

Turning Testers into Superheroes

Jonathon Wright

me@jonathon.ai

1. Abstract

In this paper I will share my experience journey from a novice tester in the 1990s to a quality assurance professional, exploring the evolution of testing approaches over decades. The author highlights the redundancy and inefficiency of repetitive manual test design. However, the advent of Generative AI and Large Language Models (LLMs) marks a transformative era in software testing. While generative AI offers vast potential, its current limitations include a lack of domain-specific knowledge and issues with data privacy. Despite these challenges, offline generative AI can leverage proprietary knowledge without risking data security, enhancing automated testing through model-based approaches.

The paper argues for the value of testing as a critical business enabler, emphasizing that QA should be perceived as adding value rather than a cost. The shift from rigid IT structures to user-centric designs has made testing more complex, necessitating faster and cheaper solutions without compromising quality. Generative AI, when combined with human expertise, promises to overcome these challenges by integrating testing into the development phase, reducing cognitive biases, and accelerating test design processes. The author contends that AI complements human testers, enhancing productivity and enabling a better work-life balance. Ultimately, the paper envisions a future where generative AI transforms testers into superheroes, facilitating collaboration, and significantly elevating the role and perception of QA in business environments.

2. Biography

Jonathon Wright is a strategic thought leader and distinguished technology evangelist. He specializes in emerging technologies, innovation, and automation, and has more than 25 years of international commercial experience within global organizations. Jonathon combines his extensive practical experience and leadership with insights into real-world adoption of Cognitive Engineering (Generative AI). In his spare time, he is a member of Harvard Business Council, A.I. Alliance for the European Commission, chair of the review committee for the ISO-IEC 29119 part 8 "Model-Based Testing" and part 11 for the "Testing of A.I. based systems" for the British Computer Society (BCS SIGiST). Jonathon also hosts the QA lead (based in Canada) and the author of several award-winning books (2010 – 2022) the latest with Rex Black on 'AI for Testing'.

Copyright Jonathon Wright, 2024

1 Introduction

The landscape of software testing has undergone significant transformations over the past few decades, driven by technological advancements and evolving business needs. In this paper, I chronicle my journey in software testing, offering insights into the challenges, developments, and future prospects of the software testing domain. This paper begins with a personal narrative from the 1990s, a time when the author, a novice tester, embarked on a summer job at a large telecommunications company. The task at hand was to manually test a hefty book of test requirements and test cases. This introduction serves to set the stage for a broader discussion on the evolution of testing practices, the role of automation, and the emerging influence of Generative AI in this field.

During the initial phase of the author's career, software testing was a tedious, manual process. The six weeks spent running thousands of tests without encountering a single failure illustrated a fundamental inefficiency: repetitive manual testing that failed to reveal new insights or add value. This realization sparked a determination to automate these processes. Automation promised not only to expedite the testing process but also to liberate testers from monotonous tasks, allowing them to focus on more value-adding activities. This shift towards automation marked a significant milestone in the author's journey and in the broader field of software testing.

The narrative then transitions to the present, highlighting the advent of Generative AI and Large Language Models (LLMs) as a revolutionary force in testing. Over the past year, these technologies have demonstrated extraordinary potential in various domains, from code generation to answering complex queries. However, their application in software testing is not without limitations. Generative AI currently lacks domain-specific knowledge and cannot verify the validity of its responses without human expertise. Furthermore, concerns about data privacy have emerged, particularly with instances of proprietary code being inadvertently used as training data for AI models. Despite these challenges, the author posits that offline Generative AI, which operates within secure environments, can effectively utilize proprietary knowledge, optimizing and prioritizing test execution through model-based approaches.

The core argument presented in this paper is that testing should be viewed as a critical business enabler rather than a cost. The IT landscape has shifted from rigid, workflow-centric applications to user-centric designs, significantly complicating the testing process. Organizations today demand faster release cadences, better quality, and lower costs—a trio of requirements that traditional testing struggles to meet simultaneously. Agile and DevOps methodologies have introduced concepts like chaos engineering, yet testing often lags behind these rapid development cycles. The author asserts that integrating Generative AI with human expertise can address these challenges, enhancing collaboration, reducing cognitive biases, and accelerating the testing process.

This paper explores the transformative potential of Generative AI in software testing, arguing that it can turn testers into superheroes by handling the heavy lifting of test design and execution. While AI can manage vast amounts of data and generate comprehensive test scenarios, human testers bring essential wisdom and contextual understanding to the process. By leveraging both AI and human strengths, the testing function can significantly enhance its value, ensuring better quality software delivered faster and at lower costs. The introduction concludes by envisioning a future where AI and human collaboration redefine the role and perception of quality engineering in business environments.

2 Experiences in Software Testing

2.1 Start of My Testing Journey

3. Picture the scene. It's the 1990s, and it's the first day of my summer vacation job at a large telecommunications company. My manager hands me a huge book of test requirements and associated test cases and tells me that my next six weeks will be spent using them to test a PABX (Private Automated Branch Exchange) system and some VoIP (Voice Over IP) phones.
4. So that's what I did. I spent six weeks sitting in a lab for 12 hours a day, running thousands of manual tests. I wasn't a domain expert, so I wasn't sure what PABX or VoIP meant. I wasn't even an expert manual tester with a stack of testing patterns. I was just a rookie following each test step, expected result and writing out the actual test result.
5. At the end of the six weeks, I went back to my manager. I told him I'd completed all the tests, and none had failed. I assumed he'd be happy his software had passed every single test. As you can guess what he said. "I know. We've run them every year for the last decade. The tests have never failed."
6. But because I didn't know any better, my heart sank. At the time, it felt like I'd just wasted my entire summer. I hadn't added any value because I'd only proved something that had always been true was still true. None of my work had revealed anything new. All I'd done was take a horrible chore off someone else.

2.2 Automation was the Future Once

7. Once my disappointment had faded, I determined that I would automate the process, so it never had to be done manually again. I spent the next three years fully automating the tests so they could be run more frequently and more quickly. Most importantly, they would never have to be done manually again.
8. Until very recently, automated testing has been our best possible solution to maximizing test coverage, freeing testers to focus on work that adds more value. What's changed in the past year or so, of course, is the advent of Generative AI and Large Language Models (LLMs).

2.3 Generative AI is the Future Now – But Not as We Know It

9. I'm guessing that, like most people, you've spent the past year or so exploring what gen AI has to offer. You might be using it to write code. You might be using it to write essays or articles. You might be using it like you used to use Google to get answers to questions.
10. It's an extraordinary tool that clearly has huge potential for testers – but in its current form it has huge limitations.
11. It has no domain-specific knowledge and doesn't know any of the answers. For now, it won't tell you where it's got information from so, without your own domain-specific knowledge, you have no way to assess the validity (or otherwise) of the response. This will undoubtedly change in the next year as the EU AI Act comes into force, forcing LLMs to reveal their sources and give data owners the power to restrict the use of their data.
12. It doesn't offer any data privacy either. Doubtless we've all seen the stories of software engineers feeding proprietary code into ChatGPT and then finding the code has been used as training data for the system. The reality is that general tools like ChatGPT don't have much (if any) practical application in our world right now. It doesn't know what you know so it's in effect a generalist

advising a specialist. You can't use it as a co-pilot because you can't – or shouldn't – feed it any proprietary data or mission-critical information.

2.4 Use Your Proprietary Knowledge with No Risk

13. The offline nature of Generative AI is super-important because it means you can feed it your IP-sensitive data. You can give it your confluences, specification documents, and entire test repository, safe in the knowledge it will all stay within your own environment.
14. It gives you a tool that understands testing, your industry vertical, and your specific environment. It can generate fully automated tests and optimize and prioritize test execution through a digital twin model-based testing approach.

3 Unlocking Superpowers for Testers

3.1 Value Driven Testing

15. At the heart of this paper is my simple concern that testing is too often seen as a cost to the business, not an enabler that adds value. I'll explore why that's the case – and show why and how gen AI holds the answer to improving the QA function's reputation.
16. But let's start by considering how much the IT landscape has shifted because I think that holds the key to understanding where it's all gone wrong.

3.2 From IT as King to User as King

17. It's a truism because it's true: the world of IT regularly changes out of all recognition. At the start of my career, the web was just getting going, and IT in business largely consisted of mainframe applications, client service, and thick client systems.
18. It was a completely different landscape in another sense too. Users were expected to follow the workflow of the application not vice versa. If you didn't follow the workflow and something didn't work, it was your fault, not the system's fault.
19. Testing built up around similarly rigid architectures, with exacting governance, standards, and organizations around them.
20. Fast forward to today, and we have multiple devices, multiple applications, and multiple connectivities. The importance of user experience (UX) means applications must work how users want them to, not vice versa.
21. It's all made testing a thousand times more complicated than it used to be. Yet in the same time, testing hasn't really evolved. We still have relatively rigid testing architectures, governance, and standards.
22. And it gets worse too.

3.3 Choose Two of Better, Faster, Cheaper

23. Competition is fiercer than ever. Whether it's your internal IT systems or your customer-facing systems, your organization needs faster release cadences to enhance customer satisfaction and drive competitive advantage. It also needs you to deliver at lower cost with better quality.
24. Yet, as we all know, when it comes to the well-known product triangle of better, faster, cheaper, you can only choose two.
25. Agile and DevOps helped us to speed up release cadences and introduced the concept of the minimum viable product.
26. But testing struggles to keep up with DevOps speeds, and that's a problem that isn't easily solved.
27. You can increase speed by lowering testing requirements, but this reduces quality. You can increase quality by maintaining testing requirements, but this reduces speed. You can throw resource at testing to increase speed and maintain quality, but this increases cost. In the first two

scenarios, you damage the user experience. In the third, you cost the business more than it wants.

28. Ultimately, the result is that the business starts to lose faith in the IT function's ability to deliver what's required – solutions that are better, faster, and cheaper. At the heart of this reputational issue is whether the business views the QA function as a cost or an enabler.
29. Too often, it's seen as a cost. Aside from the difficulty in delivering on better, faster, and cheaper, there's an even more fundamental presentational problem. Put simply, you spend time and money running a thousand tests. You find 20 defects – which will need time and money to fix.

3.4 Testing is a Critical Business Enabler

30. In fact, QA should be viewed as an enabler. We're doing what the business needs us to do – driving the quality that will enhance the UX, increase customer satisfaction, boost reputation, and deliver competitive advantage.
31. The question is, then, how do we deliver testing better, faster, and cheaper so we can restore testing's reputation and have it viewed as a critical business enabler?
32. The answer is that you use Generative AI. Which turbo-charges the human-in-the-loop so the QA function becomes the value-adding team that gives the business what it wants.

3.5 Supercharge the Human-in-the-Loop

33. The critical component of a truly successful DevOps approach is someone who is a specialist in the business side of things and a specialist in the technical requirements of implementation and testing. There aren't many of those around! Most people are either a specialist in the business or a specialist in the technical side of things.
34. Generative AI is effectively a specialist in both. It has business knowledge and it has technical testing knowledge. When you layer humans-in-the-loop on top, you add in the knowledge worker with wisdom to be able to validate and verify that everything expressed by every stakeholder and every permutation is correct. It gives you the ability to translate the subject matter of the business into something the IT and the technical teams can use to build an application and automate its testing.
35. It speeds up the pipeline and helps people on both sides of the equation to collaborate and work together effectively.
36. In short, it means we can properly shift left, bring testing right into the requirements phase of development, embed UX from the start – and elevate the value of testing. It's a game-changer that helps us deliver what the business wants and needs.

4 Turning Testers into Superheroes

4.1 Gen AI Can Do the Heavy Lifting of Test Design

The real value of testers is our ability to think critically and design the tests that get to the heart of the software and what it needs to do. Of course, this test design process is where the heavy lifting really happens. It's also one of the areas where Gen AI can add the most value.

Let's take this example to understand how.

Imagine you need to test braking specifications for a car. The specifications are just a tiny segment of the tests that need to be run for the vehicle and its overall function. At the same time, the specifications are also safety-critical, so it isn't an area you can skim and hope for the best.

Gen AI has the capability and the capacity to do all your thinking in a fraction of the time. It can develop a model with all the required axioms and delve into a level and combination of incredibly powerful detail. It can look at every scenario for heat, ice, rain, and more, individually and in combination. It can factor in road material and assess how tolerances will change.

This ability to interrogate every nuance has the potential to save incredible amounts of time in test design. Its comprehensiveness brings another benefit too, which I'll look at next.

4.2 Gen AI's Lack of Wisdom Can Be a Benefit

You can think of Gen AI as knowing everything and knowing nothing. That's because it lacks the wisdom that we bring to a problem as human testers.

To take the braking specification scenario again, Gen AI will give you every single possible permutation. Of those, some will be physically impossible or simply not reflective of the real world. Your wisdom brings the necessary X factor to the process – you'll be able to see which tests don't make sense or won't be needed.

At the same time, Gen AI's comprehensiveness can add value. As humans, we all have cognitive biases, which can lead to problems in software testing.

There's a story, probably an urban myth, about the F-16 Fighting Falcon fighter aircraft. The story goes that it flew perfectly well in the northern hemisphere – but it flipped 180 degrees the moment it crossed the equator. Why? Because cognitive biases meant the northern hemisphere developers and testers hadn't considered the southern hemisphere in their thinking.

It's an extreme example, but it's the sort of thing that happens all the time. Software designed for the 5G and full-fiber connectivity that every city enjoys but which hasn't yet reached rural communities. Technology that works in flat Illinois but can't cope with mountainous Colorado.

In each of these cases – and countless others – the developers' and testers' cognitive biases created issues. Gen AI helps eliminate those cognitive biases because it lacks the wisdom to have them in the first place.

4.3 AI Won't (And Can't) Take Your Job

The popular media perception is that gen AI is coming for all our jobs. Let's be clear: it can't, and it won't. Gen AI and LLMs draw on vast amounts of information and data – more information and data than a human could ever digest (or remember). What they lack is the next level – the wisdom and experience that humans bring to that information and data.

I liken it to Neo and Morpheus in The Matrix. Neo knows everything there is to know about kung fu, but he's never done it in the real world. It's why Morpheus, who has no knowledge but real-world wisdom and experience, can beat him so easily.

It means AIs can do a lot of simple tasks incredibly powerfully and therefore turbo-charge your productivity. But fundamentally, the human-in-the-loop will always be needed because they bring the wisdom that's needed.

But let's not underestimate the power of gen AI to do the simple things well. This leads us to the next point...

4.4 LLMs Help You Achieve Work-Life Balance

Think for a moment about how brilliantly collaborative the world of testing is. If you're struggling with a line of code or don't know the answer to something, you just ask. You might ask Google and trawl through the results. You might ask a colleague on Slack. You might ask a question in a user community.

Whichever way you choose, 99% of the time you'll get the answer you need. The trouble is, it takes time. You've had to wait to get the answer you need, so your task has taken longer than you'd like. Your colleague or the people in the user community who answered your question took time out of their days to help you.

You can think of LLMs as the latest iterations of this way of working. You ask them a question and they give you an answer – instantaneously. It gives us the potential to achieve a level of work-life balance that's eluded us in the past.

But to achieve that work-life balance, you have to know what you're doing – something we'll cover in the next point

4.5 Getting the Best from Gen AI Depends on How You Talk to It

As we just explored, gen AIs lack your context-specific wisdom. If you want to get the most from them, you need to know how to talk to them to give them that context. This is where prompt tuning comes in.

Firstly, you'll need to give them the persona they need – by telling them you're a Gherkin engineer or an MIT professor, they know the level they need to be operating on. You'll likely need to refine your questions – or prompt tune – to get to the real answer you need. At the same time, you can relax knowing you can ask it a stupid question without fear of being judged.

What's interesting, though, is how open LLMs are evolving – and getting lazier. Where earlier iterations would give you the entire code base you needed, today's iterations may only give you a snippet. To get the full code base, you need to explicitly ask for it, even to the point of saying, "I'm feeling lazy today, I don't want to write any of the code and want you to do it for me."

4.6 Think of Gen AI as the Specialist That Complements You as a Generalist (And Vice Versa)

Throughout software development history, the pendulum has swung from specialist to generalist and back again. One minute, everyone wants generalist full stack developers. Then next you need to be a specialist security engineer, a software tester, or a front-end developer.

4.7 Gen AI and LLMs can help you be better at being both.

When first started utilizing Generative AI, we fed it a wealth of testing-specific information. We then needed to help it understand relationships and dependencies so we could use RAG (retrieval augmented generation) architectures to enable it to deliver users the right information at the right time.

This means Generative AI is a generalist in that it knows – and can access – everything testing-related. When you're new to a test field, it's the specialist you need to get you up to speed. What's more, it's quicker than Googling and less embarrassing than asking a colleague.

It also means that when you're a specialist in a field it can be the generalist you need when something crops up that's outside your knowledge domain.

This brings us back to the collaborative nature of our world – and my final point...

4.8 A Test Pilot in a Post-Pandemic World

Pre-2020, we were all sat together in offices. We could celebrate the wins and commiserate the fails with our colleagues around us.

These days, we're likely remote workers. Who do you celebrate and commiserate with? Answer: your gen AI buddy. It's endlessly patient when you're working through the problem together, and it'll celebrate with you when you get something to work. Your gen AI stops being the monster that's going to take your job and starts being the selfless colleague who'll help get you your promotion.

Because, to come full circle: an AI won't take your job, but a human using an AI will.

5 Conclusion

This paper provides a reflective and forward-looking analysis of the evolution of software testing, underscoring the journey from manual processes to the integration of advanced technologies like Generative AI. This paper has delved into personal anecdotes and broader industry trends to illustrate the shifts and transformations that have shaped the current landscape of software testing.

The introduction of Generative AI and Large Language Models (LLMs) marks an even more transformative era in software testing. While automation addressed the inefficiencies of manual testing, Generative AI offers unprecedented potential to enhance and revolutionize the testing landscape. Despite its current limitations, such as the lack of domain-specific knowledge and concerns over data privacy, Generative AI has demonstrated its capacity to handle vast amounts of data, generate comprehensive test scenarios, and reduce cognitive biases. Its ability to operate offline, using proprietary knowledge securely, presents a significant advantage in maintaining data integrity and confidentiality.

The core argument of this paper is that testing should be viewed not merely as a business cost but as a critical enabler of quality, user experience, and competitive advantage. The evolution from rigid, workflow-centric applications to user-centric designs has significantly increased the complexity of testing. Organizations today demand faster release cadences, superior quality, and cost-efficiency — a trio that traditional testing methods struggle to balance. Integrating Generative AI with human augmentation offers a solution to this conundrum, enabling more efficient, comprehensive, and nuanced testing processes.

Generative AI can significantly reduce the heavy lifting involved in test design and execution. By generating exhaustive test scenarios and permutations, it ensures comprehensive coverage and helps eliminate human cognitive biases. However, the irreplaceable value of human testers lies in their wisdom and contextual understanding, which are crucial for validating and verifying AI-generated outputs. This human-in-the-loop approach ensures that while AI handles the data-intensive aspects of testing, human expertise provides the necessary oversight and nuanced judgment.

The future of software testing lies in this symbiotic relationship between AI and human testers. AI will not replace human testers but will augment their capabilities, making them more efficient and effective. This collaboration will enable testers to focus on strategic, high-value activities, ultimately enhancing the reputation of the quality engineering function as a critical business enabler.

In conclusion, this paper envisions a future where the integration of Generative AI transforms the testing landscape. By leveraging AI's data processing power and human wisdom, organizations can achieve better, faster, and cheaper testing outcomes. This evolution will not only restore the reputation of the quality engineering function but also elevate it to a strategic role within the business, driving quality, innovation, and competitive advantage. The journey from manual testing to AI-enhanced processes underscores the dynamic and ever-evolving nature of software testing, promising a future where technology and human ingenuity work hand in hand to deliver exceptional software solutions.

37. References

Adam Leon Smith (Edited by), Rex Black (author), James Davenport (author), Joanna Olszewska (author), Jeremias Rößler (author), Adam Leon Smith (author), Jonathon Wright (author). 2022. *Artificial Intelligence and Software Testing: Building systems you can trust*. British Computer Society

Dorothy Graham (author) / Mark Fewster Graham (author), Jonathon Wright (author), Experiences of Test Automation: Case Studies of Software Test Automation, 2012, Addison-Wesley Professional

Eran Kinsbruner (Edited by), Jonathon Wright (author), The Digital Quality Handbook, 2006. Infinity P