



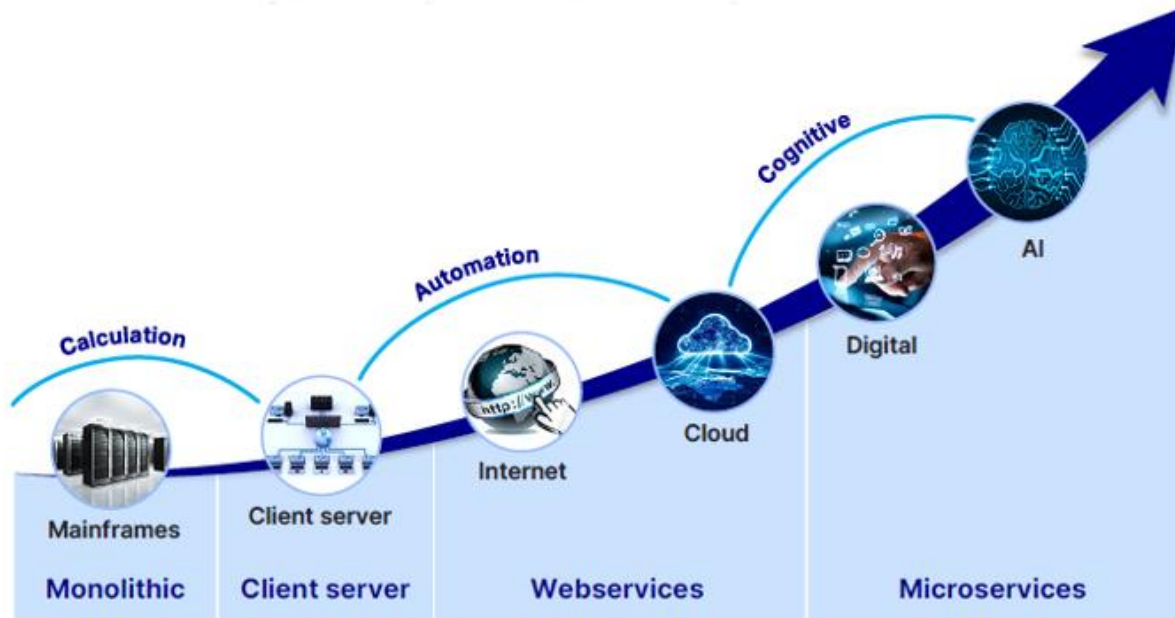
AI will transform how application development and testing is conducted?

Application Delivery Track

Kim Ries Westergaard
Lead Solution Consultant | DevOps Cloud Nordics



Generative AI – All on board !



George Colony
CEO Forrester

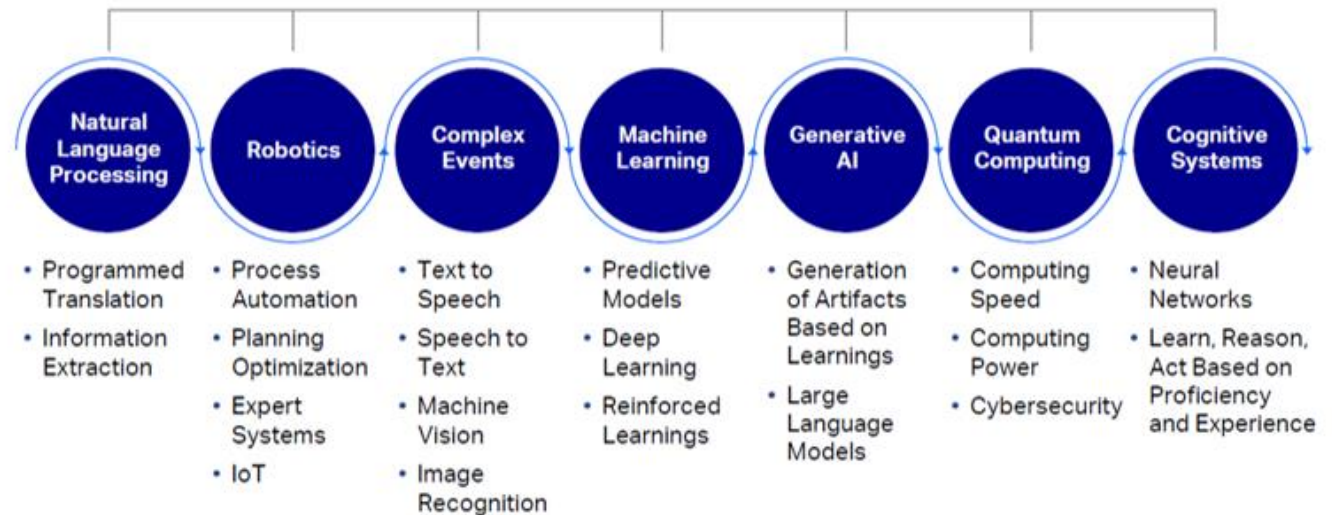
"Generative AI will replace much of the web."

Chris Howard
Distinguished VP Analyst Gartner

"Gartner sees generative AI becoming a general-purpose technology with an impact like that of the steam engine, electricity and the internet."

Mark J. Barrenechea
CEO & CTO OpenText

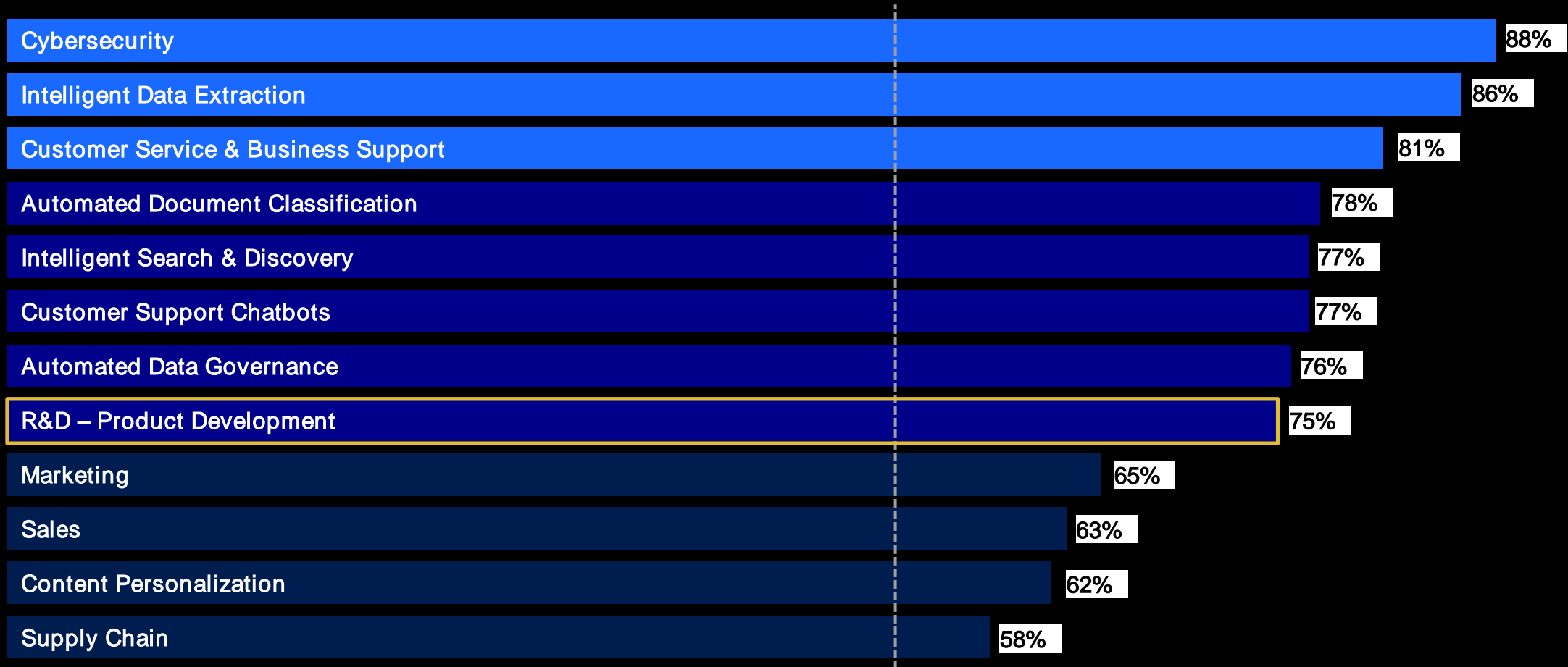
"AI will impact what it means to be a company, what it means to work, and what it means to be human."



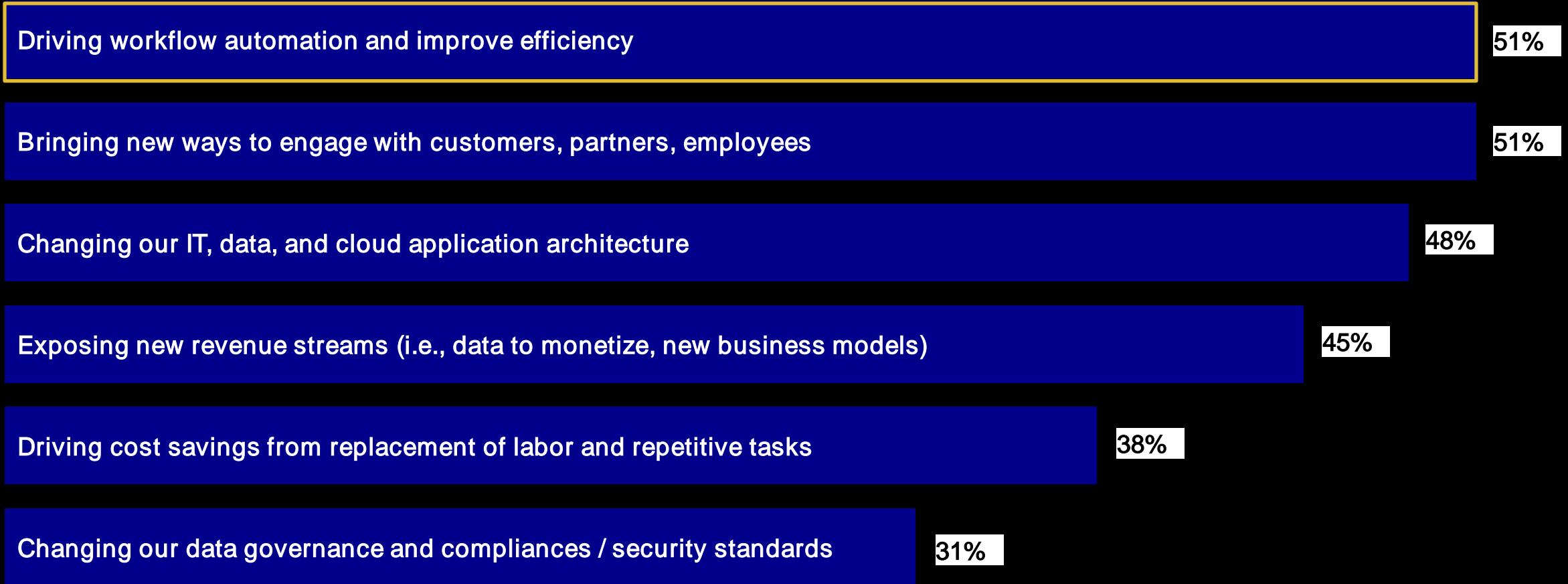
Where are you on generative AI adoption curve?



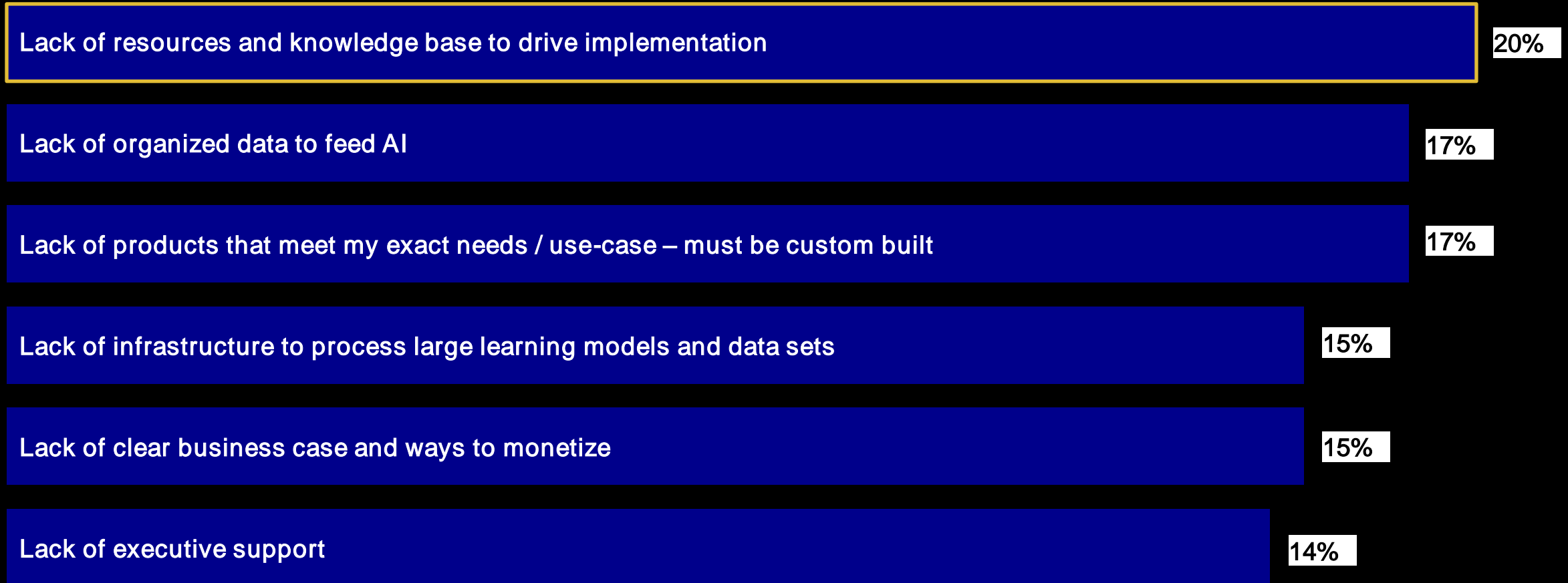
Which AI based solutions are most essential for you?



How will AI make an impact for your business?

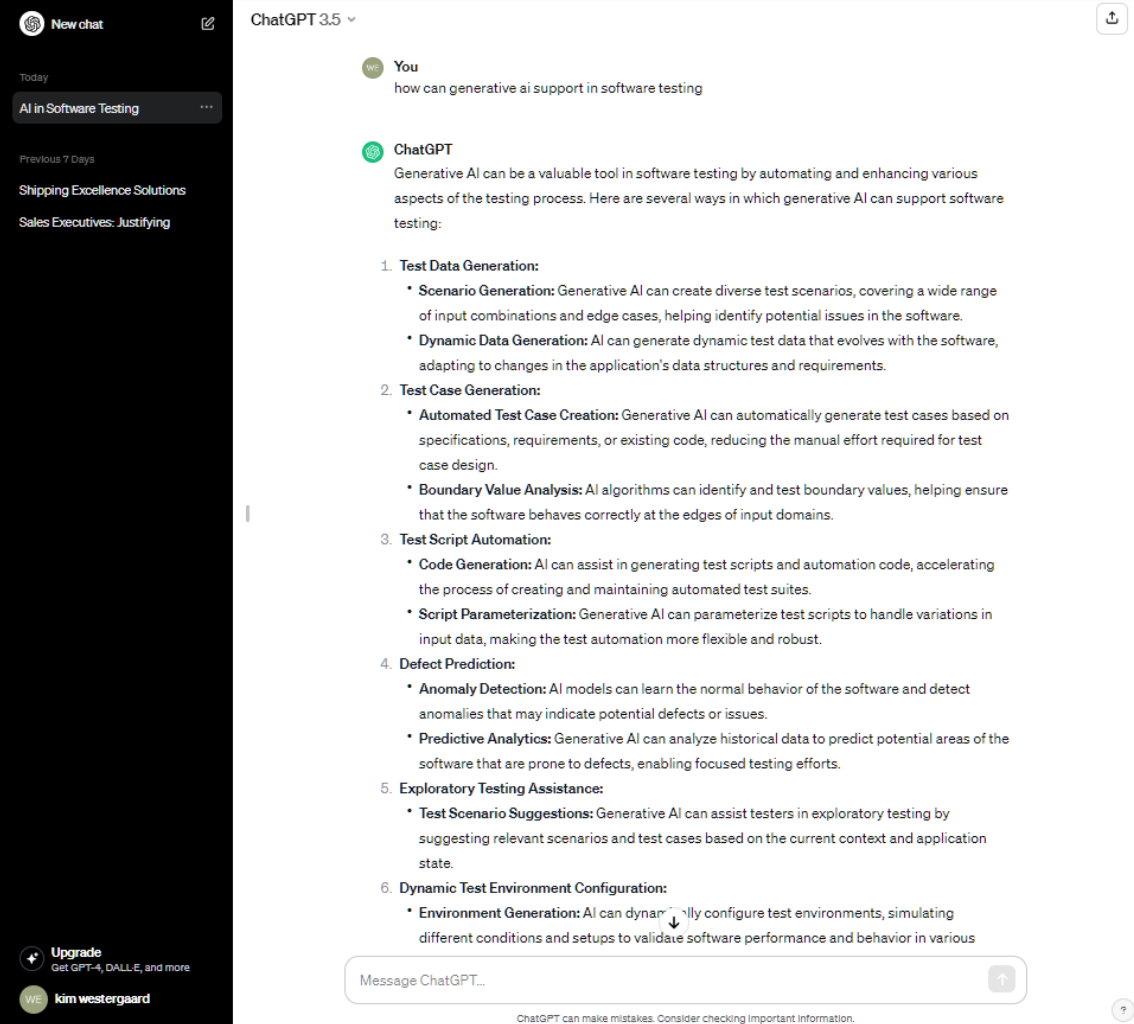


What do you anticipate as hurdles to benefiting from AI?



Power of Generative AI and LLM

-how can generative ai support in software testing?



The screenshot shows a chat window for ChatGPT 3.5. The user asks, "how can generative ai support in software testing?". The AI responds with a detailed list of six ways generative AI can support software testing:

You
how can generative ai support in software testing

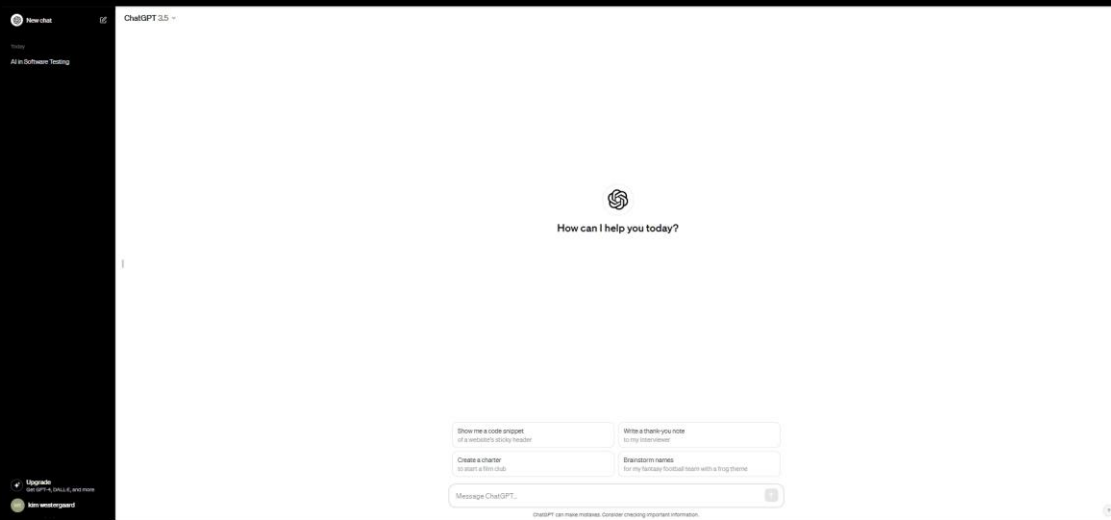
ChatGPT
Generative AI can be a valuable tool in software testing by automating and enhancing various aspects of the testing process. Here are several ways in which generative AI can support software testing:

- 1. Test Data Generation:**
 - **Scenario Generation:** Generative AI can create diverse test scenarios, covering a wide range of input combinations and edge cases, helping identify potential issues in the software.
 - **Dynamic Data Generation:** AI can generate dynamic test data that evolves with the software, adapting to changes in the application's data structures and requirements.
- 2. Test Case Generation:**
 - **Automated Test Case Creation:** Generative AI can automatically generate test cases based on specifications, requirements, or existing code, reducing the manual effort required for test case design.
 - **Boundary Value Analysis:** AI algorithms can identify and test boundary values, helping ensure that the software behaves correctly at the edges of input domains.
- 3. Test Script Automation:**
 - **Code Generation:** AI can assist in generating test scripts and automation code, accelerating the process of creating and maintaining automated test suites.
 - **Script Parameterization:** Generative AI can parameterize test scripts to handle variations in input data, making the test automation more flexible and robust.
- 4. Defect Prediction:**
 - **Anomaly Detection:** AI models can learn the normal behavior of the software and detect anomalies that may indicate potential defects or issues.
 - **Predictive Analytics:** Generative AI can analyze historical data to predict potential areas of the software that are prone to defects, enabling focused testing efforts.
- 5. Exploratory Testing Assistance:**
 - **Test Scenario Suggestions:** Generative AI can assist testers in exploratory testing by suggesting relevant scenarios and test cases based on the current context and application state.
- 6. Dynamic Test Environment Configuration:**
 - **Environment Generation:** AI can dynamically configure test environments, simulating different conditions and setups to validate software performance and behavior in various

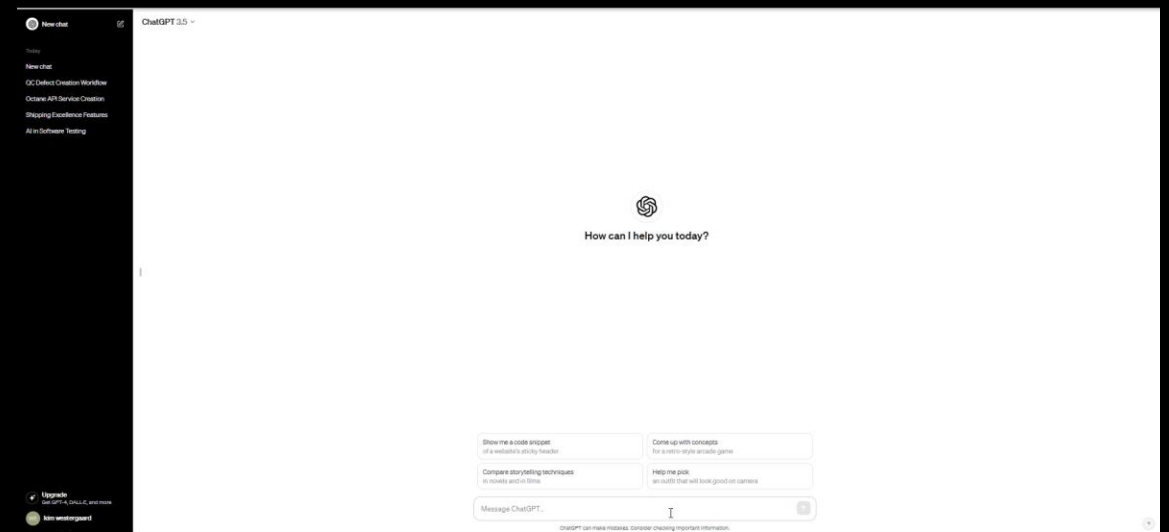
Message ChatGPT...

ChatGPT can make mistakes. Consider checking important information.

Use of open LLMs like ChatGPT



Generate Tests

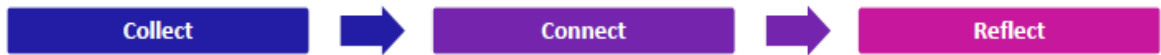
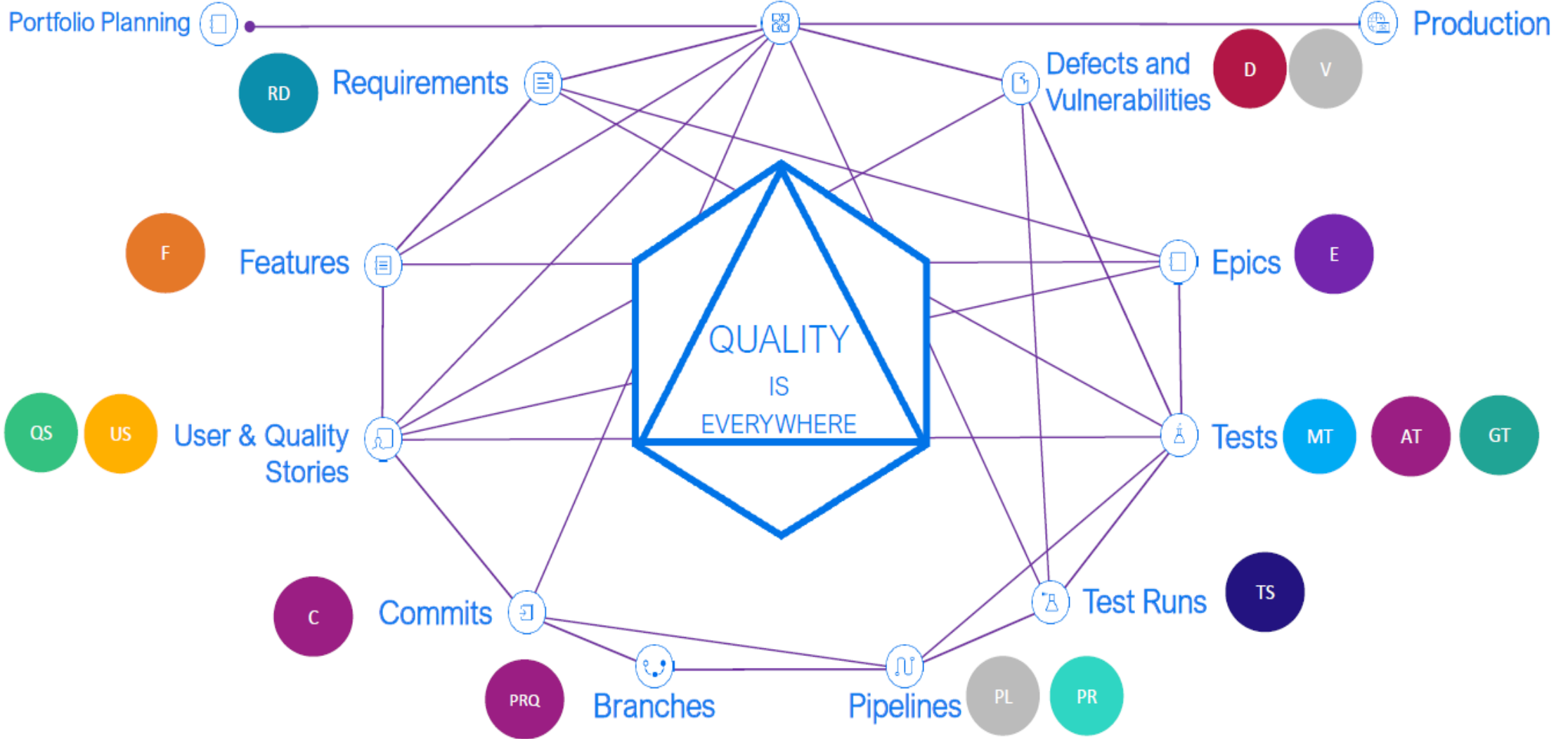


Generate Testdata

But what is missing for an AI to become
truly powerful

CONTEXT

Application Modules





AI Embedded in Testing Tools

Disclaimer

- Certain statements in this presentation constitute forward-looking statements or forward-looking information within the meaning of applicable securities laws (“forward-looking statements”). Such forward-looking statements involve known and unknown risks, uncertainties and other factors that may cause the actual results, performance or achievements of OpenText, or developments in OpenText’s business or in its industry, to differ materially from the anticipated results, performance, achievements or developments expressed or implied by such forward-looking statements. Forward-looking statements include all disclosure regarding possible events, conditions or results of operations that is based on assumptions about future economic conditions and courses of action. Forward-looking statements may also include any statement relating to future events, conditions or circumstances. OpenText cautions you not to place undue reliance upon any such forward-looking statements, which speak only as of the date they are made. Forward-looking statements relate to, among other things, changes in the ECM market; the market focus of OpenText, OpenText’s revenue mix and margin targets; OpenText’s operations priorities; and OpenText’s strategy for its products and solutions. The risks and uncertainties that may affect forward-looking statements include, among others, the completion and integration of acquisitions, the possibility of technical, logistical or planning issues in connection with deployments, the continuous commitment of OpenText's customers, demand for OpenText's products and other risks detailed from time to time in OpenText's filings with the Securities and Exchange Commission and Canadian provincial securities regulators, including OpenText's Annual Report on Form 10-K and the Quarterly Report on Form 10-k. Forward-looking statements are based on management’s current plans, estimates, projections, beliefs and opinions, and the Company does not undertake any obligation to update forward-looking statements should assumptions related to these plans, estimates, projections, beliefs and opinions change.
-

What can we imagine from a software testing tool perspective

1 AI
Assistant

2 AI
Assisted
Analytics

3 AI
Automation



1 AI Assistant

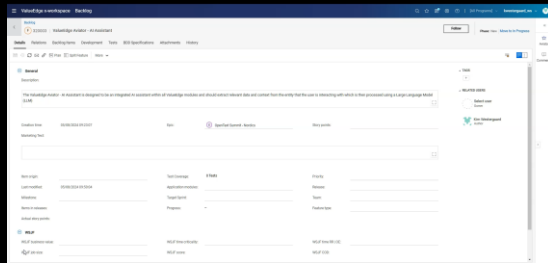
Overview: In a fast-paced corporate environment, efficiency and productivity are paramount. To streamline tasks and enhance employee effectiveness, the implementation of an AI assistant proves to be a valuable solution. This use case explores the deployment of an AI assistant in a corporate setting to assist on routine tasks, help with complex tasks and augment decision-making processes.

AI Assistant

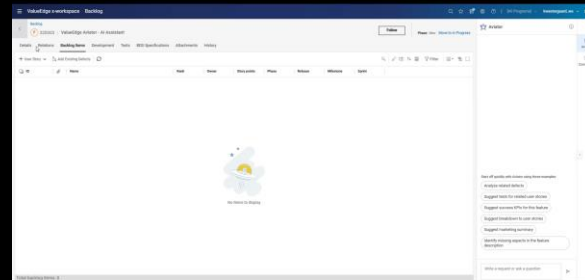
Aviator Smart Assistant



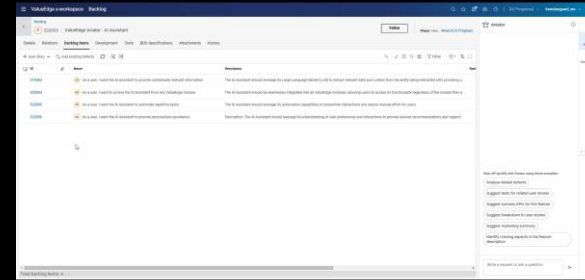
Your new virtual advisor – ask me anything! In context, beyond the visible



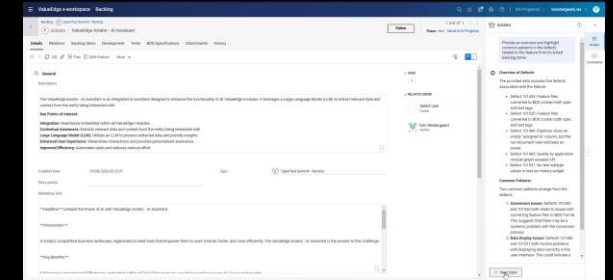
Feature Creation



User Stories and
Development Tasks
Creation



Tests Creation



Analyse User Stories
and Defects

AI Aviator Smart Assistant

Your new virtual advisor – ask me anything! In context, beyond the visible



ValueEdge x-workspace Backlog

Backlog

320003 | ValueEdge Aviator - AI Assistant

Follow Phase: New Move to In Progress

Details Relations **Backlog Items** Development Tests BDD Specifications Attachments History

+ User Story Add Existing Defects

ID	Name	Description	Rank
319004	US As a user, I want the AI Assistant to provide contextually relevant information.	The AI Assistant should leverage its Large Language Model (LLM) to extract relevant data and context from the entity being interacted with, providing u...	
320004	US As a user, I want to access the AI Assistant from any ValueEdge module.	The AI Assistant should be seamlessly integrated into all ValueEdge modules, allowing users to access its functionality regardless of the module they a...	
320005	US As a user, I want the AI Assistant to automate repetitive tasks.	The AI Assistant should leverage its automation capabilities to streamline interactions and reduce manual effort for users.	
320006	US As a user, I want the AI Assistant to provide personalized assistance.	Description: The AI Assistant should leverage its understanding of user preferences and interactions to provide tailored recommendations and support	

Aviator

Review the user stories and propose additional tests for the feature. Each test should consist of 5 steps with a detailed description

Additional Tests for ValueEdge Aviator - AI Assistant Feature:

Test 1: Automation of Repetitive Tasks

- Step 1:** Create a ValueEdge module with a repetitive task, such as data entry.
- Step 2:** Enable the AI Assistant within the module.
- Step 3:** Configure the AI Assistant to automate the repetitive task.
- Step 4:** Execute the task and verify that the AI Assistant successfully automates it.
- Step 5:** Measure the time saved by automating the task.

Test 2: Contextually Relevant Information

- Step 1:** Create a ValueEdge module with a specific entity, such as a customer record.
- Step 2:** Enable the AI Assistant within the module.
- Step 3:** Interact with the entity and observe the AI Assistant's response.
- Step 4:** Verify that the AI Assistant provides relevant data and context from the entity.
- Step 5:** Evaluate the accuracy and completeness of the information provided.

Test 3: Personalized Assistance

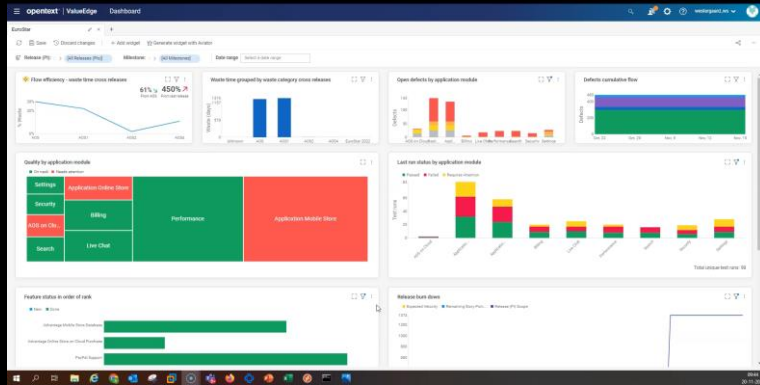
+ New topic

Total backlog items: 4

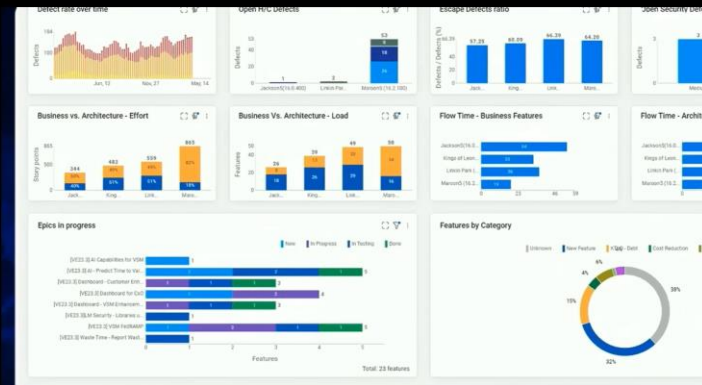
2 AI Assisted Analytics

AI analytics offers a range of benefits, including enhanced data processing capabilities, improved accuracy, automation of repetitive tasks, and the ability to uncover valuable insights. These advantages contribute to more efficient and effective decision-making processes for organizations across various industries.

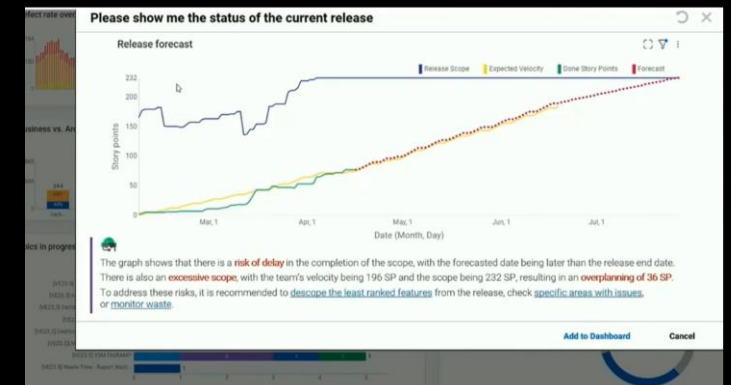
AI Assisted Analytics



Prompting instead of configurations

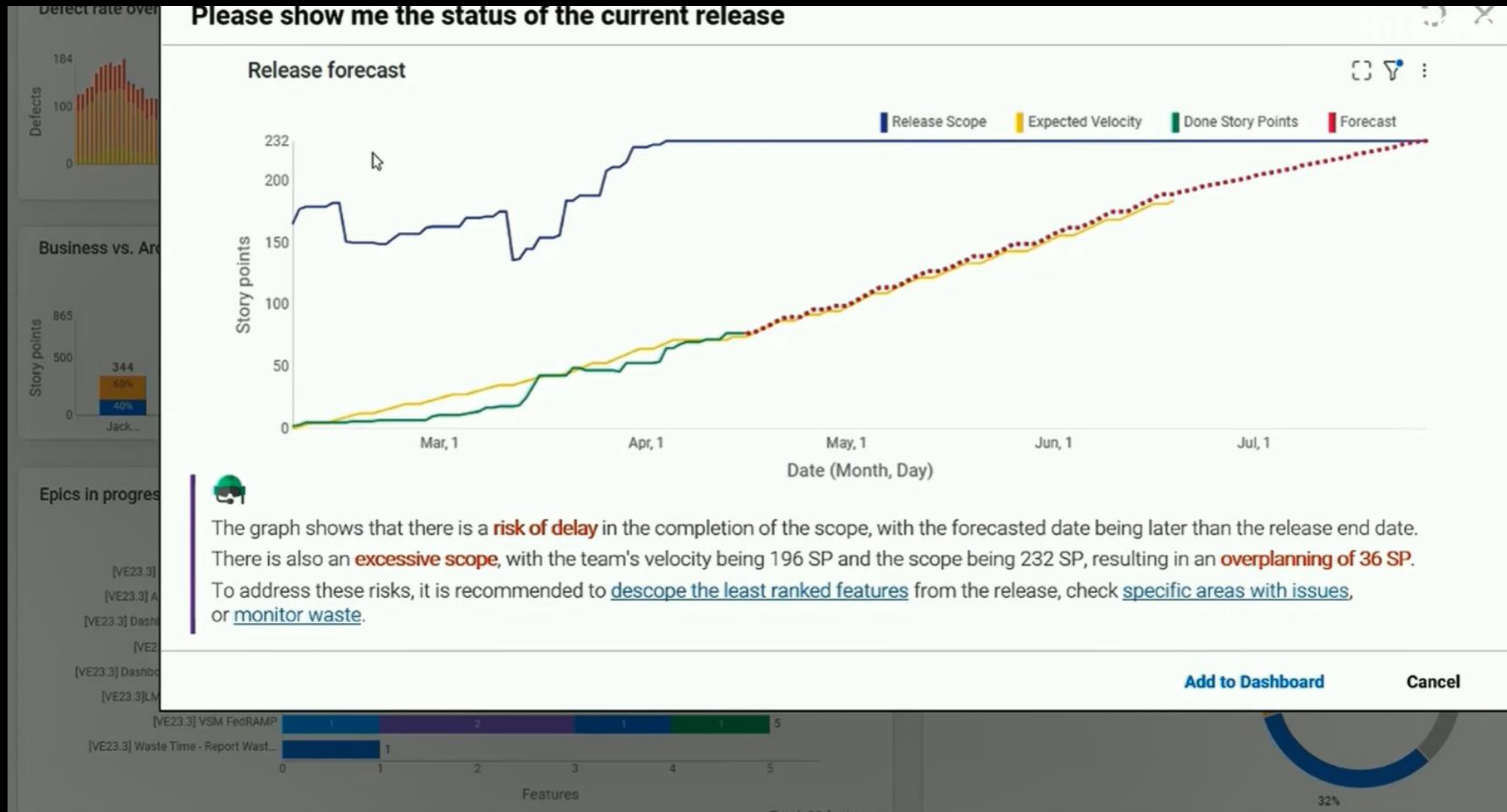


Advanced forecasting



Correlation of data to pinpoint problem areas

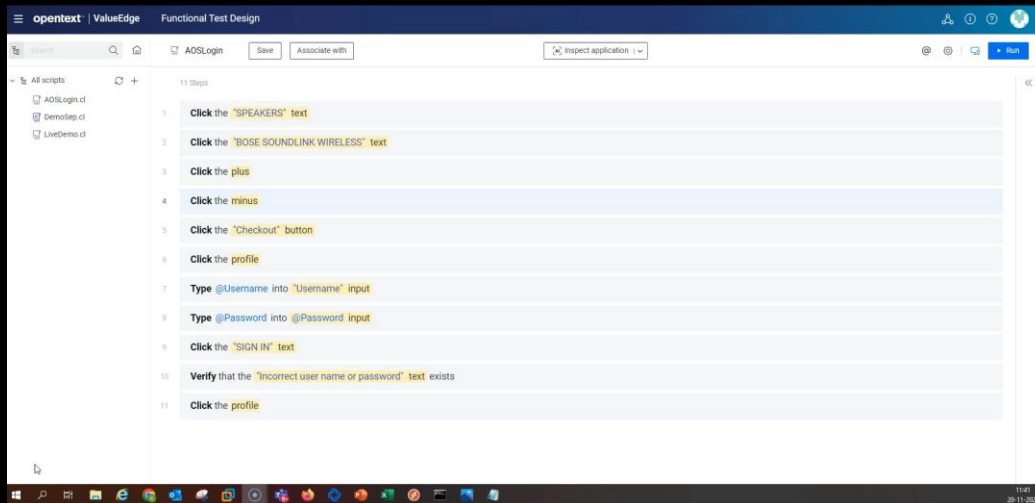
AI Assisted Analytics



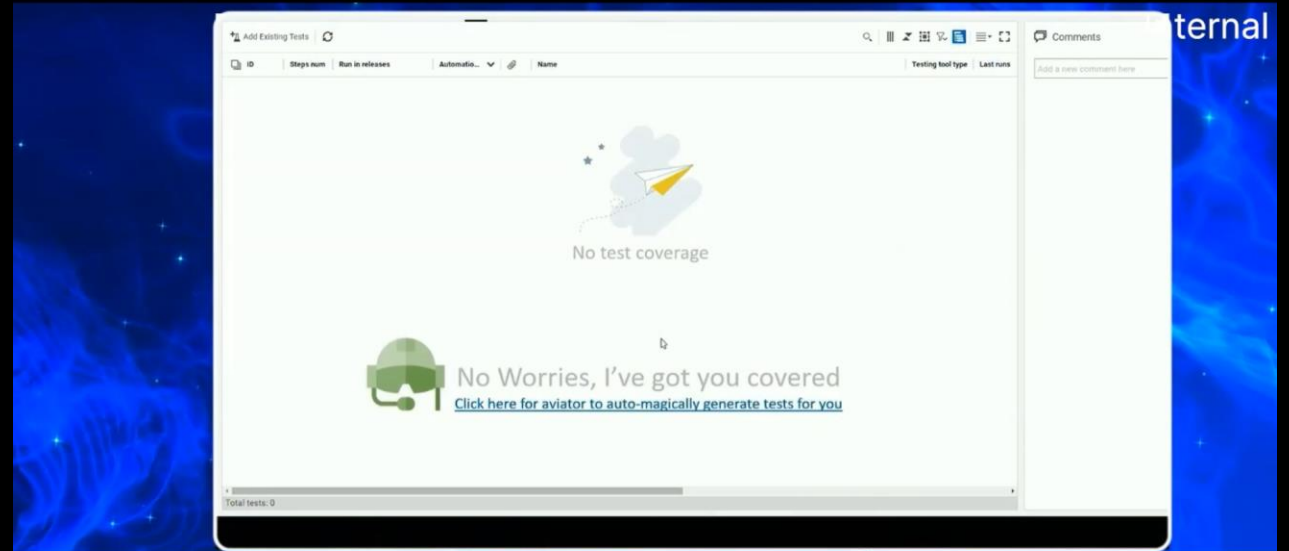
3 AI Automation

Integrating AI into automated testing processes offers advantages such as increased efficiency, broader test coverage, adaptability to changes, and intelligent decision-making. These benefits contribute to the overall effectiveness of the testing lifecycle, enabling organizations to deliver high-quality software in a timely manner.

AI Automation



Codeless AI based test



Creation of automated AI based test

AI Automation

opentext | ValueEdge | Functional Test Design

Inspect

Live applications | Mockup images

Show Visual Elements Texts

Select a different application | Re-inspect |

11 Steps

- 1 Click the
- 2 Click the
- 3 Click the
- 4 Click the
- 5 Click the
- 6 Click the
- 7 Type @U
- 8 Type @P
- 9 Click the
- 10 Verify the
- 11 Click the

The screenshot displays a functional test design interface for an e-commerce website. The main content area shows a grid of product images with labels: 'SPEAKERS' (a circular speaker), 'TABLETS' (a tablet), 'LAPTOPS' (a laptop), 'MICE' (a computer mouse), and 'HEADPHONES' (a man wearing headphones). A central modal window is overlaid on the 'TABLETS' image, containing a login form for 'advantage'. The form includes a 'SIGN IN WITH FACEBOOK' button, an email input field, a password input field, a 'SIGN IN' button, and a 'FORGOT YOUR PASSWORD?' link. The interface also features a top navigation bar with 'opentext | ValueEdge | Functional Test Design' and a right sidebar with utility icons. A left sidebar lists 11 test steps, with the first four being 'Click the' and the last being 'Click the'.

opentext™

**Come have a chat with me or reach out
to one of my 20.000+ colleagues around the
world**

<https://www.opentext.com/>



opentext™

