

Software Quality  
and Java Automation  
Engineer Survival Guide

Jagdish Babu Munta

Pragmatic Software Quality Engineer

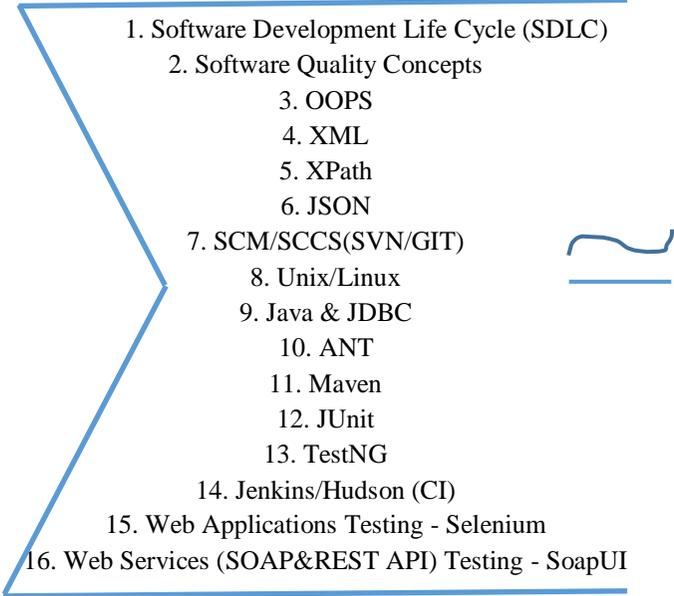
# Software Quality and Java Automation Engineer Survival Guide

(A completely packaged guide with right skills)

500+ Questions & Answers

Basic Concepts + Self Review + Interview Preparation

N = 16, M=550

- 
1. Software Development Life Cycle (SDLC)
  2. Software Quality Concepts
  3. OOPS
  4. XML
  5. XPath
  6. JSON
  7. SCM/SCCS(SVN/GIT)
  8. Unix/Linux
  9. Java & JDBC
  10. ANT
  11. Maven
  12. JUnit
  13. TestNG
  14. Jenkins/Hudson (CI)
  15. Web Applications Testing - Selenium
  16. Web Services (SOAP&REST API) Testing - SoapUI

S=1, Q=1

QA & Java Automation Core Skills

**Jagadesh Babu Munta**

# Copyright Notice

Software Quality and Java Automation Engineer Survival Guide

Copyright © Jagadesh Babu Munta, 2016

Published by Everydayon Inc.

First published in November 2016

Cover page: Curtesy from Createspace

Contact Email: [jagadesh.munta@everydayon.com](mailto:jagadesh.munta@everydayon.com) (or)

[swqualityautosurvivalguide@gmail.com](mailto:swqualityautosurvivalguide@gmail.com)

All rights reserved. Without limiting the rights under copyright reserved above, no part of this publication may be reproduced, stored in or introduced into a database and retrieval system or transmitted in any form or any means (electronic, mechanical, photocopying, recording or otherwise) without the prior written permission of both the owner of copyright and the above publishers.

Although the author have made every effort to ensure that the information in this book was correct at press time, the author do not assume and hereby disclaim any liability to any party for any loss, damage, or disruption caused by errors or omissions, whether such errors or omissions result from negligence, accident, or any other cause.

Software Quality and Java Automation Engineer Survival Guide

Jagadesh Babu Munta



Print ISBN-10: 1-63572-000-1

Print ISBN-13: 978-1-63572-000-6

*Dedicated to the current and aspiring  
Computer Software Engineering Professionals!*

Blank Page

## How is this book?

"This book provides the best answers to the most frequently asked questions in testing job interviews. It is a unique book that includes all essential information in one single place covering manual as well as automated testing and builds the confidence. Jagadesh Munta is an authority in the testing field, and he has interviewed many candidates at his company. He knows what the hiring manager/team is exactly looking for, and he has distilled all that knowledge into this book. If you are preparing for a software testing job interview, then this is the book that you are looking for!"

-- Satya Dodda

(Senior Software Engineering Director, Oracle America Inc., Santa Clara)

"This book is the reason I was able to become a Test Automation Engineer without having any background in software. Not only did it help me master highly technical interviews and get me a high paying job, but it also instilled principles that have allowed me to excel above my coworkers."

(Systems Integration Automation Engineer, San Jose)

"It is such a wonderful guide and I really liked the way you've covered all the topics. This book is extremely helpful for people like me and also for those who are interested in starting a career as a Quality Engineer."

-- Sindhuri Mutyala

(Software Engineer in Quality, Intuit, Mountain View)

"It will be of real help for interviews."

(Senior QA Engineer, Santa Clara)

"It is an incredible Hands-on guide and helpful for people like those who are interested in starting a career as a Quality Engineer/Testing Engineer. Very beneficial in preparing for the software testing/QA job interviews."

-- Malik MA.

(QA Manager, Acentia Software Solutions, Hyderabad)

“This book is very well done and well organized. There are several good books in the market for Quality Engineering but this book focuses on Quality Engineering concepts from an interview perspective, and I can say this is certainly one of the best books around for Quality Engineers looking for a job. This book is clear, concise and offers much more information. I am confident this book can serve any person looking for Quality Engineering jobs.”

-- Praveen Morusupalli  
(Software Engineering Director, Oracle America Inc., Frisco)

“Reviewed the book, and overall, it looks pretty good. These questions/answers are truly a bible for anyone entering QA/QE/automation, and you have explained each very well with good examples.”

(Automation Engineer, San Jose)

“Content is very practical/hands-on, I liked this fact.”

-- Basha Sayed  
(Software Development Manager, Oracle America Inc., Redwood City)

“This is a MUST HAVE book for each and every software engineer (DEV or QA doesn't matter). The knowledge acquired by reading this book is practically useful in day to day activities of a software engineer. Though there are some specific chapters relevant to Java/JDBC/Maven etc., this book overall is a great resource to every software engineer whether or not he works on Java. It covers most of the methodologies/technologies we use every day. The book is up to date on present practices (Agile/Scrum/Automation) and also elucidating the pros and cons of it. In a one liner, this book helps one becoming a great Software Engineer.”

(Senior Software Developer, Fremont)

“This book provides more help to the people because it is highly informative.”

-- Vijaya Bhaskar Tanamala  
(Senior Software Developer, Travelers, Hartford)

“As technology keeps changing, we tend to forget the past. This book is like one place for all. Well explained with examples from end to end from all perspectives.”

-- Sreenivasulu Naidu  
(Lead Software Developer, Hennepin County)

“The concepts explained in this book will help developers handle their day to day job duties effectively. This book is a good starting point reference for the Software Quality Engineers and Developers.”

-- Sriramulu Lakkaraju  
(Software Engineering Director, Oracle America Inc., Santa Clara)

"I had a chance to review a copy of this book, and I think Mr. Munta has put a broad experience into it. This book is a must read guide for beginners to mid-level experts to prepare themselves for the global market."

– Thirupati Panyala  
(Director, VMware, Palo Alto)

"Great attempt to consolidate all the essential and related content for test automation to help developers and QA engineers. Practically, having implemented many test frameworks and worked with many test developers, I can confidently say you will be pleased to have this guide. This book is a go-to guide if you are looking for where to start and as a quick reference for experienced. An excellent review guide for those looking for the change of jobs as well and one stop for all testing phase/process and test automation."

-- Hari Prasad Kuppala  
(Senior Software Engineering Manager, Oracle America Inc., Redwood City)

# Acknowledgements

This book has been made with all supporting of people who interacted with me directly or indirectly.

First of all, I would like to express my indebtedness of gratitude to my parents (*Malakondaiah Munta & Narasamma Munta*) & family for being here on the Earth and growing up. I would like to thank my wife (*Sreelatha Munta*) and sons (*Lokaranjan Babu Munta & Chetan Munta*) for their continuous support in carryout this effort. Special thanks to brothers, *Malik Munta & Jayaprakash Munta* for their extended inputs.

Next, I am grateful with special thanks to *Satya Dodda, Cheuk Chau, Ganesh Kirti, Praveen Morusupalli, Shah Rahman, Chitti Keswani, Shesha Allavarapu, Xiaolin Deng, Luke Su, Ramesh Dimba, Raghav S. Nandhyal, Chandrasekhara Reddy Chennuru, JD Krishnamurthy* for their guidance, support and giving opportunity in gaining the related knowledge.

My sincere thanks to *Sriramulu Lakkaraju, Hari Prasad Kuppala, Tirupathi Panyala, Anjaneyaswamy Varada, Basha Sayed, Venkat Venkatmarayan, Jayanthi Kulkarni, Sujatha Arikapudi, Sudha Pamuru, Padmasri Vallabhaneni, Rika Dodda, Sri Rajesh Ajarapu, Sarada Kommalapati, Srinivasulu Naidu, Vijaya Bhaskar Tanamala, Sindhuri Mutyala, Venkatesh Apsingekar, Sridhar Dhavala* for their encouragement and extended book reviews.

My apologies to those I have missed mentioning here but deserves to express my gratitude!

Thank you!

*Jagadesh Babu Munta*

# Preface

My book authoring is one of my dreams since the year 2009, and especially I was trying to share my accumulated knowledge to the world of people for their benefit. This book is aimed to help most of the software professions and especially to Quality Engineering folks who don't have much support in terms of books and training compared to developers. This book should be helpful for anyone to understand the basic testing concepts, interview preparation in getting a new QA automation engineer job, or switching from another discipline to Software testing. If you feel otherwise, please feel free to contact at [jagadesh.munta@everydayon.com](mailto:jagadesh.munta@everydayon.com) or [swqualityautosurvivalguide@gmail.com](mailto:swqualityautosurvivalguide@gmail.com) so that I can improve and take care of it as updates and new editions.

The goal is to help you.

Jagadesh Babu Munta

# Contents

## Table of Contents

<b>COPYRIGHT NOTICE .....</b>	<b>III</b>
<b>ACKNOWLEDGEMENTS.....</b>	<b>VII</b>
<b>PREFACE.....</b>	<b>X</b>
<b>CONTENTS.....</b>	<b>XI</b>
<b>INTRODUCTION.....</b>	<b>1</b>
<b>SKILL#1. SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)</b>	<b>8</b>
1. <i>What is Software? .....</i>	<i>8</i>
2. <i>What is SDLC? Why do you need such models? What are different models? 8</i>	
3. <i>How is SDLC related to Testing? .....</i>	<i>9</i>
4. <i>What is Waterfall model? What are different phases involved? ....</i>	<i>9</i>
5. <i>What is Agile Scrum process? Can you describe at a high level? .</i>	<i>9</i>
6. <i>When does the testing activity gets started in each development model? 10</i>	
7. <i>What is meant by standup meetings in the scrum process? What are the three questions (typically) you need to answer in Agile Scrum process?.....</i>	<i>10</i>
8. <i>Who is Scrum Master? What is his/her role and responsibilities? 10</i>	
9. <i>Are you aware of any Agile tools? .....</i>	<i>11</i>
10. <i>What is a sprint? What is a typical duration for a sprint?.....</i>	<i>11</i>
11. <i>What is meant by product backlog? .....</i>	<i>11</i>
12. <i>What is a sprint backlog? .....</i>	<i>11</i>
13. <i>Where do you add the changed requirements? Whether in a sprint backlog or product backlog?.....</i>	<i>11</i>
14. <i>Can you change a sprint backlog during a sprint? .....</i>	<i>12</i>

15. What is a use case? .....	12
16. What is a user story? .....	12
17. What is Test Driven Development(TDD) process? .....	12
18. What are the advantages and disadvantages of waterfall model? 12	
19. What are advantages and disadvantages of Agile Scrum process?	
<b>13</b>	
20. What is the typical team size in different SDLC models?.....	13
21. Which SDLC model suits for rapidly changing requirements? ...	14
22. Which model suits for incremental product delivery? .....	14
23. What is meant by refactoring? Where do you typically do?.....	14
24. What is meant by alpha and beta products? Who are the consumers? What is the main objective? .....	15
25. What is a PRD (Product Requirements Document) or BRD (Business Requirements Document)? What it contains and who owns this? .....	15
26. What is a functional specification? What it contains and who owns this? <b>15</b>	
27. What is meant by soft code freeze (SCF)? When does it happen in the SDLC process?.....	16
28. What is meant by hard code freeze (HCF)? When does it happen in the development process? .....	16
29. What is product release candidate (RC)? What is a release? .....	16
30. What is release notes? Who drives this?.....	17
31. What is a release criteria? Who sets this up and who has to follow? <b>17</b>	
32. What is meant by a release showstopper? When does it happen? <b>18</b>	
33. Who are all project stakeholders? What are those teams? .....	19
34. What is a test cycle at product level? .....	20
35. How does the QA team track the weekly status or progress?.....	20
36. What is Go/No-Go for product release?.....	21
<b>SKILL#2. SOFTWARE QUALITY CONCEPTS .....</b>	<b>23</b>
37. What is meant by software Testing?.....	23
38. What is meant by software quality? .....	23
39. What is a software bug or defect?.....	23

40. Why do you need to Test the software? What happens if you don't test separately than the developer? .....	24
41. How Quality Engineering different from Quality Assurance or simply execution?.....	24
42. What is the need for having a separate Quality Engineering Team?	24
43. Can we eliminate all bugs in a given product through extensive testing? What should we do? .....	25
44. What are functional requirements and nonfunctional requirements?.....	25
45. What are the different types of Testing? Can you list some?.....	25
46. What is meant by white box testing? List down some testing types for white box testing. Who does this type of testing?.....	25
47. What is meant by black box testing? List down some of the testing types for Black box testing. Who does this type of Testing?.....	26
48. Differentiate white box vs. black box testing. ....	26
49. What is Unit Testing? What is its relation to whitebox and black box testing. ....	26
50. What is Functional Testing? What is its relation to whitebox and black box testing? Who does this testing?.....	26
51. What is Security Testing? Is it functional or nonfunctional requirement testing?.....	27
52. What is Regression Testing? When and how often do you run these tests?	27
53. What is Performance Testing? Who does this testing - QA or separate team? .....	27
54. What is a Test Plan? Describe briefly the contents of a Test Plan. Have you ever written a Test plan?.....	27
55. What is a Test design and specification? Describe briefly contents of Test Spec or Test Design spec. Have you ever written a Test Spec?	30
56. What is meant by a Test scenario? Where do you put test scenarios? .....	31
57. Can you describe the bug life cycle? What are different states of a bug?	31

58. What is meant by bug Priority? Who sets the priority of a bug? Can we change it after filed the bug? .....	32
59. What is meant by downgrading a bug and upgrading a bug? .....	32
60. What is meant by bug Severity? Who sets the severity of a bug? Can we change it after filed the bug? .....	32
61. How do you file a bug? What details would be added? Give an example.	33
62. What is Stress or load testing? What is longevity testing? How does stress and longevity testing are related? What kinds of bugs can be found? Who does this testing - QA or Dev or any other team?	35
63. What is System testing? Who does this testing? What are the key elements in system testing? .....	35
64. What is a topology? How does it relate to testing? .....	36
65. What is High Availability(HA) testing? What are different types of failures? Who does this testing? .....	36
66. What is meant by code coverage? What is its main purpose? When does this activity happen during the product life cycle? Is it a testing activity? Who lead this? What kind of tests executed?.....	37
67. What are different code coverage tools for Java? .....	38
68. How does the Quality is being tracked? What are different levels?	38
69. What is meant by testing artifact? What are different artifacts? .	39
70. What is the test development process? What are key steps? .....	39
71. What is the difference between manual testing and automated testing?	41
72. What is the test execution process? What do you need to focus? .	43
73. What is smoke testing? When do you run? Who run these tests? .	44
74. What is Basic Acceptance Testing (BAT)? Who and when do you execute BAT tests? .....	44
75. How do you sign-off from QA? What do you do? .....	45
<b>SKILL#3. OBJECT ORIENTED PROGRAMMING &amp; SYSTEM (OOPS)</b> .....	<b>47</b>
76. What are OOP and OOPS? .....	47

77. What are fundamental principles or characteristics or properties of OOP? 47	
78. What are the benefits or advantages of using OOP? .....	47
79. What is an Object? What it contains? .....	47
80. What is a Class? Why do we need? .....	48
81. What is an instance? .....	48
82. What is a method?.....	48
83. What is meant by object state? .....	48
84. What is Interface? Why do we need? .....	48
85. What is Abstraction? Generalization? Concreteness? .....	49
86. What is Encapsulation? How can be achieved? .....	49
87. What is Polymorphism? How can be achieved?.....	49
88. How do you differentiate Object Oriented Programming with Procedural Programming? .....	50
89. What are the differences between Object-Oriented vs. Object-based programming .....	50
90. What is Abstract Class?.....	51
91. What is Inheritance? What are the benefits? What are different types of Inheritance? Describe them. ....	51
92. What is the difference between Abstract class and an Interface? .....	52
93. How do you create Objects? .....	53
94. What is a constructor? What is a destructor? .....	53
95. What is default constructor? What is parametric constructor? ...	53
96. What is overloading? What are different types?.....	54
97. What is overriding? How to do? .....	54
98. What is data hiding? Why do you need it?.....	55
99. What is “this” and “super” keywords w.r.t objects? .....	55
100. What is meant by method signature? Does it include return type?	

55

101. Can an abstract method be private? Why or why not?.....	55
102. What are the class relationships? .....	55
103. What is UML and why do you need? .....	56
104. What is UML notation for relationships? .....	56

<b>SKILL#4. XML</b> .....	<b>59</b>
105. What is XML? Is it a standard? .....	59
106. What are the main objectives of XML? .....	59
107. What is the structure of XML? .....	60
108. How do you define the structure of XML? How do you write XML?	
60	
109. What are the differences between XML and HTML? .....	60
110. What is meant by well-formed XML? .....	60
111. What is meant by validation of XML? .....	61
112. What is DTD? What is XML Schema or XSD? What are the differences? .....	61
113. What are some of the popular techniques to parse XML files? ...	61
114. What are XML declaration and processing instruction in XML?	62
115. What is namespace in XML? .....	62
116. What are real applications of XML? .....	63
<b>SKILL#5. XPATH</b> .....	<b>65</b>
117. What is XPath? Why does it matter? .....	65
118. Is it a standard specification? Who defined? .....	65
119. How do you represent XML for navigation? .....	65
120. What are different node types in XPath? .....	65
121. What are different node relations? .....	66
122. What is meant by atomic node value or value? .....	66
123. What is the XPath expression for selecting an element from root node? 66	
124. What is the XPath for selecting any node under the tree? .....	66
125. What is the XPath for selecting a node with an attribute? .....	66
126. What is a predicate? What is its purpose? .....	66
127. What is the XPath for selecting the first node? And also last but one node? Or at a particular position? .....	67
128. What is the XPath for selecting nodes with expected attribute value? 67	
129. What is the XPath wild characters to match any node and nodes with any attribute? .....	67

130.	How to select several XPath in a single selection/expression? ..	67
131.	What is axis? Why do you need? Please provide few expressions with axes. ....	67
132.	What are different XPath operators? .....	69
133.	What are some of the inbuilt/standard functions in XPath?.....	69
134.	What are different XPath evaluators? .....	69
135.	Where do you use XPath in real applications?.....	69
<b>SKILL#6. JSON</b>	.....	<b>72</b>
136.	What is JSON? How does JSON format look like (snippet)? .....	72
137.	Is JSON a programming language? Or a protocol? Or a message format? Is it language dependent?.....	73
138.	What are the key advantages with JSON over XML format? .....	73
139.	Where do you use JSON?.....	73
140.	What is the HTTP header used for passing the JSON data? .....	73
141.	How do you process (parse, transform, generate, query) the JSON text?     74	
142.	How do you query the data from JSON file from shell command or in a simple script?.....	74
<b>SKILL#7. SOURCE CODE CONTROL SYSTEM(SCCS)/SCM - SVN &amp; GIT</b>	.....	<b>76</b>
143.	What is source code control system (SCCS) and SCM? Why do you need?     76	
144.	Describe the key elements in SCM such as repository, workspace, versioning?.....	76
145.	What do you store in the repository? .....	77
146.	What is source code? Why is it important? What happens if you keep the code at individual machines? .....	77
147.	What are the primary functions of SCCS software?.....	77
148.	What are different popular SCCS software available?.....	77
149.	What is meant by the checkout, check-in or commit, merge, delete, log, status? .....	78
150.	What is meant by the trunk or mainline? Branching? And tagging?	

151. What is meant by merging conflicts? What happens if you don't resolve the conflicts? How do you resolve the conflicts?.....	79
152. Can you describe SVN? What are the supported protocols?.....	79
153. What kind of architecture/design used in SVN? Any other architectures?.....	80
154. What do you need for interacting with SVN server? .....	80
155. What is the svn command for checkout workspace? .....	80
156. What are the svn commands to do a fresh checkout, check-in or commit all files in the current directory?.....	80
157. What is the command to see the list of files modified locally? ....	81
158. What is the svn command to see the history of changes done on a particular file?.....	81
159. What is the svn command to see the differences between the locally changed file vs. file in the repository?.....	81
160. Which svn plugin used in Eclipse?.....	81
161. Can multiple users checkout and check-in in parallel? Is there any lock?	81
162. What are the key features that you need for any version control systems like Git? .....	82
163. What are the typical steps to work with Git?.....	82
<b>SKILL#8. UNIX/LINUX OS.....</b>	<b>85</b>
164. What is an Operating System (OS)? Brief on its typical functions.	85
165. What are the different OS architecture types?.....	85
166. What is meant by a Platform? .....	86
167. What is Unix/Linux and how is it different from Windows?.....	86
168. How do you see Linux and windows differences at usage level?.	87
169. What are different parts of linux OS?.....	87
170. How do you connect to a remote unix/linux machine? .....	87
171. How do you search given string or word in a file? .....	88
172. How do you run a command or process in the background? How do you bring it to the foreground? .....	88
173. What is the command to find running processes and process ids (PID)?	89

174. What is the command to kill a particular process id? .....	89
175. What is the command to list the hidden files? .....	89
176. What is meant by file permissions? How to set the permissions? What is meant by 644? .....	89
177. Why do you get the error 'command not found' while executing a command/file (with right directory path)? How to make it work? 91	
178. What is the command to see the contents of a file? How to pause after each page filled? .....	91
179. What is the command to get first 10 lines of a file? And getting first 5 lines? 92	
180. What is the command to get last 10 lines of a file? And getting last 5 lines? 92	
181. What is the command to monitor a running log file (such as a tomcat server.log or other file where content is getting modified all the time)? .....	92
182. What permission to be set to a directory so that 'cd dir' command would work? .....	92
183. What is meant by pipeline ( )? Where do you use? .....	92
184. What is meant by symbolic link and the command to create a symbolic link to a file or directory? .....	93
185. How do you find given file and directory sizes? .....	93
186. How do you find a file in a given directory? .....	93
187. What is 'vi' editor? Describe on how to edit files (insert mode, save, copy lines, paste lines, delete lines, etc.)? .....	94
188. What is an alias? How do you set an alias and how to remove alias? 95	
189. How do you count lines, words, and bytes in a file? .....	95
190. What is the command to check which user being used in the prompt? 95	
191. What is a 'touch' command do? Where do you use? .....	96
192. What is the file separator character and path separator character in unix/linux? .....	96
193. What is the kernel? What is the command to see the kernel version? 96	
194. What is the command to see the machine name or hostname? ....	96

195. How to find the OS release version? .....	96
196. How do you see the system details like total memory(RAM), processors and monitor the processes? .....	97
197. How do you redirect the output of a command to a file (such as log file)?	97
198. How do you append text to a file without overwriting the existing content?	97
199. What is the command to pause the screen scrolling temporarily and resume again? .....	98
200. How do you suspend a running process? And then send to the background or bring to the foreground? .....	98
201. What is a 'tar' command? How to use it and how to untar a file? 98	
202. Write a single command to count all java files in a given directory and have a word 'test'. .....	99
203. Write a command to get the 3rd word (separated by comma) of a string in a variable. ....	99
204. What is a sudo command? Why do you need? .....	99
205. What is a ping command and how it helps in checking the host reachability? .....	100
206. What is curl command and why do you need? .....	101
207. How do you write associative arrays in bash? .....	103
208. How do you run a command on the terminal without terminating when the shell closed or terminal closed? .....	103
209. How to see the help for commands in unix/linux? .....	103
210. What is a shell script? Write a script to do 1) Given Java code compilation and execution of class 2) Print hostname and how many java files are there in a given directory. ....	103
<b>SKILL#9. JAVA</b> .....	<b>106</b>
211. What is Java? What are benefits of Java language? .....	106
212. What is Java Platform? Why it is called a platform? .....	106
213. What are key features of Java? .....	108
214. Is Java language cross-platform? .....	109
215. What is JVM? Describe more on this. ....	109

216. What is JRE? Describe more on this. ....	110
217. What is JDK? Describe more on this. ....	110
218. Does JVM know about Java language? If yes, how much or what API it knows? If not, then what it can do? .....	110
219. What is the relation between JDK, JRE and JVM? .....	111
220. Where to get the Java or JDK? .....	111
221. Could you describe in general how is the Java code development process or lifecycle? .....	111
222. What is meant by compilation? Why do you need to do? .....	112
223. How do you compile the Java code? .....	113
224. What is the file extension to be used for Java source code? .....	113
225. What is meant by WORA? .....	113
226. What is the Java program's runnable format? .....	113
227. What is the class file structure? .....	113
228. How to determine the java and javac versions? .....	114
229. How do you run Java programs using the command line? .....	115
230. How do you resolve classes dependency during compilation and runtime? What are different ways? .....	115
231. What is a Java package? Why do you use? .....	115
232. How do you package the classes into a single file? What is the format of the file? .....	116
233. What do you set in the CLASSPATH (or) -cp (or) -classpath options? Like class file path/class files directory path/jar files directory path/exact jar file path? .....	116
234. What is JVM stack? When does this be created? .....	117
235. What is StackOverflowError? When do you get it and how to resolve? .....	117
236. What is OutOfMemoryError? When do you get this and how to resolve? .....	117
237. What is ClassNotFoundException? When do you get this and how to resolve? .....	118
238. What is JVM heap? When does it gets created? .....	119
239. What is Garbage Collection? What is its role in JVM? .....	119
240. Does the heap storage have to be contiguous? .....	121

241. What is a variable? Where do you define?.....	121
242. What is meant by variable assignment and initialization?.....	121
243. What are different types of variables?.....	122
244. What is a static variable, and its scope? How do you reference it?	122
245. What is an instance variable and its scope? How do you reference? .....	122
246. Are all non-static variables are instance variables?.....	123
247. What is a local variable, and its scope? How do you reference?	123
248. What are the differences between static variable and instance variable?.....	123
249. What is a field? How is it same or different (or same) from other types of variables? .....	124
250. What is a data type? Where do you use? .....	124
251. What are the default values for instance or static variables? ...	125
252. What is meant by Parameter? How is it different from a variable or field? .....	125
253. What are different techniques to pass the parameter values from caller to callee method?.....	126
254. What kind of parameter passing technique is supported in Java?	126
255. How does the “Pass by value” works?.....	126
256. How does the “Pass by reference” works?.....	127
257. What are the differences between “Pass by value” and “Pass by reference”? Please give an example. ....	127
258. What is a literal? Any examples?.....	128
259. What is an expression?.....	128
260. What are the unary & binary operators?.....	129
261. What are prefix & suffix unary operators? What are the differences? .....	129
262. What is a statement? What are different kinds of statements? ..	130
263. Can we have an empty statement? How do you write an empty statement? .....	131

264. What is a block? Can we have an empty block? Do we need to have ";" at the end for a block? .....	131
265. Where do you use block statements? .....	131
266. What are looping statements? Could you briefly explain? .....	132
267. How do you represent infinite loops with "for", "while" or "do-while" statements? .....	133
268. What is meant by branching? What are different branching statements? Where do you use? .....	134
269. What is the purpose of switch statement? How do you write switch statement? .....	135
270. How do you write an interface? Can you implement multiple interfaces from single class? .....	137
271. How do you write abstract class? Can you extend from multiple classes? .....	139
272. What are the differences between an interface and abstract classes? .....	140
273. When do you use interfaces and abstract classes? .....	141
274. How do you write constructors? .....	141
275. Can you use a default constructor without defining it in a class? Please explain why or why not? .....	143
276. In class, if you have defined a parametric constructor only, then can you construct the objects with default constructor? In this context, does java provides the default constructor? .....	143
277. What is an inner class? What it represents in OOP? Describe more on nested classes? .....	143
278. How does the generated class look like for an inner class? .....	145
279. What is meant by access modifier? Where do you use? What are different access modifiers? .....	145
280. How do you provide access when you have private access members? .....	147
281. Can you assign multiple access modifiers to a single member? .....	147
282. What is final keyword significance? Where do you use? .....	147
283. Can you define a private constructor? If yes, why do you do? ..	147
284. What is a Java exception? Could you please brief on different types of exception? .....	148

285. How is the Exception and Error classes tree?.....	149
286. How to handle Java exceptions? Which exception to handle first when multiple exceptions to be caught? .....	149
287. What are the differences between checked exceptions and unchecked/runtime exceptions? .....	151
288. How to resolve when there is an exception stacktrace? .....	152
289. How to raise exceptions? .....	152
290. How to create a new custom or user defined checked exception?	152
291. What is a conditional operator, “?:”?	153
292. How and where to use “for-each” statement? .....	153
293. What is “enum” data type and how to use? .....	154
294. How to write a recursion method? What is the core advantage if it? Where to pay attention? .....	154
295. What are the different Java I/O classes? .....	155
296. What are Java Collection and Java Collection frameworks? ...	156
297. What are the advantages of Java Collections?.....	156
298. What are the core interfaces in Collection framework? .....	156
299. What are the common methods in the Collection (interface)?...	157
300. What is Set interface, and its implementation classes? .....	158
301. What is List interface, and its implementation classes? .....	159
302. What are different algorithmic methods available in the List collection?.....	160
303. Could you explain the Map collection? .....	161
304. How should you write the classes (the rules to be followed) in order to use custom Java beans as elements in a collection?....	162
305. If there is a requirement where a million records should be read from DB and would like to make them unique and sorting. Which collection data structure is appropriate?.....	163
306. If there is a requirement where some kind of paired objects to be stored, then which collection data structure is appropriate? ....	164
307. How to choose a specific Collection data structure? What are different performance characteristics (Big O-complexity)?.....	164

308.	<i>What is JDBC? Could you please explain about different JDBC driver types.....</i>	166
309.	<i>Could you please explain end-to-end steps in a typical JDBC program? .....</i>	167
310.	<i>What is JDBC URL or Connect string? Can you give examples?</i>	171
311.	<i>How to create a datasource in WebLogic? .....</i>	171
	<b>SKILL#10. ANT .....</b>	<b>173</b>
312.	<i>What is Apache's ANT? What are the core elements? .....</i>	173
313.	<i>Why and where do you use? Any alternatives? .....</i>	173
314.	<i>How do you design your project workspace and how do you build the project? What do you write? .....</i>	173
315.	<i>Is ANT cross platform? Can we run on all platforms? .....</i>	174
316.	<i>What is minimally required to run ANT build file? .....</i>	175
317.	<i>How to setup ANT? .....</i>	175
318.	<i>What is meant by ant target? .....</i>	175
319.	<i>How do you define a target in the build xml? .....</i>	176
320.	<i>What is ant task? .....</i>	176
321.	<i>How do you write and define a task in the build xml? .....</i>	177
322.	<i>How do you load the parameters from a file? Also from Java system properties? .....</i>	178
323.	<i>How do you reference properties? What are different in-built properties? .....</i>	178
324.	<i>How to group multiple targets to run from single target? .....</i>	179
325.	<i>What is the depends attribute? What is description attribute? ..</i>	179
326.	<i>Do you need have all the time build.xml? Or any file is ok? What is the difference having the build.xml or another file name? .....</i>	179
327.	<i>What are the minimal things required in build.xml? .....</i>	180
328.	<i>How do you run a target from cmd or terminal? Can you run a task directly from ant command at the terminal? .....</i>	180
329.	<i>How to set a default target to be executed without specifying with ant command? .....</i>	180
330.	<i>Can you run any program from ANT? If yes, how? .....</i>	180
331.	<i>What do you need to run build.xml? Any downloads? .....</i>	180

332.	<i>How do you set the classpath?</i>	180
333.	<i>What is classpathref in the ANT build and what is its significance?</i>	181
334.	<i>What do you need to integrate any software?</i>	182
335.	<i>How do you integrate Java compilation and execution (snippet) process in the ant build file?</i>	182
336.	<i>How do you create a jar in ant framework (snippet)?</i>	182
337.	<i>How do you integrate JUnit runtime in ant framework (snippet)?</i>	183
338.	<i>How do you integrate TestNG runtime in ant framework (snippet)?</i>	183
339.	<i>Can you include one build.xml in another build.xml?</i>	185
340.	<i>How can you call another target from same build file?</i>	185
341.	<i>How can you call a target from one build.xml in another build.xml?</i>	185
342.	<i>How do you print messages on the console? How do know supported ant tasks?</i>	185
343.	<i>How do you know supported ant targets from a ant build.xml?</i>	186
344.	<i>What is fileset? Where do you use?</i>	186
345.	<i>What is the end-to-end flow (in build.xml) to build and run a Java application?</i>	186
<b>SKILL#11. MAVEN</b>		<b>191</b>
346.	<i>What is Maven? Why and how do you use? What are other alternatives?</i>	191
347.	<i>What are key objectives and features of Maven?</i>	191
348.	<i>How does Maven use Convention over Configuration?</i>	192
349.	<i>How to install and setup Maven - steps?</i>	192
350.	<i>What is the latest Maven version?</i>	192
351.	<i>How do you create a new project workspace using archetype or project template?</i>	193
352.	<i>Describe the default workspace structure created by Maven? ..</i>	193
353.	<i>What is POM? What is its core value?</i>	194
354.	<i>Describe briefly the skeleton of POM file.</i>	195
355.	<i>What is groupId? What is artifactId? What is version?.....</i>	196

356. What is the jar name look like based on the above? .....	197
357. What is Maven repository? Describe both local and central repositories?.....	197
358. What is the default local repository path? .....	197
359. Where do you set the dependency information? .....	197
360. How to set the dependency to download testng 6.9.6 version? Also selenium java 2.46.0 jar?.....	198
361. What are the commands to compile sources, build jar, and run tests? .....	198
362. What is the main sources directory path? What is the test sources directory path? .....	199
363. Where are the classes get generated for both product and tests? .....	199
364. Do you need to download the dependent jars all the time from the internet (central repository)? How to force the update? .....	199
365. What is the command 'mvn install' does?.....	199
366. How to publish the artifacts to central repository?.....	200
367. Where do you check if any third party dependent jar is needed? .....	200
368. How to generate the product documentation? Where does it generate the files (index.html etc.)? .....	200
369. What is a maven plugin? What is main functionality? .....	200
370. What is a maven goal? Where is it used? .....	200
371. What is a build phase? How do phases relate to plugins?.....	201
372. What are default phases in the maven build lifecycle?.....	201
373. How do you clean up the build? .....	202
374. What is meant by maven local settings and global settings? And where do they located? .....	202
375. What do you need while publishing an artifact to a repository? .....	202
376. How to generate eclipse project files?.....	203
377. How do you generate ANT build files automatically from existing Maven project?.....	203
378. What project files do you check into source code repository? ..	203

379. Do you generate pom.xml and other files again if you accidentally deleted the files? .....	203
380. What happens if you accidentally deleted target folder? .....	204
381. How do you compare Maven with ANT? .....	204
<b>SKILL#12. JUNIT</b> .....	<b>206</b>
382. What is JUnit? .....	206
383. What kind of testing framework is it? Who uses it - Developers or QA/QE? .....	206
384. Where do you use JUnit? .....	207
385. What are all major versions in JUnit? Which version have you used? Differences between JUnit 4 or 4.x vs. JUnit 3 or 3.x? ...	207
386. What do you need to start working with JUnit? .....	208
387. How to write, compile and execute a simple JUnit test program? .....	208
388. What are different ways that you can run a JUnit test program? .....	213
389. How do you run JUnit tests from the command line? What do you put in the classpath? .....	214
390. Does JUnit support annotations? Which version? .....	215
391. Describe the key annotations supported in JUnit4? .....	215
392. What are the differences between JUnit3.x and JUnit4.x? .....	216
393. Why do you import JUnit assert methods as static? What is its advantage? .....	217
394. What is “no runnable methods” exception message? When do you get this? .....	217
395. How do you generate a single HTML report from multiple JUnit results xml files? .....	217
396. What happens when a test method or code inside it throws an exception? .....	218
397. Can a test method contain multiple assertions? What happens if any assertion failed? .....	218
398. What is your strategy to have assertions as one assertion per method or multiple assertions in a test method? Are there any advantages and disadvantages? What is the recommendation? .....	218

399.	How do you create test suites? .....	218
400.	Can you group the JUnit tests like in TestNG?.....	219
401.	How do you enable and disable test methods? .....	219
402.	How do you distinguish a regular Java method vs. a JUnit test method? What are the rules to be followed for a test method?..	220
403.	Is JUnit a Data Driven framework? How do you write data driven test?	220
404.	Do we need to follow naming rules for test methods in JUnit 4?	222
405.	Do you write a test class per a dev class? What is your strategy?	222
406.	What happens if a test method is declared as “private” and also it returns a value rather than void? .....	222
407.	Do you need the main() method in the JUnit test class? Why or why not?	222
408.	What is a JUnit Test Fixture? When do you use? .....	223
409.	What is the Java package for JUnit? What is the main test runner program class? .....	223
410.	How do you run JUnit tests as a batch without creating explicit suite class creation?.....	223
411.	What are other frameworks in similar to JUnit? .....	223
412.	How do you develop the JUnit Tests? Can you explain end to end process for automating these tests?.....	224
	<b>SKILL#13. TESTNG .....</b>	<b>228</b>
413.	What is TestNG framework? what are the key goals with TestNG?	228
414.	What are key features of TestNG? .....	228
415.	What are all required software and libraries to work with TestNG?	229
416.	What are all different annotations supported by TestNG?.....	229
417.	What version of TestNG have you used? What kind of plugins have you used? .....	230
418.	What are the typical TestNG test development steps? .....	230

419. How does the report format look like? Does this generate XML report?	231
420. How do you execute TestNG tests?	231
421. How do you integrate with ANT?	232
422. What are the steps in writing a data-driven test class with TestNG?	233
423. How do you group the tests?	235
424. How do you set the dependency of test method on groups or other test methods?	236
425. How do you enable and disable the tests?	237
426. What is testng.xml? How do you set the test suites and test classes?	237
427. Can you run multiple testng.xml files at a time? How?	238
428. What is the default report directory? What is default report format?	238
429. What is the TestNG API java package?	238
430. What are different assertions supported by TestNG Assert? ....	238
431. How do you import methods than classes in TestNG class (as an example for assertEquals())?	239
432. What are the similarities between TestNG and JUnit 4?	240
433. What are the differences between JUnit4 and TestNG? Which one do you use? Why?	241
<b>SKILL#14. HUDSON/JENKINS</b>	<b>243</b>
434. What is meant by continuous integration (CI)? What are the key benefits of CI?	243
435. What are different CI tools/software?	243
436. Describe Hudson and Jenkins software?	244
437. What is the difference between Hudson and Jenkins? How to choose which one?	244
438. How do you setup Jenkins?	245
439. How do you change the default http 8080 port while starting the Jenkins?	245
440. What is a Jenkins job? How do you create jobs in Jenkins? Who creates the Jenkins job?	245

441. How long does it take to create a simple job for running your tests?	247
442. How do you schedule and/or run the builds?	247
443. How do you schedule the build periodically? What field values set to schedule at 11pm on every day?	248
444. How to clean up old builds and workspace?	248
445. How do you configure the SCM (SVN) in a job?	248
446. How to install a Jenkins plugin?	249
447. What do you need to do if Git source code is needed in Jenkins?	249
448. What is the Jenkins dashboard? And what actions you can perform?	249
449. How do you configure ANT build file and target to be executed?	250
450. How do you archive the results? And send the build artifacts or results in an email?	250
451. What happens when multiple jobs or same job scheduled multiple times?	251
452. How to see the console output while a job is running?	251
453. How do you configure JDK, ANT, Maven in Jenkins?	252
454. How does one can extend the Jenkins functionality?	252
455. How many plugins are there in Jenkins? Can you list some? ...	252
456. How do you get the help while creating job?	252
457. How do you do distributed builds? What are Jenkins master and slave nodes?	253

## **SKILL#15. WEB APPLICATIONS TESTING AND SELENIUM**

.....	255
458. What is the Web Application/UI testing means to you?	255
459. What are the key challenges in Web UI testing?	256
460. What automation framework(s) have you used?	256
461. What is Selenium and why do we use? Any alternatives?	257
462. What are the differences between Selenium IDE, Selenium RC, Selenium1.0, Selenium2.0, WebDriver?	257
463. What Selenium versions and drivers you had used?	258

464. How to do you develop automated tests for a given web site? Explain end-to-end process flow.....	259
465. Can you briefly explain on how to write a simple Selenium test? 260	
466. What are the different web elements that you can find from Selenium and the list of actions performed? .....	262
467. How does the page loading happen in WebDriver? How long does the test code to wait? .....	263
468. What are different WebElement locators? How do you find those elements? What tools or strategies do you use? .....	264
469. How to maximize the browser window? .....	265
470. How to switch between windows and frames?.....	265
471. What are the challenges that you had faced with Selenium automation? .....	266
472. How do you test Ajax requests?.....	266
473. What is implicit wait and explicit wait?.....	267
474. What happens if we have both explicit wait and implicit wait together in the same test? .....	268
475. What is the default wait time while finding an element?.....	268
476. What are different expected conditions to be used in the explicit wait?   268	
477. Does Selenium WebDriver supports any test report/driver framework? .....	269
478. What automation test report/driver framework that you had used with Selenium? .....	269
479. How do you automate Ajax auto-suggestion or auto-complete UI? 269	
480. How do you trace or log the web requests access? Can we have event listeners? How?.....	271
481. What is WebDriver Actions class? How to use it?.....	271
482. What are the common issues with cross browser testing? .....	272
483. What is meant by headless driver and an example?.....	272
484. How do you set the browser specific settings? .....	272
485. What is a proxy server and how to configure it in the automated tests?   273	

486. What are the selenium design patterns? What design strategies that you had used? .....	274
487. What is PageObject? How to write design PageObject based tests? .....	275
488. What are the different browser versions and platforms you have used in your environment? .....	276
489. How did you integrate Selenium tests with continuous integration system? .....	276
490. What is Selenium Grid2 and why do we need? .....	277
491. How do you configure the Selenium Grid and run tests in parallel? .....	277
492. How do you test Flash objects with Selenium WebDriver? .....	278
493. Explain and differentiate Data Driven, Keyword Driven, Hybrid test design frameworks. ....	279
494. Can you explain the high-level design process for the test automation? .....	280
495. How to read the properties file from Java code? .....	281
496. How do you take the screenshots/images for the failed tests? Give the code snippet. ....	281
497. How do you handle drag-and-drop web UI scenario in the automation? .....	283
498. How do you do handle html SELECT element automation? .....	284
499. What is HTTP protocol? Please explain. ....	286
500. What are different HTTP status codes? .....	287
501. Can we get HTTP status codes from Selenium WebDriver? .....	287
502. Can you explain the end-to-end process for a given website testing with selenium? .....	288
<b>SKILL#16. WEB SERVICES (SOAP&amp;REST API) TESTING AND SOAPUI TOOL .....</b>	<b>291</b>
503. How do you describe Web Services? What are key elements? ..	291
504. What are some of the distributed technologies? .....	291
505. What are the benefits or key advantages of web services? .....	292
506. What are key elements in web service architecture? How is it different from traditional client/server? .....	292

507. What are different kinds of web services?.....	293
508. What kind of protocol used between the service provider and consumer?.....	293
509. What is a de-facto standard web service?.....	293
510. What is a SOAP based web service? .....	294
511. What is the transport protocol used for SOAP web services? And what is the Payload? .....	294
512. What are different versions available in SOAP? What is the latest? .....	294
513. What are some of the WS implementations around? .....	294
514. What is meant by API? How do you describe APIs or characteristics of APIs?.....	295
515. Why are Web Services called as APIs? .....	296
516. How would you test an API (say for address validation)? How many tests do you write? Can you write single test case covering all different scenarios? .....	296
517. What are the things to be tested in Web Services API testing?..	296
518. What is a RESTful Web Service? How do you describe these services? .....	297
519. What are the key elements in REST APIs? .....	298
520. What are the different message formats supported in SOAP and REST WS? .....	299
521. How is a SOAP envelope and what it contains? .....	299
522. How do the REST API request and REST API responses look (give a snippet)?.....	299
523. What is WSDL? And it's significance? .....	301
524. What is UDDI, and its significance? .....	302
525. What is WADL, and its significance? .....	302
526. What is Service Oriented Architecture (SOA)? .....	302
527. How do you compare SOAP web services with RESTful web services? Which one is more secure/standard/reliable? Which one is more simple to use? .....	303
528. What are some of the example SOAP web services and RESTful APIs? .....	304

529.	<i>What are some of SOAP based WS-* specifications?.....</i>	<b>304</b>
530.	<i>What is a protocol or protocol stack? What is meant by over HTTP protocol?.....</i>	<b>305</b>
531.	<i>What is HTTP protocol? What are different HTTP methods?...</i>	<b>305</b>
532.	<i>How do you map HTTP methods with CRUD operations in REST?</i>	<b>307</b>
533.	<i>What are some of the web services testing tools or frameworks available for the automated testing? .....</i>	<b>307</b>
534.	<i>What is SoapUI tool? Why do you choose SoapUI tool? What are latest versions? .....</i>	<b>308</b>
535.	<i>What are different platforms supported? .....</i>	<b>308</b>
536.	<i>What kind of testing can be done with SoapUI tool? What are some of the key features of SoapUI tool? .....</i>	<b>309</b>
537.	<i>Does the SoapUI tool support the data-driven framework?.....</i>	<b>309</b>
538.	<i>How do you do the data driven in SoapUI tool?.....</i>	<b>309</b>
539.	<i>Describe the end-to-end steps for testing of web services using SoapUI tool? .....</i>	<b>309</b>
540.	<i>What are the data stores that can be associated with data sources?</i>	<b>310</b>
541.	<i>What is basic input required to test SOAP and RESTful web services? .....</i>	<b>311</b>
542.	<i>How do the tests developed in SoapUI projects getting saved? What are the different types of SoapUI tests execution?.....</i>	<b>311</b>
543.	<i>Can you generate JUnit test reports from SoapUI tests? And how?</i>	<b>315</b>
544.	<i>How do you integrate SoapUI tests in ANT and Jenkins?.....</i>	<b>315</b>
545.	<i>Without SoapUI tool, how can you test web services? .....</i>	<b>315</b>
546.	<i>How do you use the properties in SoapUI projects? .....</i>	<b>318</b>
547.	<i>Can you create a mock service from SoapUI tool? And how? ..</i>	<b>319</b>
548.	<i>How do you assert the responses with xpath in SoapUI project?</i>	<b>319</b>
549.	<i>What is the hierarchy of test workspace structure with SoapUI?</i>	<b>320</b>
550.	<i>What kind of automation framework used for both web application and web services testing?.....</i>	<b>320</b>

<b>HANDS-ON/GETTING STARTED.....</b>	<b>322</b>
<b>JAVA SAMPLES CODE.....</b>	<b>325</b>
<b>SELENIUM SAMPLES CODE.....</b>	<b>360</b>
<b>CONCLUSION.....</b>	<b>397</b>
<b>REFERENCES.....</b>	<b>398</b>
<b>READER'S TRACKING INDEX .....</b>	<b>399</b>

# Introduction

The book is about Software Quality & Java automation skills with basic concepts, self-review and interviews preparation related to the Java based projects in a practical sense with questions-answers. This book is targeted mainly at beginners to the software quality and development engineers. It is also useful for experienced quality engineers to do self-assessment and to be on top of relevant automation skills.

In the software industry, the quality related professional role is called as QA (Quality Assurance) or QE (Quality Engineering) engineer. In general, the "Quality Assurance" and "Quality Engineering" roles carry out the similar effort. The QA is a traditional role originated from industrial manufacturing assembly where the final product quality tested by verifying against a checklist, inspected for any damages and finally put a quality sticker. The similar QA role came to IT Software but associated with Engineering background. So, QE role makes sense for Software industry, and it stresses more on the importance of applying the Engineering principles rather than simply repeating the quality assurance actions.

The QA role involves the tasks such as understanding of the software product features & domain knowledge, writing test plans, writing test specifications, manual execution of tests and interpreting results. Later, one has to do the development of automated tests, automated execution and generation of final results summary for release level regression testing. Note that the QA role need average coding skills, but it takes more time and energy than in general as expected by many people. Because, for each product release in the product line, the developed tests should be continuously executed and verified to make sure no regressions at any point in time. The Quality Engineering skills have to be applied all the time during the product development. The quality has to be measured, assessed and concluded to prove that product is meeting the customer expectations. Without the engineering principles, quality testing might be happening and can discover some bugs but it may not

ensure the proper proof to justify the on the quality and might leave some gaps. In summary, apply the core engineering principles such as measuring, making consistency, repeatability and correctness during the application of testing techniques.

It is difficult to do the above principles merely with the manual approach because of limitations such as human errors in producing consistent actions & output, testing coverage with quick, agile product development cycles and making sure each build is verified for regressions. The Quality Automation is one of the important things to address this continuous regression testing. The automation effort needs programming skills in the same software domain and relevant skills to achieve the high automation, which saves time and test coverage.

It is hard to survive without having the fundamentals and automation skills in software development and quality as a QA or dev engineer. Also, many skills need references to many other books that were not targeted at QA professions. Now, this book is an attempt to create such survival guide for QA professional with all relevant skills at one place.

This book should help in making sure that you get the basic core concepts, working knowledge and in summary as a survival guide for programming and automation with all required skills. The goal is not to aim at making you an expert at one skill or entirely on these skills. The focus is on finding the needed skills and at what level is being used for 80% of the time (80-20 rule) in the Java programming and Quality engineering profession.

For the Manual QA engineer, this book helps in understanding Quality concepts, SDLC (Software Development Life Cycle), technical terminology, etc. Also, this helps in moving the engineer from manual testing to automation.

This book also helps junior developers working on Java projects in SDLC, OOPS, Java programming, unit testing and most of the other skills are in common with automation. Also, it gives understanding some of the test frameworks and terminologies in the test development.

One of the key things to observe is that it is not difficult to learn anything but to see what is relevant and why to learn. Learn the things most important to you in the work or life. If someone asserts on what you know, then it builds more confidence in the current profession or newly pursuing one.

At a high level, this book starts with the areas such as processes (SDLC/Quality/OOPS), data (XML/XPath/JSON), code repositories (SVN/GIT), and operating systems (Unix/Linux). Then enters into programming (Java&JDBC), and later build frameworks/tools (ANT/Maven). After that it covers more on unit testing frameworks/tools (JUnit/TestNG) and continuous integration tools (Jenkins/Hudson). Next, it includes the functional testing of web applications and web services with frameworks/tools (TestNG/Selenium/SoapUI) covered. Finally, included sample code (for Java/Selenium) to reference offline, and hands-on/getting started in the job.

There are about 500+ questions and answers to ease on understanding the concepts and for review purpose.

The below are 16 core skills covered in this book.

1. **Software Development Life Cycle (SDLC):** The SDLC is to understand the core methodology of the software development process. This process is one of the essential concepts for Dev/QA individual and the teams to collaborate effectively in the creation of the software products or IT services.
2. **Software Quality Concepts:** The quality concepts are the fundamental knowledge on the testing part of the SDLC process that must be possessed by every QA and Dev team for the delivery of a high-quality software product.
3. **OOPS:** The majority of the software products development has been done with Object-oriented languages such as Java, C++, and other OO high-level languages. OOP concept is very much essential for any Java/C++ programmer to maintain or develop new software in Java.

4. XML: Most of the existing dynamic content, e-commerce sites and many software configurations have been done using XML file format. It is essential to understand this universal XML file format to survive in the Software dev and QA profession.
5. XPath: As long XML is there, the XPath also plays a key role in performing the navigation, extraction of XML, and transformation of the content to various other forms (HTML/text). So better to acquire this skill by developers and QA professionals.
6. JSON: The JSON file format is a newly evolving and modern universal file format. JSON format is being used widely in REST WS, Internet/Web applications, Social sites, etc. Having hands-on with JSON files is crucial for all developers and QA automation professionals.
7. SCM/SCCS(SVN/GIT): The source code is one of the major intellectual property of any software organization. It is essential to collaborate with the developing teams using a source code control system such as SVN or GIT. It is important to understand the SCM concepts and working knowledge on SVN and recently popular GIT software for both developers and QA automation engineers.
8. Unix/Linux: In Today's World, most of the enterprise software, website hosting, and any Internet-hosted software or applications are on the Linux/Unix platform. Having hands-on experience with this platform is fundamental for everyone.
9. Java & JDBC: Most of the existing or new, whether small or larger software projects are still written in Java language. The Java should be at the heart of every software professional and thereby essential to have this language to survive in the software development and QA automation profession.

10. ANT: The ANT is a build framework. Once it was popular and still is being used as Java build framework for both product code and test code in most of bigger software organizations. Having an understanding and hands-on experience for both developers and QA is good.
11. Maven: The Maven is a build framework used everywhere in the recent Java projects. It addresses some of the dependency resolution problems with Java project build process that was a burden with ANT. Maven makes the build process simple by using conventions. Thereby Maven is a significant survival skill for both developers and QA automation engineers.
12. JUnit: The unit testing should be one of core tasks for any Software developer. The JUnit is an old but simple and popular framework for Java projects. This test framework is still being used in many organizations and lot of IDEs support the automatic test code generation and execution. Having JUnit experience is an essential skill for developers.
13. TestNG: The TestNG is one of the common framework used across the unit testing and functional testing. TestNG is being used in most of the recent projects and specifically in Selenium based test projects. Having hands-on with this skill is essential for both developers and QA automation professionals.
14. Jenkins/Hudson (CI): In the Continuous Integration (CI) category, Jenkins/Hudson play a vital role with automated jobs for creating the product builds/bundles and also end-to-end regression test cycles. The Hudson/Jenkins functionality is extendable with plugins, and many of the plugins are already available. It is a widely used tool. Thereby having this skill by both developers and QA engineers gives a real strength.
15. Web Applications Testing – Selenium: The Selenium is a browser-based website testing automation framework. Many of the organizations need this skill to create automated test suites

going through the browser. This framework acts as a remote to the browser. The selenium automation hands-on experience is a must for QA automation engineers.

16. Web Services (SOAP&REST API) Testing and SoapUI: The Web Services are loosely coupled and distributed technology that is widely used. The WS are being used in Service Oriented Architecture (using SOAP-based web services), or with the Internet resources addressed as RESTful WS or REST API or REST endpoints. Nowadays, the REST is very popular and is everywhere. It is essential to understand and test these REST APIs. This skill is a must for both developers and QA engineers to survive in the development of current software and IT services.

Hands-on/Getting started: Once joined the job, many times it will be a confusion where and how to get started the day-to-day work. This section helps to understand those basic expectations from your new company/team and walks through the flow.

Java samples code: Learning by an example is a quick way to learn new things. The Java sample code helps developers and QA engineers to get comfortable in Java programming.

Selenium samples code: The Selenium sample code helps QA engineers in learning the browser-based website test automation using Selenium. Having examples is a good way to quickly getting into coding.

**How to read this book?**

For beginners and junior software engineers, it is recommended to go from Skill#1 to Skill#16 so that each learned concept or skill would be used in the following skills.

For senior engineers and leaders, just jump to any skill to review the concepts. There is no direct link between the skill chapters rather the previous skills will be used in the later skills.

Overall, these skills review should get you at the level about five years of experience in the Software quality and Java automation profession.

Finally, this book is an attempt to share and build confidence in you with core skills of Software quality and Java automation.

All the best for your journey!

## **Skill#1. Software Development Life Cycle (SDLC)**

The SDLC is to understand the core methodology of the software development process. This process is one of the essential concept for Dev/QA individual and the teams to collaborate effectively in the creation of the software products or IT services.

### **1. What is Software?**

- The Software or Computer Software is a set of programs and data that process information and controls underlying hardware. It includes programs, libraries, and non-executable data.
- Some of the examples include the following.
  - Operating Systems such as Windows, MacOS, Linux, Solaris, etc.
  - Browser software such as Firefox, Chrome, Internet Explorer, Safari, etc.
  - Office software such as Microsoft Office, Star Office, etc.
  - IDE software such as Eclipse, NetBeans, IntelliJ, etc.
  - Any other programs that operate on devices and data.
- The main purpose of software is to have the flexibility to control a general purpose solution, business solution or devices such as mobile, desktop or large computers.
- The software is everywhere in the Today's evolving technologies. Understanding and gaining of software skills would help in professional advanced technology careers.

### **2. What is SDLC? Why do you need such models? What are different models?**

- SDLC stands for Software Development Life Cycle. It is the process of developing any software. There are many processes around from a traditional "Waterfall model" to the latest "Agile" processes. There are many types in Agile itself such as XP (eXtreme Programming), Scrum, TDD

## 1. Software Development Life Cycle (SDLC)

(Test Driven Development), etc.

**3. How is SDLC related to Testing?**

- Testing is the part of the SDLC and understanding it will help effective testing of a software product. The testing is the process that comes as verification phase in Waterfall model and in testing part of agile Scrum.

**4. What is Waterfall model? What are different phases involved?**

- The Waterfall model (as the name indicates) is like water flowing from upper ground to next lower ground where development (along with information) passes through different phases as one after the other. It involves five different phases in an order and goes to next phase only after completing the prior phase.
- The phases are
  - i) Inception
  - ii) Design
  - iii) Implementation
  - iv) Verification
  - v) Maintenance.
- It is hard to inject the customer requirements (either with new requirements or changes to old requirements) while development is in progress. It usually takes about six months to 1-5 years to complete such projects and most of the time used in old stable/matured projects and also in larger projects.

**5. What is Agile Scrum process? Can you describe at a high level?**

- Agile Scrum is an iterative process of developing software with the goal of getting incremental intermediate product delivery. In the agile scrum process, there will be a product backlog, Sprint backlog and short cycles of the development process (typically seven days length of a sprint with the goal

## 1. Software Development Life Cycle (SDLC)

of delivering product in 4-5 weeks) and release. Sprints are repeated until entire product backlog is completed. Every release is towards an improvement or finish towards the end product. Customer requirements can be injected into product backlog and will be picked up by next sprint (through sprint backlog). There will be daily stand-up meetings of approx. 15 mins in length on the scrum sync' up.

6. **When does the testing activity gets started in each development model?**
  - In the waterfall model, testing is started only in the verification phase after completing the implementation phase.
  - In the agile model, testing is done in every sprint or scrum cycle. Testing is started since beginning with the features in that sprint and work closely with the dev on the completed features within that sprint.
  
7. **What is meant by standup meetings in the scrum process? What are the three questions (typically) you need to answer in Agile Scrum process?**
  - The following three questions are to be answered and should be getting ready for the meeting.
    - i. What have you done since Yesterday?
    - ii. What are you going to do Today?
    - iii. What are the blocking issues or stumbling blocks (from you)?
  
8. **Who is Scrum Master? What is his/her role and responsibilities?**
  - The scrum master is like project manager in the agile scrum software development process. Scrum master handles the sprint schedule, product backlog, sprint backlog and conducts the daily standup meetings with all stakeholders.

## 1. Software Development Life Cycle (SDLC)

**9. Are you aware of any Agile tools?**

- Rally software is one of the tools (<http://www.rallydev.com>) to used for tracking the Agile process.
- Atlassian's JIRA also being used to track the tasks in a sprint and have the epics with JIRA dashboards.
- Finally, simple wikis/spreadsheets can also be used as a tracking tool.

**10. What is a sprint? What is a typical duration for a sprint?**

- The sprint is the short cycles of software development lifecycle, and typically it is a weekly or 2 or 3 or 4 weeks period to code, verify, document a set of user stories or fix and verify the bugs.

**11. What is meant by product backlog?**

- The product backlog is the list of tasks and requirements for a product under development. The backlog gives the full list of prioritized requirements or tasks for the product at any given time. If any new or changes in the old requirements, those will be updated in this product backlog and prioritized.

**12. What is a sprint backlog?**

- The sprint backlog is the list of prioritized tasks and requirements to be completed in a particular sprint cycle. It is a subset of top requirements or tasks picked from the product backlog.

**13. Where do you add the changed requirements? Whether in a sprint backlog or product backlog?**

- The changes in the requirements would be added to the product backlog and will not be added to the current sprint backlog. These changes would be picked up in Next sprint cycle (through sprint backlog) from the product backlog.

## 1. Software Development Life Cycle (SDLC)

**14. Can you change a sprint backlog during a sprint?**

- No. Not in the current sprint but the next sprint cycle would catch up the sprint backlog derived from product backlog with those changes.

**15. What is a use case?**

- Use case is the sequence/flow of actions and events performed by an actor to achieve a business goal. The actor could be an end-user using the system or the system or device itself interacting with some other system. There will be many use cases for a given product under development.

**16. What is a user story?**

- The user story, the term used in the agile SDLC process and is a simple description of a software requirement in the end user's perspective and his/her business job function. It should be scoped for an iteration and should provide a business value or add the value to the overall product once implemented.

**17. What is Test Driven Development(TDD) process?**

- The Test-driven development is an agile software development process under extreme programming paradigm. In the TDD, the developer first writes an initial, which is a failed automated unit test as per user story and repeatedly modified the code until the test passed. I don't think TDD is that popular or widely used now.

**18. What are the advantages and disadvantages of waterfall model?**

- Waterfall model advantages:
  - It can give better project schedule estimation and planning.
  - It provides the re-use of Resources because of sequential phases.
  - Good for large projects with clear planning, design,

## 1. Software Development Life Cycle (SDLC)

architecture before actual implementation. The team is not under time pressure (in comparison with Agile) as there can be sufficient time buffer for planned tasks. Typical project release duration is higher and is between 6 months to 1-5 years compared to Agile monthly project releases.

- Waterfall model disadvantages:
  - Customer requirements can't be accommodated in the middle of the project.
  - This takes more time and incurs more cost.
  - There is no incremental product delivery. The final project is available only after the final phase, which is a longer in duration.
  - It is difficult to predict if the project is doomed to failure until the project completion or out of resources.

**19. What are advantages and disadvantages of Agile Scrum process?**

- Scrum process advantages:
  - Adapt to the changes where customer requirements can be honored in the middle of the project development (that is in the next sprint cycle).
  - It helps in doing incremental product delivery.
  - It takes less time comparing with waterfall model.
  - It gives an early indication of the project status and can be stopped quickly or can be transformed to a different project.
- Scrum process disadvantages:
  - It can consume more resources because sprint cycles need to cover all dev/test/doc activities at same time.
  - Typically, it is good for shorter projects with less complex projects (where fewer alternatives/architectures to be considered).
  - It demands high productivity and constant effort be done by the team and thereby stress or time pressure felt by the team.
  - Also it will have more quick sync' up meetings.

**20. What is the typical team size in different SDLC models?**

- A typical number could be viewed as below, and it can very much be based on the matured SDLC process in the

## 1. Software Development Life Cycle (SDLC)

organization.

- Waterfall model >10 team size
- Agile model < 10 team size
- Nowadays, more team size projects (like more than 50) also doing well with Agile Scrum process. It is becoming a de-facto process for most of the startups and new development teams in larger organizations.
- Also, it could be some kind of mixed as waterfall and agile together in the development process.

**21. Which SDLC model suits for rapidly changing requirements?**

- Agile SDLC process (say scrum) is most suitable because the changes can be uptaken in the next sprint cycle. Thereby customer requirements can be considered during the product development time itself rather than waiting till the product release.

**22. Which model suits for incremental product delivery?**

- Agile SDLC process (say scrum) because it is an iterative process contributing to the overall product in each sprint cycle. Thereby product progress can be seen as sprints in progress and can be demo-ed or given to the sales and trial customers before the final release of the product.

**23. What is meant by refactoring? Where do you typically do?**

- Refactoring is the general software engineering process in coding and is to make code changes in such a way that it can improve readability, structure and optimization while keeping the use case/behavior intact.
- This refactoring process is very typical in the software programming and also especially in the agile process as it is iterative because getting the perfect code is not possible with all optimizations at a time.

## 1. Software Development Life Cycle (SDLC)

**24. What is meant by alpha and beta products? Who are the consumers? What is the main objective?**

- The Alpha product is the very initial product available after code completion but exists with many bugs (all features might not work as expected).
- The Beta product is the next milestone product after Alpha release with more bugs fixed but some bugs can still exist in the end-user functionality and also still the quality improvement should be done.
- Few Alpha customers would be selected for Alpha testing, which is done at the customer site to get the very initial feedback on the features understanding.
- Few Beta customers would be selected for Beta testing, which is done at the customer site to get the feedback so that those can be fixed before the release of the RTM (Release to Manufacture) product.
- The main objective of these intermediate releases is to make sure that the customer acceptance is in-line with the product development and make corrections with their feedback.

**25. What is a PRD (Product Requirements Document) or BRD (Business Requirements Document)? What it contains and who owns this?**

- PRD is the product requirements document with the list of customer features gathered and the documented by the product management (PM) team. The PMs interact with customers on the product requirements and acceptance criteria later.
- BRD is the Business requirements document, which is similar to PRD but used in the context of IT projects or consumer projects (like retail or financial) than Product research projects.

**26. What is a functional specification? What it contains and who owns this?**

- The Functional specification document is a detailed

## 1. Software Development Life Cycle (SDLC)

requirements analysis document created by the senior development team for each feature listed in the PRD/BRD.

- This document contains a detailed specification of features and owned by the Dev team.

**27. What is meant by soft code freeze (SCF)? When does it happen in the SDLC process?**

- Soft code freeze or Feature complete is a considerable milestone that occurs after completing the implementation of all the features during the development process.
- After SCF, no more features would be added, and instead, only critical, high and medium bugs would be fixed.
- A separate branch or tagging of source code repository would be created for this SCF milestone and keep using this branch until release is completed.
- QA team would continue to test and file bugs as Dev team continue to fix the bugs (but no new feature).

**28. What is meant by hard code freeze (HCF)? When does it happen in the development process?**

- Hard code freeze is a milestone and happens after SCF milestone when most of the critical/high/medium bugs fixed and ready for final QA test cycle to find any showstopper bugs (that can stop the release).
- After HCF, only show stopper (customer impact is very high) bugs would be fixed.
- Please note that these show stopper bug fixes are allowed only with product management approvals.

**29. What is product release candidate (RC)? What is a release?**

- Release Candidate (RC) is the final milestone in the product release cycle timeline.
- RC is created after HCF is completed and QA would do the final test cycle before RTM release.
- RC would be released as RTM product when no show stopper bugs found on this final build.

## 1. Software Development Life Cycle (SDLC)

- The product release is nothing but bits are ready to handover for manufacturing (i.e., to cut the CD/DVD) or download from the internet by the customers.
- For the release bits testing, QA and dev teams would do the final sanity testing like checking binary sizes, documentation links, basic installation, downloads check, etc., so that customers don't see any basic issues.

**30. What is release notes? Who drives this?**

- Release notes is a document that is made available to the customer along with product release where customer is aware of such as basic version, features summary, critical known issues with workarounds, documentation references etc.
- Release notes document is integrated into the product bundle and also downloadable from the internet. Any updates in future would be updated in the online documentation.
- Release Engineering (RE) team drives the release notes but contributed and reviewed by all the project stakeholders.

**31. What is a release criteria? Who sets this up and who has to follow?**

- The Release criteria are the minimum checklist or goals to be achieved to release the product. This effort is driven by RE team, and all the project stakeholders would contribute to making this release to happen.
- Below is a sample release criteria/checklist and can be taken as a template if none is available.
  - Bugs criteria
    - No P1, P2, P3 open bugs
  - Quality criteria
    - 100% feature coverage and test development
    - 100% test execution
    - 95-100% pass rate
    - >90% automation for regressions

## 1. Software Development Life Cycle (SDLC)

- Stress/Load/Concurrency criteria
  - 100 users/sec
- Performance criteria
  - Minimum throughput numbers related to the product.
  - Comparison to other competitive products
- Code coverage
  - >90% class level
  - >50% method level
  - >40% instruction/statement level
- Internationalization (I18N) & Localization (L10N)
  - Internationalization support
  - Specific languages to translate
  - NOTE: I18N is short form as there are 18 characters between I and N in the “InternationalizationN”. Similarly, L10N is short form as there are 10 characters between L and N in the “LocalizationN”.
- Documentation (books/online)
  - Release notes
  - User Manuals
    - Installation Guide, Admin Guide, Developer Guide, Troubleshooting guide, etc.
- License and license text
- Support/Maintenance plan

**32. What is meant by a release showstopper? When does it happen?**

- The showstopper is a bug that has a visible & high impact to the customer and must be fixed before the product release. If the fixes are not possible within the release time, then corresponding bugs and workarounds should be added to release notes.
- This kind of bug fixes should go through focused reviews and approvals to minimize the regressions (if any) in this

## 1. Software Development Life Cycle (SDLC)

final release of the product.

- The showstopper happens after the HCF and RC milestones or during the final sprint of the development process.
- The regressions tests should be selected based on the bug fixes instead of running all the tests because of the need for quick turnaround time.

**33. Who are all project stakeholders? What are those teams?**

- In general, below are the Project stakeholders in a product development organization.
  - Product Manager
    - Acts as a bridge between product and customers
    - Responsible to create PRD (Product Requirement Document) or BRD (Business Requirement Document).
  - Project Manager/Dev Manager/Dev Director
    - Owner of the product execution (who creates and executes the Project Plan)
    - Acts as a driver for successful project release.
  - Program Manager
    - Co-ordinates with all stakeholders during the project execution and helps project driver.
    - Conducts release meetings
  - Architects
    - Responsible for performing product architecture considering the big and strategic goals like scaling, availability, performance, security, technology, etc.
    - Creates architectural diagrams and documents.
    - Work closely with the design/development teams for making the implementation successful.
  - Designers
    - Responsible for the product design and work closely with Architects and developers.
    - Creates FSD/FS (Functional Specification Document)

## 1. Software Development Life Cycle (SDLC)

- Please note that sometimes Architects can act as designers or senior developers can be designers.
  - Developers
    - Responsible for implementation (coding) as per FS.
    - Do bug fixes
    - Conducts unit testing
  - QA/QE
    - Responsible for Testing ( tasks like write test code, execute tests, file bugs, create Test Plan and Test Specification).
    - Review documents - release notes, manuals/guides
  - Doc writers
    - Responsible for all sorts of documentation like user guides/manuals/help content.
  - Release Engineers (RE)/Devops
    - Responsible for doing the integration of modules and creating the builds.
    - The DevOps would host and monitor the stage and production sites.
  - Support Engineers/Front-end support (customer facing)
    - This team would triage the customer issue and pass on to backend support if needed.
  - Support Engineers/Backend support
    - They would triage and fix the customer issues escalated by front-end support engineers.
- 34. What is a test cycle at product level?**
- A test cycle is the duration of time, where all the test activities can be completed on a given product binary/build.
  - A typical cycle could be varied from 1 week to 4+ weeks.
- 35. How does the QA team track the weekly status or progress?**
- Typically, QA team representative (Director or Manager or Lead) would prepare the answers to the following questions

## 1. Software Development Life Cycle (SDLC)

while tracking the status summary.

- How many open bugs and list of these bugs?
- How many blocking bugs (P1)
- How many tests are failing (count and %)? Example: 30% failing
- How much test development is pending? Example: 40% tests
- When will be the test automation is going to be completed?
- What are the bugs inflow (new bugs came since last reported time) and outflow (fixed bugs since last reported time)? (Example: Weekly)
- What are the total number of bugs fixed and bugs opened? Bugs trend.
- How many bugs yet to be verified?
  - Please note that the bugs verification & closing usually done by the bug filer.
- Get all the answers in a wiki or document and update in the email for the weekly status report and send to stakeholders.

**36. What is Go/No-Go for product release?**

- “Go/No-Go” is the decision voting conveyed/sent after their respective release criteria met by each project stakeholder participated in the release meeting.
- Once all votes of GO received, then RE would take it as the GO for release. Any No-Go voting should be discussed and mitigated or fixed before release.
- Once it is a GO, then it is a celebration time for the entire project team;)

← End: Skill#1. Software Development Life Cycle (SDLC) ←

Good!

Keep going and never give up!!

Please re-read again for more clarity and feel free to contact for help!!!

→ Start: Skill#2. Software Quality Concepts →

End of the book preview.

## Author's biography

Jagadesh Babu Munta is working as a Consulting Member of Technical Staff with Oracle America Inc. He has been with Oracle and Sun Microsystems together for over 16 years (since June 2000). Jagadesh has overall 20+ years of Software development and quality/testing experience. Jagadesh's experience has been filled recently with Cloud PaaS services, Multi-Tenancy, Security and Penetration testing. In the



past, extensively worked on Java EE servers like SailFin/GlassFish/Sun Java System/iPlanet/Netscape Application Servers testing. Jagadesh has gained extensive expertise in software automation, designing frameworks, writing tools, scripts, creating tests, writing specs/plans,

etc. Jagadesh is interested in developing and testing complex software useful to up level the humanity.

Jagadesh Munta holds M.S. in Software Engineering from San Jose State University, California, USA; B.Tech. in Computer Science and Engineering from J.N.T.U., Hyderabad, India; Special Diploma in Electronics with Specialization in Computer Engineering, G.I.O.E, Secunderabad, India.

Jagadesh Munta was born at Chejerla, Nellore, AP. India and lives with family in Fremont, California, USA.

## Do you also think this book can help?

Please buy a full book at any of the below sites.  
(Search for “Software Quality and Java Automation” in amazon.com or in google.com). Visit <http://www.everydayon.com/>

### Buy Paperback in USA:

1. Amazon: <http://www.amazon.com/Software-Quality-Automation-Engineer-Survival/dp/1635720001/>
2. Create Space: <https://www.createspace.com/6647429>
3. eBay: <https://t.co/7EUT8rw171>

### Buy Digital:

4. Amazon kindle (mobi) : <http://www.amazon.com/Software-Quality-Automation-Engineer-Survival/dp/1635720001/>
5. Apple's iBookstore  
(ePub): <http://itunes.apple.com/us/book/id1180399694>
6. Lulu (ePub): <http://www.lulu.com/content/e-book/software-quality-and-java-automation-engineer-survival-guide-basic-concepts-self-review-interview-preparation-%28500+-questions-answers%29/19975685>

### Buy Paperback in India:

<http://www.everydayon.in>



THANK YOU!