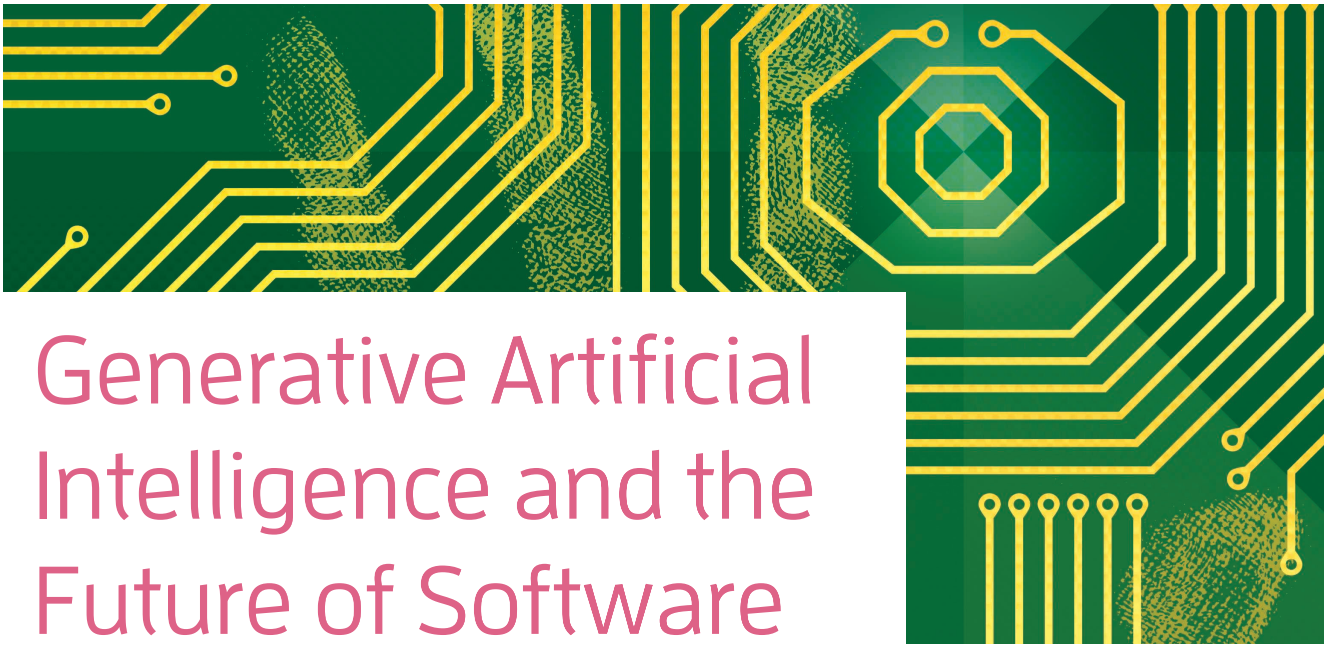
VIRTUAL ROUNDTABLE



Generative Artificial Intelligence and the Future of Software Testing

Lucas Layman and **Ron Vetter**, University of North Carolina Wilmington

This virtual roundtable focuses on applications of generative artificial intelligence (GenAI) to software testing with four leading experts from the field. Our experts reflect on transforming the work of software testing with GenAI, its impact on quality assurance engineers, and privacy concerns.

Generative artificial intelligence (GenAI) is a branch of AI focused on models that create new content. GenAI models have been used to generate text from prompts, create images, formulate new molecules, and write program source code.[1] As GenAI's capability grows, it must address computational challenges, the scale of training data, and new aspects of trust, compliance, privacy, and ethics.[2,3]

The software engineering field is ripe for GenAI applications, including authoring specifications, generating test data, and writing program source.[4] Products such as GitHub Copilot[5] and Meta's CodeCompose[6] have already entered developers' tookits.

IMPACT ON SOFTWARE ENGINEERING

COMPUTER: In what ways do you anticipate GenAI will change how we engineer software?

THOMAS DOHMKE: It will make programming more fun, allow engineers to be more ambitious, and make participating in software development—including testing of course—accessible to more people. Everyone who wants to should have the opportunity to be a developer. Given that tools like ChatGPT and GitHub Copilot allow us to interact with them in human language, in almost any human language, will allow more students to learn how to write software earlier in their lives. As such, AI will democratize access to software development and will significantly increase the number of people that have the skills to accelerate human progress.

Digital Object Identifier 10.1109/MC.2023.3306998
Date of current version: 5 January 2024

ROUNDTABLE PANELISTS

Thomas Dohmke has been fascinated by software development since his childhood in Germany and has built a career building tools developers love and accelerating innovations that are changing software development. Currently, Thomas is chief executive officer of GitHub, where he has overseen the launch of the world's first at-scale artificial intelligence developer tool, GitHub Copilot, and now GitHub Copilot X. Before his time at GitHub, Thomas cofounded HockeyApp and led the company as CEO through its acquisition by Microsoft in 2014. He holds a Ph.D. in mechanical engineering from the University of Glasgow, U.K.

Paul Gerrard earned Masters degrees from the universities of Oxford and Imperial College, London. He has worked in software development and testing since the early 1980s as a developer and project manager and, for 30 years, a leading test consultant. He has chaired several international conferences, won three international awards, and founded the Test Management Forum, Technology Knowledge Base, and the Test Engineering Society. He has worked in projects of all sizes and criticality. Author and coauthor of several books, he focuses on professionalism and artificial intelligence in testing, and improving his poetry, drawing ability, and golf swing.

Adam Porter is a professor of computer science at the University of Maryland and the University of Maryland Institute for Advanced Studies. He holds appointments in the University of Maryland Institute for Systems Engineering and the University of Maryland Applied Research Laboratory for Intelligence and Security. He also serves as the executive and scientific director of the Fraunhofer USA Center MidAtlantic, a University of Maryland-affiliated applied research and technology transfer center specializing in software and systems engineering.

James Walker holds a Ph.D. in data visualization and machine learning in the field of visual analytics, a topic that combines human problem solving skills with the vast processing power of computers. James has given talks worldwide on the application of visual analytics and has several articles in high-impact journals. He has since applied these approaches to quality assurance, focusing particularly on model-based testing and test data management. He is the cofounder of Curiosity Software, a fast-growing start-up that helps enterprises drive quality throughout their software development lifecycle.

PAUL GERRARD: Improvements in the logical analysis of text will enable more effective critical evaluation of textual requirements, whether written in natural language or domain-specific languages like Gherkin. They will use examples to illustrate feature gaps, ambiguities, conflicts, and missing behaviors. These tools know more about business and application domains as those models will emerge over time. Security, reliability, availability, and failover testing will be supported or even performed by AI-based tools using model-based approaches. AI may also offer trustworthy guidance to stakeholders on the documentation, prioritization, and even repair of defects, and potentially the release-readiness of whole systems.

ADAM PORTER: GenAIs and other AI technologies are finding a wide range of applications in software engineering, just as they are in many other industries. In fact, we have already seen impressive applications of AI in code generation. More improvements are certainly on their way. I am particularly interested to see how GenAIs might be applied in other parts of the engineering process. Some candidates include performing business intelligence/gathering requirements for consumer applications based on mining open source data, creating better support for finding and configuring reusable software for specific use cases, and creating more effective software development education and team onboarding.

JAMES WALKER: Top engineers are integrating GenAI daily to accelerate writing code. As these AI models enlarge, and are trained on larger, richer datasets, their problem-solving capabilities and knowledge will grow. Numerous use cases exist for engineering tasks to automate code reviews, enhance code, query databases, and more. The understated problem of requirements quality often leads to unsuitable solutions and technical debt. These two areas present significant opportunities for AI: refining the formulation of complete requirements and addressing the problem of technical debt and understanding at a fine level through domain-specific large language models.

OPPORTUNITIES FOR TESTING

***COMPUTER*:** What are the most exciting opportunities GenAI created for software testing? Can GenAI accelerate testing activity and improve test quality?

GERRARD: The short answer is "yes, but…" I have used ChatGPT to scan HTML code to identify form fields, test data, and boundary values, create covering test cases and Python code to automate tests of simple transactions. But there are limitations in accuracy and comprehensiveness in such test design. Random/statistically based outputs mean responses are inconsistent, and the tool can forget what it has previously reported earlier in the same conversation. The tool can generate "ideas" for tests but needs careful prompting and supervision to check that it does not stray from the mission. It is almost human in its frailties.

WALKER: The immediate opportunity is as an accelerator for quality, assisting with writing tests and code. The assets produced are not perfect, but they provide a great starting point. The longer-term opportunity lies in addressing technical debt. In large enterprises, the biggest challenge is understanding legacy systems/processes; there are pockets of knowledge, but they are siloed between teams and subject matter experts. Training AI in an organization can assist with understanding the landscape, allowing it to be tested appropriately. This is immensely empowering: AI would effectively become the hub of knowledge for driving understanding and promoting quality in an organization.

PORTER: In its current state, GenAI seems to be very strong at conversation, summarization, and transformation (among many other things). Therefore, I expect that the initial applications of GenAI to software testing may revolve around these capabilities. For example, GenAIs could support conversational end user feedback and troubleshooting, providing highly contextualized data to the developers of a given software system. GenAIs can summarize large quantities of heterogeneous data, such as that found in software repositories, user and team Q&A forums, YouTube videos, requirements documents, and more. Finally, GenAIs transform information in one format to another, such as transforming usage scenarios and requirements statements into test artifacts and test code, generating test code for multiple different end user personas and goals, and translating test assets across different testing frameworks and toolsets.

DOHMKE: GitHub Copilot has learned testing conventions from public code and various other texts in the model training set, such as blog posts, wiki pages, and documentation. It also has your project as added context. Whether you write a unit test first or the method, GitHub Copilot can use it to suggest the code for the respective other side. And this is just the beginning. With the help of GitHub Copilot, developers can generate many tests at the same time, and we will soon see the automatic generation of whole test suites.

PRIVACY AND CONFIDENTIALITY

COMPUTER: How will privacy and confidentiality concerns change when GenAI services are integrated into software testing?

WALKER: Organizations' back-end systems contain business rules, trade secrets, and the fundamentals of how an organization operates. Privacy and confidentiality should be of the greatest concern when they are trained and exposed to software testing data. Security risks include aiding hacking and potential exposure of trade secrets if models leak. Furthermore, unlawful use of sensitive information, for example, personally identifiable information, within applications is a concern. Legal and regulatory extensions, like the General Data Protection Regulation, need to be extended to cover AI use. Potential technical solutions may include on-premises [large foundation models] or sandboxed smaller models, and options for nonweight adjusting queries, safeguarding against breaches.

GERRARD: The training data that AI requires to deliver meaningful, reliable services to testers would need to include much proprietary data (code, usage patterns, architectural models, defect histories, etc.) collected across many organizations and systems. It's unlikely this will happen of course. It may be possible that some products appear trainable and usable within single organizations. But it seems unlikely a global "AI test model" could be created. Organizations sensitive to exposing their intellectual property and commercial activity to the outside world, will probably insist tools and models are for internal use only, within their own cloud infrastructure.

DOHMKE: Ensuring user privacy and protecting user data are critical with the GenAI services in the market today, and it will remain critical when these services are integrated into software testing. Developers should take the time to understand how data flow through the GenAI services they use and make sure it fits their privacy needs. For example, with GitHub Copilot we never retain prompt data or suggestions for business users, and individual users must explicitly opt-in for us to retain prompt data. And, as GitHub is part of Microsoft, we adhere to the strict guidelines of the Microsoft Trust Code.

PORTER: Privacy and confidentiality are critical concerns for this technology. Multiple public articles have shown cases in which GenAI users have effectively given their private information to the GenAI provider. This information was then used by the GenAI provider in ways that essentially made it public. One likely response will be that users create and manage their own private GenAIs, rather than rely on public providers. Interestingly, the open source community around GenAIs is flourishing and quite successful, lessening the need to interact with large GenAI providers.

BARRIERS TO ADOPTION

COMPUTER: What are the current barriers to GenAI adoption for software testing? What is required to address these challenges?

PORTER: As with many trendy technologies (e.g., Blockchain is one recent example) there's a real lack of understanding about what GenAI is, how it might actually be used, its benefits over existing technologies, and its potential downsides. This leads to magical thinking about potential use cases and applications in which GenAI can solve every problem that exists. There will need to be a careful examination of our software testing needs and processes, a thorough identification of GenAI strengths and weaknesses, a widespread exploration of specific use cases, and a data-driven comparison against existing solutions. We are only in the beginning stages of GenAI use. Much more experience and hard data will be needed before GenAI adoption becomes widespread.

> I envision that testing and QA will become more focused on the end-user experience and less focused on code-centric activities, such as writing unit tests, as GenAI technologies continue to mature.

DOHMKE: Brains and GPUs. It'll take creativity to integrate GenAI into testing workflows and to build new AI-powered testing applications. It will also require calm consideration of risk and reward from companies and policymakers to not artificially block adoption. And of course, the world needs more GPUs to simply meet demand from software testing and every other field.

GERRARD: For too long, tool vendors have focused on the logistics of testing: test case management, test execution, defect reporting and management, and so on. With AI, vendors see low-hanging opportunities to, for example, make it easier to generate test automation code or test data. Help with such logistics is useful of course, but this does not help with the intellectual challenge of building test models from varying sources of knowledge, defining coverage measures, balancing test utility, coverage, and cost. We need to understand how testers think to identify requirements for true AI-based test assistants.

WALKER: GenAI has the potential to hinder the testing industry. Testing aims to provide confidence to stakeholders that software functions correctly and adheres to requirements. AI is a black-box algorithm, harvesting inputs and providing outputs. Applying this to testing provides less transparency into the testing process, a lower understanding of methodologies applied, and no way to assess the quality of the test cases used (e.g., their coverage). The greatest barrier is comprehending the reasoning behind results and visualizing the generated data for user evaluation. Feedback loops, allowing users to input their subject-matter expertise and understanding to guide solutions, are crucial.

HOW WILL QUALITY ASSURANCE SKILLS CHANGE

***COMPUTER*:** How will the skills required of software testers and quality assurance (QA) engineers change as GenAI tools integrate into the software engineering process? Will software testers and QA engineers become nonexistent?

GERRARD: With the right tools, the skills profile of testers will change. They will become more valuable to software teams but that will mean fewer testers. The best testers will develop a collaborative relationship with their AI partner. Testers will shift left to build relationships with stakeholders to refine system requirements and stakeholder needs for information from testing. New tools will capture models and data across the technical stack, the test team, test outcomes, for all time. The tools will develop both exploratory and advisory capabilities. They will make recommendations and with permission, run tests autonomously when they see opportunities.

PORTER: GenAIs are just one of many technologies that have an impact on software testers and QA engineers (and nearly every other work category as well). Over time, we have repeatedly seen technology automating cognitively lower-level tasks, which pushes testers and QA engineers to focus on cognitively higher-level tasks. Software testers and QA engineers are not going away any time soon. I envision that testing and QA will become more focused on the end-user experience and less focused on code-centric activities, such as writing unit tests, as GenAI technologies continue to mature.

WALKER: Testers reaping the benefits of GenAI have mastered effective prompt design. As AI integrates into testing tools, the barrier for leveraging AI will lower. However, the early adopters will hold a dominant position. I believe there will always be a place for QA engineers. QA engineers will always have a role in assuring stakeholders, fostering confidence, and applying critical thinking. Automation/testing/AI is a mechanism to provide confidence and answers. QA teams might diminish; however, there will always need to be owners of quality who make sure quality is addressed using the appropriate means.

DOHMKE: GenAI makes software more useful, so it will increase demand for software and in turn drive demand for the people who help build it. The fundamental nature of roles and skills will change as we move toward testing GenAI-powered applications. Nearly everyone involved in software testing will be using GenAI in some form. Being skilled at prompting and understanding the output of the [machine learning] model or AI assistant will move the different roles closer together.

ETHICAL CONSIDERATIONS

***COMPUTER*:** What are the ethical considerations that need to be addressed when deploying GenAI for software testing? How can organizations ensure fairness, transparency, and accountability in the testing process?

PORTER: GenAIs lack meaningful theoretical or empirical "guarantees" of many essential system properties, such as correctness, safety, fairness, high performance, and more. While their output is seductively human-like, no technology professional should be comfortable completely turning over critical functions to GenAIs. Without such guarantees in GenAIs themselves, additional safeguards will need to be built into GenAI applications. In some cases, these will be implemented as automated checks, safety shutoffs, manual reviews, and other approaches.

WALKER: The internal workings of GenAI are opaque to the user, obscuring the decision-making process (i.e., black box). Reasoning and transparency are crucial for understanding why a specific output is given from a prompt, which can then be used to ensure fairness and accountability. As AI progresses, I anticipate a growing emphasis on visualization to help communicate these algorithms' inner workings. This improved transparency could subsequently allow us to better comprehend aspects of fairness and foster a sense of accountability within these systems.

GERRARD: Setting aside the obvious challenges of using, for example, production or personal data for testing, AI may have a role in protecting sensitive data. The bigger effect will be how we measure and improve the effectiveness of testers, our developers, and our processes. The product of testing is information and only testing captures evidence of achievement. If "integrated systems intelligence" becomes available, AI can evaluate the performance of the test process against stakeholder needs for high-quality information. The value of testing is how insightful and actionable the product of testing—information—is to stakeholders.

DOHMKE: Every use case is different, but broadly speaking I would encourage organizations to look to Microsoft's Responsible AI Standard[7] when deploying GenAI for software testing. It offers a clear path for methodically evaluating critical areas, like accountability, transparency, fairness, reliability, and safety.

FIVE–TEN YEAR OUTLOOK

***COMPUTER*:** How do you see GenAI integrating into software testing processes five years in the future? 10 years?

DOHMKE: That will depend on what developers build. I believe we'll see a wave of tools that will transform every aspect of software testing within five years, if not faster. It will help with writing test cases, generate test cases automatically while checking test coverage, and identify untested areas of the codebase. It will also determine which tests to run against the set of changes, for example in a pull request, to shorten the turnaround times of large test suites. Adoption still takes time, but demand for more robust software and competitive pressures will result in GenAI being nearly universally adopted more quickly than, for example, [continuous integration/continuous deployment]. And will require little to no migration effort.

GERRARD: A tester uses their knowledge and experience, communication, and analytical skills to model usage patterns, failure modes (risks), required and conventional behavior, and scenarios to demonstrate the software "works" to enable testing stakeholders to make better-informed decisions. AI tools for testers will require integrated training data from code, changes, the old system, real-world data, usage patterns, test, and defect histories. AI could become a trusted partner of testers who explore knowledge sources and direct AI to perform much of the legwork of testing. But these tools need "integrated systems intelligence" and a focus on the thought processes of testers.

PORTER: Although GenAIs applications are impressive now and destined to improve rapidly, I think their near-term use in software testing will be limited and narrow in scope. In particular, GenAIs lack meaningful theoretical or empirical safety "guarantees." That said, in the longer term, GenAIs and future AI innovations will be used to automate more and more currently manual tasks. Most interestingly, I believe that during this transition period, GenAIs will enable technically knowledgeable people to review and curate GenAI output in ways that leapfrog their productivity over less technically knowledgeable people.

WALKER: Generative AI's future lies in specific models trained on organizational data to facilitate intelligent algorithms. The biggest barrier to that is a lack of structured data, which largely doesn't exist in the software domain. Over the next five to 10 years, I anticipate a shift toward prioritizing the harvesting of AI-training data from across the development lifecycle. Despite a current focus on models/algorithms, data are fundamental. I predict models will be trained on assets from throughout the software development lifecycle, enabling a comprehensive organizational AI. This could drive autonomous testing and quality assessment.

***COMPUTER*:** Many thanks to our panelists, who all agree that GenAI will transform the software engineering profession. In the near term, software testing will benefit from AI-based test case specification, test code authoring, and test data generation. However, test engineers will remain the ultimate authority who assure that AI-assisted software testing results in a reliable, safe, secure, and functionally correct system. The potential of GenAI for software testing, as in other disciplines, will be realized once organizations identify and overcome the limits of this technology for improving, rather than replacing, the practices of engineering.

REFERENCES

1. M. Jovanovic and M. Campbell, “Generative artificial intelligence: Trends and prospects,” *Computer*, vol. 55, no. 10, pp. 107–112, Oct. 2022, doi: 10.1109/MC.2022.3192720.
2. H.-Y. Lin, “Large-scale artificial intelligence models,” *Computer*, vol. 55, no. 5, pp. 76–80, May 2022, doi: 10.1109/MC.2022.3151419.
3. Z. Akata et al., “A research agenda for hybrid intelligence: Augmenting human intellect with collaborative, adaptive, responsible, and explainable artificial intelligence,” *Computer*, vol. 53, no. 8, pp. 18–28, Aug. 2020, doi: 10.1109/MC.2020.2996587.
4. I. Ozkaya, “Application of large language models to software engineering tasks: Opportunities, risks, and implications,” *IEEE Softw.*, vol. 40, no. 3, pp. 4–8, May 2023, doi: 10.1109/MS.2023.3248401.
5. “Your AI pair programmer.” GitHub Copilot. Accessed: Jul. 13, 2023. [Online]. Available: https://github.com/features/copilot
6. V. Murali et al., “CodeCompose: A large-scale industrial deployment of AI-assisted code authoring,” May 2023. [Online]. Available: http://arxiv.org/abs/2305.12050
7. “Microsoft responsible AI standard, v2.” Microsoft. Accessed: Aug. 30, 2023. [Online]. Available: https://blogs.microsoft.com/wp-content/uploads/prod/sites/5/2022/06/Microsoft-Responsible-AI-Standard-v2-General-Requirements-3.pdf

LUCAS LAYMAN is an assistant professor of computer science at the University of North Carolina Wilmington, Wilmington, NC 28403 USA. Contact him at laymanl@uncw.edu.

RON VETTER is a professor of computer science and founding dean of the College of Science and Engineering at the University of North Carolina Wilmington, Wilmington, NC 28403 USA. He is a Senior Member of IEEE. Contact him at vetterr@uncw.edu.

IEEE Pervasive Computing seeks accessible, useful papers on the latest peer-reviewed developments in pervasive, mobile, and ubiquitous computing. Topics include hardware technology, software infrastructure, real-world sensing and interaction, human-computer interaction, and systems considerations, including deployment, scalability, security, and privacy.

Author guidelines: www.computer.org/mc/pervasive/author.htm

Further details: pervasive@computer.org

www.computer.org/pervasive

Digital Object Identifier 10.1109/MC.2023.3338642