Ranorex | White Paper

AI AND ML IN TESTING: BEYOND THE HYPE

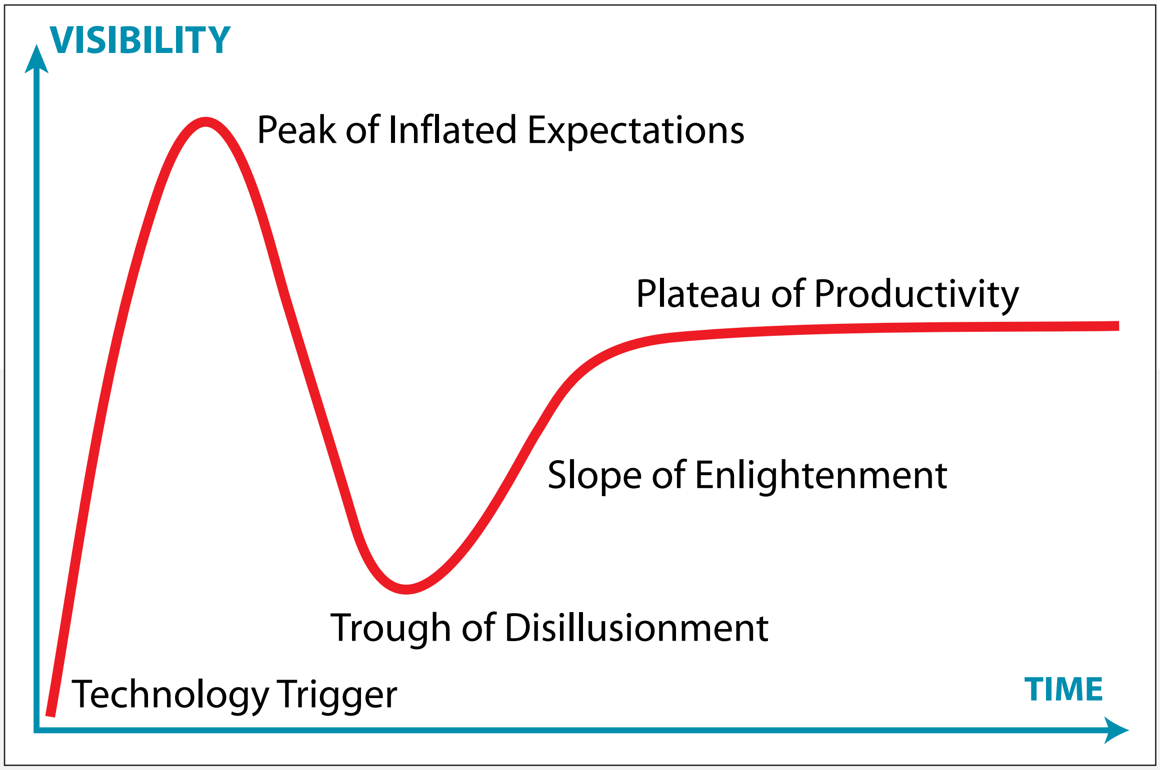
by Cameron Laird

INTRODUCTION

"AI" and "ML" are frequent topics of business discussion. The article which follows summarizes the essentials you need regarding

> The potential for AI and ML,
>
> Current use of AI and ML in testing,
>
> Testing of AI, and
>
> AI claims compared to reality

This article equips you to analyze AI's applicability to testing for yourself. Think in terms of the Gartner hype cycle. You don't want to invest in a technology inflated by premature expectations; far better to "ride the wave" as it emerges from disillusionment and advances toward productivity. Make the cycle work for you, rather than drown in an unfavorable tide. The descriptions below will help you decide how best to boost your own testing programs.



By Jeremy Kemp at English Wikipedia, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=10547051

AI AND ML: HOW IT WORKS, WHAT THE POTENTIAL IS

One possible meaning of "AI"— **artificial intelligence** — is "the computer makes a **decision**." That isn't what most people hear in the initialism, though. While it's valuable when a computer, for instance, correctly computes a sales tax based on the locale of the customer, decisions like "if this zip code then that tax-rate" hardly deserve the AI label. "Artificial intelligence" has been a term of art in computing research at least since 1956. Most commentators expect a deeper kind of AI, one that **learns**, **reasons**, and **solves problems** to reach novel conclusions or decisions, often decisions no human anticipated ahead of time. We'll help you distinguish simple decisions from deeper learning, and how these distinctions shine a light on promotions of AI. At that point, you'll be in a better position to judge for yourself how accurate, reasonable, and realistic AI claims are, and what to do about them.

Machine learning (ML) labels all the kinds of AI that emphasize **training**, or progressive improvement through time. The simplest form of ML is **supervised ML**, which takes a set of data and makes predictions. It does this through training on samples experts have judged. Supervised ML is much like a "fit to curve" algorithm. You can show the computer a hundred thousand examples of hands of poker and which player won, then have the software infer the game's underlying rules. An example closer to our QA target: given adequate data, supervised ML can compare published testing schedules vs. actuals and predict future timelines.

Unsupervised ML, in contrast, undertakes such chores as identification of affiliations in groups. You might invoke unsupervised ML to find, for example, your most profitable customers in order to market to that segment. Casinos are known for using these sorts of techniques. More mundanely, market identification is one of the subjects numerous "reality TV" business shows emphasize: trim a menu, cater to favored customers, and promote customer retention. Unsupervised ML has great potential to analyze such decisions systematically and objectively. Unsupervised ML might, for example, sort through implementation source or test runs to find "neighborhoods". Labeling neighborhoods can lead to a decision that closely-related modules need to be tested together. Note that unsupervised ML often identifies relations or neighborhoods that humans don't notice on our own. The patterns are real, even though not ones to which we're attuned.

Most commentators expect a deeper kind of AI, one that learns, reasons, and solves problems to reach novel conclusions or decisions, often decisions no human anticipated ahead of time.

This low bar for human participation is characteristic of ML. The experts of supervised ML don't have to justify verbally that a particular photograph captures a cute dog, or explain that a flavor is appealing because of the hints of vanillin present; ML achieves its results even if the data are as reductive as "good" vs "bad". Successful ML projects most often draw their data from human actions and assessments that are already familiar before the ML begins.

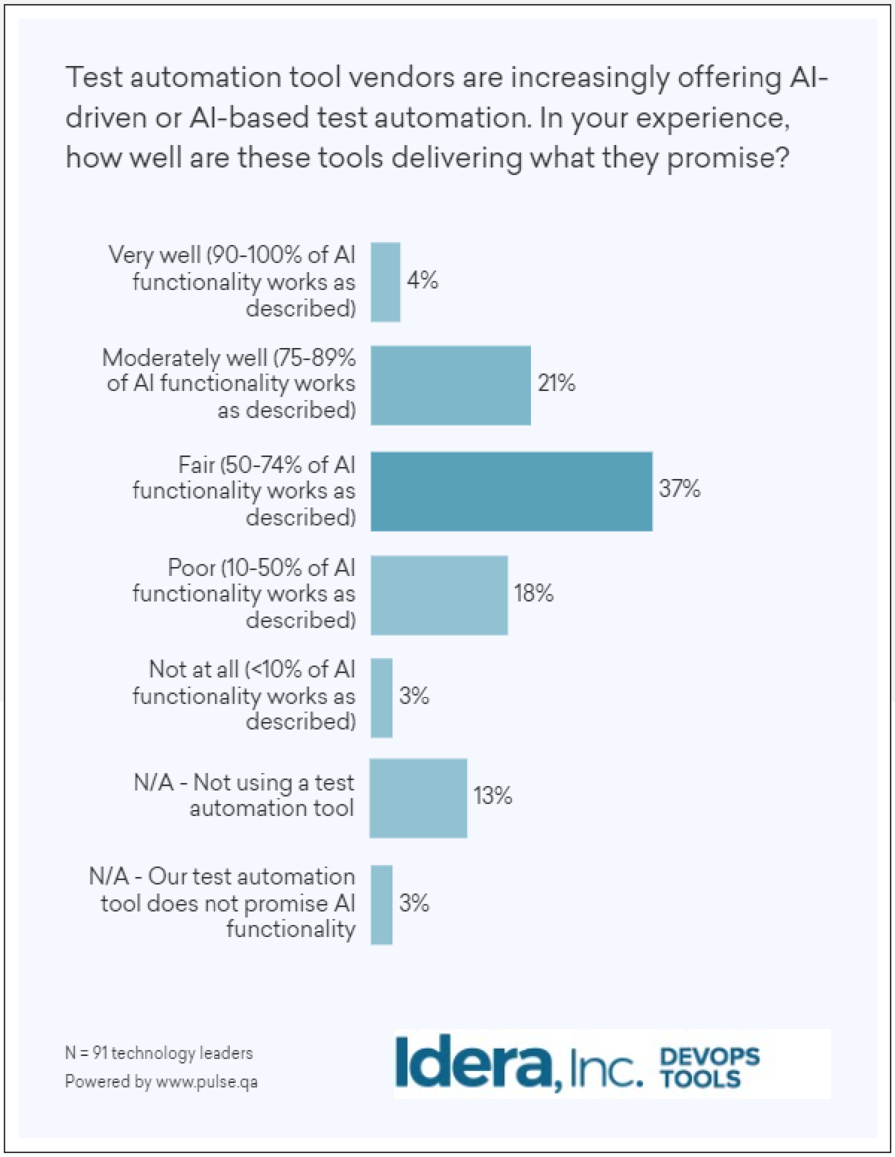
Another form of ML is the **neural net**. A neural net is a particular computer technology which can be trained to recognize, for example, the kind of emails that are spam, by constant reference to what is spam and what is not. This "training" may take millions of examples, but an email company can simply have the customers mark things as spam, then use that as training data! One of the most important lessons from AI's decades of experience to this point is how to generate and manage the data AI needs. Creation of data sets "for free"— incidental to other business operations — is always a welcome achievement, of course.

Yet another key theme in AI research is NLP: **natural language processing**. Think for a moment about testing operations: to test that an application's login works correctly, for example, might involve a specialist writing a computer-language test that exercises the login in automated, reproducible, and useful ways. NLP enthusiasts sometimes advertise a vision that we'll soon be able just to tell the computer "in English" what we want, and the AI figures out the test for itself. Is that realistic? The frequency at which humans misunderstand each others' "plain English" suggests how difficult effective NLP by computers is.

HOW AI AND ML ARE USED IN TESTING TODAY

Several of these AI domains already play a concrete role in computer testing. **Object-guessin**g or **self-healing** automations help with **regression testing**. Regression tests frequently are fragile in the aspect that, for instance, conceptual items move unpredictably. Think back to the earlier example of an automated test of a login sequence. The test likely includes login along the lines of "push the submit button". This can be a problem when a new visual style relocates the submit button on the screen, or otherwise changes its appearance. After the change, an unaided computer effectively doesn't know what or where the submit button is. Humans find submit buttons often without conscious effort; a test automation, in contrast, might have difficulty finding a button which has moved.

This is where object-guessing enters. While finding a moved button might be difficult for a computer, object-guessing and other specialized techniques bring it at least into the realm of possibility. Earlier we noted that having humans mark messages as spam is a source for valuable data AI can use. Similarly, **Software-as-a-Service** (SaaS) companies have the potential to direct unsupervised ML to collect data from millions of test runs, with hundreds of thousands of failures, and arrive at useful conclusions about how and where buttons move during application evolution. Be aware that object-guessing in practice has often reduced to "the blind squirrel" technique, where the algorithm simple-mindedly loops through all displayed objects to find the closest match for, in this case, the submit button.



Object recognition is another form of AI. More precisely, object recognition was once an AI aspiration. Now it's in such widespread use that commentators have moved the boundaries of AI to exclude object recognition. **Optical Character Recognitio**n (OCR) of document images, including documents captured as PDF (portable display format), is the most familiar object recognition. Current challenges in object recognition include automobile safety mechanisms that identify pedestrians, parking spots, and other vehicles, or automatic reviewers of industrial processes from food to clothing. Object recognition is essential in several forms of testing.

Different AI technologies grow in value when combined, of course. Consider **training scenarios** in software testing. AI can recognize common shopping cart workflows. After testing with large collections of similar workflows, AI effectively learns that the picture of the magnifying glass is **search**, while the nearby shopping cart is **cart**, and that consumers characteristically traverse search->add->cart->checkout. E-commerce software of this sort is usually tested "manually", or, at best, is scripted by specialized testers. E-commerce is sufficiently complex that even the best human testers produce tests that are expensive in preparation and rather "brittle" in use. AI-boosted tests, in contrast, leverage the computer's learning capacity to lower costs and yield more consistent and robust tests.

Anything human testers do that is mundane, tedious, or error-prone is a candidate for AI. Some of AI's best applications in testing have been "in the back office", that is, AI focused not on generation and scripting of tests, but analysis of **results** that have already been obtained. ML has already appeared above as a general-purpose aid to classification, sorting, and prioritization. Classification, sorting, and prioritization of **bugs**, in particular, or other testing items like **security vulnerabilities**, is a promising area for ML.

AI also can assist "far to the left". AI was just mentioned as a help in managing test results; different kinds of AI monitor the actions of human testers as they manually exercise an application, then, among other assistance, suggest other tests for humans to run. Even before that, certain kinds of AI can help bring new employees on and train them in their roles as testers. Every station along the testing lifecycle has a potential for an AI boost.

Continuous testing (CT) has, among other effects, exploded the volume of test results. It often takes AI to analyze CT results with the care deserved.

As already mentioned, AI is "slippery", in that its long history has ranged widely. The meaning of AI has changed with time, and even place, as different universities or technical centers have focused on different kinds of AI, and defined the field to their own advantage. One of the results is that business leaders' "AI experience" has varied over a correspondingly wide range. The survey behind the infographic above illustrates that, even when confined to the specific domain of test automation, satisfaction with AI hits both extremes. This reality puts a premium on being specific and concrete in evaluation of AI's potential in testing.

What's certain for now is that:

> Technologies like object recognition once regarded as AI have become commonplace in parts of testing
>
> New forms of AI are progressively making their way into more and more aspects of testing
>
> All AI is dependent on the quality of the data it receives
>
> Many of AI's showiest successes have depended on co-operation between two or more leading-edge AI technologies.

When you evaluate an AI proposal, be sure to verify that trustworthy data will be available to "feed" the algorithms.

TESTING AI

Most of this article is about testing's use of AI. Turn the relation between the two topics around for a moment: does testing play a role in the quality of AI?

Absolutely. Testing AI is itself a weighty subject. While it's almost entirely outside the scope of this article's focus on how AI can help test, many of the low-level concepts are common to both areas. When other parts of your organization put AI into play, you might be asked to "weigh in" with your own knowledge. Be prepared to advise others that testing AI remains in its infancy, that is, commercial users of AI in aggregate still have weak and unrefined practices for testing the AI they already put into use. Even when restricted to the more tractable topic of testing machine learning, the conclusion's largely the same: great opportunity for improvement remains ahead.

Be prepared to advise others that testing AI remains in its infancy, that is, commercial users of AI in aggregate still have weak and unrefined practices for testing the AI they already put into use.

EVALUATING AI AND ML CLAIMS

Testing advisor James Bach once underlined the need for critical thinking in testing by notoriously reducing it to the crude language of "Huh? Really? ... so?" His aim was much like the counsel of this article: emphasize the simple and tangible, especially in the face of flamboyant claims for AI. Bach himself harshly cites examples of what appear to be vacuous but long-standing offerings in testing AI.

The 2017 Conference of the Association for Software Testing was another occasion for vivid overselling. One of the featured speakers claimed that AI would be able to find defects, correct the corresponding source, re-generate the application, and re-run tests to success. Former AST board member Matt Heusser pointedly if politely walked out, in response. Humans don't have a reliable formula for debugging software as the speaker described, and certainly haven't yet been able to embody our fragmentary knowledge in computers. Is the vendor in front of you incapable of explaining an algorithm to prospective human customers? If so, he's unlikely to have succeeded at endowing computers with a working version of that same algorithm.

You don't have to have the deep background of a Bach or Heusser, though, to practice your own critical thinking about AI in testing. Concentrate on reducing claims to language simple enough to understand. Practice straight talk:

> "Let me see an example";
>
> "How would it know?";
>
> "Could a spreadsheet do that?"
>
> "Would that actually work for us? What work do we have to do to get those results?"

Plenty of testing organizations create big progress for themselves with just a little care around data and practices already at hand. 'Need to identify delicate or brittle code? Sort your existing key performance indicator (KPI) reports by defects per module or defects per line of source.

"Critical thinking" regarding AI for testing needn't be entirely negative. The marketplace already has valuable ML-based solutions, especially in analysis of logfiles and code churn. These analyzers provide advice on where to invest the next efforts of testing to yield the greatest return in quality. Recognize that, while this is a successful domain for ML, it's not yet mature enough to run "hands-off". These analyzers are infrastructure and analysis tools which demand expert installation. Moreover, they generally depend on application-specific instrumentation, that is, locally-written ML-savvy custom code.

Testing also is a favorable area for AI application simply because as testers we're accustomed to thinking in terms of measurement, statistics, and cooperation with other teams. These are all common elements of the vocabulary of business AI, and make for a productive basis for conversations about AI's applicability.

CONCLUSIONS

Testing has seen lofty claims for a very long time. Even today, some advertisements claim magical results--"magical" in having a basis that is more misdirection than result. Armor yourself against these deceptions with the basic business analysis of asking yourself and any vendor:

> What is the actual claim?
>
> How will you know objectively whether it's true or false?
>
> What actions can you take in either case?
>
> How does the role proposed for AI compare to human action? Will the AI exploit computers' usual advantages of being able to work at all hours, at scale, and consistently?

Plenty of specific aspects of testing are ripe for improvement from AI. Look for solutions with definite, measurable payoffs in particular phases of testing, rather than foggy promises to take over all of testing.

A technology in the trough of disillusionment doesn't need your money or months of pilot-program effort to discover that it can't keep its promise. Advance your testing projects instead with enlightened, productive, and realistic bets.

Ranorex®

US Headquarters

10801 N Mopac Expressway
Building 1, Suite 100
Austin, TX 78759
USA

+1 512 226 8080

sales@ranorex.com