

AI AND THE FUTURE OF TESTING



eGUIDE



TECHWELL™

3

5 Things That Will Impact the Future of Software Testing

7

Why Testers Should Take Control of the AI Narrative

11

What's Our Job When the Machines Do Testing?

12

Leveraging Machine Learning to Predict Test Coverage

14

The AI Testing Singularity

15

4 Advantages of Applying AI in Software Testing

16

Testing the Ethics of AI

17

Insight from the Industry

18

Additional Resources

At today's rapid pace of software delivery, manual testing alone is simply inadequate. Artificial intelligence will soon be a necessity for testers to keep up—and in many ways, that reality is already here. But what do these AI capabilities look like, and how can they augment and benefit our test activities? This eGuide rounds up a collection of resources that explain what truly qualifies as AI, the different types of AI, and how quickly AI is advancing. It's time to embrace the future of testing.

In this AI and the Future of Testing eGuide

5 Things That Will Impact the Future of Software Testing

From the way we look at software, evaluate risks, think about complexity, design our test approach and strategy, and help to release a stable product to the customer, technology has had an influence on how we test software. And that influence will only continue as technology advances. On a high level, here are five key things we're already seeing that are going to shape the future of software testing.

Why Testers Should Take Control of the AI Narrative: An Interview with Tariq King and Jason Arbon

In this interview, Tariq King, the senior director and engineering fellow for quality and performance at Ultimate Software, and Jason Arbon, the CEO of test.ai, explain the role artificial intelligence plays in modern testing and why you should establish a foundation right now.

What's Our Job When the Machines Do Testing?

It's a safe bet that testing jobs won't be taken over by machines anytime soon. However, those of us in the test industry would be wise to heed cross-industry applications of analytics and machine learning and begin staking out the proper role of the machine in our testing domain. What could AI mean for testing?

Leveraging Machine Learning to Predict Test Coverage

Test coverage is an important metric within test management, and as technology evolves, we're able to leverage new trends to predict coverage. Weka, an open source suite of machine learning software, can take your test management beyond spreadsheets to the latest AI technologies, letting you predict your test coverage earlier with greater accuracy.

The AI Testing Singularity

Machine learning is rapidly growing more powerful, already sometimes imitating the actions and judgments of humans better than humans. In the near future, even before machines are conscious, they will be able to mimic human software testers. What will be the impact of AI on testing? Jason Arbon has a bunch of ideas.

4 Advantages of Applying AI in Software Testing

We're always looking for smarter, faster, better ways of testing. As the popularity of artificial intelligence grows, more and more testers are realizing its capacity to make cumbersome and time-consuming tasks simpler. AI is coming, so we should take advantage of it. Here are four benefits to applying AI in testing.

Testing the Ethics of AI

AI is a double-edged sword. When it's being used in situations involving sensitive personal data, such as health care, banking and finance, and real estate, security is of the utmost importance—and so are ethical implications. It's up to testers to mitigate risks and make sure AI is used responsibly.

Insight from the Industry

Additional Resources

5 Things That Will Impact the Future of Software Testing

By Raj Subramanian

In the past few decades, software testing has evolved in terms of both the tools used to perform different activities and the mindset of the people using those tools. There used to be only a handful of tools used in software testing, but now we have a plethora of tools to choose from, from proprietary to open source. Likewise, people have started thinking of testers as information brokers instead of gatekeepers, and there have been a lot of positive developments in the agile world that have contributed to significant changes in the processes that teams follow in their software development lifecycle. Advancements in technology are to thank for these evolutions.

From the way we look at software, evaluate risks, think about complexity, design our test approach and strategy, and help to release a stable product to the customer, technology has definitely had an influence on how we test software, and that influence will only continue as technology advances. On a high level, here are five key things we're already seeing that are going to shape the future of software testing.

1. Artificial Intelligence

About five years ago, everyone was talking about "mobile first" and giving the user a mobile experience using mobile web, native, and hybrid applications. Now, the new buzzword is AI. It is in self-driving cars, home assistants (people sure do love their Alexa), computer vision, health care, finance, and now in software testing.

Currently, there are very few reliable tools in the market that use machine learning to help in the authoring and execution of func-



tional testing, end-to-end testing, and regression testing. They are primarily concentrated in UI-based test automation—the more tests the user creates, the smarter the algorithm becomes, which makes the tests more stable.

Here are some of the benefits we could expect to start seeing in testing thanks to AI:

- Easier authoring of tests
- Lower maintenance work on test scripts
- Fewer flaky tests
- Having nontechnical people start doing automation
- Easier CI/CD integration
- More reusable tests

3

5 Things That Will Impact the Future of Software Testing

7

Why Testers Should Take Control of the AI Narrative

11

What's Our Job When the Machines Do Testing?

12

Leveraging Machine Learning to Predict Test Coverage

14

The AI Testing Singularity

15

4 Advantages of Applying AI in Software Testing

16

Testing the Ethics of AI

17

Insight from the Industry

18

Additional Resources

3

5 Things That Will Impact the Future of Software Testing

7

Why Testers Should Take Control of the AI Narrative

11

What's Our Job When the Machines Do Testing?

12

Leveraging Machine Learning to Predict Test Coverage

14

The AI Testing Singularity

15

4 Advantages of Applying AI in Software Testing

16

Testing the Ethics of AI

17

Insight from the Industry

18

Additional Resources

For example, I built an automation framework with Cucumber, Java, and Appium. Although I had a robust framework and a lot of flexibility in writing custom code to perform various actions, I often suffered with the common problem of maintenance. When a developer changed attributes of an element that were already covered by my automated tests, the tests started failing. As a result, I spent a lot of time just maintaining these tests instead of writing new automation code to cover new functionalities that were implemented.

This problem can now be solved by using dynamic locators the AI extracts from the Document Object Model (DOM). In real time, the AI analyzes all the object trees and properties from the DOM and can create a list of different attributes for a particular element. So, when an attribute of an element changes, the AI tries to go to the next attribute in the list to locate the element and keeps going through the list until the element is located. The tests are more stable, the authoring and execution of tests are much faster, and the tester has to spend less time on maintenance.

2. DevOps

DevOps has helped software development teams and operation teams to better collaborate, thereby ensuring there is constant automation and monitoring throughout the software development lifecycle (SDLC), which includes infrastructure management as well.

You may ask, how is this going to influence software testing? The answer: Everything we do as part of testing is going to change. The changes I foresee include:

- A need to start automation right at the beginning of the SDLC and ensure nearly all test cases are automated
- All the QA tasks would need to be aligned to ensure a smooth CI/CD cycle
- A high level of collaboration would be needed to ensure there is continuous monitoring in the production environment
- All the QA environments would need to be standardized
- The testing mindset changes from “completed testing on this module” to “What are the business risks that have been mitigated in the release candidate?”

The key to all the above changes is automation. DevOps and automation go hand in hand—without one, the other won't work. This is where smart people and tools can help in bringing shorter and more dependable release cycles.

I worked at a company where there was minimum collaboration between the development, testing, and operations teams. We were seeing a lot of gaps in the SDLC cycle in terms of more bugs getting into production, unstable CI/CD infrastructure, and less visibility into production monitoring and statistics. Noticing these gaps, the team decided to implement a DevOps practice, and everyone started collaborating and contributing in each phase of the SDLC. This started from requirements gathering and extended all the way to production release and monitoring.

This increased culture of collaboration started having positive effects in team morale, more automation coming into place, and the whole team starting working together as one unit.

Technology has definitely had an influence on how we test software, and that influence will only continue as technology advances.

3

5 Things That Will Impact the Future of Software Testing

7

Why Testers Should Take Control of the AI Narrative

11

What's Our Job When the Machines Do Testing?

12

Leveraging Machine Learning to Predict Test Coverage

14

The AI Testing Singularity

15

4 Advantages of Applying AI in Software Testing

16

Testing the Ethics of AI

17

Insight from the Industry

18

Additional Resources

It's important to change our mindsets about how we view systems and to test them accordingly. We can choose to either ignore it or embrace it. What will you do?

3. QA as a Service

Just like how we have SaaS (software as a service), IaaS (infrastructure as a service) and PaaS (platform as a service), we now have QAaaS. Over the past few years, this has become a popular way for companies to meet their software testing needs.

Companies that have QAaaS solutions make different aspects of your software testing process easier by providing:

- Test case management and maintenance solutions
- Test automation tools with less need for coding
- Robust test reporting features with logs, video recording, and screenshots
- Easy integration with CI systems

Resources such as mobile phones, virtual machines, secure networks, and human testers. In the past seven years of doing automation, one big problem I have always run into is having to maintain my own server machines to run my automated tests. The server machines have different problems, like running out of storage space, a flaky internet connection, slow processing speed for the number of tests that are being run continuously throughout the week, and the need to be frequently updated with the latest OS, build tools, security patches, IDEs, and so on. These kinds of problems could be solved with QaaS providers, as they can do all these activities for you, so team members can concentrate on more critical tasks.

In the future, QaaS providers are going to think about more ways to improve their offerings to stay ahead of their competitors, which will benefit software testers, too.

4. The Internet of Things

With the advent of wearables, smart homes, connected cars, and other cloud-based technologies, the internet of things (IoT) has become a big topic of discussion. The amazing thing about these devices is that there are so many communications and integrations taking place every second.

Let's analyze, on a high level, the different communications taking place with a wearable fitness tracker. First, the mobile app and fitness tracker need to communicate with each other. The data captured by your mobile app should seamlessly integrate with the desktop, mobile web, and tablet versions of the app, and all these communications across devices should happen in real time. All the data travels to and from the cloud, the devices, and the apps. People also can form groups and compete with each other via the app, so these calculations and communications need to happen in real time, too. Based on different events triggered, the proper notifications need to be sent to the right user at the right time. All these communications happen over the internet.

3

5 Things That Will Impact the Future of Software Testing

7

Why Testers Should Take Control of the AI Narrative

11

What's Our Job When the Machines Do Testing?

12

Leveraging Machine Learning to Predict Test Coverage

14

The AI Testing Singularity

15

4 Advantages of Applying AI in Software Testing

16

Testing the Ethics of AI

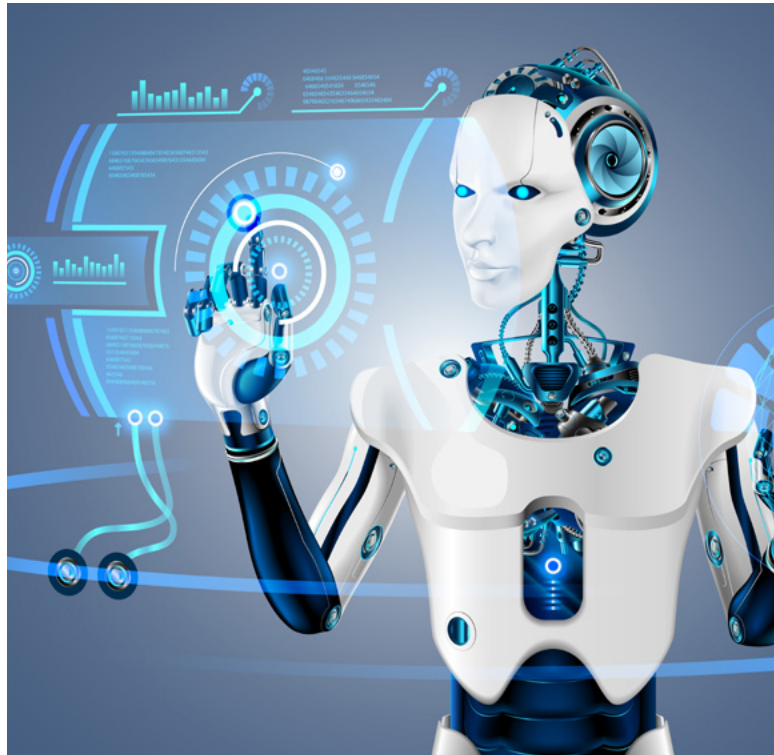
17

Insight from the Industry

18

Additional Resources

Imagine you are a tester testing this fitness tracker. Where do you start? How would you design your test strategy and approach? The IoT introduces its own level of complexity into software testing. It is going to affect the way we think about testing, particularly because more concentration needs to be given to integration testing than the old approach of testing each component separately. For example, when I was working for a travel booking company, we came up with a new app for the Apple Watch built using WatchOS (when it was first introduced by Apple). The application had limited but useful functionalities, like the ability to view notifications and rewards information, reservations, and locations of hotels, flights, and car rentals. While testing this application, I noticed that when the Apple Watch app was connected to the same app on my phone,



there were weird issues: When I minimized the app on the phone, the Apple Watch went blank, showing only a black screen; but when I opened the app on the phone again, the black screen disappeared, and the Apple Watch app behaved normally.

This is a perfect example of why it is important to do integration testing. With more and more devices coming into the market, this is going to be critical for organizations and users.

5. Robots

There are now [robots that do testing](#). Some people may think this is scary in terms of job security, but I still believe that the human mind can never be replaced. There would still need to be humans to monitor the robots to ensure they are doing what is expected of them and to actually program them to do things. How far is this scalable? Only time will tell.

In summary, advancements in technology have already started influencing the way we are doing software testing. It has also caused companies to rethink their organizational structure: The QA team is moving toward being embedded within the development team, and the whole team will own quality. It also will become important for the research and development group to be in frequent interaction with the development group in order to make products smarter and more useful for their customers.

There will also be a need to have procedures in place to handle large amounts of data, as well as appropriate computing power to comb through this data to get useful information and feedback. Finally, to make all this a reality, companies need to adopt lean processes and be much more transparent to prevent being an obstacle for innovation. Lean transformation will be vital for effective growth.

It's important to change our mindsets about how we view systems and to test them accordingly. We can choose to either ignore it or embrace it. What will you do?

Why Testers Should Take Control of the AI Narrative: [An Interview with Tariq King and Jason Arbon](#)

By Jennifer Bonine

In this conversation, Tariq King, the senior director and engineering fellow for quality and performance at Ultimate Software, and Jason Arbon, the CEO of test.ai, talk with Jennifer Bonine about the role artificial intelligence plays in modern testing and why you should establish a foundation right now.



So, when you think about it, artificial intelligence is really a hot topic these days. Whether you're in the business of building software or consuming software, it's all over. For me, I think that one of the biggest things in terms of the testing community is we've seen folks who are using AI for testing. I've seen folks who are kind of averse to it, like I said, but there's these different pockets of people. And I really wanted to just get everyone familiar with what AI means and what AI brings, and start to get people comfortable with thinking about, what does the future look like?



Tariq King

There's, quite frankly, a lack of innovation when it comes to pushing that boundary, and testing seems to lie behind other aspects of software engineering. And I just wanted to try to open up people's minds to say, hey, look. Let's not think that these things are impossible and that they're far away. They're actually here and they're coming. And as a community, we have to get in front of it.

And so, my message was really around, let's stop and think about what we can do to leverage this technology to make what we do better. But let's also notice that in terms of, for AI to be successful, as with any paradigm, testing has to be a key part of that.

And so, the message was really to try to bring those two worlds together: Have people try to get comfortable with thinking about, "Hey, if our future looks different, it's okay." Right? But then also, what ideas can we bring as testers? What things can we bring to this forum to kind of make our impact and make sure that at the end of the day, AI doesn't take over? We need testers to kind of help be that ground and that foundation.

3

5 Things That Will Impact the Future of Software Testing

7

Why Testers Should Take Control of the AI Narrative

11

What's Our Job When the Machines Do Testing?

12

Leveraging Machine Learning to Predict Test Coverage

14

The AI Testing Singularity

15

4 Advantages of Applying AI in Software Testing

16

Testing the Ethics of AI

17

Insight from the Industry

18

Additional Resources

3

5 Things That Will Impact the Future of Software Testing

7

Why Testers Should Take Control of the AI Narrative

11

What's Our Job When the Machines Do Testing?

12

Leveraging Machine Learning to Predict Test Coverage

14

The AI Testing Singularity

15

4 Advantages of Applying AI in Software Testing

16

Testing the Ethics of AI

17

Insight from the Industry

18

Additional Resources

What we found is that some people are visionaries, right? Like, they see something before it comes, and you see it coming, and having talked to you, I've seen this coming as well for a while. But it's interesting to watch the curve of other folks starting to get involved, get on board and see that this is, it's here. It's here.



Jennifer Bonine



Jason Arbon

Inevitable. You've seen it actually kind of evolve, right?

I have.



Jennifer Bonine



Jason Arbon

Actually, you said "awhile," it made me a little bit painful for a second because I'm like, I wish I could just code faster.

Yeah. But, it's two things, I think, right? It's not even just having the technology and the code, but it's the adoption, right? Can people—we have a lot of technology that's ready today, but people aren't ready for it. Right? So, it's not even that it's not we couldn't have it; there's lots of stuff out there we could have. But, people aren't ready for it, right? To absorb it, to accept it, to embrace it. We have to get them to that stage of being okay with it.



Jennifer Bonine



Jason Arbon

Right. That's perfectly put. You're not supposed to upstage the people you're interviewing. But that's better said than either of us had said it this week.

3

5 Things That Will Impact the Future of Software Testing

7

Why Testers Should Take Control of the AI Narrative

11

What's Our Job When the Machines Do Testing?

12

Leveraging Machine Learning to Predict Test Coverage

14

The AI Testing Singularity

15

4 Advantages of Applying AI in Software Testing

16

Testing the Ethics of AI

17

Insight from the Industry

18

Additional Resources

I was just talking to one of the other folks, right before you guys, about the fact that technology and change comes faster than ever before. Things are moving and changing. And if we bury our heads in the sand and pretend stuff isn't here, it's not coming, or don't get aware or educated, you get—that gap gets bigger and bigger, and the opportunity gets wider spread. And people get more afraid, I think, sometimes, when they don't get information. It progresses, it gets more mature. They haven't been involved.



Jennifer Bonine



Jason Arbon

The more they adapt, the more angst they have.

Yeah. And it just creates a bigger angst. And so, as stewards of information, it's good that you're creating opportunities for education and awareness. Because I think a lot of times what people don't realize is, change creates some level of fear, and a lot of times it's fear because they don't know, right? It's the unknown. So, get some information, and that can really help a lot with that gap, of just creating information and having, since it is a small community, people that have some good information, have been involved in it, know it's here, what it's doing, to educate others.



Jennifer Bonine



Jason Arbon

The really smart thing that's gonna happen five to ten years from now is what Tariq was saying, and that is that systems will eventually test themselves from the inside out, right? Rather than from the outside in.

And the testing I do is from the outside in, it's very nontraditional, with swapping the brain out with machine learning. But his idea and vision is software that actually tests itself. And that's the stuff that stands between us and the singularity and the robots taking over. So, I'm really excited to see where he takes that research.

3

5 Things That Will Impact the Future of Software Testing

7

Why Testers Should Take Control of the AI Narrative

11

What's Our Job When the Machines Do Testing?

12

Leveraging Machine Learning to Predict Test Coverage

14

The AI Testing Singularity

15

4 Advantages of Applying AI in Software Testing

16

Testing the Ethics of AI

17

Insight from the Industry

18

Additional Resources

Yeah. There's really a lot of research that started from when I did my Ph.D. That's kind of the area that we looked at. But, that's 2009; the time and the technology has really gone further. And so, for us, like right now we're trying to see with all the advances in technology, microservices, artificial intelligence—can we really design systems in the way that they can test themselves?

Because, when you think about software development, yes, we do a lot of testing from the outside. But we also do a lot of testing from the inside, and if we actually think about designing these systems with these features, then they can be a lot more thorough and actually get to a point where systems may not only be able to detect when they have issues, but they may be able to fix themselves.



Tariq King



Jason Arbon

AI is just to bring people together. Everyone's curious about AI for testing. Some are concerned about it, some really want to dive into it. The idea is to have one place where we can get together and talk about AI. ... Tariq and I were talking about how there's—we don't get together very often, there's very few of us working on the problem today. We're basically lonely people, we try to meet on the internet, and there's a mailing list. We find things, we have papers, we run into sort of interesting issues—or we survey the community, too, to see what they think about AI. ...

Because, there's, I think, a combination of people who are probably super excited about it, a smaller group. A group that just doesn't know much and really wants to get educated. And then a group that's probably a little bit fearful of what that means. I've even heard some testers saying, "I just hope I retire before it happens." ... And a lot of that comes from lack, I think, of awareness and education. So, having places where people can get information, education, awareness, is always good.



Jennifer Bonine

What's Our Job When the Machines Do Testing?

By Geoff Meyer

The application of analytics, artificial intelligence (AI), and machine learning are transforming jobs in industries once thought to be “safe” from automation.

[Law firms are using AI-based machines](#) to do the research that used to define the role of first-year lawyers. Trading decisions for our retirement accounts are managed by algorithmic robots crunching through massive amounts of historical data and real-time market metrics. The [progress of the autonomous vehicle](#), underpinned by AI, is on the verge of replacing drivers across the transportation sector. What businesses in industry after industry have uncovered is this: Processes that have large amounts of data, a representative model, and a generally understood set of rules are candidates for automation.

Geoff Colvin, the author of *Humans Are Underrated*, points to the first-year lawyers described earlier as a model for those of us whose job consists of analysis, subtle interpretation, strategizing, and persuasion. These jobs are gradually—sometimes entirely—being transformed as increasing sets of tasks are delegated to a “smart assistant.”

Our job in software testing is also composed of many tasks. From the nuances of requirements elaboration, the back-and-forth in establishing “expected behavior,” test case analysis, and the other numerous cognitive tasks that we deal with, it’s a safe bet that our jobs won’t be taken over by machines anytime soon. However, for the reasons cited by Colvin, those of us in the test industry would be wise to heed cross-industry applications of analytics and machine learning and begin staking out the proper role of the machine in our testing domain.

Jason Arbon published a test autonomy maturity model, assigning a measure of an organization’s automation capabilities. L0 (manual testing) is entirely in the mode of manual testing. L1 (scripted testing) has a firmly entrenched culture of scripting regression suites. L2 and above (exploratory bots, as well as human-directed, generative, and fully autonomous testing) are for organizations that have started to move into the realm of automating their cognitive and complex tasks.

For an L1 test organization well-versed in a world of automation, whether it be scripting of regression tests or the automation of build and deployment processes, analytics and machine learning represent the next generation of automation. And although it’s just another tool to be considered for our testing toolbox, it’s one that opens up the possibility of automating tasks in our job that we may never have imagined.

While many of the tasks across our diverse test practices are similar, each of our jobs has unique challenges, so priority will be determined by the specifics of our organizational context. So your job, when the machines can do testing, is to [figure out what tasks you want them to do for you](#).

Whether your high-priority pain points include triaging a bounty of automated test failures, avoiding wasted effort researching a defect only to find it is a duplicate, rapidly predicting the root cause of a failure, deciding whether to automate or retire a test case, or increasing regression test coverage in a time-constrained period, there’s a good chance that [employing a machine as your smart assistant](#) might be the next automation approach for you to consider.

3

5 Things That Will Impact the Future of Software Testing

7

Why Testers Should Take Control of the AI Narrative

11

What's Our Job When the Machines Do Testing?

12

Leveraging Machine Learning to Predict Test Coverage

14

The AI Testing Singularity

15

4 Advantages of Applying AI in Software Testing

16

Testing the Ethics of AI

17

Insight from the Industry

18

Additional Resources

Leveraging Machine Learning to Predict Test Coverage

By Bhavani Ramasubbu

Testers should be involved in the requirements collection phase of the software development lifecycle because it benefits both the QA and business teams to understand the requirements better. In test management, we analyze those requirements, prepare test cases, execute test cases, do bug tracking, and get QA to sign off on test coverage.

At my company, a digital commerce agency, we work on e-commerce technologies and serve enterprise clients. Test coverage is an essential metric to measure the quality delivery of our projects. Our test management was initially handled in spreadsheets, but multiple versions of different files ended up underused. Maintaining these spreadsheets and extracting value became complex, and the data was quickly obsolete.

Next, we tried a shared Google spreadsheet to work in a collaborative environment. However, this led to work products being accidentally modified or deleted. Preparing reports and presenting projects to stakeholders consumed more and more of the test leads' time.

We decided to explore the world of test management tools. These tools have evolved over the years, and they resolved some of our problems with features like a centralized repository of test work products, role-based access, reuse of work products, monitoring test activities, and tailor-made reports. They helped us maintain best practices and helped our test leaders manage more effectively at a higher level.

However, they still lacked the ability to predict tasks such as test estimation, test coverage of a release, and quality issues. Test managers



were still spending too much time having to manually combine the predictions made using the reports from these test management tools with statistics. We had to find another technology to monitor and track the progress of the project's delivery timeline.

We discovered [Weka](#): the Waikato Environment for Knowledge Analysis. Weka is an open source collection of machine learning algorithms for data mining tasks developed by the University of Waikato, New Zealand. It is available with an easy GUI, and the library is simple to learn compared to other tools for a beginner in machine learning and AI. The algorithms can either be applied directly to a dataset or called from our own Java code.

3

5 Things That Will Impact the Future of Software Testing

7

Why Testers Should Take Control of the AI Narrative

11

What's Our Job When the Machines Do Testing?

12

Leveraging Machine Learning to Predict Test Coverage

14

The AI Testing Singularity

15

4 Advantages of Applying AI in Software Testing

16

Testing the Ethics of AI

17

Insight from the Industry

18

Additional Resources

3

5 Things That Will Impact the Future of Software Testing

7

Why Testers Should Take Control of the AI Narrative

11

What's Our Job When the Machines Do Testing?

12

Leveraging Machine Learning to Predict Test Coverage

14

The AI Testing Singularity

15

4 Advantages of Applying AI in Software Testing

16

Testing the Ethics of AI

17

Insight from the Industry

18

Additional Resources

There are four main tools in the Weka GUI Chooser:

- Explorer is the primary graphical user interface that gives you access to most of the functionality
- Knowledge Flow allows you to process, view, and visualize your data
- The Experimenter helps you answer questions, like whether one classifier is better than another on a particular dataset
- The Workbench is the unified UI for Weka

Weka has multiple options for generating predictions of any kind of data. Here's how we use its libraries to predict test coverage.

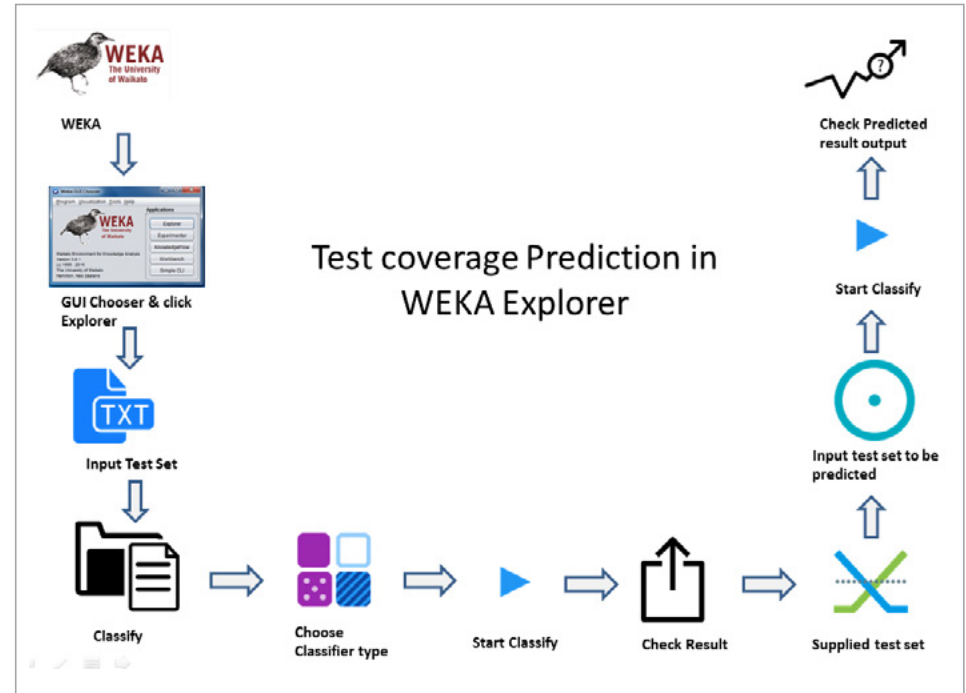
Test Coverage Prediction in Weka

The Weka Explorer is an easy-to-use GUI that harnesses the power of the Weka machine learning software. Each of the major Weka packages—Filters, Classifiers, Clusterers, Associations, and Attribute Selection—is represented in the Explorer, along with a Visualization tool that allows datasets and the predictions of Classifiers and Clusterers to be visualized in two dimensions.

The flow of test coverage prediction in Weka Explorer looks like the snapshot on the right.

We have historical data of test coverage metrics in Excel spreadsheets. For evaluation of the data, we use a .ARFF file as a data source and prepare the .ARFF file with the available existing data of our projects. After inputting the file, we do the classification. This process is known as training, with two data sets. We analyze the training results and track the accuracy. Once we train the system, then we have prepared the .ARFF file with the data that needs to be predicted.

In initial execution, the accuracy was poor—10 percent or so. We had a discussion and analyzed the results. After that, we performed the training with multiple data sets, and it improved the accuracy up to 70 percent in prediction of test coverage results.



At this point we save this model to reuse, which saves time. When the existing saved model is readily available, we load it and then upload the .ARFF test set file to be predicted.

We can load saved models in Weka and reuse them with new datasets, informing updated predictions. It helps to present the data to the stakeholders of the project earlier in the schedule, and we also can make predictions around project estimation and defects with the same accuracy.

If you're struggling with your test management and spending too much time compiling test coverage information, download and explore Weka. Being able to predict your coverage earlier in the software development lifecycle allows you to better prepare for your project delivery.

The AI Testing Singularity

By Jason Arbon

Artificial intelligence is the next exponential technology trend, and it's knocking on our front door.

AI is broken into two major categories of study. Artificial general intelligence is the effort to make machines that are conscious like humans and can reflect on their own existence. The other branch is narrow AI, also known as machine learning, which is focused on computer algorithms that can be trained with data to mimic human thinking—without actually thinking.

Machine learning is rapidly growing more powerful, already sometimes imitating the actions and judgments of humans better than humans in many fields. In the near future, even before machines are conscious, they will be able to mimic human testers. What will be the impact of AI on testing?

The AI testing singularity is the moment software can test itself without human intervention. This moment will come, and we all have questions before that happens: How will machines learn to test better than humans? Which testing activities will succumb to AI first? Which will be last?

There are those who say AI is just slightly smarter software and will be unable to ever behave like a human tester. Some say AI can never



replace humans in testing because machines cannot think, and they would require human judgment and cognition to operate properly. Let's politely say it is far easier to be a skeptic than to actually dive into the technology and understand its implications.

The reality, though, is that there are several factors that point to the AI testing singularity:

- The tech itself. Unlike previous technological revolutions (desktop, web, cloud, mobile), AI is designed to learn from humans and replicate their judgment and actions. AI is focused on solving human input and output functions, and testing is just one of them.
- The world is spending billions of dollars on AI research. The number of papers published on the topic is growing exponentially. For a long time our field was dependent on the R&D budget of open source "charity" projects such as Selenium and Appium and low-priority testing efforts of tooling companies and vendors. As we build AI-based testing approaches, we will ride an unfathomable wave of new R&D.
- AI by definition abstracts concepts and feeds on large amounts of data and/ computing. AI is itself just data and computing, so AI can learn from the very data it generates in a reinforcing feedback loop, and executing this loop can only take minutes to improve itself. This means AI will get smarter far faster than humans can get smarter.

The AI testing singularity will not happen overnight. But it is inevitable; the question is just how soon, and how it will unfold. It's important that you know how and when to bring AI into your own testing efforts, as well as where to invest in your personal learning to stay ahead of the technology curve.

3

5 Things That Will Impact the Future of Software Testing

7

Why Testers Should Take Control of the AI Narrative

11

What's Our Job When the Machines Do Testing?

12

Leveraging Machine Learning to Predict Test Coverage

14

The AI Testing Singularity

15

4 Advantages of Applying AI in Software Testing

16

Testing the Ethics of AI

17

Insight from the Industry

18

Additional Resources

4 Advantages of Applying AI in Software Testing

By Ester Brierley

Since the effectiveness and efficiency of your testing process are crucial to the success of your product, there's no surprise that we're always looking for smarter, faster, better ways of testing. As the popularity of artificial intelligence grows, more and more testers are realizing its capacity to make cumbersome and time-consuming tasks simpler.

Artificial intelligence is coming, so we should take advantage of its abilities. Here are four benefits to applying AI in software testing.

1. Unfailing accuracy

Even the most experienced testers sometimes make mistakes, especially when dealing with repetitive tasks. This is the main reason automation became so popular. Unlike humans, AI always performs the necessary tasks exactly as intended, completing the same repetitive tasks successfully, time after time. While AI works on repetitive tasks, testers are able to focus on creating effective automation solutions and on exploratory activities that only humans can perform.

2. Improved flexibility

Even the simplest changes in an application can lead to test failures in automation tools because traditional testing scenarios consider a singular selector or path. Therefore, such testing approaches are somewhat rigid. Machine learning and AI allow for a more flexible testing process, learning relationships between various segments of documentation. Such systems can automatically adapt to any changes in real time, being both flexible and reliable.

3. Increased test coverage overall

AI allows you to increase the scope and depth of tests significantly. It can check the file contents, memory, data tables, and internal program

states, being able to quickly determine whether or not the program works as intended. AI-powered test automation allows for executing over a thousand test cases in one test run, which is impossible through manual testing.

Even though AI still cannot perform software testing with no help from humans, it is already capable of improving the testing process significantly.

4. Visual validation

Pattern recognition and image recognition enable AI to detect visual bugs by performing visual testing of applications and making sure that all the visual elements look and function properly. AI can recognize dynamic UI controls regardless of their size and shape, analyzing them on a pixel level.

Even though AI still cannot perform software testing with no help from humans, it is already capable of improving the testing process significantly. The main advantage of AI is that it takes automation to a new level so that testers don't need to deal with repetitive tasks anymore, but it also improves the flexibility and accuracy of software tests and, through pattern recognition and machine learning, allows computers to perform tasks that used to require human work. That leaves us more time to do the creative, exploratory aspects of testing.

3

5 Things That Will Impact the Future of Software Testing

7

Why Testers Should Take Control of the AI Narrative

11

What's Our Job When the Machines Do Testing?

12

Leveraging Machine Learning to Predict Test Coverage

14

The AI Testing Singularity

15

4 Advantages of Applying AI in Software Testing

16

Testing the Ethics of AI

17

Insight from the Industry

18

Additional Resources

Testing the Ethics of AI

By Rajini Padmanaban

Artificial intelligence is becoming more and more widespread in the software world. The potential is mind-boggling: AI is now where digitization was a decade ago, touching and impacting every domain. The technology that started in niche areas, chiefly entertainment, has extended by leaps and bounds.

But before you think this is an ideal scenario with endless possibilities for reach and benefit, consider that AI is a double-edged sword. When AI is being used in situations involving sensitive personal data, such as [health care](#), banking and finance, and real estate, security is of the utmost importance—and so are [ethical implications](#).

How can testers play a role in mitigating risks?

Testers understand that automation is only as smart as we design it to be. This also holds true for AI and similar practices such as machine learning, image recognition, and natural language processing. Plenty of sci-fi films have examples showcasing what can go wrong when undue powers are given to robots, and AI is often seen as a core piece in humanizing technology. It's up to testers to make sure AI is used responsibly—especially considering we're already at a point where people are discussing whether AI will become [powerful enough to be worshipped](#).

Such close monitoring is necessary not just because of the potential AI holds, but also to keep adverse effects under check. Just as ethical hacking has gained popularity, ethical AI solutions are on the rise. At a simplistic level, these solutions mainly focus on ensuring AI as a technology does not support immoral causes from societal, environmental, or political influences, and on limiting the power given to the AI algorithms we engineer.



Testers have a critical role here in vetting solutions, even from the ideation stages. Areas that have traditionally been reserved for human intervention due to high cognitive involvement, such as accessibility engineering, are now open to the technology touch. A lot of learning—and unlearning—is needed for testers to be able to bring new perspectives about how solutions will impact stakeholders and users.

Primarily, a lot of investment in quality data feeds that cover positive, negative, null, and boundary values will make all the difference in the test effort and outcomes. Testers will need to employ live monitoring, connect current solutions and new players, and consider what implications third-party integrations with an app will have. Ethical negative tests may also need to be taken up in the right doses to maintain checks and balances on the adverse potential of AI systems.

Interestingly, we can also consider leveraging AI itself. Testers should automate tests as much as possible, making them smart and lean and enabling automation in areas that haven't been possible until now, leaving more time for strategic manual tests.

As AI continues to be used in more and more cases, everyone in the software industry has a collective responsibility to build responsible and ethical AI systems. Testers have a huge role in this process, so start thinking about the implications.

3

5 Things That Will Impact the Future of Software Testing

7

Why Testers Should Take Control of the AI Narrative

11

What's Our Job When the Machines Do Testing?

12

Leveraging Machine Learning to Predict Test Coverage

14

The AI Testing Singularity

15

4 Advantages of Applying AI in Software Testing

16

Testing the Ethics of AI

17

Insight from the Industry

18

Additional Resources

Insight from the Industry

The future is people working on unique, new things. Because same-old, same-old can be handed over to the machine.

“Not only are UX variations continuing to proliferate, but release cycles get getting shorter. For example, even if you manage to run a 60 percent coverage manual UI test, regression will start to occur almost immediately. This does not even account for the error-prone nature of manual QA on your UX in the first place. Artificial intelligence technology that can mimic the human eye and brain has opened the door to automated visual testing at scale, and behind that first door is a whole house full of applications and use cases that will be addressed with visual AI.”

—*Gil Sever*

“I’ve met with various leaders who think automation is the magic pill, but it’s not. How you automate and how much you automate is the key. Skills and the team effectiveness is also a large ingredient in this calculation.”

—*Jennifer Scandariato*

“That’s what I imagine the future of software development being. Right now, yeah, there’s a lot of people who are building the same, or kind of variants of the same. ... We keep doing the same, kind of trying it in different ways. That’s temporary. That will go away. The future is people working on unique, new things. Because same-old, same-old can be handed over to the machine.”

—*Daria Mehra*

“I really made a call to innovate, and the thing is that if we really think about testing and how it really requires us to be independently thinking creatively, can we think of a future where we have systems that actually can test themselves? Is that too big of a dream? We put a man on the moon. We had the Wright brothers fly. Are those things bigger dreams or smaller dreams than just kind of solving this problem? ... [Can we] take testing and what is automated testing from this small checking and verification into, can we use machine learning, artificial intelligence? Can we leverage all that research to make testing better?”

—*Tariq King*

“I see AI having a huge, huge impact in agile and DevOps environments, especially, say, for example, when we talk about test automation. Test automation starts right from the requirements phase early in the software development lifecycle, all the way through production and monitoring, right, so that’s kind of the shift-left paradigm. I can see AI playing a huge part in that.”

—*Raj Subramanian*

“When you think about software development, yes, we do a lot of testing from the outside. But we also do a lot of testing from the inside, and if we actually think about designing these systems with these features, then they can be a lot more thorough and actually get to a point where systems may not only be able to detect when they have issues, but they may be able to fix themselves.”

—*Jason Arbon*

“I see this as a huge opportunity. I see plenty of practitioners asking themselves, ‘Is AI and machine learning going to take my job away?’ Totally no. Actually, that’s the biggest opportunity—I wouldn’t say ever, but biggest opportunity in years—that business testers, manual testers have to get back on stage and support continuous testing activities. ... Codeless testing based on AI and machine learning can allow them to gain back trust and become involved more in test automation activities. By doing that, they also contribute and add more coverage into the overall test suite.”

—*Eran Kinsbruner*

3

5 Things That Will Impact the Future of Software Testing

7

Why Testers Should Take Control of the AI Narrative

11

What’s Our Job When the Machines Do Testing?

12

Leveraging Machine Learning to Predict Test Coverage

14

The AI Testing Singularity

15

4 Advantages of Applying AI in Software Testing

16

Testing the Ethics of AI

17

Insight from the Industry

18

Additional Resources

Additional Resources

MORE INFORMATION FOR SOFTWARE PROFESSIONALS



StickyMinds is home to thousands of software testing resources, including informative articles about all aspects of testing and interviews with industry notables. StickyMinds offers how-to advice and views on the latest ideas and practices from experienced software professionals and thought leaders. Join the community to gain access to exclusive members-only content such as conference presentations, Q&A discussions, a weekly newsletter, and more.

JOIN
HERE

NARROW YOUR SEARCH TO A SPECIFIC TYPE OF RESOURCE:

StickyMinds Articles

StickyMinds articles cover a wide range of software testing topics, including artificial intelligence, the future of testing, testing tools, test automation, agile testing, test process improvement, and much more. **Click here** to read articles about AI and the future of testing on StickyMinds.

Interviews

Each year, TechWell interviews dozens of software professionals, including well-known thought leaders, seasoned practitioners, and respected conference speakers. **Click here** to read, listen to, and watch interviews with testing experts about the future of testing.

TechWell Insights

Find stories about the future of testing, AI, agile, DevOps, teams, and more, all written by software industry professionals. New stories are added each weekday, so **click here** to read the latest—or sign up to receive the weekly newsletter roundup of the newest stories.

TechWell Conference Presentations

Couldn't make it to a TechWell conference to sharpen your testing knowledge? TechWell conference presentations are available to StickyMinds members for free soon after conferences end. **Click here** to join StickyMinds and access conference presentations related to the future of testing and AI.

STAR Conferences

The STAR conferences feature keynote presentations, tutorials, and classes covering test techniques, performance testing, test automation, agile testing, mobile testing, test management, and more. Learn from experts in the field and network with your peers to get the most immersive conference experience possible. **Click here** to learn more.

The TechWell Hub

Join the active and engaging TechWell Hub community on Slack to keep in touch with what's happening in the industry and ask questions about your day-to-day challenges. The Hub offers a safe space to get real-time answers from other software professionals in your shoes, and even experts in the field! **Click here** to join for free.



In the modern world, where speed of release time is accentuated and pressure to frequently release value to end-users is high, sound testing practices and practical automation frameworks are necessary to fully integrate quality into the development and delivery process. Coveros has been a pioneer in testing and QA training, offering a broad spectrum of testing courses for all levels. Our coaches and consultants can also pair with and mentor your staff to accelerate their improvement, now and into the future.

[Software Testing courses](#) | [Automation courses](#) | [Dev & Testing Tools courses](#) | [Agile Testing services](#)

3

5 Things That Will Impact the Future of Software Testing

7

Why Testers Should Take Control of the AI Narrative

11

What's Our Job When the Machines Do Testing?

12

Leveraging Machine Learning to Predict Test Coverage

14

The AI Testing Singularity

15

4 Advantages of Applying AI in Software Testing

16

Testing the Ethics of AI

17

Insight from the Industry

18

Additional Resources