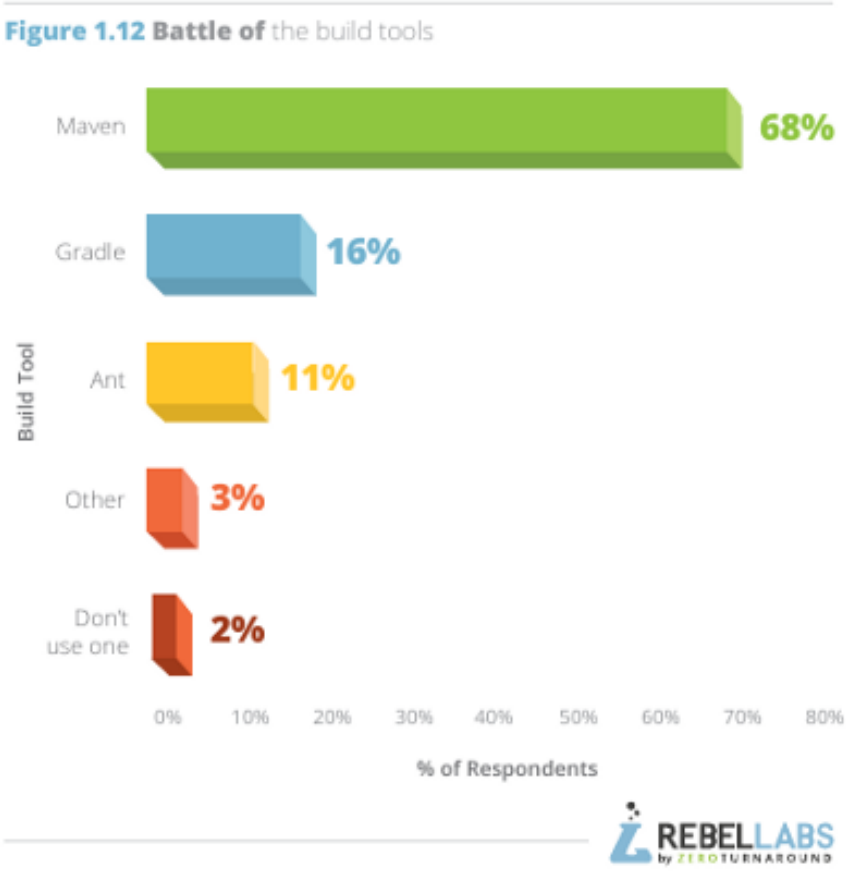
JUnit 5 User Guide—Official Documentation & Tutorial

It is also important to understand how to perform **assertions** using the **assert** keyword in the Java programming language.

Build Tools—Automating The Build Process

Build tools are automation tools used to manage the dependencies of software, describe how software is to be built, and automate the process of compiling,linking, & producing an executable. They are a standard staple of enterprise projects, which will almost always contain numerous dependencies that will need to be managed over the evolution of the software.

According to Rebel Labs [27], Maven is still holding a pseudo-monopoloy in Java mind-share with ~7/10 developers opting for the tool. Gradle notched a distant second, with ~ 1/6th of mind-share.



Where to learn, Recommended Resources:

Tutorial's Point on Apache Maven

Apache Maven Official Documentation

Spring: Building Java Projects with Gradle

Grade Build Tool Official Documentation

Tutorial's Point Gradle Tutorial

Version Control—Collaborative Development

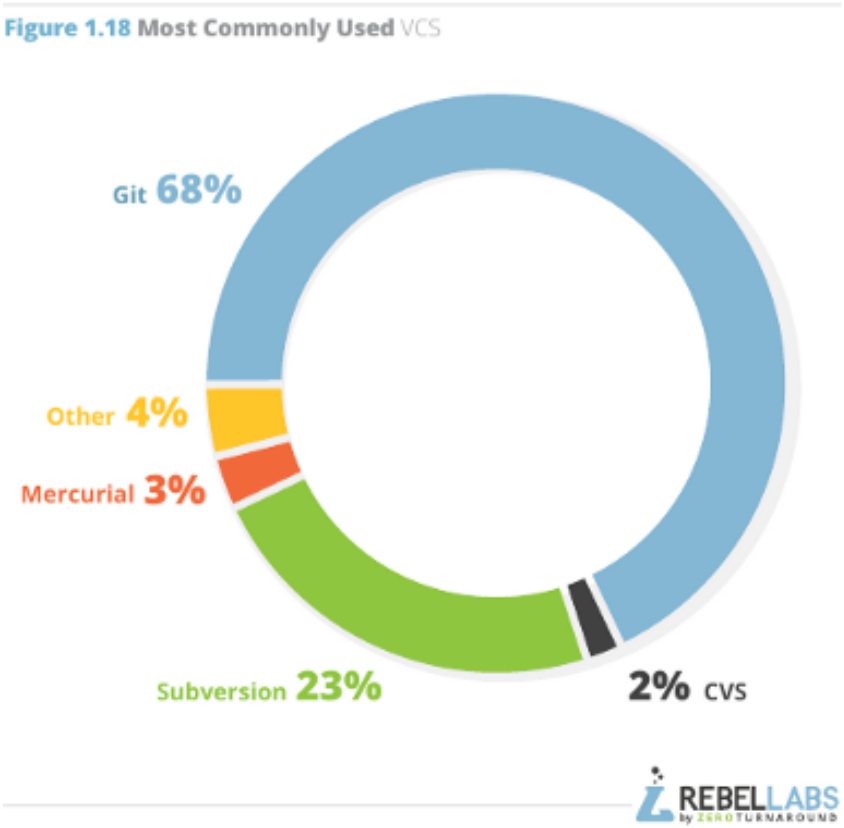
> *"Alone we can do so little; together we can do so much" — Helen Keller*

Version Control is important because it is the means by which managing versions of software in the development process takes place. This is relevant to real-world Enterprise Development where **several people work on various components of a single application**, often in teams. Whether you like it or not, real-world development will require you to work in teams, or at least, to collaborate with other people.

There are many benefits of version control:

- A Complete History of the Application and it's changes
- Ability to collaborate
- Ability to merge changes
- Ability to maintain different versions of the same application
- Branching
- Much more….

Git seems to have a commanding lead with 68% market share, while Subversion seems to hold a distant yet sizable second place spot with 23% [27].



According to the Stack OverFlow Developer Survey 2017 [35], Git had a 69.2% & Subversion had a 9% mind share among developers for choice in version control.

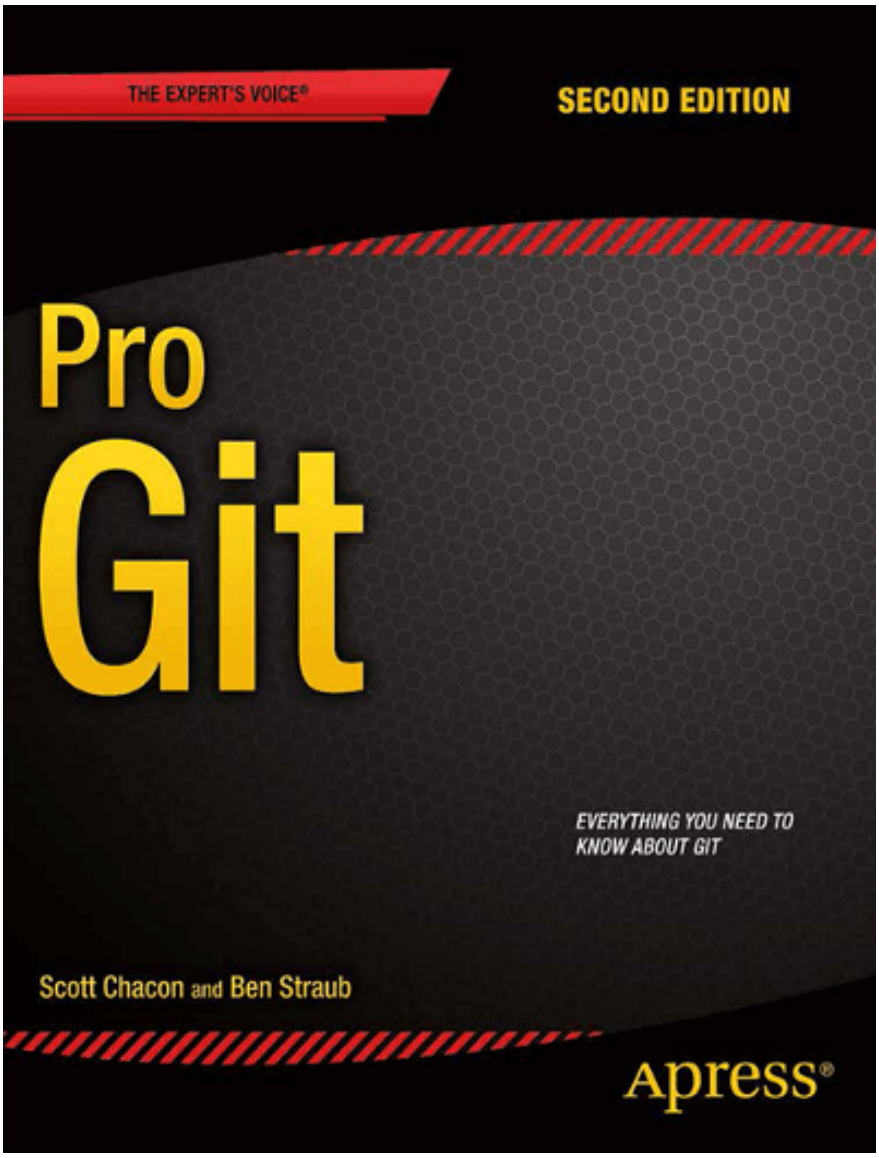
A Final Note: You should also be aware of version control repositories such as **GitHub** or **BitBucket**. These two repositories are among the most common used in development, with the former having similar levels of market share as it's companion, **Git**.

Where to learn, Recommended Resources:

TutorialsPoint—Git Tutorial

Git Official Documentation

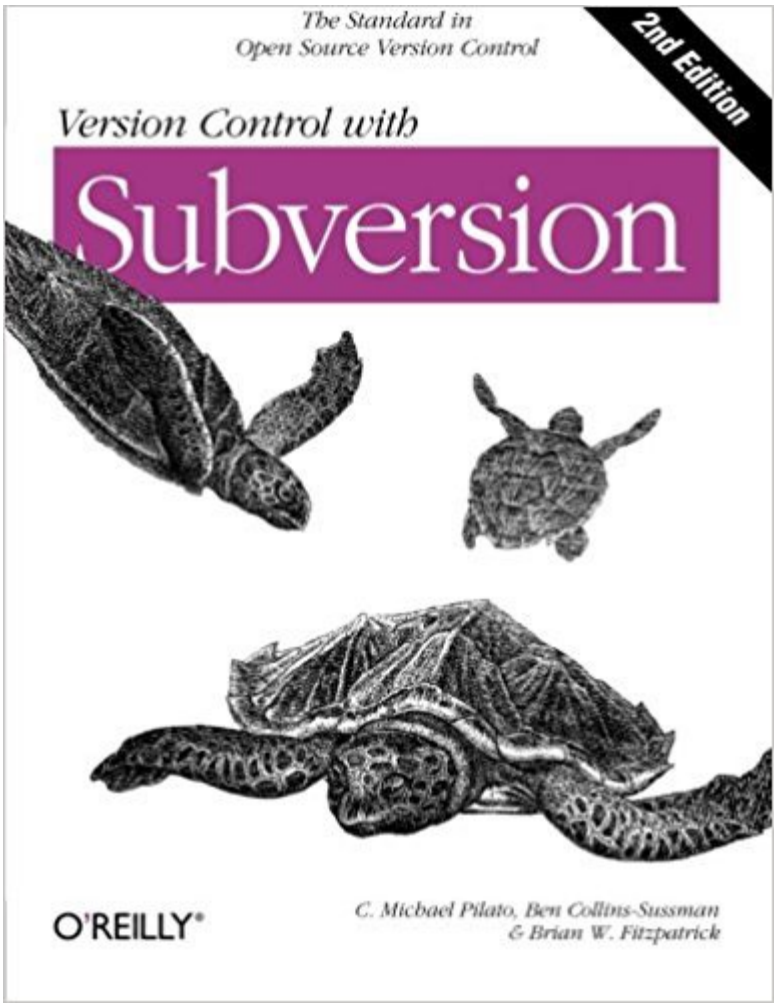
Pro Git (Free Book)



GitHub's Official List of Tutorials

TutorialsPoint—Subversion Tutorial

Version Control with Subversion (Free Book)



Debugging—If the bug's big enough, it's a feature

> *" The wages of sin is debugging."—Ron Jeffries*

> *"If debugging is the process of removing software bugs, then programming must be the process of putting them in."—Edsger Dijkstra*

> *"The most effective debugging tool is still careful thought, coupled with judiciously placed print statements."—Brian Kernighan*

Debugging is the process of removing bugs, otherwise known as errors in software, from an application. There are many tools and techniques involved in debugging, enough to fill many books. The objective of this

portion is to point you in the right direction regarding debugging in Java.

For more, see here.

Enterprise Developers should be aware of how to use the **JDB** / JPDA, best practices, as well as how to use **breakpoints** and **IDE tools** to perform debugging.

Where to learn, Recommended Resources:

Java Platform Debugger Architecture (JPDA) SE 8

Tutorials Point—Java Debugger Tool

Java Platform Debugger

IntelliJ IDEA Java Debugging Tutorial

Code Quality—Practice Makes Perfect

> *" Quality is not an act, it is a habit."—Aristotle*

Poorly written code can be devastating to an organization in terms of the cost (both in time and money) of managing the consequences of said code. Obscurely written code can lead to developer inefficiency which leads to more time spent trying to understand and/or restructure the code which costs the organization money and which ultimately hampers progress.

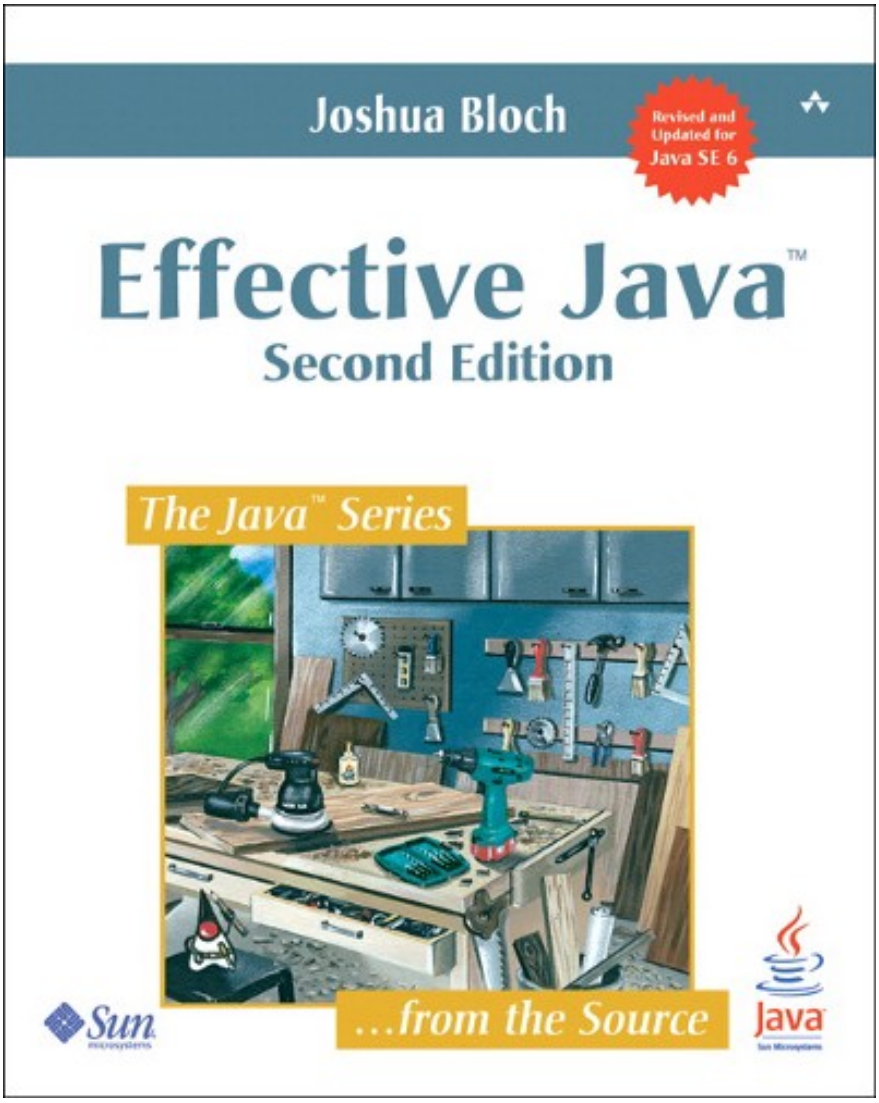
Badly written code can lead to performance penalties which may (depending on the nature of the organization's aims) lead to less adoption of their product/service which could lead to the organization loosing business or ceasing to operate.

Badly written code can lead to security vulnerabilities which may lead to the breach of sensitive information which will assuredly lead to demise and utter destruction.

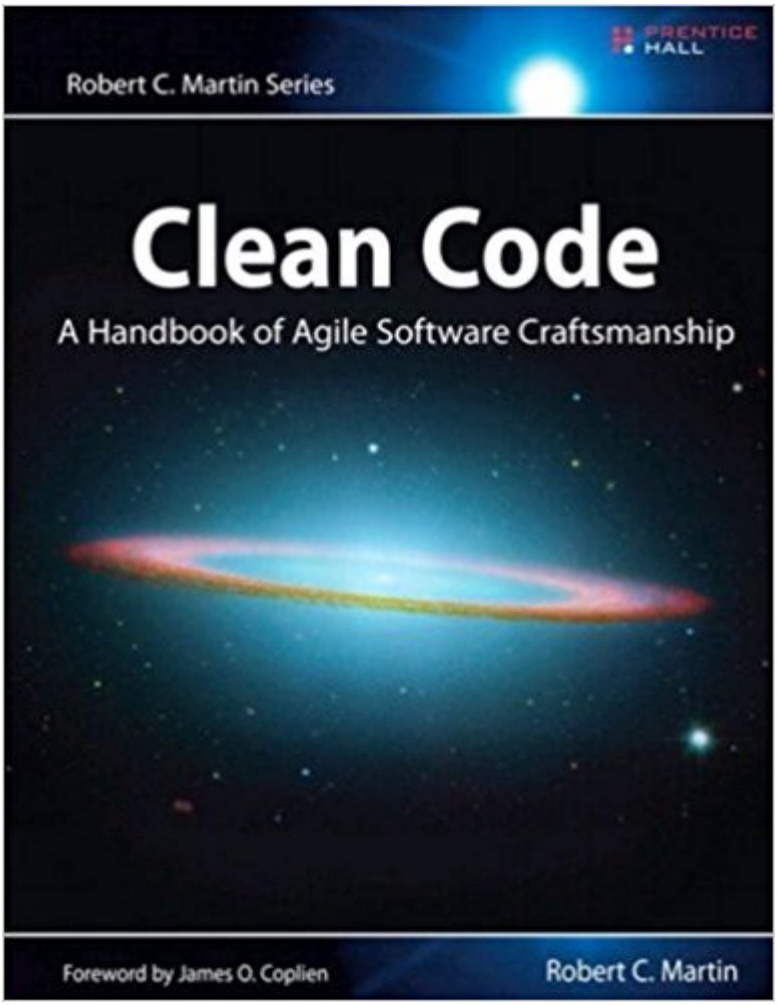
Get the point? It's not enough to know the elements of a programming language—one needs to know how to use the tools the language provides in a way that is safe, effective, & efficient.

With that in the mind, I recommend the following books for writing quality code:

Effective Java, 3rd Ed.—Joshua Bloch (**Release:** December 29, 2017)



Clean Code



Another good book that is often recommended is Code Complete by Robert Martin. The reader should be cautioned, however, that it is a lengthy read. At 960 pages it is much longer than Clean Code, about an additional ~500 pages by comparison.

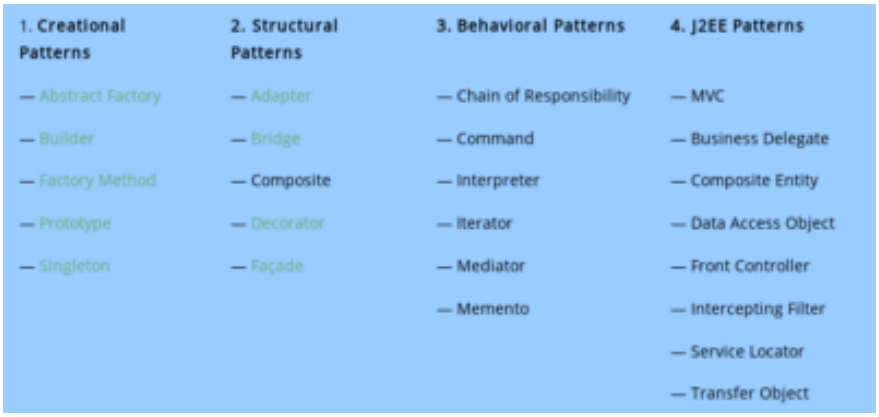
Design Patterns and Best Practices in Java 9,Adrian Ianculescu (**Dec 6, 2018**)

Design Patterns—Templates for Solving Common Problems

Design Patterns are useful because they save time and (arguably) effort. They are the result of observing a general, repeatable means by which to solve commonly occurring problems. **Enterprise Design Patterns** save time by providing a template for common Enterprise Tasks.

In 1994, Erich Gamma, John Vlissides, Ralph Johnson, and Richard Helm, cordially referred to as the ***Gang Of Four***, wrote the groundbreaking book **Design Patterns: Elements of Reusable Object-Oriented Software**. Though the text was written with C++ in mind, the concepts may be applied to many Object Oriented languages.

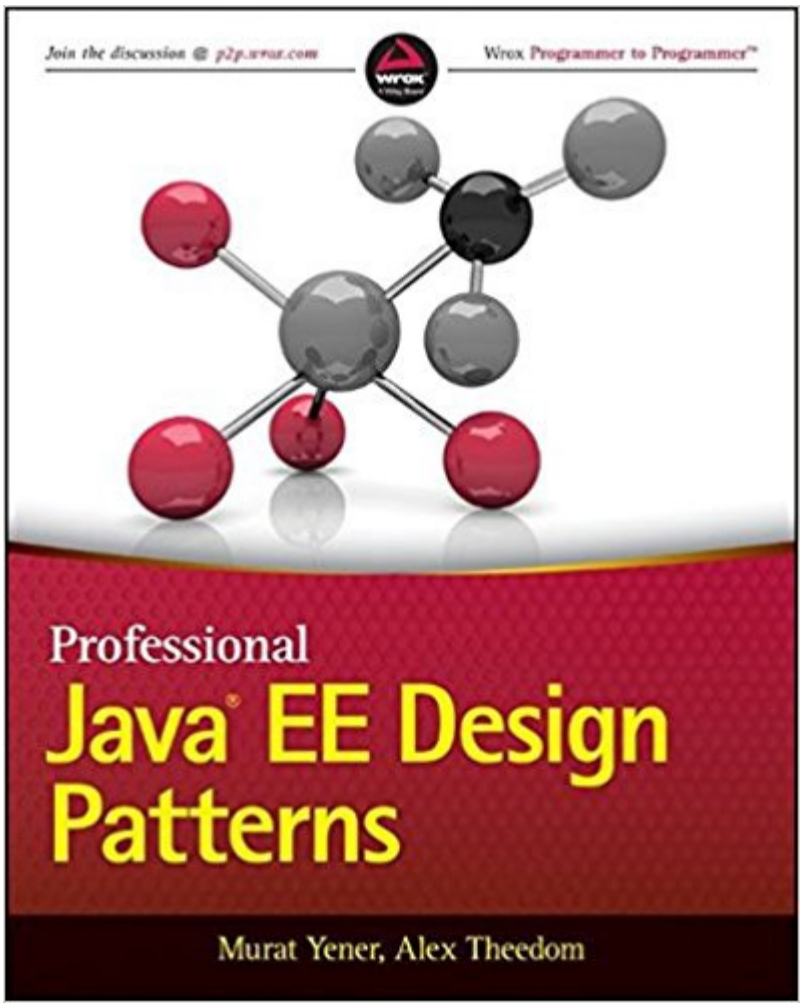
It is valuable to be familiar with **The 23 Design Patterns** of the Gang Of Four Design Patterns (Elements of Reusable Object-Oriented Software)[30].



Though many of the concepts of the *GOF Design Patterns* are timeless, significant technical evolution has occurred since then which merit (also) reading a more modern book. In particular, I recommend reading: **one of the first four resources** listed below, **Clean Code,** and if developing with the Spring Framework, **Spring 5 Design Patterns**. A great person to follow is Martin Fowler and his work. The sequel to the original Design Patterns book can be found here.

Where to learn, Recommended Resources:

Professional Java EE Design Patterns



uDemy Course: Java EE Made Easy—Patterns, Architecture and Frameworks

Martin Fowler's Patterns in Enterprise Development

Modern Java EE Design Patterns: Building Scalable Architecture for Sustainable Enterprise Development, RedHat(Markus Eisele)

Patterns of Enterprise Application Architecture

Clean Code: Design Patterns Videos by Robert Martin

Spring 5 Design Patterns, Dinesh Rajput (**December 11, 2017)**

The Last Step—BUILD THINGS. Build in tandem with learning.

> *“Knowledge without practice is useless. Practice without knowledge is dangerous.”—Confucius*

> *“Tell me and I forget . Teach me and I remember. Involve me and I learn.” —Benjamin Franklin*

This portion is purposefully brief. I believe the aforementioned quotes by Confucius & Benjamin Franklin could not be more fitting.

When one attempts to build something, say an Android App, without fully understanding the Java programming language or the Android APIs, it is, again, akin to playing basketball while blindfolded. Sure, you may make a basket here and there, but you may be ultimately clueless as to what it is you are truly doing. Worse still, you may make some dangerous code.

Conversely, if one has knowledge but does not apply their knowledge to practice, it is almost useless. Anyone who has built even the most primitive systems can attest to the fact that programming is best learned in practice. Learning without practice can in many cases be akin to learning French without engaging in verbal or written dialogue.

Hopefully I have successfully described what I believe to be fallacies on camps that ignore the importance of either theory or practice. The truth is, a unique blend of both is essential. **Reading & building go hand in hand**. There is a problem with reading without enough application and building without sufficient understanding.

To market yourself to organizations with which you wish to work with, it is useful to build applications based on what you have learned and post them to a public repository like GitHub to demonstrate your competence and passion. A blog or vlog is useful as well ;).

Beyond The Core Skills—Things to explore when need necessitates it

> *“Once you stop learning, you start dying.”—Hans Albert Einstein*

At this point I believe that we have touched on the core skills of Enterprise & Web Developers. **The truth is, a career as a Software**

Developer is a never-ending journey in learning. These are merely the core skills. This is very likely to be an incomplete list of all the skills that you may need for your specific job. This is also likely to be an incomplete or outdated list of skills as evolution in technology barrels forward. While these are the core skills, it is useful to pay attention to **technology trends.** In addition to trends, there are many valuable tools and concepts that are useful for various application domains. Among them:

Continuous Integration (i.e. Jenkins)
Virtualization (i.e. Docker)
Agile Practices
Microservices Architectures (This is one is especially popular.)
Visual Profiling (i.e. Visual VM)
Advanced User Experience & User Interface Tools
Advanced Performance & Security Techniques
NoSQL

And so on. These are things to check out when the need for them arises.

[34]—KEEP CALM and CODE JAVA

This publication is the last part of a 3-part series. Should you prefer to read the full article it is available here.

The works cited for the publication is available below

Works Cited

[1]—http://www.oracle.com/technetwork/articles/javame/bluray-142687.html

[2]—https://www.tiobe.com/tiobe-index/java/

[3]—http://www.codingdojo.com/blog/9-most-in-demand-programming-languages-of-2017/

[4]—https://zeroturnaround.com/rebellabs/reports/

[5]—https://springframework.guru/how-do-i-become-a-java-web-developer/

[6]—http://www.asanet.org/journals/ASR/Dec12ASRFeature.pdf

[7]—https://en.wikipedia.org/wiki/Java_%28programming_language%29#cite_note-21

[8]—https://en.wikipedia.org/wiki/SQL

[9]—Grady Booch (1996) *Object Solutions: Managing the Object-Oriented Project*. p. 277

[10]—http://www.decimustechub.com/expertise/java-expertise/

[11]—https://en.wikipedia.org/wiki/SQL:2016

[12]—https://howtodoinjava.com/spring5/spring5-features-and-enhancements/

[13]—Flanagan, David. *JavaScript—The definitive guide* (6 ed.). p. 1.

[14]—Java EE 7, The Big Picture ~ Danny Coward

[15]—https://en.wikipedia.org/wiki/JAR_(file_format)#Manifest

[16]—https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/serialization/

[17]—https://en.wikipedia.org/wiki/Serialization

[18]—https://www.slideshare.net/mgrebac1/java-api-for-json-binding

[19]—https://en.wikipedia.org/wiki/XML

[20]—https://en.wikipedia.org/wiki/JSON

[21]—https://www.w3schools.com/js/js_versions.asp

[22]—https://www.techopedia.com/definition/8842/persistence-computing

[23]—https://zeroturnaround.com/rebellabs/java-tools-and-technologies-landscape-for-2014/10/

[24]—https://www.techopedia.com/definition/4528/protocol

[25]—https://www.techopedia.com/definition/2460/transmission-control-protocolinternet-protocol-tcpip

[26]—http://www.theserverside.com/news/1363671/What-is-an-App-Server

[27]—Java Tools And Technologies Landscape 2016 ~ Rebel Labs by Zero Turn Around

[28]—https://plumbr.eu/blog/java/most-popular-java-application-servers-2017-edition

[29]—https://www.techopedia.com/definition/9847/unit-test

[30]—https://www.dineshonjava.com/design-patterns_25/

[31]—https://www.consumer.ftc.gov/blog/2017/09/equifax-data-breach-what-do

[32]—https://www.nbcnews.com/tech/tech-news/yahoo-must-face-litigation-data-breach-victims-judge-rules-n797871

[33]—https://www.nbcnews.com/storyline/hacking-in-america/more-4-billion-data-records-were-stolen-globally-2016-n714066

[34]—https://www.keepcalm-o-matic.co.uk/p/keep-calm-and-code-java/

[35]—https://insights.stackoverflow.com/survey/2017

[36]—https://www.alexa.com/siteinfo/stackoverflow.com

[37]—https://www.usnews.com/education/best-colleges/articles/2011/08/05/how-higher-education-affects-lifetime-salary

[38]—https://undergrad.cs.umd.edu/what-computer-science

[39]—https://en.wikipedia.org/wiki/Computer_science

[40]—Learn Java for Web Development ~ Vishal Layka

[41]—https://zeroturnaround.com/webframeworksindex/

[42]—https://www.tutorialspoint.com/object_oriented_analysis_design/ooad_uml_analysis_model.htm