Russell Roberts

CSE 270

W12 Testing Issues

Article title: The Future of software testing

Article Author: Angie Jones

Article Reference: Found on Angie’s Website the link is the following:

<https://angiejones.tech/the-future-of-software-testing/>

**IDENTIFY ISSUE CATEGORY**

I think this article is in the category of Academia / Corporations as it has to do with both. This article reflects on the methods of Software Testing and the future of it.

### BRAINSTORM ISSUE

1. The first issue I think this article reflects on is how software testing is being viewed in our day. The issue is to focus on how it will evolve and become more sophisticated over the next few years using AI technology.
2. How will Software Engineers stay relevant?
3. How will these future smart systems of our generations change modern testing?
4. How will people who have been in the industry be able to keep up with lightning speed testing?
5. How will one maintain quality code in this fast-paced testing world?

### IDENTIFY THE RISK

1. The risks of the first issue are that many people will not trust the AI and want to do it the way they do testing now. That is the risk people may reject the future of Software Testing.
2. The risks of the second issue are that many software engineers will no longer know the niches of the business.
3. The risks of the third issue are that many new methods and ideas of testing will have to be adapted and the future smart systems themselves. If we can ever accomplish that.
4. The risk is many software engineers will be left in the dust in their old ways which may slowly become more and more less affective.
5. The risk is the quality of the code that is being testing will become less and less affective.

### BRAINSTORM MITIGATION

1. The first way we can use mitigation in the first issue is acceptance. We obviously know the risk is going to happen, but we can prepare people to slowly let AI take over. Teaching them how to use the AI programs.
2. Prevention is what I would do in the second issue. We as software engineers need to know niches can change and we must adapt with the changes. We must be the ones who prepares ourselves, so we no longer know the niches.
3. Transference might be the best way to mitigate this risk by making sure the AI we are using is correctly working and we put disclaimers there may be bugs in the AI.
4. Acceptance would be the best way to mitigate this risk so these engineers would need to adapt their methods and mesh them with the new way to test.
5. Prevention would be the best way to limit the risks of less capable code. We would need to put other tests in place still to test other parts of the code.

### TEST FOR MITIGATION

To measure all these tests, I think surveys will be needed to gather the data that is needed to be analyzed. With this data we can truly analyze how new testing methods have affected Software engineers in their testing process. We could also ask how incorporating the new features of AI has affected them in their daily work.