The Deloitte On Cloud Podcast

Nitish Jain, Managing Director, Deloitte Consulting LLP

Title: **GenAI can make quality engineering smarter and faster. Deloitte's Nitish Jain explains how**

Description: In this Knowledge Short, Deloitte's Nitish Jain explores how Generative AI (GenAI) is reshaping quality engineering through automation, intelligent test generation, and dynamic validation processes. He discusses tools and practices teams can deploy to unlock value. Nitish also explains how teams can leverage GenAI to improve speed, coverage, and adaptability. Finally, he addresses challenges in using GenAI, such as limiting bias, ensuring quality output, and building system resiliency.

Duration: **00:12:50**

Welcome to this Deloitte On Cloud Podcast Knowledge Short, where I will do a quick dive into a specific topic related to emerging technologies. By way of introduction, I am Nitish Jain, managing director at Deloitte. I have been a career consultant in quality engineering, helping our clients transform and go to market while focusing on managing quality and risk and the expectations. Thank you for joining me. Today, we are exploring how quality engineering is evolving in the era of Generative AI, why it matters, what's changing, and what you need to know. In short, we will be looking at the evolution of IT and testing, then we take a brief look of what is Generative AI, then we will look at the impact of Generative AI on quality engineering and testing. We will also visit some of the considerations in testing Generative AI systems, and then we will end with a quick summary and the key takeaways.

I will quickly start with the evolution of information technology, which is IT. And this field has undergone dramatic transformation since its inception, driven by advances in hardware, software, networking, and the needs of businesses and society. Starting from the mainframe era of the 40s through the 70s, which was characterized by large, centralized computers primarily used by governments and research institutions and large enterprises, which then gave rise to mini computers and early networking in the 70s and the 80s, which was essentially smaller, more affordable mini computers that departments and early business applications could use.

And then came the PC, or the personal computer, revolution in the 80s and the 90s that gave rise to the desktop computers and applications like word processing, spreadsheets, and early graphics and games. Who can forget the Internet and the web era of the 90s and the 2000s, which was the explosion of the Internet and the World Wide Web that allowed for global connectivity, web servers, browsers, and search engines. And then that evolved into mobile and cloud computing with the advent of smartphones, tablets, and mobile apps.

Then we went into the era of digital transformation and modern IT, which was essentially integration of digital technologies in all areas of business and society allowing for massive data collection and real-time analysis. That leads us to where we are today in the era of emerging technologies, which is a combination of quantum computing, 5G, and advanced networking and autonomous systems that are focusing on immersive experiences for work and entertainment. They also allow for sustainable IT focusing on green computing and energy efficiencies.

Likewise, as information technology has evolved, so has software testing and quality engineering. From the manual testing era of the 70s and the 80s to structured testing and test case design of the 90s, which led to the automation test cases and scripting that drove efficiencies and coverage that allowed us for agile testing in the 2000s. Which was essentially integration of testing into the agile iterative development, continuous testing, test driven development that allowed testers to become part of cross-functional teams and shift left the testing effort.

At a faster pace, things evolved, and in the 2010, we started with the DevOps and continuous testing era where testing was embedded into the development OPS and CI/CD pipelines, which allowed for continuous integration and deployment of testing. Which brings us to where we are today in a world of AI-driven and autonomous testing that leverages AI and ML to optimize and automate testing process. It also allows for self-healing tests, intelligent test case generation, predictive analytics for defect detection.

So, with all this said, what is Generative AI? In very simple terms, Generative AI refers to a category of artificial intelligence models and techniques designed to create new content, data, or solutions that resemble human-generated output. Unlike traditional AI, which typically classifies and only predicts or analyzes existing data, Generative AI can produce original materials, such as text, image, audio, video, code, and much more. Generative AI consequently has a profound impact across industries, business functions, and society. And now what I am going to get into is a little bit of how Generative AI is impacting some of these. As we are seeing, it is helping transform businesses by automating content creation.

It is also changing the software development and testing ecosystem by assisting developers in writing, reviewing, and debugging code and generating automated test cases, scripts, and maintaining the automation. Generative AI is also allowing for content creation and personalization. It's improving customer experience by conversational AI and personalized recommendations, data analysis and decision support by summarization of large volumes of data and scenario simulation. And it's also giving rise to innovation and new business models by rapid prototyping and creative collaboration.

So, this brings us to the fundamental question of what does Generative AI mean in the context of quality engineering and software testing? There are two aspects of that. One is how can Generative AI help the process of testing and quality engineering and the second is how do you test Generative AI systems. So, let me touch briefly about how Generative AI is advancing the software testing and quality engineering ecosystem where it is transforming software testing by making it faster, smarter, and more adaptive. It's enabling automated test creations where Generative AI models analyze the requirements and user stories and automatically generate test cases.

It's also allowing for self-healing test scripts where AI is able to detect changes in the application and automatically update test scripts to prevent failures. It's also helping with intelligent defect prediction as a large volume of data is able to be analyzed, historical data included for code changes and test results and be very focused about where testing needs to occur. Natural language test creation is allowing testers and business analysts to describe and write test cases in plain English, and then Generative AI can convert them to executional scripts. Test data generation is an important area where Generative AI is helping create realistic and more diverse data sets that are based on production patterns.

GenAI is also allowing for exploratory and autonomous testing and the overall optimization and prioritization of quality engineering and testing. So, in summary, the key benefits of GenAI in the testing process are speed, coverage, adaptability, and collaboration. If you look at the second aspect of how you test Generative AI systems, then in essence, testing Generative AI systems is very different from traditional software testing due to the probabilistic data-driven and often non-deterministic nature of the ecosystem. Hence it is very important to define clear objectives and metrics. In essence, what are you expecting the GenAI system to do? As far as test data preparation is concerned, it is imperative that real-world scenario, Edge cases are used where possible to reference outputs for comparison.

When you think about functionally testing GenAI systems, it is important to check outputs and make sure that they are relevant, correct, and meet requirements. And at the same time, the quality evolution is important where you use expert or crowdsource reviewers to rate output quality, and you also use aspects of A/B testing, compare different models, versions, and real-time users. A very important and evolving aspect of testing GenAI systems is testing it for bias, fairness, and safety testing where you test it for unwanted bias in its outputs, you make sure that automated tools allow you to compare it with manual reviews, and ensure outputs comply with legal and ethical standards.

GenAI systems should also be tested for robustness and stress testing where tricky and intentionally misleading inputs are provided to make sure that the system still continues to operate in the way it's designed to be. The user experience is an important aspect for clarity, usefulness, and then the feedback loops. Obviously, last but not the least, the monitoring in production for ongoing evolution and the logging and alert systems allow it for quicker response. So, if I were to summarize this, the key takeaways from GenAI as it applies to the process of software testing and testing of GenAI-based systems is.

In closing, artificial intelligence and machine learning have been changing the way software is developed, tested, and maintained. The discussion about their influence on software testing is no longer about if it will happen, but about the extent of it. This will depend on the level of investment in AI and ML technologies and the willingness of organizations to adapt them. As enterprises further incorporate digital technologies into processes and businesses, quality engineering practices are transforming to keep pace.

Thanks for listening to this On Cloud podcast Knowledge Short, I am Nitish Jain. If you enjoyed this podcast, make sure to like us, rate us, and subscribe. Until next time, best of luck with your cloud journey and stay safe.

Operator:
This podcast is produced by Deloitte. The views and opinions expressed by podcast speakers and guests are solely their own and do not reflect the opinions of Deloitte. This podcast provides general information only and is not intended to constitute advice or services of any kind. For additional information about Deloitte, go to Deloitte.com/about.

This publication contains general information only and Deloitte is not, by means of this publication, rendering accounting, business, financial, investment, legal, tax, or other professional advice or services. This publication is not a substitute for such professional advice or services, nor should it be used as a basis for any decision or action that may affect your business. Before making any decision or taking any action that may affect your business, you should consult a qualified professional advisor.

Deloitte shall not be responsible for any loss sustained by any person who relies on this publication.

Visit the On Cloud library
www.deloitte.com/us/cloud-podcast

About Deloitte

As used in this podcast, “Deloitte” means Deloitte Consulting LLP, a subsidiary of Deloitte LLP. Please see www.deloitte.com/us/about for a detailed description of our legal structure. Certain services may not be available to attest clients under the rules and regulations of public accounting. Please see www.deloitte.com/about to learn more about our global network of member firms.
Copyright © 2025 Deloitte Development LLC. All rights reserved.