VJX

VERITYJX PUBLICATION 001

Rehumanising Quality

The QA Role as Orchestrator in AI-Accelerated Testing

Compact Full Edition · A faithful literary edition of the presentation delivered at the BCS SIGiST Summer Conference 2026

Niyi Ibironke

VerityJX · 2026

CONTENTS

In this publication

Foreword	
Preface	
Acknowledgements	
Opening	
Chapter 1	**Users Experience Journeys, Not Components**
Chapter 2	**How Did We Get Here?**
Chapter 3	**We Protected the Wrong Things**
Chapter 4	**Delegatable and Non-Delegatable**
Chapter 5	**From Gatekeeper to Steward**
Chapter 6	**Quality is a Property of Journeys**
Chapter 7	**How We Evolve**
Chapter 8	**The Orchestrator**
Chapter 9	**Rehumanising Quality**
Chapter 10	**An Invitation**
Closing	
Afterword	
About the Author	
About VerityJX	
References	
Publication Notes	

VERITYJX PUBLICATION 001

Publication Information

TITLE	Rehumanising Quality: The QA Role as Orchestrator in AI-Accelerated Testing
AUTHOR	Niyi Ibironke
FIRST PUBLISHED	2026
PUBLISHER	VerityJX
EDITION	Compact Full Edition

This publication is a literary edition of the presentation delivered at the BCS Specialist Interest Group in Software Testing (SIGiST) Summer Conference 2026.

Rather than serving as a transcript of the presentation, this edition seeks to preserve its ideas, structure and voice while adapting them for long-form reading. It reflects the presentation delivered on the day together with editorial refinement appropriate to the written medium. New concepts have not intentionally been introduced; where additional explanatory material has been included, its purpose is solely to improve clarity for the reader.

Foreword

Conference presentations are, by their nature, fleeting.

They exist for those who happen to be present in the room. They are shaped as much by the conversations that precede them as by the slides that accompany them, and afterwards they often survive only through photographs, brief social media posts and the memories of those who attended.

I wanted to leave behind something more enduring.

Not because I believe the presentation itself was extraordinary, but because I believe the conversation surrounding it was important.

Artificial Intelligence is changing the way software is conceived, developed and tested at an unprecedented pace. Much has already been written about the technologies themselves. Comparatively less has been written about what those changes mean for the people whose professional identities have been built around the work that technology is increasingly capable of performing.

This paper is my contribution to that conversation.

It is intentionally faithful to the presentation delivered at the BCS SIGiST Summer Conference in 2026. Rather than expanding the presentation into a white paper or academic publication, I have sought to preserve the experience of attending the session while allowing the ideas to breathe within the written word.

The central question remains exactly the same as it was on the day:

> What remains our responsibility when more and more of what we do can be delegated to machines?

I do not claim to offer definitive answers.

Instead, I hope these pages encourage reflection, discussion and, above all, continued conversation.

If they succeed in doing that, they will have achieved exactly what I intended when I first stood before the audience at SIGiST.

Preface

Every profession experiences moments that cause it to pause and reconsider the assumptions upon which it has built its identity.

For software quality professionals, I believe Artificial Intelligence represents one of those moments.

The purpose of this paper is not to argue against technological progress. Quite the opposite. Artificial Intelligence is already transforming software engineering in remarkable ways, and I have embraced many of those capabilities within my own work. The opportunities it presents are substantial and continue to grow.

Yet technological capability inevitably prompts another question.

> If more and more of the activities traditionally associated with quality assurance can be performed-or significantly enhanced-by intelligent systems, where does that leave the profession itself?

This paper explores that question.

It argues that the enduring contribution of Quality Assurance has never been defined solely by the execution of tests or the detection of defects. Those activities remain important, but they have always existed in service of something larger.

Ultimately, quality is about helping people achieve confidence in the systems they build, the organisations they serve and the journeys their users experience.

That responsibility cannot simply be delegated.

The ideas presented here were originally shared as a conference presentation. They have been deliberately preserved in the same sequence in which they were presented, allowing the reader to follow the progression of thought as closely as possible.

My hope is that this publication captures not only the content of the presentation, but also the spirit of the discussion that surrounded it.

Acknowledgements

This paper owes its existence to a community that continues to ask thoughtful questions at precisely the right moment in our profession's history.

My sincere thanks go to the organisers of the BCS Specialist Interest Group in Software Testing for creating a conference that consistently brings together practitioners, researchers and industry leaders willing to challenge established thinking whilst remaining grounded in practical experience.

I am equally grateful to my fellow speakers.

Throughout the conference, each presentation explored different aspects of a common theme: how our profession is evolving in response to extraordinary technological change. Individually, our talks offered distinct perspectives. Collectively, they became a much larger conversation.

Many of the ideas explored in this paper were shaped not only by what happened on stage, but also by the discussions that took place between sessions, over coffee and in the conversations that continued long after individual presentations had concluded.

Those exchanges reinforced something that lies at the heart of this publication.

The conversations we have today will influence how our profession responds to Artificial Intelligence tomorrow.

More importantly, they will help shape the ways in which those capabilities are ultimately applied in service of people.

Finally, my thanks go to everyone who attended the presentation.

Whether you agreed with the ideas presented, challenged them or simply found yourself reflecting on them afterwards, you became part of the conversation this publication seeks to preserve.

I hope these pages allow that conversation to continue.

Opening

When I was invited to speak at the BCS SIGiST Summer Conference, I found myself returning to a question that had quietly occupied my thinking for some time.

> Not a question about technology.
> Not a question about testing techniques.
> Not even a question about Artificial Intelligence itself.
> Instead, it was a question about us.

> What remains our responsibility when more and more of what we do can be delegated to machines?

It is an uncomfortable question because it reaches beyond software engineering. Every profession built upon knowledge, judgement and repeatable practice is now beginning to ask similar questions of itself. Some approach them with excitement. Others with apprehension. Most, perhaps, with a mixture of both.

The software quality profession is no exception.

Throughout its history, our profession has continually evolved alongside the technologies it supports. New tools have emerged. New delivery models have reshaped the way software is produced. Automation has transformed practices that once depended almost entirely upon manual effort. Each transition has required us to reconsider not only how we work, but also why our work matters.

Artificial Intelligence represents the latest-and perhaps the most profound-of those transitions.

It is capable of analysing information at extraordinary scale, generating software, identifying patterns, proposing solutions and supporting decisions in ways that would have seemed remarkable only a few years ago. Every month brings new capabilities. Every month challenges assumptions that previously felt secure.

For many people, the obvious response is to ask what Artificial Intelligence will eventually be capable of doing.

I believe there is a more important question.

> What should remain uniquely ours?

That question became the foundation of the presentation delivered at SIGiST, and it remains the foundation of this paper.

My purpose is neither to celebrate Artificial Intelligence uncritically nor to resist its adoption. I use these technologies extensively myself because I believe they offer extraordinary opportunities to improve the way we build, test and deliver software.

The challenge lies elsewhere.

As Artificial Intelligence assumes an increasing number of activities that once defined our profession, we are compelled to reconsider where our greatest value truly resides.

I believe that answer lies not in the activities themselves, but in the responsibilities that surround them.

That is the journey this paper explores.

CHAPTER 1

Users Experience Journeys, Not Components

Every discussion about quality must begin with the people we ultimately serve.

It is surprisingly easy, particularly within large and complex technology organisations, to become preoccupied with the health of individual systems. We monitor services. We verify interfaces. We assess performance. We measure availability. We examine logs, dashboards and test results, often in remarkable detail.

> Each of these activities is valuable.
> Yet none of them represents what users actually experience.
> Users do not experience databases.
> They do not experience APIs.

They do not experience microservices, cloud infrastructure or deployment pipelines.

They experience journeys.

They experience whether they were able to accomplish what they set out to do.

From the user's perspective, every underlying technical component becomes part of a single continuous interaction. They neither know nor care which application processed a request or which service returned a response. They care only whether the journey felt coherent, dependable and trustworthy.

This distinction appears simple, yet I believe it changes the way we think about quality.

When we define quality primarily in terms of technical components, we naturally optimise those components. Teams become increasingly effective at improving the systems for which they are directly responsible.

> Paradoxically, the overall user experience can still deteriorate.
> Each component may perform exactly as intended.
> Each team may achieve its objectives.
> Each service may satisfy its technical measures.

Yet the user may still experience confusion, interruption or frustration because no one has maintained responsibility for the continuity of the journey itself.

This is why I believe quality cannot be understood simply by examining the individual parts of a system.

Quality emerges from the relationships between those parts.

It emerges from continuity.

It emerges from the confidence that every decision, made by every discipline throughout delivery, contributes towards a coherent experience for the person at the end of the journey.

This idea became the foundation upon which every subsequent argument in the presentation was built.

If we lose sight of the journey, we risk optimising everything except the experience that ultimately matters.

CHAPTER 2

How Did We Get Here?

The questions we face today did not appear suddenly.

They are the latest chapter in a much longer story of continuous evolution.

Over several decades, our profession has repeatedly adapted to changes that initially appeared disruptive but ultimately became accepted practice.

Many of us began our careers working within highly structured development models, where quality assurance was often viewed as a distinct phase occurring towards the end of delivery. Testing largely followed implementation. Feedback frequently arrived late. Responsibility often appeared concentrated within specialist teams.

Then came Agile.

Software delivery became iterative. Feedback cycles shortened. Collaboration increased. Quality ceased to belong exclusively to testers and became a shared responsibility across multidisciplinary teams.

Soon afterwards, DevOps extended that philosophy even further.

Development and operations became increasingly connected. Deployment accelerated. Continuous Integration and Continuous Delivery transformed release cycles from infrequent events into routine activities.

Each transition challenged assumptions that had previously felt fundamental.

Each required new skills.

Each encouraged us to rethink where quality was created.

Artificial Intelligence now represents the next stage of that journey.

Unlike previous transitions, however, Artificial Intelligence is not simply changing how we perform existing activities.

> Increasingly, it is performing some of those activities itself.
> It can assist with requirements analysis.
> It can generate implementation.
> It can propose tests.
> It can analyse outcomes.
> It can identify anomalies.
> It can summarise findings.

Capabilities that once demanded significant human effort are becoming increasingly accessible through intelligent systems.

That reality should not surprise us.

Nor should it threaten us.

Every technological advance has altered the balance between human effort and technological capability.

What distinguishes this moment is the speed with which those changes are occurring.

It is therefore understandable that many professionals find themselves asking where they fit within this emerging landscape.

For me, however, the more interesting question is not whether our activities will change.

History suggests they always do.

The more important question is whether we have inadvertently defined our profession too narrowly.

> Have we come to identify Quality Assurance primarily by the tasks we perform rather than by the outcomes we seek to create?

If so, perhaps Artificial Intelligence is not exposing the decline of our profession.

Perhaps it is inviting us to rediscover its true purpose.

That thought leads naturally to the next question.

> If quality has never been defined solely by the execution of activities, what exactly have we been protecting all along?

CHAPTER 3

We Protected the Wrong Things

The conclusion reached in the previous chapter led me to an uncomfortable reflection.

For many years, we have become exceptionally good at protecting the artefacts of software delivery.

> We protect requirements through review.
> We protect code through testing.
> We protect releases through governance.
> We protect infrastructure through monitoring.
> We protect services through resilience engineering.
> Every one of these activities remains valuable.
> None of them should disappear.

Yet I increasingly found myself wondering whether they had subtly distracted us from something much larger.

Perhaps we had become so effective at protecting the individual components of software delivery that we had overlooked the thing they collectively exist to serve.

The user's journey.

That is not to suggest that the profession has somehow failed.

On the contrary, many of the practices that have transformed software engineering over recent decades have done so precisely because they have improved the reliability of those individual components.

The difficulty is that users never encounter those components independently.

They experience the whole.

Consequently, they evaluate quality very differently from those who build the systems.

They rarely ask whether an individual interface complied with a specification.

They rarely celebrate that a deployment pipeline completed successfully.

They rarely distinguish between one supplier, one department or one application and another.

> Instead, they ask much simpler questions.
> Did the service work?
> Could I achieve what I came to do?
> Did the organisation understand what I needed?
> Could I trust the outcome?
> These are questions about journeys rather than technology.

They remind us that quality is ultimately experienced rather than engineered.

That distinction matters because it challenges the way success is often measured.

> Many organisations possess impressive collections of technical metrics.
> Availability.
> Performance.
> Coverage.
> Defect density.
> Deployment frequency.
> Recovery times.

Each provides valuable insight into the health of the systems being delivered.

None, by itself, confirms that the user has experienced a successful journey.

It is therefore entirely possible for an organisation to demonstrate outstanding technical performance while simultaneously disappointing the people it exists to serve.

I believe this is one of the defining challenges of modern quality assurance.

Our responsibility is not simply to determine whether individual components perform correctly.

It is to help ensure that every component contributes towards an experience that users experience as coherent, trustworthy and complete.

Once viewed through that lens, quality assurance begins to look rather different.

It becomes less concerned with isolated technical success and more concerned with preserving continuity across increasingly complex systems, teams and technologies.

That, I believe, is what we should have been protecting all along.

CHAPTER 4

Delegatable and Non-Delegatable

Having reached that conclusion, I wanted to introduce what I believe is one of the most important distinctions our profession must now make.

Not between manual testing and automation.

Not between traditional engineering and Artificial Intelligence.

But between activities that can increasingly be delegated and responsibilities that ultimately cannot.

> Artificial Intelligence is already transforming software delivery.
> It can generate implementation.
> It can analyse requirements.
> It can propose test cases.
> It can execute tests.
> It can identify anomalies.
> It can summarise findings.

It can recognise patterns across volumes of information that would overwhelm even experienced practitioners.

These capabilities are remarkable.

More importantly, they continue to improve at an extraordinary pace.

I believe we should embrace that progress.

It enables us to build better software, to reduce repetitive effort and to devote more attention to the problems that genuinely require human understanding.

The question is not whether these technologies are valuable.

They clearly are.

The question is what happens if we continue to define our professional value primarily by activities that technology is becoming increasingly capable of performing.

During the conference, I paused at this point.

Not because I wished to create uncertainty, but because I wanted the audience to consider the question personally.

If my contribution is defined principally by executing tests…

If my value lies in detecting patterns…

If my professional identity depends upon activities that increasingly become automated…

> What remains uniquely mine?
> This is not a criticism of our profession.
> Nor is it an argument against Artificial Intelligence.
> It is simply the natural consequence of technological progress.

Every profession eventually reaches a point where technology changes the activities through which value has traditionally been created.

The challenge is deciding whether we continue to define ourselves by those activities.

Or whether we redefine ourselves around something more enduring.

For me, that distinction becomes clearer when viewed through the language of responsibility.

> Artificial Intelligence can increasingly assist us in reaching conclusions.

> It cannot accept responsibility for those conclusions.
> It can recommend actions.
> It cannot own the consequences of those actions.
> It can identify possibilities.
> It cannot determine what an organisation should value.
> Those responsibilities remain fundamentally human.
> They require judgement.
> Context.
> Empathy.

Ethics.

Experience.

They require an understanding of people rather than simply information.

This is where I believe the profession finds its future.

Not by competing with Artificial Intelligence.

Not by defending activities that technology now performs more effectively.

But by embracing those capabilities whilst recognising that responsibility itself cannot meaningfully be delegated.

Technology increasingly enables us to know more.

Human judgement determines what matters.

I believe that distinction will define the next generation of quality professionals.

Not because their technical expertise becomes less important.

But because their human responsibilities become significantly more valuable.

CHAPTER 5

From Gatekeeper to Steward

If the responsibilities of the profession are changing, then so too must the way we describe its purpose.

For many years, Quality Assurance has often been characterised as the final safeguard before software reaches production.

> Sometimes this has been been described as quality control.
> Sometimes as governance.
> Sometimes, rather unfairly, as gatekeeping.

The terminology varies, but the underlying assumption has remained remarkably consistent.

Quality exists to determine whether software should proceed.

That description reflected the environments in which many of us learned our profession.

> Development was frequently sequential.
> Responsibilities were clearly separated.
> Testing often occurred after implementation had largely completed.

The quality function naturally became associated with inspection and approval.

Modern software delivery bears little resemblance to that world.

Today, products emerge through continuous collaboration between disciplines whose boundaries are increasingly difficult to distinguish.

> Product managers.
> Developers.
> Architects.
> User researchers.
> Security specialists.
> Operations.
> Clinical or business experts.
> Customer support.
> Service management.

Governance.

Each contributes decisions that ultimately influence the experience delivered to the user.

Quality is therefore no longer something that can be inserted at the end of delivery.

Nor can it belong exclusively to one profession.

It emerges continuously from the interactions between people, technology and organisations.

If that is true, perhaps our own description of the profession requires reconsideration.

Rather than imagining Quality Assurance as standing at the end of delivery deciding whether work should proceed, perhaps we should see ourselves as maintaining coherence throughout delivery itself.

I deliberately chose the word steward because stewardship carries a very particular meaning.

A steward does not own what has been entrusted to them.

Nor do they exercise authority simply because they occupy a particular position.

Instead, stewardship is responsibility exercised on behalf of something larger than oneself.

The steward's concern is not whether an individual activity has succeeded.

It is whether the integrity of the whole has been preserved.

> That distinction matters enormously.
> The steward asks different questions.
> Not:
> "Did testing complete?"
> But:
> "Do we still understand the experience we are creating?"
> Not:
> "Has every requirement been implemented?"
> But:

"Will people experience this service in the way we intended?"

> Not:
> "Have individual teams fulfilled their responsibilities?"
> But:
> "Have those responsibilities remained connected throughout the journey?"

Those questions cannot be answered solely through dashboards, reports or automated pipelines.

> They require conversations.
> Shared understanding.
> Curiosity.
> Professional judgement.

Increasingly, I believe these become the defining characteristics of Quality Assurance.

> Not ownership of quality.
> Stewardship of quality.
> Not acting as the final checkpoint.

Helping every discipline understand how its decisions influence every other discipline.

Helping organisations preserve a shared understanding of the journeys they are collectively creating.

Ironically, the more capable our technology becomes, the more valuable that stewardship becomes.

Technology can accelerate delivery.

It cannot ensure that people remain aligned around a common purpose.

That remains our responsibility.

CHAPTER 6

Quality is a Property of Journeys

Throughout this paper I have returned repeatedly to a simple observation.

> Users experience journeys.

Not components.

Having established that principle, it is worth exploring its implications more fully.

When organisations measure quality, they naturally begin with the artefacts they can most easily observe.

> Applications.
> Interfaces.
> Infrastructure.
> Availability.
> Performance.
> Security.
> Defects.

Each provides valuable evidence about the health of individual parts of a system.

Collectively, they provide a detailed understanding of technical performance.

> Yet users evaluate something subtly different.
> They rarely distinguish between one organisational boundary and another.
> They do not separate suppliers from departments.

They do not consciously navigate architectures or organisational structures.

> Instead, they experience one continuous interaction.
> To them, every technical boundary is invisible.
> Only the journey remains visible.

This explains why organisations can sometimes become genuinely surprised when users report poor experiences despite every operational measure appearing healthy.

> The organisation has measured the health of the components.
> The user has experienced the continuity of the journey.
> Those are not always the same thing.

This distinction becomes increasingly significant as software ecosystems continue to grow in complexity.

Today's services frequently span multiple suppliers, cloud platforms, products and organisational teams.

No single individual possesses complete visibility of every technical component involved.

Yet the user still experiences only one journey.

Someone, therefore, must retain responsibility for understanding the whole.

> That responsibility has never belonged to technology alone.
> Nor has it belonged exclusively to testing.
> It belongs to every discipline that contributes to delivery.

The role of Quality Assurance, I believe, is to help maintain that shared understanding.

To keep the conversation connected.

To ensure that local optimisation never undermines the integrity of the wider experience.

Viewed in this way, many of the measures we traditionally associate with software quality remain enormously important.

> Automated testing.
> Performance monitoring.
> Security assurance.
> Observability.
> Reliability engineering.
> None of these become less valuable.

Instead, they take their place within a broader understanding of what quality ultimately seeks to achieve.

> They become indicators rather than destinations.
> Evidence rather than purpose.
> The destination has always been something larger.
> Confidence.
> Confidence that the technology behaves as intended.
> Confidence that organisations understand the people they serve.

Confidence that every decision made throughout delivery contributes towards experiences that users experience as coherent, dependable and trustworthy.

That is why I believe quality is not ultimately a property of software.

> Nor of processes.
> Nor of individual teams.
> Quality emerges from the relationships that connect them.
> It emerges from shared understanding.
> It emerges from continuity.
> It emerges from journeys.

CHAPTER 7

How We Evolve

Whenever a profession experiences profound technological change, there is an understandable tendency to think in terms of replacement.

> Will existing roles survive?
> Will entirely new professions emerge?
> Will Artificial Intelligence eventually perform everything we do today?
> These questions are natural.
> History suggests they are also incomplete.

Every major technological advance has altered the activities through which professionals create value. It has changed the tools they use, the skills they require and the expectations placed upon them.

Rarely, however, has it removed the underlying purpose of the profession itself.

> Medicine evolved without abandoning care.
> Architecture evolved without abandoning design.
> Education evolved without abandoning learning.
> The activities changed.
> The purpose endured.

I believe the same will prove true for Quality Assurance.

> The profession will undoubtedly change.
> Some responsibilities will diminish.
> Others will expand.

Many routine activities will increasingly be delegated to intelligent systems capable of performing them more quickly, more consistently and at greater scale than any individual practitioner.

We should welcome that progress.

Every repetitive activity delegated to technology creates new space for human contribution.

The opportunity before us is not to preserve yesterday's definition of Quality Assurance.

It is to discover tomorrow's.

I do not believe that future will be defined primarily by the execution of tests.

Nor by the production of reports.

Nor by the ownership of particular tools or methodologies.

Instead, I believe it will be characterised by the ability to connect people, disciplines and perspectives that would otherwise remain disconnected.

To ask questions others have not yet considered.

To recognise unintended consequences before they become failures.

To help organisations understand not only whether systems function, but whether they continue to serve the people for whom they exist.

These are capabilities that cannot be reduced to a checklist.

> They emerge through experience.
> Through curiosity.
> Through judgement.

Through a willingness to see the whole rather than merely the parts.

That is not a retreat from technology.

It is a deeper partnership with it.

Artificial Intelligence becomes an extraordinary amplifier of capability.

Human beings become increasingly responsible for purpose.

This is not the end of Quality Assurance.

It is the beginning of a broader understanding of what quality has always been.

CHAPTER 8

The Orchestrator

If I were asked to summarise this paper in a single word, it would not be testing.

Nor would it be automation.

It would be orchestration.

An orchestra is not defined by the excellence of its individual musicians.

Each performer may possess exceptional technical ability.

Each may execute their own part flawlessly.

Yet beautiful music emerges only when every individual contribution remains connected to a shared interpretation of the whole.

Software delivery has become remarkably similar.

Modern organisations bring together specialists from disciplines that scarcely existed a generation ago.

> Product management.
> Cloud engineering.
> Cyber security.
> Data science.
> User research.
> Artificial Intelligence.
> Operations.
> Architecture.
> Business change.

Clinical or domain expertise.

Quality Assurance.

Every discipline contributes knowledge that the others do not possess.

No single individual can now hope to master them all.

> Nor should they.
> The challenge is no longer individual excellence alone.
> The challenge is collective coherence.

That, increasingly, is where I believe Quality Assurance can make one of its greatest contributions.

Not by attempting to control every decision.

Not by becoming the owner of every aspect of quality.

But by helping each discipline understand how its decisions influence every other.

By preserving context as complexity grows.

By recognising emerging risks before they become entrenched.

By continually returning conversations to the journeys experienced by users.

The orchestrator does not perform every instrument.

The orchestrator enables every instrument to perform together.

Perhaps that is the role towards which our profession has quietly been moving all along.

CHAPTER 9

Rehumanising Quality

The title of this paper was chosen carefully.

Some readers may initially assume it implies that technology has somehow made quality less human.

That is not the argument.

Technology has always extended human capability.

Artificial Intelligence simply represents the latest, and perhaps the most significant, expression of that continuing evolution.

The challenge is not that software engineering has become less human.

The challenge is that we sometimes risk measuring success using criteria that are increasingly detached from the people our systems ultimately serve.

To rehumanise quality is therefore not to diminish technology.

It is to reconnect technology with purpose.

It is to remind ourselves that software exists because people need to accomplish something.

> Every architectural decision.
> Every automated pipeline.
> Every deployment.
> Every algorithm.
> Every model.
> Every test.
> Every line of code.
> Exists in service of a human outcome.
> That outcome may be safety.

> Efficiency.
> Convenience.
> Understanding.
> Communication.
> Confidence.

Whatever form it takes, the technology remains a means rather than an end.

Artificial Intelligence reinforces this principle rather than weakening it.

The more capable our technology becomes, the greater our responsibility to ensure those capabilities continue to serve people wisely, fairly and responsibly.

In that sense, rehumanising quality is not about returning to the past.

It is about ensuring we do not lose sight of the future we are trying to create.

CHAPTER 10

An Invitation

When I delivered this presentation at the BCS SIGiST Summer Conference, I never expected everyone to agree with the conclusions I reached.

Nor did I hope they would.

Healthy professions evolve through constructive disagreement.

They mature because practitioners challenge one another's assumptions, refine each other's ideas and collectively develop richer ways of thinking.

> My hope was much simpler.
> I hoped the presentation would encourage people to pause.
> To reconsider.

To ask themselves where they believed the profession was heading.

If this paper achieves the same effect, it will have fulfilled its purpose.

> Artificial Intelligence will continue to evolve.
> New tools will appear.
> Existing roles will adapt.

Practices we currently regard as fundamental will eventually become ordinary.

That process is inevitable.

The more enduring question is how we choose to evolve alongside those changes.

> Do we define ourselves by activities?
> Or by responsibilities?
> Do we focus primarily upon technical correctness?
> Or upon the experiences our work ultimately enables?
> Do we seek ownership?
> Or stewardship?

These are not questions that can be answered by any individual paper.

Nor should they be.

They belong to the profession itself.

My contribution has simply been to begin a conversation that I hope continues long after this publication has been read.

If you find yourself agreeing with these ideas, I hope you will develop them further.

If you disagree, I hope you will explain why.

Either way, the conversation continues.

And perhaps that is the most important outcome of all.

Closing

Throughout this paper I have explored a question that I believe every profession will increasingly be required to answer.

> What remains our responsibility when more and more of what we do can be delegated to machines?

For Quality Assurance, I do not believe the answer lies in preserving familiar activities simply because they have defined us in the past.

Nor do I believe it lies in resisting technological change.

History has shown repeatedly that progress reshapes every profession.

Our responsibility is not to prevent that evolution.

It is to ensure that, as our methods change, our purpose does not become obscured.

> Artificial Intelligence will continue to transform software engineering.
> It will continue to improve.
> Its capabilities will continue to expand.
> I hope they do.

Because every advance creates new opportunities for us to redirect our own attention towards questions that technology alone cannot answer.

> Questions of purpose.
> Questions of judgement.
> Questions of responsibility.
> Questions of trust.
> Those have always been human questions.
> They remain so today.

If there is a single idea I hope readers carry away from this paper, it is not that Quality Assurance has become less important.

> Quite the opposite.
> I believe its importance is increasing.
> Not because we own quality.
> Not because we perform particular activities.

But because we help organisations retain sight of the people their technology exists to serve.

Users experience journeys.

Those journeys are created by increasingly complex ecosystems of people, organisations and technology.

Artificial Intelligence now becomes part of those ecosystems.

So do we.

Our responsibility is to ensure those ecosystems continue to serve people with integrity.

Perhaps that has always been the true purpose of Quality Assurance.

Artificial Intelligence has simply made it easier for us to recognise it.

The future of our profession will not be determined solely by the capabilities of our tools.

It will be determined by the quality of the questions we continue to ask.

I hope this paper contributes, in some small way, to that continuing conversation.

Afterword

When I left the conference that evening, I found myself reflecting less on the presentation itself than on the conversations that followed it. People approached me (at the conference as well as online) to say that the talk had articulated something they had been sensing but had not yet found the words to express. Others challenged aspects of the argument, and those conversations have already influenced how I think about the ideas presented here.

If this paper encourages even one reader to look beyond the mechanics of testing and to think instead about the stewardship of journeys, the responsibility of professional judgement and the purpose that lies behind our work, then standing on that stage will have been worthwhile.

About the Author

Niyi Ibironke is a quality engineering leader, software testing practitioner and technology strategist whose career has spanned software development, solution architecture, implementation consultancy and quality assurance across complex, highly regulated environments.

His work explores the intersection of technology, human-centred design and professional practice, with a particular interest in how emerging technologies-especially Artificial Intelligence-reshape the responsibilities of software professionals.

Drawing upon experience across software engineering, healthcare technology and digital transformation, he advocates for approaches to quality that extend beyond technical correctness to embrace stewardship, collaboration and the experiences of those ultimately served by technology.

He is the founder of VerityJX, an independent body of work dedicated to exploring the future of quality engineering through thoughtful writing, practical experimentation and professional dialogue.

VERITYJX PUBLICATION 001

About VerityJX

VerityJX exists to explore how technology can be developed, tested and governed in ways that remain fundamentally centred upon people.

Rather than focusing solely upon tools, methodologies or emerging technologies, VerityJX examines the broader principles that underpin trustworthy digital systems: stewardship, shared understanding, human judgement and the experiences created through software.

Its publications are intended to encourage thoughtful discussion amongst practitioners, leaders and organisations navigating the continuing evolution of software engineering in an increasingly AI-enabled world.

Conference papers, essays, implementation notes and ongoing research published through VerityJX seek not merely to document current practice but to contribute constructively to the conversations shaping its future.

Further publications and resources are available through the VerityJX website.

References

The ideas presented throughout this paper have been informed by decades of evolution within software engineering and quality assurance together with contemporary developments in Artificial Intelligence, continuous delivery, socio-technical systems and human-centred design.

Rather than functioning as an academic review, this publication reflects professional practice and personal reflection. Readers interested in exploring the wider body of work surrounding these themes may wish to consult literature relating to:

- Human-Centred Design
- Socio-Technical Systems Theory
- Systems Thinking
- DevOps
- Continuous Delivery
- Quality Engineering
- Artificial Intelligence in Software Engineering
- Professional Ethics in Computing

A planned companion publication, Extended Notes & References, will expand upon many of the ideas introduced here and provide a fuller bibliography together with additional discussion and commentary.

VERITYJX PUBLICATION 001

Publication Notes

EDITION	First Edition
PUBLICATION	VerityJX
AUTHOR	Niyi Ibironke
COPYRIGHT	© 2026 Niyi Ibironke. All rights reserved.

This publication may be shared in its entirety for personal and non-commercial educational purposes provided appropriate attribution is maintained and the work is not modified.