HOW TO BECOME A SOFTWARE TESTER

A Complete Guide

ASKTESTER.COM

Table of Contents

It's 2016 now and this year is special to me. I will celebrate my 10-year anniversary being a software tester.

Hurrah!!! (You don't see, but I'm opening champagne :-))

Time flies...

10 years ago, I was a webmaster of a local shipping magazine while I was a last year student in software engineering. My job was to translate articles from English into Vietnamese, posted them to the website or customized the website when needed. While the job was great for a student, it's not the type of work I would love to do (To be honest, I didn't know what I really loved to do either). I was just thinking that I was young and that I need to do something new, something relating to software and something really challenging.

I then browsed job sites and found a company was hiring a software test engineer. "What the heck is software test engineer doing" I wondered. The job description listed some requirements, but I had no ideas what they were. ... but I liked it. So I decided to write a CV and applied for the job.

Luckily, I was called for the interview.

"How was the interview?" you may ask.

Well, the interview went so bad that I wished I hadn't applied for the position. I failed to answer most of the questions on software testing. However, there's one thing I think I did well in the interview was I showed that I was willing to learn new things....and I passed the interview and have become a software test engineer.

Fast forward to today, I've been in software testing long time enough to realize that how lucky I was to be accepted for the job.

Now if you want to get into software testing, you can't just solely rely on luck. Well, luck is part of life, but now you need to prepare and do a better job than I did when I first started because things have changed. Technology has changed, software testing market has changed, competitiveness has changed.

For that reason, I decided to write a guide to help you become a software tester. If you are planning to head into software testing career this year, this post is for you. In this post, I will walk you through things you should do, learn and prepare to come closer to the goal of becoming a tester.

Here we go.

Phase 1: Awareness:

This is one of the most important phases, but most of people skip it. People jump quickly right into the "How" without clearly understand the "Why" and the "What". Understanding the "Why" and "What" will help you realize that if software testing is for you or not and if it's something you want to do and get along. This awareness will also save you from spending time on doing the wrong thing.

I will show you the "How" down the road, but for now, bear with me with this part.

Basically, you have to answer following questions before going further:

1) Why do you want to become a software tester?

Now you want to be a software tester. Great!...but why?

Why would you like to choose software testing as your career?

Why software testing, but not programming, graphic design, IT, or database administrator, etc..

Keep asking more "Why" until you find what really motivates you to move into software testing

It's okay if you don't know the reason why you want to become a tester... and I understand. This is especially true if you are totally new and you have no idea what software testing is, how it looks like, it has a good career path or not, etc.

In order to help you have a better understanding and see if software testing is for you or not, let me do a real quick introduction about software testing and answer some of common software testing questions:

2)What is software testing?

Here is a definition from Wikipedia:
Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. [1] Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects).

Here is a definition from ISTQB Exam Certification:
Software testing is a process of executing a program or application with the intent of finding the software bugs.
It can also be stated as the process of validating and verifying that a software program or application or product:
Meets the business and technical requirements that guided its design and development
Works as expected
Can be implemented with the same characteristic.

... another one from James Bach:
Testing is the process of evaluating a product by learning about it through exploration and experimentation, which includes to some degree: questioning, study, modeling, observation, inference, etc.

Here are some simplified and fun definitions:
Someone cooks something, and I'm the one who tastes to see if this's delicious or not :)
"I get paid to tell people they have made an error"
I'm like a proofreader for a book - I go over everything with a fine tooth comb and make sure that it works the way it should.
I'm like a house inspector, but for software.

I hope you get an idea of what software testing is, but if you don't, it's okay. You will learn more about it later. For now, let's just agree that software testing is to verify that software works as intended and is in good shape before releasing to customers or find defects in the application.

3) What are software tester requirements?

Even though software testing is not an inborn skill, software testing is not for everyone. In other words, it requires some specific skill sets which you hardly do the testing effectively without it. According to State of Testing Report 2105, they are:
+Team player.
+Technical and basic programming skill
+Business oriented, open mindset
+Self-learner
+Knowledge of mobile technology
+Critical thinking

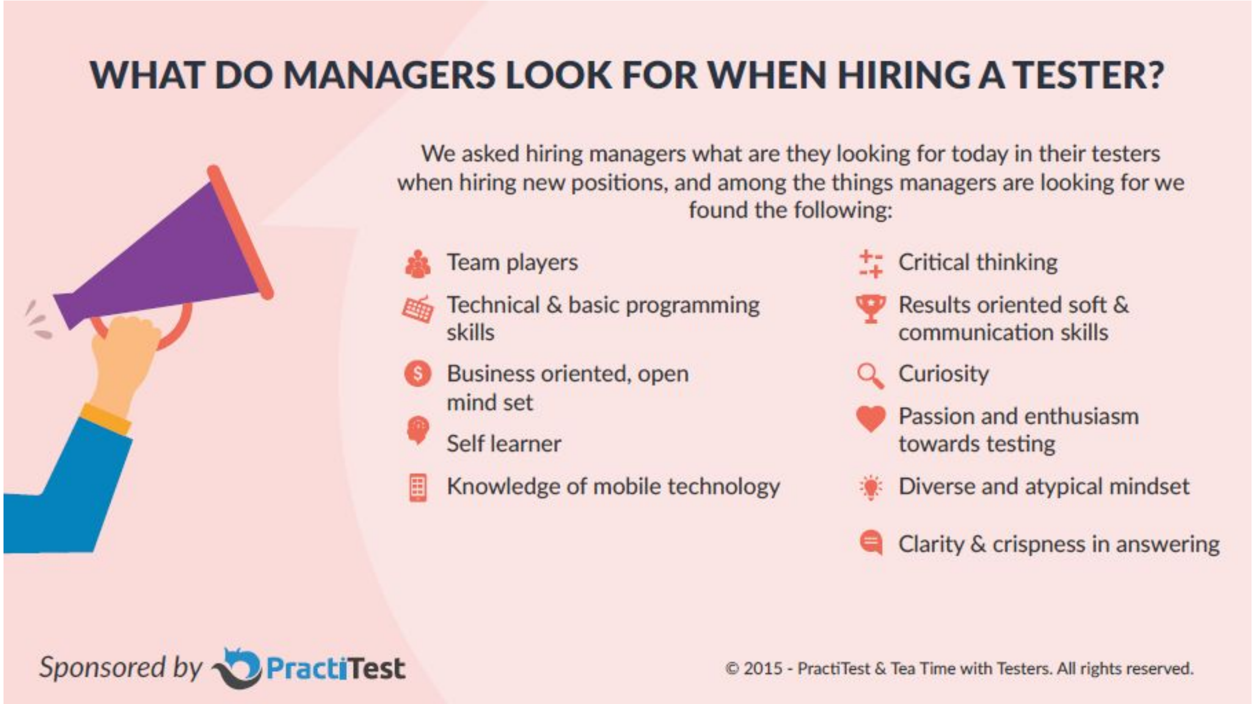
+Result oriented and communication skills

+Curiosity

+Passion and enthusiasm towards testing

+Diverse and atypical mindset

+Clarity and crispness in answering



WHAT DO MANAGERS LOOK FOR WHEN HIRING A TESTER?

We asked hiring managers what are they looking for today in their testers when hiring new positions, and among the things managers are looking for we found the following:

- Team players
- Technical & basic programming skills
- Business oriented, open mind set
- Self learner
- Knowledge of mobile technology
- Critical thinking
- Results oriented soft & communication skills
- Curiosity
- Passion and enthusiasm towards testing
- Diverse and atypical mindset
- Clarity & crispness in answering

Sponsored by PractiTest

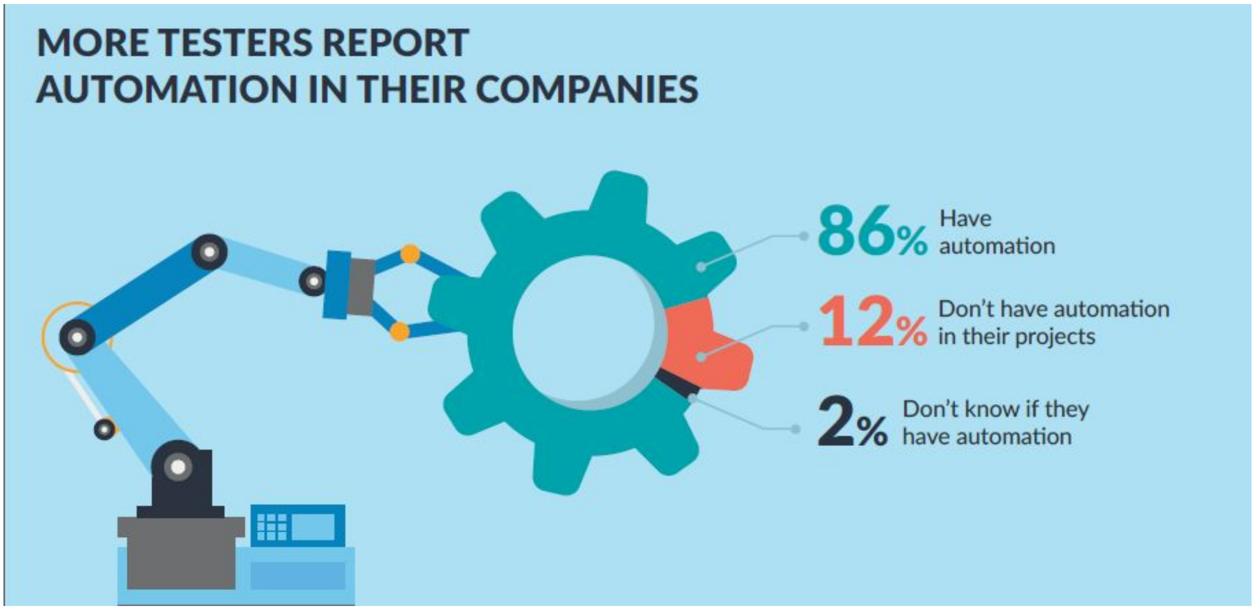
While they are all important skills, I put strong emphasis on "Critical thinking" and "Curiosity".

4) Do you need programming to start your career as software tester?

No, you don't have to be an expert in programming or computer gurus to start in software testing.

I'm myself not a technical guy. Well, I learned software engineering, I can write code but I'm far from expert and to be honest, I haven't touched any lines of code for 5 or 6 years. However, knowing how to program can help you understand how software is built (and how and where the code can be broken). Of course, you can still break the software without knowing how software is built, but if you know how the software is built from the coding's perspective, you will find these bugs much more easily and systematically.
Also, knowing programming skill can really help you if you want to become an automation test engineer where you write scripts to execute the test cases automatically or build small tools to support you to test the software better.

Simply put, knowing how to code is nice-to-have, not a must.



5) Communication skill is a must

You may be not good at programming or technical skill, but you need to be really good at communication skills if you want to become a software tester.

Actually, communication skills are important not just in software testing but also in other fields. However, it becomes especially important in software testing because one of the goals of testing is to explore the system, find problems in the system and communicate these problems to managers/stakeholders. It doesn't make any sense if you find a problem and you keep it yourself or nobody understands what your problem is about.

Actually, there are many skills under the roof of communication skills, however, I would like you to focus on these two skills:

+Bug reporting: Simply put. When you find a problem in the system, you need to report the problem to people in charge accordingly. Your bug report is very important because it not only tells the managers what are the risks in the system with the problem you've found so that they can make decisions, but also tells developers what exact problem is about,

step-by-step how they can reproduce the problem, in what condition the problem occurs so that they can fix the problem.
[Related read: Why Your Bug Report Sucks (and How To Fix It)]
+Asking questions: Actually, software testing is all about asking the right question to expose the problem of the system under test. This is especially true when you test and explore the software as an end user or in case when you have no documents about the system. In such cases, your product becomes a blackbox.

What you would do is to use the product and then ask: what if I click this button? What if I edit the information and then I close the application without saving data? What if I do this...and then do that? What is this feature used for? Why we need it? etc..

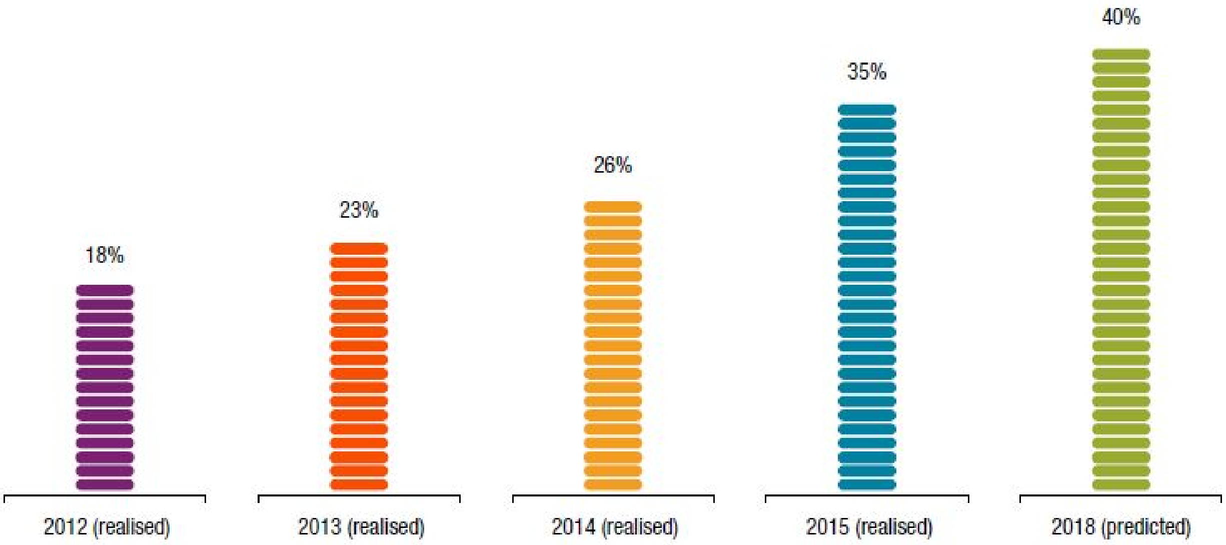
Related read: 6 Tips for Software Testers on Asking Questions

6) What about the career path? Does it have competitive salary?

According to World Quality Report 2015, budget for QA and Testing have risen to an average of 35% of total IT spend.

Budget allocation as percentage of IT spend

FIGURE 10



It means that software testing is still in need and most organizations still have budgets for it. So, don't worry for now, there's enough room for you to grow as far as you want to be.

7) Does software testing require technical skills?

If you've ever heard someone say that software testing is very easy and you don't need any technical skill to do it.

I would say that they are dead wrong.

Like it or not, you still need decent level of technical skill so that you can effectively test the software. However, you don't need to be an expert in programming, networking, database to do software testing. All you need is basic level of those skills when starting out.

Now, here is what I want you to do:

Stop reading for a few seconds and answer yourself the question:

What's your gut feeling about software testing so far?
Is software testing really for you?
Is this something you want to do?
Is this something that interests you?

If your answer is a big YES, keep reading because I will go into more details and guide you how to become a software tester.

(If you don't find software testing interesting so far, stop reading more. You now can open YouTube and find some funny videos to watch :-))

Phase 2: How to get into software testing

Basically, you need to do these 3 things:

First: Learn about it
Second: Earn hands-on experience
Third: Go and find a full-time job

"Hey, but there's nothing special in those steps? " You may ask.
I agree...but here's the thing:

You have to do these basic steps...the right way.

I'll go into the point right now.

1) How to learn software testing?

You can learn software testing by different ways and each way has its own advantage and disadvantage. What you should do is to choose the one that is suitable for you and fits your context.

Below are some options you can choose from:

+Certification in software testing:

I'm quite sure this is one of the first things you think of when you want to learn software testing and start your career as software tester.

Even though there are a few negative feedback about software testing certification, I find software testing certification is a good way to start.

Why? Because it has a lot of advantages:

-You will get basic software testing knowledge
-You (may) earn experience
-You get credit. E.g.: You will become a "certified tester" when you get the certification.

However, there are a few issues with certification and you need to consider:

First, it will cost you money: This is obvious. I haven't found any certification in software testing for free.

Second, certification may be misleading you (and turns out more harm than good). This is a more important point than the cost itself. Certification may mislead you and makes you think that you are now "certified" and what you learned is standard. As a result, you may limit yourself to the standard you learn and stop learning more things or accept new ways of doing software testing.

The whole point about certification I want to make here is that:
If you find certification helpful, go and get it...but don't just stop there.

Here are a few certifications you can get:
International Software Testing Qualifications Board (ISTQB):
ISTQB® has created the world's most successful scheme for certifying software testers.
As of March 2015, ISTQB® has issued almost 400,000 certifications in over 100 countries world-wide, with a growth rate of more than 13,000 certifications per quarter. - As per ISTQB

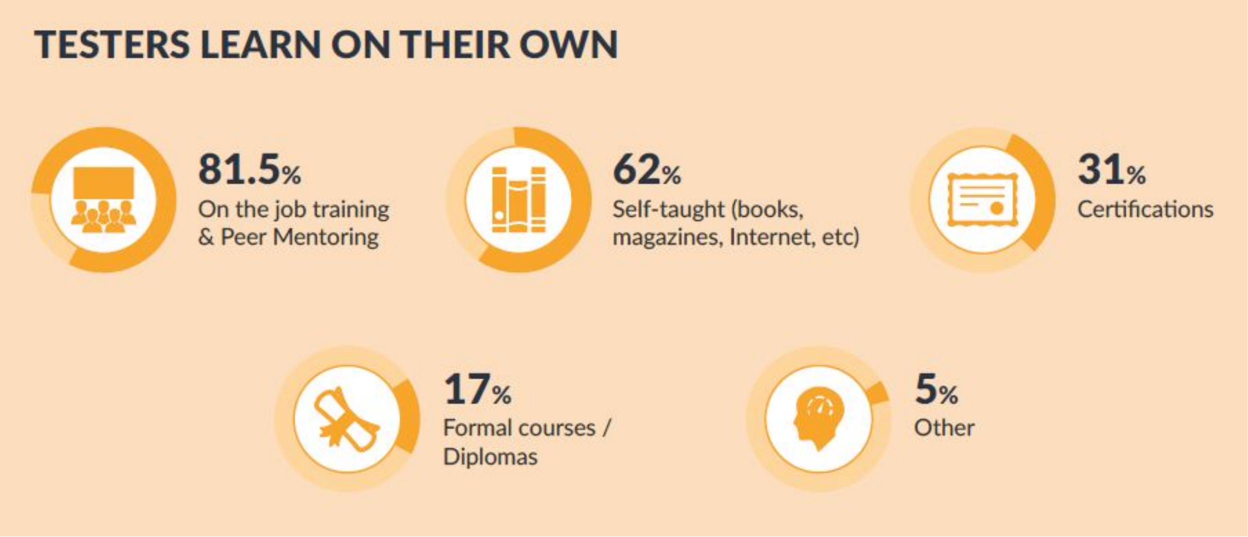
ASTQB - a US version of ISTQB (for those who are in the US).

If you don't want to go with ISTQB, you can try these options which are recommended by many software testing experts in the world:

Blackbox Software Testing (BBST) by Cem Kaner
Rapid Software Testing (RST) by James Bach

+Self-learning:

If you don't want to get certification for some reasons and all you have is time and ready to go by yourself, self-learning is not a bad option at all. 62% testers choose self-learning path:



One of the great things about self-learning is that you will have a chance to explore and learn software testing from different sources. As a result, you have a wider view of software testing. The disadvantage is that it will take you time to learn things out and if you are new you may be distracted and get lost direction. But, if you can manage your time and schedule your learning properly, this is not a problem at all.

If you would like to go on "self-learning" path, check out best online resources to learn software testing. These are resources I myself used to learn software testing and I find they are helpful.

2) Earn hands-on experience:

What's the number#1 complaint from Freshers when they search for jobs? Well, most of the job ads require 2-3 years of experience in software testing even in a junior position

"I'm a beginner...how can I have years of experience" you are frustrated, you are nervous and this type of job ads turns you off completely.

This is a "famous" vicious circle we all know about.

Set aside the job ads ridiculously requires years of experience for a junior position (believe me or not, many employers do not know what they really want in the job ad), you have to agree with me that in any professional jobs, hands-on experience will play the most important role to make or break the game. However, I find most of beginners make this mistake:

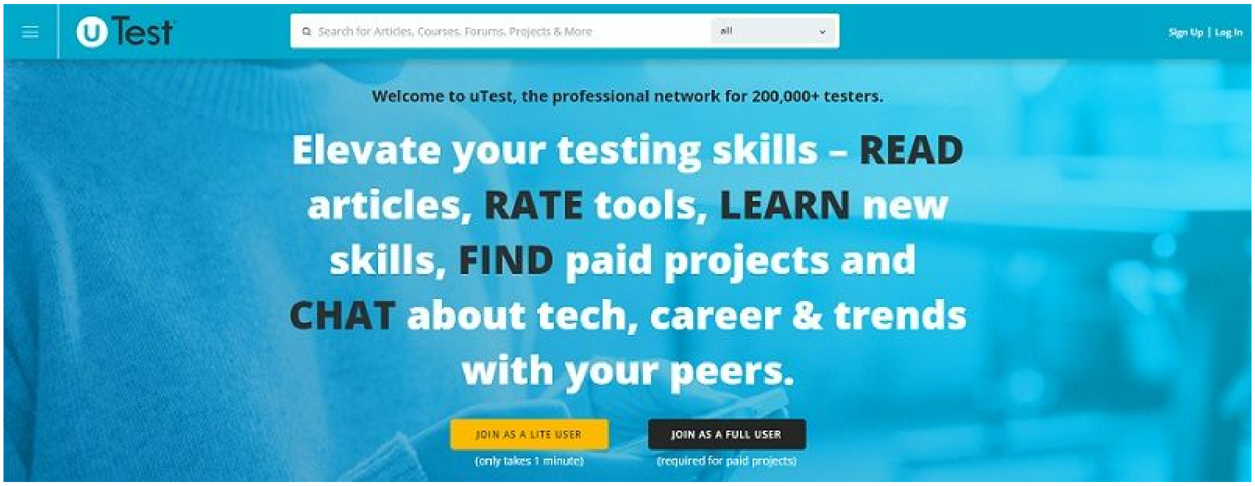
They think they can only get software testing experience only when they have a full-time job...and they set themselves in their traps and that vicious circle.

Don't make that mistake!

Here are a few ideas you can try to earn experience even before you have a full-time job:

+Looking for the opportunity as a freelance software tester
While applying to find a freelancing job is still challenging, but it's much easier in comparison to full-time tester. There are a few trusted sources where you can find freelancing jobs as software tester:

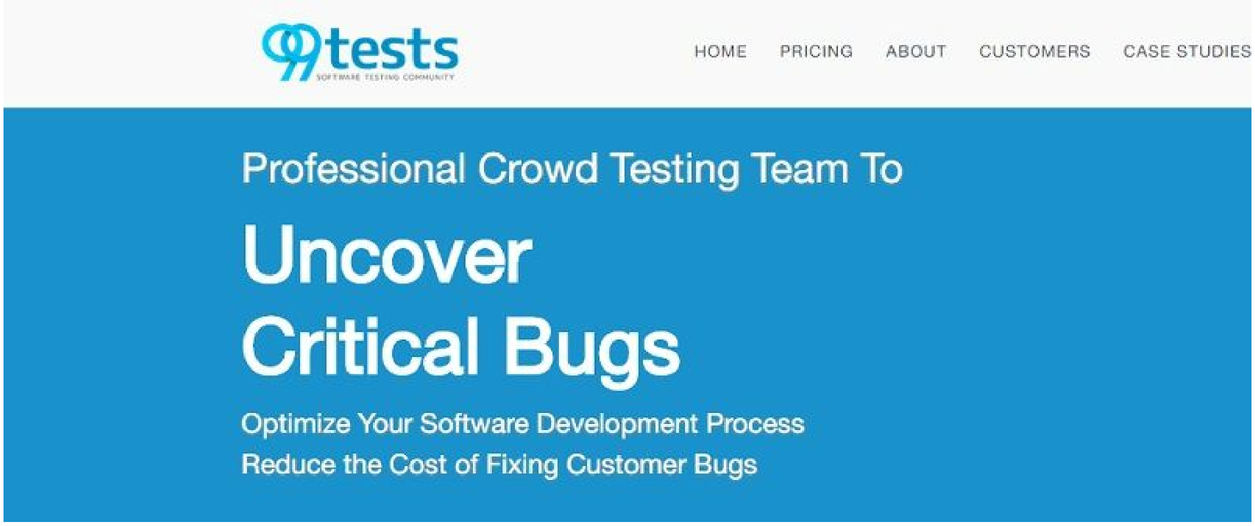
uTest: uTest is considered one of the most crowded communities of software testers in the world. With 200,000 professional testers and many freelancing job opportunities offered, you will surely learn a lot of things and earn experience from there.



Welcome to uTest, the professional network for 200,000+ testers.

Elevate your testing skills – READ articles, RATE tools, LEARN new skills, FIND paid projects and CHAT about tech, career & trends with your peers.

>99Tests: 99Tests is less well-known and based in India, there are many freelance opportunities for you there.



Professional Crowd Testing Team To

Uncover Critical Bugs

Optimize Your Software Development Process
Reduce the Cost of Fixing Customer Bugs

In order to become freelance software tester, what you simply do is to sign-up your account, update your profile with your skills, add your available testing environments such as your devices such as your phone, tables, laptops, PC, your OS such as iOS, Android, Windows, iOS, etc. When there's a project that meets your device requirements, you will be invited for the job and be paid for the work accordingly.

You will need some time and be patient to be invited for your first project. It sounds challenging, but it's worth a try.

+Practice to test your everyday applications

All applications have bugs. You can spend your time on exercising the application you use every day such as Facebook, Twitter, LinkedIn, Google Chrome browser, FireFox and see if you can find bugs.

Next step is to report this bug to customer supports. Once they confirm it's a bug and they will fix, you are going good. Note this bug down and add this achievement in your CV.

If you can find bugs in these applications, congratulations! You have proved that you know how to find bugs too.

(I've recently found a bug on LinkedIn. If I can do that, you can do that too :-))

+Join workshop or local Meetups

Search for your local workshops or Meetups and join them. This is perfect opportunity for you to meet with other testers and practice your software testing skills.

There are some good workshops run by well-known testers in the software testing community you can choose to join:

>Weekend Testing Workshop

>TestBash - Conference, workshop event from MinistryOfTesting

Even though I haven't got a chance to join these workshops, these workshops are well-known and received good reviews and feedback from the software testing community.

+Ask for one-on-one coaching:

If you have friends or someone you know has many years of experience in software testing, you can ask them for a one-on-one coaching. You will be surprised that there are many great testers out there willing to help and share their testing experience. It would be great if you have a skill that they want, you can exchange with them. If not, you can buy them coffee, lunch or pay to ask for help.

[Note: Of course, if you need any help from me, you can shoot me an email at thanh[at]asktester.com]

Those are a few ideas you can try, but don't just stop there. Be creative and proactive to earn experience in software testing.

Don't go to the interview with your empty-hand.

You have already built your groundwork so far and you should be ready to go to the next important step:

Phase 3: Find a full-time job as software tester

Now, one of the common mistakes most of people (testers) make is to do the following to find a full-time job:

Here is what most people (testers) do to find a full-time job:

1) Browse job sites

2) Pick up several jobs that sounds fit

3) Write one CV and send it to all the job ads

4) Hit Send

5) Pray

It's no wonder why they hardly receive calls for interviews.
Here is what you can do instead:

1)Warm-up and networking

+Join LinkedIn and ...be active on that.

LinkedIn is considered the number #1 network of professionals in the world. It means that your employers may be already there. It means that hiring team may be already there. It means that most of professional testers already there. This is great place for you to let those people know more about you, your software testing skills and your professional.

Joining LinkedIn is easy:

>Create account

>Update your profile by adding your skills, achievement, education, certification, things you care about, etc. (Don't just create profile for its sake, take it seriously because you need to be professional here)

>Find and add connection. Don't just add any connection, choose to invite and add those who are testers or who are working in the company you want to apply for.

Join groups and start discussion: if you are testers, you should know the following LinkedIn group (introduce group here). Join these groups and start asking questions, answers questions. This is important because this shows you really care about software testing and you are taking your career seriously.

Hey, I'm on LinkedIn too, let's connect!

+Attend local conferences:

Search any software testing related local conferences and attend them. Conference is a great place to attend and learn software testing knowledge. However, the most useful part of the conference is networking.

You will have a chance to network with other testers, employers. In some conferences, they have booths to interview and hire fresher too.
So, do not miss this great opportunity.

2)Write a CV that works

There are a lot of tips you can find on how to write a good CV. You may not need more guidance on this. However, regardless of the guide, template you use, make sure you do this in your CV:

+No spelling and grammar error.

People make this mistake all the time. I basically reject candidates make spelling mistakes in their CVs.

Why?

This shows that the candidate is too lazy and careless. As of 2016, we all know that how easy it is to do a spell check on MS Word program. So, having a spell check on a CV is not an excuse anymore.

+Design and Formatting

It's okay if you get creative and decorate your CV a little bit, but don't make it like you are applying for a graphic designer position.

Keep your CV clean and consistent with good font size, font face.

+Short and scannable

It depends on the position and job markets you are applying for, more often hiring team receive many CVs from applicants. The goal here is to make your CV short and scannable.

>Short:

Don't make your CV become an essay. Keep it short at 1-2 page.

"Why? I have a lot of things to tell the employer and I don't want to leave them out" you may ask.

Sure, you have a lot of things to say...but the hiring team also has a lot of work to do too. They don't have enough time to read your CV word by word to dig gold.

So, keep your CV short

>Scannable:

By scannable I mean that you have to make your CV in a way that readers can easily find what they want.

The key thing is you have to figure out what the hiring want, what they are really looking for in the CV.

How?

First, you need to read the job description carefully and understand what skills, requirements they are looking for.

Second, customize your CV so that your CV looks matched with what they are looking for. It's recommended to use the same terms and language as in the job description.

Ex: if the job description says they are looking for someone can "write test cases", make sure you use "test cases" phrase to describe your experience even though you are familiar with phrase "test procedures". If the job description says they are looking for someone can "File defects", use "file defects" phrase in the CV even though you are familiar with phrase "report bugs".

So, you got the idea right.

The goal here is not to show off your linguistic knowledge, the goal here is to make sure you and hiring team are talking the same language.

[Download my CV template. Nothing fancy, but you can refer if you like]

After you follow my guide so far, I hope that you will receive a few phone calls for interviews.

Congratulations!...but you're not done yet.

You are 50% complete to become a software tester. The other 50% is to pass the interview round :-)

3)How to have a successful interview?

I can't count how many guides, advises written about this topic, but people still fail at this step.

The reason is because having a successful interview is crucial and...because it's easy said than done.

It depends on different factors to have a successful interview, from my interviewing experience, it boils down to these key points:

+Create a positive first impression.
This is extremely important because as people says:

"There's no second chance for the first impression"

There's no one-size-fit-all advice here. All I can suggest is to dress nice and professional. Calm and clear in response.

+Research

Research about the company you're applying for to see what business they are in, their products, their team, their environments...the more details you know about the company the better you are. You will surely impress the interviewer how well you know about their organization. This shows that you are serious when you apply for the position

+Be honest in the interview

When answering interview questions, be confident and try your best to answer those questions but you have to be honest. Don't brag or fake it until make it.

+Show the ability to learn

Prove that you are willing to learn new thing even though you don't have experience before. Actually, nobody knows everything. The distinction is the ability to learn new thing.

+Be confident

I know it's hard to be confident when you're new and don't have much experience, but being confident is important. Don't hesitate to ask questions during the interview to clarify the question, to know more about the company. This not only helps you answer the questions better, but also help you see if the company fits you or not.

Remember, you are looking for a job not begging for a job.

What if I failed...

What if you follow all these things and things still do not work out themselves and you do not become a software tester?

If that's the case, you're not alone. It's not that you're doing something wrong or software testing is not for you.

Sometimes, you still need luck to achieve the goal, but you know what? Luck will find you if you keep working hard and consistently.

Final thought

There you have it, a complete guide to how to start a career in software testing. I've tried my best to provide you all possible ideas and guides to help you step by step closer to the goal of becoming a software tester. This guide is far from the perfect guide, actually there's no such thing. All you have to do is to understand yourself, try these guides and see what works what not. It's you, not me, not your friends will make you become a software tester. The journey to become a tester is tough, but I can do, so you definitely could do that too.

What’s next?

- Shoot me an email to my email address thanh@asktester.com to let me know what you think about this post (I read all emails)

- Connect with me:
  - Twitter: @AskTester
  - LinkedIn: Thanh Huynh