



***READY,  
TESTER ONE?  
GO!!***

@MELTHETESTER

1

As with all things, sometimes there are stories this thing is no different....

***HELLO!***



***I'M  
MELISSA EADEN***



***YOU CAN FIND ME  
@MELTHETESTER***



***A LONG TIME AGO... IN A GALAXY CALLED***



@MELTHETESTER

5

Some ideas start in the weirdest of places. If you are not familiar with the testing community on twitter, I suggest following it, or joining twitter and seeing what testers are talking about. A theme that pops up often is the idea of technical vs non-technical. That testers are either one or the other. I have no idea where or how this got attached to testers, no one else in the industry really suffers from this odd notion. Program managers, business analysts, customer service folk all manage to be understood and classified correctly as folks that have technical skills either in business, writing, or computers in general. Testers however have managed to have this weird non-technical label attached.

It's bullshit.

The very idea that any tester is not considered technical is kind of a farce. There is a very real and different distinction between being non-technical and a non-coder. Again, somehow, everyone else has managed to be a non-coder, but technical, except testers. You have to code as a tester to be technical. That's bullshit too.

I have some theories as to why this happens to testers more often than other roles, mostly related to gender and race, but it's a bit of guesswork at the moment.

## For Historical Purposes...



**MelTheTester**

@melthetester · 15 Aug 2017

↻ 2    ❤ 4    ▼

Is there a general confusion about "learning to code" and "being more technical"?



**MelTheTester**

@melthetester · 15 Aug 2017

↻    ❤ 4    ▼

Replying to @melthetester

My general thought about this is you can be more technical by learning to code, but that's not the only way to be a technical tester.

@MELTHETESTER

6

Here are the tweets I'm referring to and why I wrote the blog post Ready Tester One. I specifically linked it to gaming because I'm a gamer and because there is a long history with testers and games. Testers were around before video games were a huge thing, but the industry somewhat took off as software, specifically gaming software became more mainstream.

The tweets....

<https://twitter.com/i/moments/947154288698150912>

# Making the List

- What are technical skills?
- What skills do we need?
- What skills do we have which are taken for granted?

@MELTHETESTER

7

I had a working list of skills by levels I had created for a client because they were having a hard time finding talent with the right skill set. I wanted to explain how I evaluate myself and what I thought we needed as skill sets. I started with my career chronologically, made a list, rearranged them again, made another list. The original version had three levels. It eventually expanded it to five. Once I knew what I was capable of doing, I looked at others I worked with and pulled skills from them as well.

Specializations are a thing, and they are much broader and deeper for testers that people realize.

-Automation is one... but if it's just UI, then that's only one space. What about services automation, pipeline automation, tool testing development and maintenance, automation design/architect.

-Performance & Load testing have a whole lot of things you can focus on

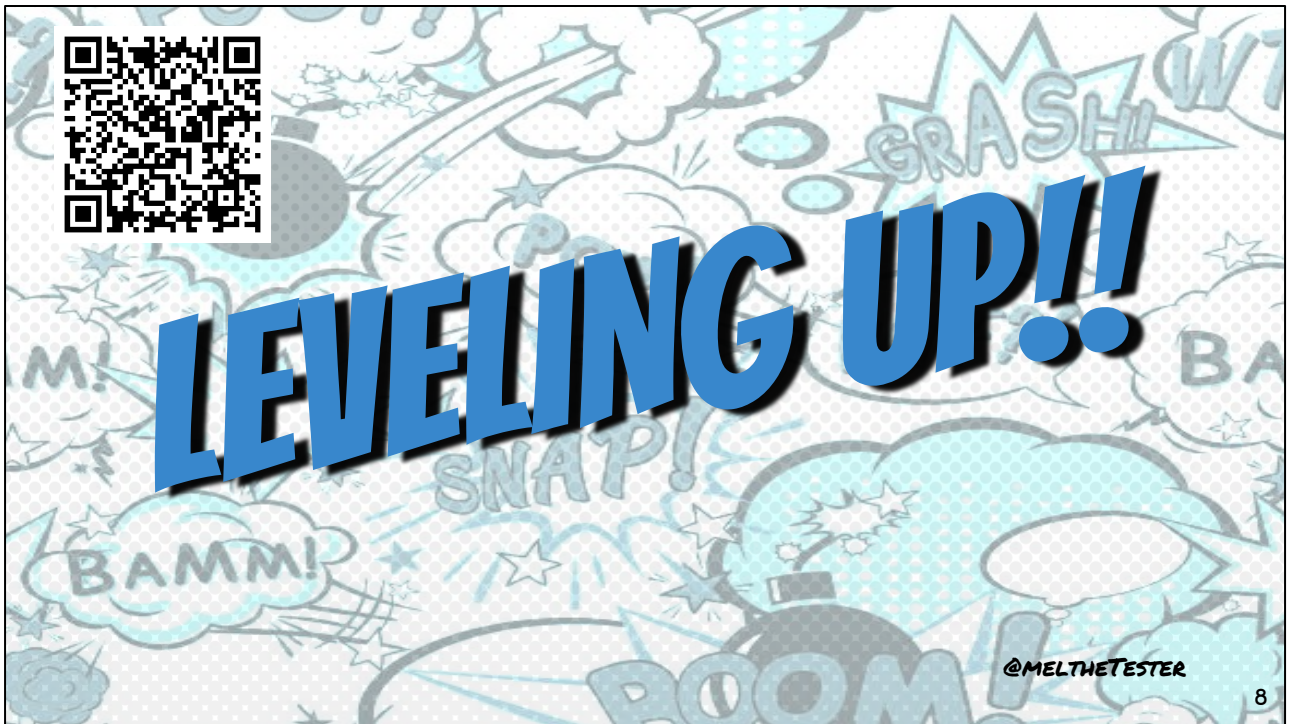
-Security testing is a whole sphere of techniques and tools

-Accessibility testing

-Mobile testing especially if you are dealing with more than platform, services, devices, then accessibility, on top of any native pay features, and then security.... On top of automation.

For my character sheet, I have skill sets mentioned, some specialities but the majority of it are around tools and general knowledge you likely need before you even get into a specialization.





<https://testingandmoviesandstuff.blogspot.com/2017/09/ready-tester-one-go.html>

I got a lot of my ideas from gaming, and games in general.

Story time!

I'm not only a gamer, once upon a time, I used to be a larper. LARP is Live Action Role Play.



# LARP!?

Vampires, Werewolves, and Fairies,  
Oh MY!

@MELTHETESTER

10

I spent about eight years playing LARP games created by White Wolf. The general idea is that you would make a character, based on some rules and roles, then act the character out. I did that for a number of years, and then I moved to being a storyteller for LARP games. This is the person that takes all these wonderful, crazy characters people invented and directed them through plot lines and stories which I or others invented.

How does this equate to a career? Well, if you think about your career like a character sheet, the leap isn't that far. You have some basic talents and skills. You learn some specializations and keep building on your "character" as time goes on. As a storyteller, through the stories I tried to tell, I directly or indirectly shaped how people added talents and skills to their characters. This is akin to what a manager or business might do for an employee through mentoring and/or career development.

The problem is, for testers, businesses and managers have a hard time figuring us out and what exactly we need to do and why we need to do it. There are no clear career paths for testers, in testing with an idea that you grow your skill sets - it's out there for sure, but companies are somewhat poor at identifying them. It comes from the idea that "any tester will do" - because testers are generally not specializing in a code language, or our talents are not tied to a particular tool set, we are seen as interchangeable units. Sometimes.

Automation Testing is where this divide gets interesting. While Generalist Testers (testers that can do manual, automation, service, usability, design, security, data...

etc) aren't always seen as value adds. Testers that squarely put themselves in an automation or SDET role are seen as valuable by companies because they are seen as producing SOMETHING. They have tangible code with visible displays of speedy interactions with UIs. It's the flash and the grandeur of automation that people are dazzled by, and because of that, they sometimes devalue other skill sets testers have. I'll make a note here that I'm not anti-automation at all, but I would find my job pretty boring if all I was doing was automation all day long every day.

Separating automation out as its own role can cause the same gender and diversity issue problems seen with developers. Keep that in mind.

Typical career paths for testers --- tester to developer, tester to product person, tester to automation then management... but there is another, more complex and helpful way to develop testers and these are by specialities. These are skill sets which are specifically geared for a task.

## ***SPECIALIZATIONS***

- |                   |                     |
|-------------------|---------------------|
| ~Management       | ~Monitoring/Logging |
| ~Web              | ~Performance        |
| ~Mobile           | ~Security           |
| ~Data             | ~Accessibility      |
| ~Build Pipelines  | ~IoT                |
| ~Design/Usability | ~Services           |

@MELTHETESTER

10

These are the specializations I've identified, and it's not even a complete list. These don't even really address what happens when you get into combinations of things like Mobile Data, or Mobile Pipelines. Web design, or IoT logging... There is a complete ecosystem out there with developers writing and building things. There are testers out there which understand these combinations and environments and put these specializations together and keep leveling up skill sets to become experts in testing all combinations of things and automate them as well!

What we need to do as testers, as managers of testers, as hiring managers: understand what you are looking for in regards to testing. Understand the combination of skills that you think you might need at the basic level, then understand what specializations might be necessary to do the job. It's impossible at times to find the perfect person, however, if someone has the willingness to learn, to continue learning, to teach others, I promise you they will learn everything you throw at them, challenge themselves to do more and work harder, and then when they leave, they will look for another challenge, and you might be crying because you won't have realized what you had until that person is onto their next job.

I say all this to say, don't pigeon hole your testers. The number one trait of most testers - curiosity. Present everything as a mystery or a challenge to be solved, and with competitive pay and a good cultural working environment, they can do amazing things!



You're never lonely in a crowd....

@MELTHETESTER

13

You might be wondering... how do I get all the skills, how do I find them, how do I... etc...

Well, crowdsource them! You have networks and contacts you probably have that are more than willing to help you along your journey to learning any of the previous skills mentioned. I wrote an article about it, and how you can get a network going, where to look for them... if you don't have a twitter account, get one. The testing community and dev communities are very active. Get out to local meetups. Get names from people. Read books then contact the author; ask questions. Get a book club going, a study group, get linked up on LinkedIn. Read articles people have written then contact the author of the article, learned who they learned from. Go to conferences like this one and collect business cards and stay in touch with people. Start writing a blog, start speaking, get visible... ask for help.

I've told this story a couple of different places. I was working on some postman automation, and I was running into a bit of a problem with an if/else statement. This was frustrating the crap out of me and I couldn't get it work. I was trying to get the test to skip if it ran out of variables. If it realized it didn't have any variables left, it would mark the test as skipped and give the response: No Timeslots left - Test Skipped. This gave us a heads up in the test environment that failures were not because of an issue, but because we were testing too much. Never a bad thing really, but it's a pain if you for pipeline fails for something as simple as a missing variable which is supposed to be missing.

I reached out to Danny Daniton to ask him about the problem. I knew him from my network. He's super good with postman and has a great tutorial up on github. I looked through his tutorial and found something useful, but it wasn't what I needed. I pinged him and asked a few more questions. He was awake to my amazement! I was on the East Coast and he was in the UK. I explained the problem without giving any state secrets away and he worked through it with me. We passed code snippets back and forth over Slack. He figured out how to handle my variable problem and I taught him that if/else still worked in Postman. It was fast, collaborative, and really rewarding for both of us.

I wouldn't have ever been able to do that without a network around me. Danny was cool enough to work with me in real time, but even if it hadn't of been real time, I'm sure he would have worked with the problem, giving me a few tips and advice to get me going, which would have worked out too.

The tech community is changing. We might all be a little introverted, but we are willing to help each other, without the snark and the attitude of previous decades. Shit is way too complicated for one person to do on their own. You might solve parts on your own, but when you get stuck, getting perspective and help is so valuable. Stack Overflow is OK, but I suggest getting to know someone who you can ask.

I recently gave a workshop with Lisa Crispin in Boston. It was the first time I had given the workshop and the whole process was new to me. Lisa was great walking me through a lot of the steps we needed to do to get to a viable workshop at the conference. Others I know who have workshopped before gave great feedback and because of that, I'm sure we'll do the workshop again. I'm also sure we have a good second version in the works.

You can crowdsource anything, really.

Let's do it right now!





***FIND OUT WHO YOU  
ARE SITTING NEXT TO  
AND WHAT FAVORITE  
TOPIC THEY HAVE***

@MELTHETESTER

12

If you don't want to do this right now, it's ok introverts, just say no thank you...

Everyone else, get a name, get a topic or project, get a business card or a twitter handle.

You have your first name of a network!

Do you want a few more?

## Topical Influencers On Twitter

Here is an example:

Software Testing Topics	Expert Software Testers
Mobile & Mobile Automation	<a href="#">Richard Bradshaw</a> , <a href="#">Evan Niedojadlo</a> , <a href="#">Jean Ann Harrison</a>
Mob/Pair Development & Testing	<a href="#">Maaret Pyhajarvi</a>
API (Postman)	<a href="#">Danny Dainton</a> , <a href="#">Mark Winteringham</a>
Automation	<a href="#">Angie Jones</a> , <a href="#">Alan Richardson</a>

@MELTHETESTER

13

Here is part of my network as an example. This took some time and getting to know different people online, having others refer me here and there to folks, or working with folks via the Ministry of Testing. I have been seeking these folks out and saving up names and contacts for when I might be able to ask questions. I'm also available for any of these people too. I might not have anything the need right this minute, but you never know. You start talking about what you've accomplished and people who have helped you, sometimes that comes back in exciting ways.



## Topical Influencers On Twitter

Here is an example:

Software Testing Topics	Expert Software Testers
Mobile & Mobile Automation	<a href="#">Richard Bradshaw</a> , <a href="#">Evan Niedojadlo</a> , <a href="#">Jean Ann Harrison</a>
Mob/Pair Development & Testing	<a href="#">Maaret Pyhajarvi</a>
API (Postman)	<a href="#">Danny Dainton</a> , <a href="#">Mark Winteringham</a>
Automation	<a href="#">Angie Jones</a> , <a href="#">Alan Richardson</a>
Exploratory Testing	<a href="#">Lisa Crispin</a> , <a href="#">Janet Gregory</a> , <a href="#">Elisabeth Hendrickson</a> , <a href="#">Del Dewar</a>
Management/ Process	<a href="#">Kate Falanga</a> , <a href="#">Dan Ashby</a> , <a href="#">Ash Coleman</a> , <a href="#">Alan Page</a>
Blockchain/Cryptocurrency	<a href="#">Rhian Lewis</a>
Security	<a href="#">Dan Billing</a>
DevOps	<a href="#">Katrina Clokie</a> , <a href="#">Mason Richins</a>
Coding/Developer	<a href="#">Angie Jones</a> , <a href="#">Amber Race</a> , <a href="#">Llewellyn Falco</a> , <a href="#">James Spargo</a>

14

Here is more of the list for context... which is always changing but this list from an article I wrote for Ministry of Testing about Crowdsourcing your Learning...

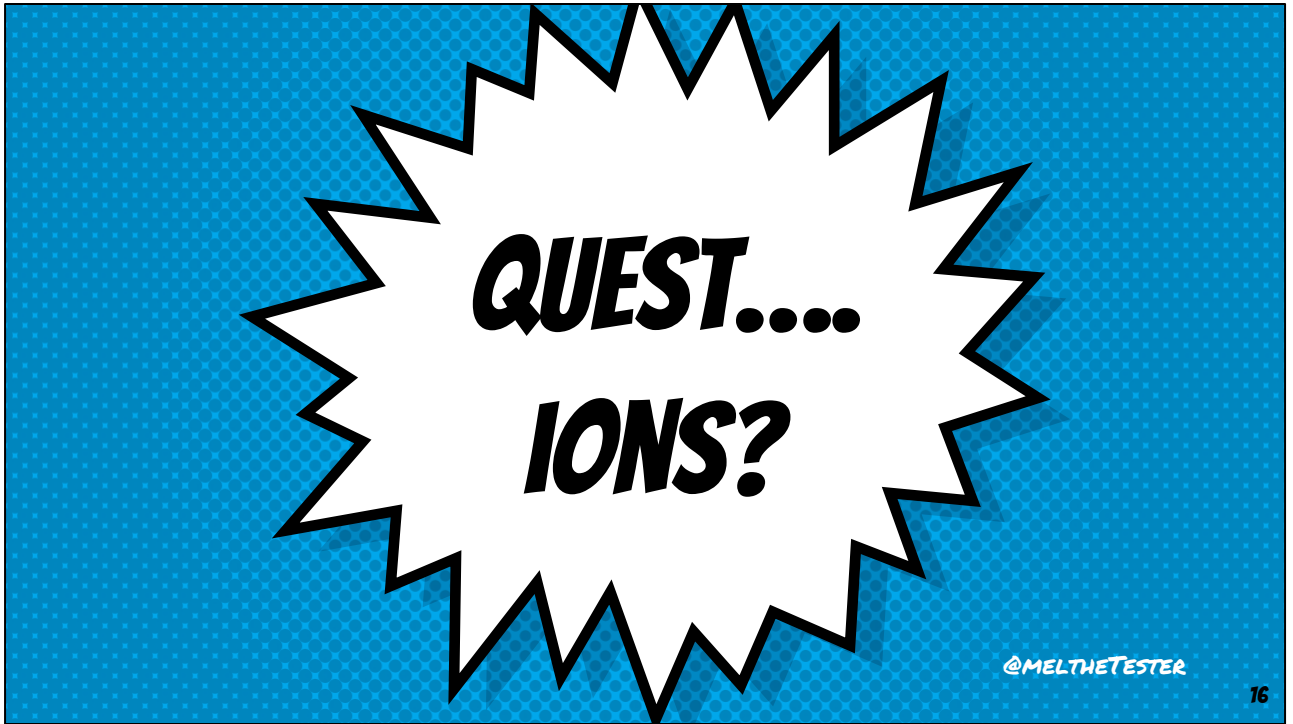


@MELTHETESTER

15

Here is the mega QR code for that article.

[https://ministryoftesting.com/dojo/lessons/crowdsourcing-your-learning?s\\_id=12181](https://ministryoftesting.com/dojo/lessons/crowdsourcing-your-learning?s_id=12181)



Questions? I might have answers - Go!!!



@MELTHETESTER

17

Would you like to fill out your career character sheet? Awesome! Check out this link and feel free to give me feedback and ideas or a submit a pull request even.