



99 THINGS YOU CAN DO TO BECOME A BETTER TESTER

Software Testing **Club**
a community for software testers


THE TESTING PLANET



Introduction

Over at the Software Testing Club we asked our lovely community of software testers whether we would be able to come up with a list of [99 Things Testers Can Do To Become Better Testers](#).

This is the result. The community has answered. We hope it will inspire you to take some action to become a better tester. Enjoy!

Please check out our sponsors, they are awesome for helping make this ebook happen!



EXPLORATORY TESTING
AUTOMATION

THE
NET-A-PORTER
GROUP

RUBY SELENIUM
JAVA iOS

MAKE YOUR MARK

*BECOME AN INTEGRAL PART
OF OUR LEADING-EDGE TEAM*

THE NET-A-PORTER GROUP is so much more than the world's premier luxury fashion destination. We are an innovative global online and mobile business at the forefront of the digital domain. A passion for technology is at the heart of our success and we continue to invest in the people who share our drive.

Working in a highly collaborative, agile environment, our standalone test team boasts over 40 of the industry's best testers and developers in test. If you are highly skilled, creative and focused on continuous improvement, we want you to join us.

**VIEW THE NET-A-PORTER GROUP
JOB OPPORTUNITIES NOW ►**



WWW.MINISTRYOFTESTING.COM



Dreaming of an on-demand software testing workforce ?

It's real, its here: passbrains crowdsourced knowledge and software testing power. For your mobile, web and enterprise apps. **On-demand, scalable, efficient.**

www.passbrains.com • crowd testing • crowd consulting • crowd recruitment



devZing.com
**Hosted Bug
 Tracking & Test
 Case Management**

**ONLINE TEST CASE
 MANAGEMENT TOOL**

Work collaboratively
 online with no user
 limits!

www.TestLodge.com

"Simplicity is the ultimate sophistication"
 (Leonardo Da-Vinci)

Test Management Tool

www.testuff.com



WWW.MINISTRYOFTESTING.COM



INTEGRATED TEST MANAGEMENT



Rational software



enterprisetester.com/ministryoftesting



XCEPTANCE
Software Technologies

The Software Testing Experts.
www.xceptance.com

House of Test

James Bach

Michael Bolton

MIND THE CONTEXT

PrettyGoodTesting



WWW.MINISTRYOFTESTING.COM

1. Get to know your customers - Chris George
2. Work first line support for a while - Chris George
3. Never stop learning - Chris George
4. ...and recognize that you will never know everything - Amy Phillips
5. Learn when to automate and when not to - Chris George
6. Question the way you work every day - Kinofrost
7. ... and have the courage to share your findings with whoever is in charge as well as fight for key changes - Erick Brickarp
8. Pair with developers. - Gareth Waterhouse
9. Test as early as possible - Gareth Waterhouse
10. Remember it is about people. - Tony Bruce



11. Learn to question. - Tony Bruce
12. Learn to explain. - Tony Bruce
13. Understand the business model and business challenges/context before determining the testing challenges. Its not all clicks and buttons. - Mohan Panguluri
14. Keep your eye on the ball (the end goal) - Kate Paulk
15. Get an understanding of Systems Thinking - Martin Huckle
16. Also get an understanding of mental modelling, the scientific method and design of experiments - Kinofrost
17. Be critical but do not criticise - Kim Knup
18. Create a MindMap - Rosie Sherry



19. Learn from other Tester mistakes - Mantas
20. True, but learn from your mistakes first :-) - Mauri Edo
21. Attend or even better speak at Software Testing and Programming conferences (there are free/low cost one that provide great value) - Stephan Kämper
22. Take the Association for Software Testing "Black Box Software Testing" course(s) - Stephan Kämper
23. Follow other testers on Twitter - Stephan Kämper
24. Addition to Stephan's suggestion: Follow testers on Twitter is a good start but make sure you don't stop there. Twitter is a great way to share ideas and experiences, get feedback and practice your debating skills not to mention getting in contact with other passionate testers. - Erik Brickarp



25. For non-native English speakers: Improve your English. For native English speakers: Learn another language. - Stephan Kämper
26. Teach testing. - Stephan Kämper
27. Learn to program, some lines can help so much - Stephan Kämper
28. Start a test blog - Erick Brickarp
29. And comment on others - Amy Phillips
30. Read Articles, Blogs, Forum posts. - Haplerinko
31. Join an open source project you like as tester. - Erick Brickarp
32. Is "Join Software Testing Club!" taken? I couldn't see it... or maybe it is too obvious :-)) - Geir Gulbrandsen



33. Listen to what your client has to say. This is, by far, the most important (and underrated) Testing skill, in my humble opinion. - Marcelo Cordeiro Leite
34. +1 for listening. Being really, really good at listening helps you not only to understand your client, but to pick up on areas where the team is confused or unsure about what we're building - a slight hesitation in speech because the dev keeps getting two different entities muddled up, perhaps.
35. Read the ISTQB syllabus from start to end - then use it as a map of the box you need to be thinking outside of! - Geekonomicon
36. Do not avoid technical discussions/information - Boipelo Mawasha
37. Every year or 2 - Refresh on basics - Read ISTQB Syllabus, TheTestEye SW Quality Characteristics, BBST material, etc. again. (Every time you have different perspective thus gain different insights from same material) - Halperinko



- 38. Always use the best methodology invented - Common Sense. Then choose your testing methodology... - Gil Bloom
- 39. Build a good relationship with the developers. - David Wardlaw
- 40. Resist the temptation to go after bogus certifications. Take the risk of thinking for yourself. I've got 99 ideas on how to become a better tester, but certification ain't one. - Johan Jonasson
- 41. Learn how to use testing techniques such as boundary value analysis, equivalence partitioning etc - Amy Phillips
- 42. Explore your ideas through blog posts, discussions, or by speaking at events. Listen to others opinions and use to broaden your own - Amy Phillips
- 43. Read the release notes for products you use (or even ones you don't use), many list the bugs they've fixed. Test whether the same bugs exist in your product. - Amy Phillips



- 44. Learn from bugs that end up in production. Try to work out how your testing missed them. - Amy Phillips
- 45. Listen. And then form your own opinion. - Amy Phillips
- 46. Keep your mind open to new techniques and tools - Graham Perry
- 47. Come up with three ways that your boss, your co-worker, and your trainee can become better at their jobs ... then apply it to yourself. - Jeff Lucas
- 48. Read as much as you can about testing. Then read as much as you can about things outside of testing; product design, software development, psychology, anthropology etc - Amy Phillips
- 49. When you analyse a system, don't forget the parts of it that are not made of code, but living, breathing, fickle, forgetful, quarrelsome, cooperative, adaptable, lovely human beings. - Anna Baik



- 50. Explore the system!! - Vishu Udayan
- 51. Look at Competitor products and benchmark the differences - Sandeep Maher
- 52. Trust no one! In sense that what people say is not always what they REALLY mean. So the more you conversate, the more you trust. But your trust level is always less than 95% - Oleksii Burdin
- 53. Ask yourself (and answer) the fundamental questions like "why do we need testers?", "what is good testing?", "what should I document, why and for who?"... The greatest value in questions like these is not the answer but the thought process of getting to an answer. - Erik Brickarp
- 54. Understand what your customer wants, what he needs, communicate the needs, agree with needs and deliver both of them. -Teemu Vesala



- 55. But remember that people don't always need what they think they need. Be prepared to think around the problem to deliver a solution to the problem, not just what was explicitly asked for. - Joseph Brannan
- 56. If it doesn't exist: Start a community of practice or other forum there you and colleges can talk testing, share ideas/experiences and improve as testers together. - Erik Brickarp
- 57. Utilise your courage (and display it) and stand up for what you believe in. Don't bend on your morals and ethics. Test the best way you know how, despite being told otherwise. Before doing this... make sure you have mortgage insurance. ;o) - David Greenlees
- 58. Embrace quality as a lifestyle, not only during work hours. - George Motoc
- 59. Breathe as a tester, Live as a tester, Be a tester. Be a tester in all areas of your life whatever it is at work, private life, relationships, hobbies or



others, 24/7, 365/year (I give you 1 day rest on leap year :)). - Gabrielle Klein

- 60. Never stop trying to become a better tester - Mauri Edo
- 61. Try to provide the solutions as well as finding the problems. - Geekonomicon
- 62. Think about as many different perspectives as possible!! - Dan Ashby
- 63. When reporting, try not to be seen as the enemy; assign blame to systems and applications rather than individuals and stick to the facts. Equally, emphasize the positives, and the efforts made to correct issues; this time ignore the systems and applications and focus on the individuals who got the defects fixed. - Joseph Brannan
- 64. Learn the difference between Severity and Priority - Dan Ashby



- 65. Would add to that and recommend that everyone in the company knows what they mean by severity and priority. If people have different understanding of these words then there's no actual communication. - Kinofrost
- 66. Accept that not all bugs you raise will be prioritised to be fixed - Steven Cross
- 67. Everyone within the project team is responsible for quality - Steven Cross
- 68. Testing should be fun so remain positive and get everyone within the team enthused about the merits of testing - Steven Cross
- 69. Accept that Developers have a different mindset - Steven Cross
- 70. Be brave. You're possibly the only person(people) saying "Are you sure you want to release this now because...?" - Vernon Richards



71. Don't repeat yourself. I learned this from the Pragmatic Programmer although it means something different for testers: don't repeat the same actions, don't follow the same path, the same order. Break your habits. - Philippe Antras
72. Ask experienced testers for feedback/help - Erik Brickarp
73. Flip that, and experienced testers should help less experience testers as much as they can. It will help you and help the other tester. - Gareth Waterhouse
74. Good point and to keep 'em coming: experienced testers can benefit from asking less experienced testers to get a new/fresh view of a problem, especially true when experienced includes very use to the product being tested (in that case as a means to fight bias). - Erik Brickarp
75. Use the box itself to help you. Look at the edges of your box. Create a bigger box, look at what is now in the box and decide what you might



want to follow up. Look at the edges again, can you push any of them out a bit? What would you find if you did? Do quick, cheap, experiments on stuff that seems unlikely to surprise you - because sometimes it will.

- 76. Use another pair of eyes to help - pair with someone, or try to grab someone for a quick debrief if your team doesn't "do" pairing. - Anna Baik
- 77. Leave your ego at home - maybe that amazing bug won't get fixed before go-live. Trust that the person making the decision knows better than you. - Amy Phillips
- 78. When you say, you're a tester, play your position, that position could involve much more than just testing in a start-up. It might even involve giving an opinion on the go/no-go decision. - Matt Archer
- 79. Respect programmers, designers, product owner and others involved.



- 80. Earn respect back by doing a great job and learn how to communicate your results. - Erik Brickarp
- 81. +1 to this, specially do not feed the anger between testers and developers, if you face it at some point in your career.
- 82. Go to non-testing technology events - Rosie Sherry
- 83. Reduce biases & unintentional blindness. - halperinko
- 84. Learn to take effective notes and document your testing in different ways - models, mind maps, sketches and other approaches will all help you gain insights and new perspectives on the system you are testing.
- 85. Build personal development time into your week - 5-10% or approximately half a day a week sharpening your skills by reading, practicing or learning a new skill will pay dividends.



86. See the bigger picture. How does your testing add value to your team, project, organisation?
87. Learn how to use regex.
88. Learn how to use the command line. Shell and batch scripts too.
89. Learn a scripting language. Use it to automate repetitive tasks or processes. Manipulation of text or data files for example.
90. Become an Excel power user. Functions, logic and conditional formatting can all be used as powerful analysis and test tools.
91. Stop following test scripts and think - Stephen Blower
92. Hang out with developers, designers, managers - Rosie Sherry
93. Understand your business and customer needs, not just the requirements - Mike Hendry



94. Understand your domain fully and as importantly your competitors - Stephen Blower
95. Do not try to find errors or bugs - try to find problems and victims. Testing is more than checking. - Thomas Lattner
96. Be prepared, you won't catch all the bugs, but keep trying - Mauri Edo
97. Be prepared, all the bugs you raise won't get fixed - Rosie Sherry
98. Test what matters - Rosie Sherry
99. Question the veracity of 1-98, and their validity in your context - Kinofrost
100. Try always what happens when you violate the rules. So here's the second sentence also.- Teemu Vesala

