zinnov

NESS

HARNESSING THE POWER OF GENERATIVE AI IN TRANSFORMING SOFTWARE ENGINEERING PRODUCTIVITY

TABLE OF CONTENTS

01

Executive Summary 03

02

Real World, Real Data: A view on High Impact Areas of Generative AI in Product Development 05

Generative AI has a significant impact on repeatable sustenance activities and reduces knowledge barriers, thus increasing the potential for globalizing a higher number of products 05

Generative AI increases the productivity gap between junior and senior developers, expediting the shift from traditional pyramid teams to diamond or flatter pod structures 08

Generative AI has a limited impact on productivity in high-complexity coding environments prompting continued reliance on skilled engineers for desired output in such scenarios 11

Generative AI enhances engineer engagement, fosters collaborative team dynamics and collective problem-solving, and reduces attrition 13

03

Implications of Generative AI on Transforming Engineering Productivity 15

Framework for Engineering Productivity Transformation 15

Launching a Generative AI Transformation Journey 17

01
EXECUTIVE SUMMARY

There is ample literature currently postulating the transformative power of Generative AI in improving engineering productivity. While some have predicted productivity gains upwards of 50%, others, slightly lower. However, most of these publications have taken an outside-in view of the potential, without a detailed empirical analysis of real-life engineers working in live engineering environments.

Ness and Zinnov have partnered to conduct a detailed analysis of the impact of adopting Generative AI tools like Copilot and CodeWhisperer, in actual development environments. Our detailed analysis of 100+ software engineers across a wide variety of use cases and development environments suggests that Generative AI could fundamentally improve software engineering productivity. However, the productivity gains may vary significantly based on several factors such as the engineer's experience, task complexity, and development environment. Furthermore, adoption of these tools may reduce the barriers to knowledge transfer between various development teams, improving the ability to operate more effectively in a distributed global model.

For the study, Ness leveraged its proprietary dynamic data-driven engineering platform, Matrix, to monitor essential engineering Key Performance Indicators (KPIs) such as quality, productivity, responsiveness, and code quality. The study highlighted that Generative AI has transformative potential when used in the appropriate way. Generative AI's impact on task completion time and engineer engagement varied, depending on factors such as the seniority of the engineer, task type, and code complexity.

KEY OBSERVATIONS	IMPACT ON PRODUCT DEVELOPMENT	OUTCOME
Generative AI (GAI) has a significant impact on repeatable sustenance activities and reducing knowledge barriers...	...increasing the potential for globalizing higher number of products	**70% reduction in task completion time for existing code updates**
Generative AI increases the productivity gap between junior and senior developers...	...expediting the shift from traditional pyramid teams to diamond or flatter pod structures	**48% reduction in task completion time for senior engineers**
Generative AI has limited impact on productivity in high-complexity coding environments...	...prompting continued reliance on skilled engineers for desired output in such scenarios	**~10% reduction in task completion time for high code complexity**
Generative AI enhances engineer engagement...	...fostering collaborative team dynamics and collective problem-solving, that reduce attrition	**70% of the engineers reported improved engagement**

Generative AI also assists in knowledge management and risk management. It will help global engineering teams to collaborate better and create a cohesive, efficient, and globally integrated product development environment. Engineering leaders can evaluate ways to enhance the globalization of product development by effectively leveraging Generative AI.

These findings have significant implications for CTOs of large enterprises, heads of global software engineering Captives (or Global Capability Centers [GCCs]), and CEOs of software product companies to transform engineering productivity. However, the findings also reveal that Generative AI needs a holistic transformation approach that is integrated with existing engineering processes, methods, and tools that are in use in the industry. Generative AI serves as a powerful catalyst for transforming how engineers work and collaborate, both in physical and virtual environments. Therefore, to unleash the full power of AI, it needs to be integrated with existing engineering processes and tools, taking into account the overall distributed development landscape of an organization.

This whitepaper proposes a methodology to maximize Generative AI's impact on engineering productivity through a comprehensive approach that considers distributed development environment, engineering processes, methods, and tools.

02

REAL WORLD, REAL DATA: A VIEW ON HIGH IMPACT AREAS OF GENERATIVE AI IN PRODUCT DEVELOPMENT

Generative AI tools have been systematically integrated into Ness Digital Engineering's real-time customer projects, involving a diverse team of 100+ engineers with varying experience levels. These projects encompass a wide range of business scenarios, spanning new product development, proof of concept projects, and existing product enhancement. Engineers on these projects undertake a variety of tasks, including code development, testing, bug fixing, and code refactoring, tailored to the unique requirements of each customer project.

100+ ENGINEERS

45+ Jr Engineers
55+ Sr Engineers

Working on 5+ customer engagements

GEO DISTRIBUTION

30% India
70% Eastern Europe

CUSTOMER ENGAGEMENTS

5+

NUMBER OF SPRINTS

14+ Sprints

2-week sprints, 7 before and 7+ after Generative AI implementation

TASK TYPES

20% Bug fixes and updates
30% Code refactoring
20% Legacy Code Development
30% New Code Development

CODE COMPLEXITY

17% Complex
25% Simple
58% Medium

To gauge Generative AI's impact, the study meticulously collected and analyzed product development metrics spanning 7 sprints, before and after its implementation. This quantitative approach allows for the measurement of Generative AI's influence on task completion time, providing valuable insights on its overall project efficiency-enhancing capabilities.

Concurrently, a comprehensive survey evaluated the engineers' engagement levels. The survey findings enrich our comprehension of the engineers' experience with Generative AI, shedding light on the human facets of technology adoption.

GENERATIVE AI HAS A SIGNIFICANT IMPACT ON REPEATABLE SUSTENANCE ACTIVITIES AND REDUCES KNOWLEDGE BARRIERS, THUS INCREASING THE POTENTIAL FOR GLOBALIZING A HIGHER NUMBER OF PRODUCTS.

The impact of Generative AI on task completion time was highest for projects that involved repeatable sustenance activities including existing code updates (bug fixes, code refactoring, legacy code development, and code maintenance). Non-sustenance activities that included developing new code, and integrations with new systems and products, had a very low impact on task completion time with Generative AI adoption.

EXISTING CODE UPDATES

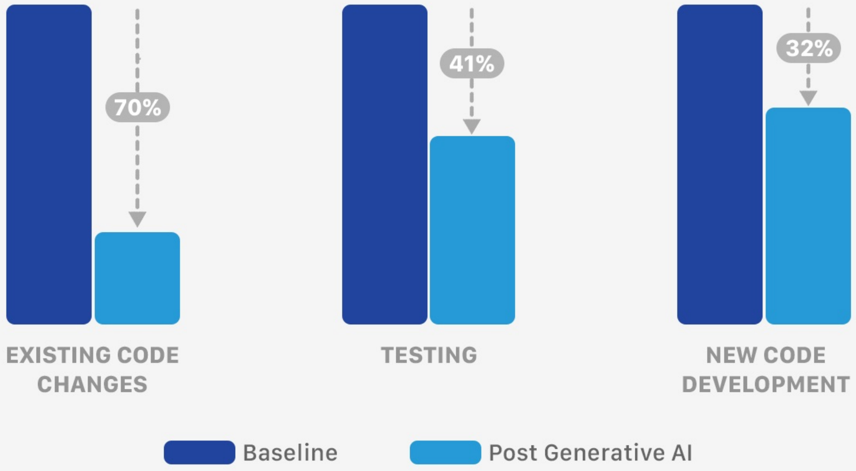
Generative AI tools significantly accelerated task completion time, particularly when engineers were engaged in modifying existing code. Generative AI was effective at reviewing and fixing issues in the code rather than suggesting new lines in the early stages of a project. However, the quality of its suggestions improved as the codebase grew.

Engineers reported the areas below to have the maximum impact when performing existing code changes:

Effective Reuse of Existing Functions:
Generative AI tools demonstrated remarkable competence in efficiently utilizing existing functions within the codebase. Engineers found that these tools eliminated the need to constantly reference existing code libraries or search for solutions online when creating new functions, thereby accelerating the development process.

Code Refactoring Support:
Ness' engineers reported that Generative AI tools were invaluable in suggesting improvements for code performance and aiding in the code refactoring process. The tools provided insights and recommendations for optimizing code structure and efficiency, contributing to the overall code quality and performance enhancement.

% REDUCTION IN TASK COMPLETION TIME WAS HIGHER FOR EXISTING CODE CHANGES



SUSTENANCE ACTIVITIES

- **Existing Code Changes**
  - Bug fixes and updates
  - Code refactoring
  - Legacy code development
- **Code Maintenance**

NON-SUSTENANCE ACTIVITIES

- New code development
- Integration with new systems and products

Generative AI's ability to reduce knowledge barriers is higher for sustenance activities...

SUSTENANCE
(Existing Code Changes & Code Maintenance)

- Providing code suggestions that are aligned with the existing code
- Coding pattern recognition and understanding while performing existing code changes
- Provide better understanding of existing code, functions, libraries, and frameworks

Generative AI has a limited impact on Non-sustenance activities

NON-SUSTENANCE
(New Code Development)

- Limited understanding of project context
- Limited availability of training data for new code development

TESTING

Generative AI has a high impact on accelerating the testing activity. Engineers reported the areas below to have the maximum impact while performing testing.

Test Case Generation:

Generative AI tools were able to automatically create test cases by reading the existing codebase. Test cases generated by the Generative AI tools captured some of the scenarios that the engineers hadn't considered.

Dummy Data Creation:

Generative AI tools were also effective in creating relevant dummy data for testing. This significantly reduced the time taken for the testing process.

NEW CODE DEVELOPMENT

While Generative AI tools were instrumental in creating entirely new code, they exhibited a relatively lower percentage of reduction in task completion time when compared to their impact on existing code updates. The limited availability of training data and limited understanding of the project context were some of the reasons why Generative AI tools had a limited impact on new code development.

QUICKER ONBOARDING FOR NEW ENGINEERS

Generative AI offers **code recommendations aligned with the existing codebase, minimizing onboarding time for new engineers**

CONSISTENCY ACROSS DISTRIBUTED TEAMS

By adhering to coding standards, Generative AI ensures **consistency across distributed teams, promoting uniformity in coding practices**

IMPROVED QUALITY OF DOCUMENTATION

Generative AI can generate high-quality documentation, making it **easier for distributed teams to understand code functionality, reducing communication gaps**

IMPROVED TEAMWORK AND COLLABORATION

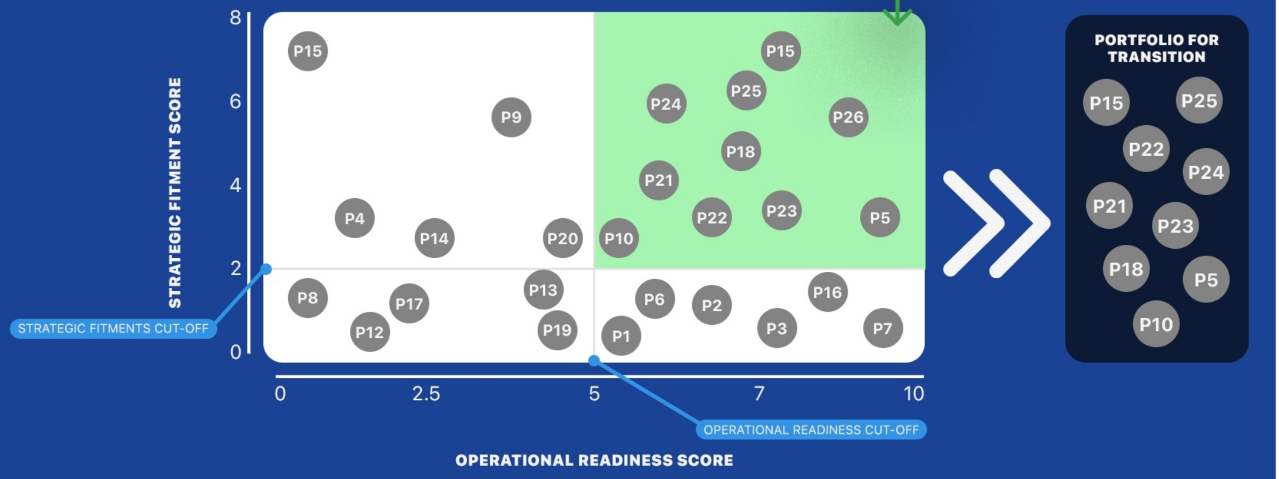
Generative AI fosters collaboration by **assisting in the comprehension of existing code, easing teamwork for distributed development**

ILLUSTRATIVE OUTPUT

DELIVERABLES

- Prioritized list of products suitable for continuation, transition to other global centers and outsourcing
- Potential challenges and drivers

Increase in the number of products prioritized for transition to other global locations and outsourcing post Generative AI



RESTRUCTURED PORTFOLIO FOR EMERGING AND ESTABLISHED LOCATIONS BASED ON:

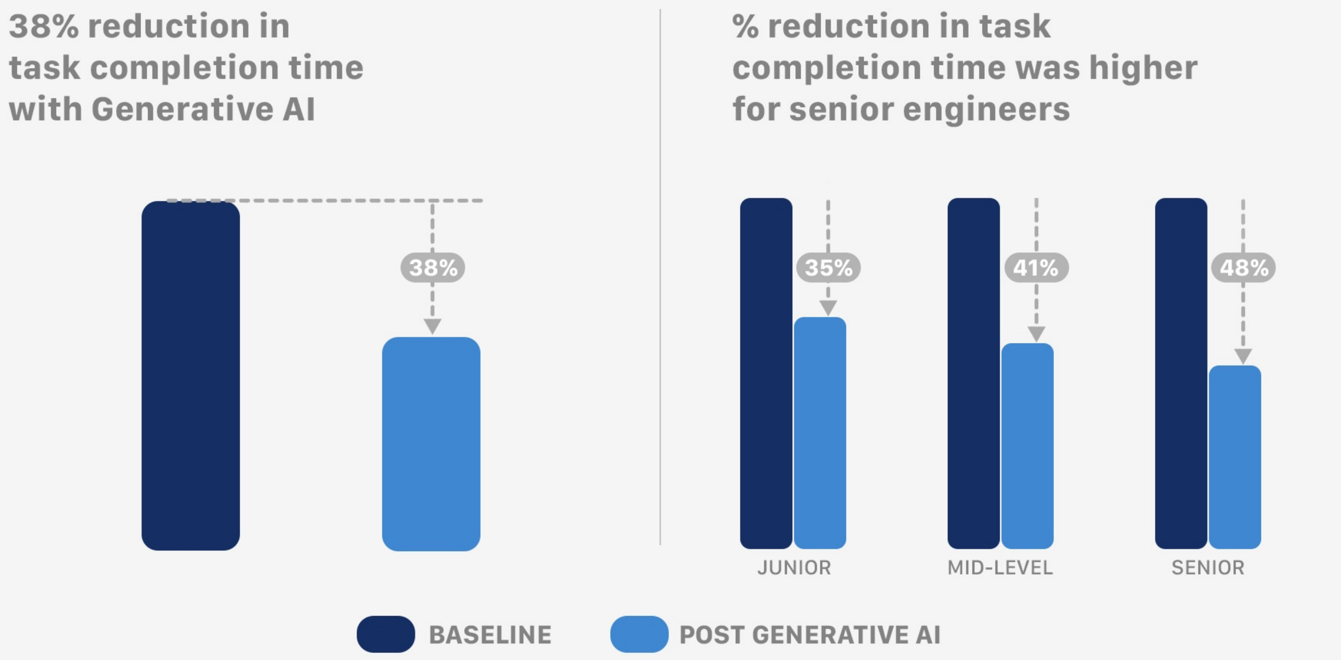
- Generative AI maturity
- Product roadmap
- Skill availability in emerging locations
- Training timelines for technology and domain skills

- Customer complexity
- Infrastructure complexity
- Regulatory requirements

Generative AI emerges as a catalyst for globalizing products by facilitating efficient and collaborative software development across distributed teams. Its ability to provide code recommendations aligned with the existing codebase significantly reduces onboarding time for new engineers, ensuring seamless integration. Moreover, Generative AI's adherence to coding standards promotes consistency across distributed teams, fostering uniformity in coding practices and simplifying cross-team collaboration. The technology's capacity to generate high-quality documentation serves as an asset, enhancing the understanding of code functionality among dispersed teams and mitigating communication gaps. Ultimately, Generative AI's role in code comprehension and documentation creation not only streamlines development processes but also expands the scope for globalizing products by facilitating effective collaboration within distributed product development teams.

GENERATIVE AI INCREASES THE PRODUCTIVITY GAP BETWEEN JUNIOR AND SENIOR DEVELOPERS, EXPEDITING THE SHIFT FROM TRADITIONAL PYRAMID TEAMS TO DIAMOND OR FLATTER POD STRUCTURES.

Generative AI implementation enabled a 38% reduction in task completion time. However, this reduction in task completion time is higher for senior engineers versus junior engineers.



38% reduction in task completion time with Generative AI

% reduction in task completion time was higher for senior engineers

Senior engineers' commit rate increased at a higher pace compared to junior engineers as the sprints progressed. In fact, senior engineers were able to achieve higher impact with Generative AI, due to the following reasons:

Ability to understand and review code suggestions provided by Generative AI: Senior developers stated that they were able to better understand the code suggestions and were able to rectify any mistakes before committing the code in the repository. On the other hand, entry-level developers had to wait until unit testing to identify these issues.

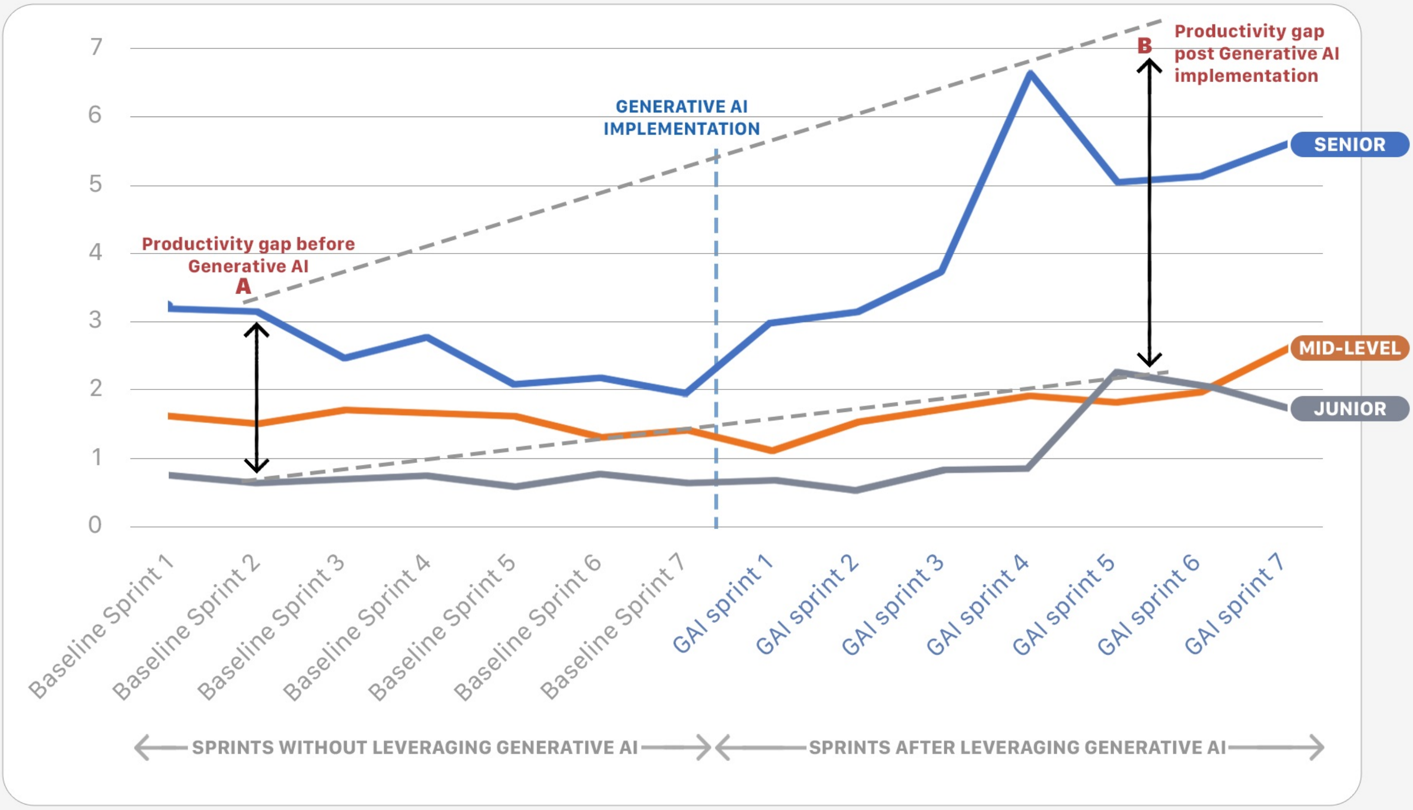
Better prompts with project context: Senior engineers needed fewer iterations to achieve the desired results as they were able to improve their prompts more efficiently for each iteration by leveraging their previous experience of working across varied coding scenarios. Entry-level engineers needed a higher number of iterations as they had to rely on trying out multiple prompts before recognizing the most efficient ones.

Senior engineers possess contextual knowledge of the existing codebase, and project requirements. Senior and mid-level engineers' detailed understanding allowed them to offer more information and context in their prompts, leading Generative AI tools to provide more accurate code suggestions.

Ability to break down complex problems: Senior engineers were able to break down complex problems into easy, manageable ones. This helped them to use Generative AI in complex scenarios compared to entry-level engineers.

Refinement and Review: Senior engineers reported that the Generative AI tools were highly helpful in reviewing and refining the code developed by their team, which helped the senior engineers to save time spent on code review.

NUMBER OF COMMITS – SPRINT-WISE



Senior engineers achieve an accelerated shift in productivity compared to junior engineers as they have a higher ability to...

- Provide better prompts with project context and clear requirements
- Understand and review the code suggestions provided by Generative AI
- Breakdown complex problems into simpler tasks

Junior engineers experience reduced productivity gains when using Generative AI because...

- AI-generated code needs more review cycles to fix identified issues
- They accept recommendations from Generative AI without thorough evaluation, increasing issues therein

Generative AI adoption offers significant advantages for senior developers, who can experience an accelerated commit rate and a higher impact on task completion time. Senior developers leverage their expertise in comprehending and refining code suggestions provided by Generative AI, optimizing their workflow efficiency. Their contextual knowledge, problem-solving abilities, and efficient use of project context enable them to outpace their junior counterparts in leveraging Generative AI for complex scenarios.

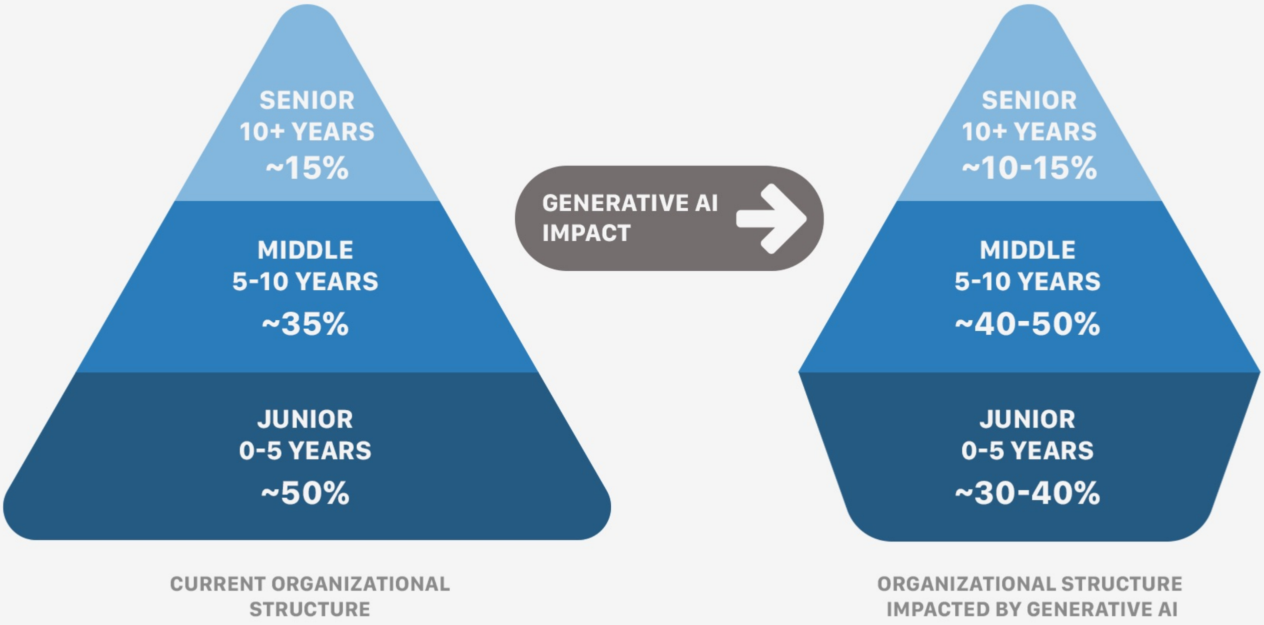
This evolution underscores a decline in the demand for junior developers and a simultaneous increase in the productivity and impact of senior developers in organizational workflows. Generative AI implementation is set to reshape organizational structures by reducing the need for junior engineers and enhancing the role of senior developers. Its impact on simplistic tasks prompts a shift towards leaner organizational structures, necessitating a smaller base of junior engineers.

The improvement in productivity of senior engineers results in a reduction in the cost per story point[1] below that of junior developers. Previously, without Generative AI, junior developers contributed fewer story points than their senior counterparts. But as their unit cost was lower, it was still cost beneficial to employ a sizeable number of junior developers. With Generative AI, the senior developers' enhanced efficiency results in more cost-effective production of story points, challenging the traditional cost paradigm and highlighting the potential cost benefits of leveraging Generative AI with a higher proportion of senior resources.

Note: 1. Story points are units of measurement used to determine how much effort is required to complete a product backlog item or any other piece of work. The team assigns story points based on the work's complexity, amount, and uncertainty.

GENERATIVE AI'S IMPACT ON TEAM PYRAMIDS

(INDICATIVE)



KEY OBSERVATIONS

- **Generative AI is expected to reduce the number of junior-level resources** across organizations as it takes on simple tasks
- **Organizational structures** are expected to **shift towards a leaner shape** with a smaller base
- Generative AI requires **expertise in overseeing, interpreting output, and optimizing performance,** reshaping team skill dynamics
- **Layer of Business Analysts**, who understand the business problems as well as technology, **become more important**
- **Reskilling of the workforce would be an additional overhead** for organizations since importance would be given to domain and not technology expertise
- **The enhanced efficiency of senior developers** results in more cost-effective production of story points, resulting in **decreased cost per storypoint** even with top-heavy talent structures

GENERATIVE AI HAS A LIMITED IMPACT ON PRODUCTIVITY IN HIGH-COMPLEXITY CODING ENVIRONMENTS, PROMPTING CONTINUED RELIANCE ON SKILLED ENGINEERS FOR DESIRED OUTPUT IN SUCH SCENARIOS.

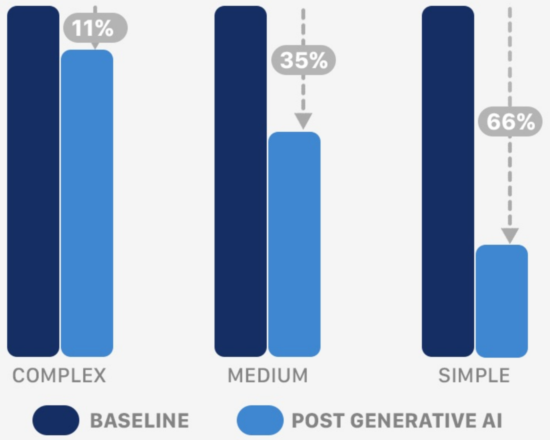
Generative AI tools have proven effective in addressing a range of code complexities, but engineers acknowledge the indispensable role of human involvement, particularly in complex coding scenarios. Task complexity is influenced by factors such as legacy code presence, upstream/downstream integrations, ease of code maintainability, and the cyclomatic complexity of the central repository/components.

Notably, Generative AI has significantly reduced task completion time and demonstrated a higher impact in scenarios with lower code complexity. However, several reasons underline the continued need for human expertise in complex coding tasks. Firstly, the pivotal role of both the quality and quantity of training data cannot be overstated. In instances that demand deep creativity and problem-solving capabilities, Generative AI may struggle if the available data is insufficient or fails to represent real-world complexities adequately.

Next, the challenge intensifies when interpreting Generative AI suggestions in complex scenarios where outputs may be intricate and difficult to comprehend, thereby affecting the practical usability of the AI system. Additionally, while Generative AI excels in the accuracy and quality of simple coding scenarios, tasks demanding profound creativity and advanced problem-solving capabilities often surpass the tool's limited abilities.

In essence, while Generative AI proves invaluable in simplifying coding processes, human expertise remains irreplaceable for navigating the intricacies of complex coding tasks. This is especially true in scenarios where creativity, interpretability, and a deep understanding of diverse situations are indispensable.

% reduction in task completion time is minimal for complex code



Code complexity was determined by factors including:
- Involvement of legacy code
- Upstream/Downstream integrations
- Ease of code maintainability
- Cyclomatic complexity of the central repository/components

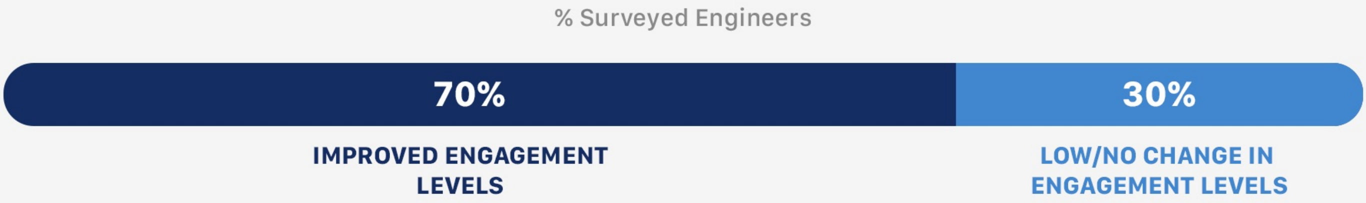
Generative AI's impact in complex coding scenarios was minimal due to...

- The limited ability of GAI tools to perform tasks requiring deep creativity and problem-solving capabilities
- Incomplete project context and information provided in the prompts to Generative AI tools
- Difficulties in handling multiple integrations and industry-specific business logic
- Limited ability of engineers to break down complex problems into simple tasks that can be managed by Generative AI

GENERATIVE AI ENHANCES ENGINEER ENGAGEMENT, FOSTERS COLLABORATIVE TEAM DYNAMICS AND COLLECTIVE PROBLEM-SOLVING, AND REDUCES ATTRITION.

From the survey and interviews conducted with Ness' engineers working on real-time customer projects, 70% of the engineers expressed improvement in their engagement levels post-implementation of Generative AI tools.

70% of engineers experienced improved engagement post Generative AI implementation



Engineers' engagement was evaluated by considering several factors, including perceived improvement in product development speed and quality, support for learning, and engineers' overall satisfaction levels.

The engineers' feedback highlighted the top reasons contributing to improved engagement levels after Generative AI implementation:

Toil Reduction: Generative AI tools automate various repetitive coding tasks, such as generating boilerplate code, refactoring, and code formatting. This reduces the need for engineers to perform these tasks manually, minimizing their mental effort. It enables them to focus on more creative, logic-driven, or problem-solving tasks, thus increasing their engagement and excitement levels.

Real-time Learning Assistance: Engineers affirmed that Generative AI tools function as real-time learning aids, offering explanations, examples, and alternative solutions. The Generative AI tools assist engineers in grasping new concepts, understanding contextual ideas, and learning techniques while programming. Several developers admitted that Generative AI tools functioned as effective pair programmers, significantly improving their efficiency and satisfaction levels.

Testing Support: Around 50% of the QA engineers acknowledged the assistance of Generative AI tools in testing. These tools aid in creating test cases, generating mock data, and automating testing. This leads to improved code coverage and quality with equal or reduced tester effort.

	85% **OF THE SENIOR DEVELOPERS**	**71%** **OF THE MID-LEVEL DEVELOPERS**	**60%** **OF THE JUNIOR DEVELOPERS**
KEY MOTIVATING FACTORS	• Generative AI tools reduce the **effort spent on repetitive, mundane coding tasks** • Generative AI tools helped to focus on **creative, complex problem-solving, while also accelerating product innovation**	• Generative AI tools contribute to the **enhancement of programming logic** by offering contextual recommendations	• Generative AI tools **offer syntax and boiler code templates,** enhancing **learning opportunities in new programming languages** • **Senior engineers**, with more time available due to Generative AI implementation, provide **valuable mentorship**
ENGINEER TESTIMONIALS	"Often **short sprint times are attributed to killing innovation** as they limit the developer's ability to create POC for new features. But with **Generative AI, an innovative POC can be created in the same sprint time interval, thus helping product innovation."**	"Copilot provides **good coding suggestions**, using which we can improve our **coding logic**. We can combine our logic with Copilot suggested logic and write code more effectively."	"It is offering **contextual code suggestions and auto-completion**. Copilot will significantly help in learning."

A higher percentage of senior engineers reported improvement in engagement levels compared to their junior counterparts. Generative AI tools helped senior engineers reduce the time taken on mundane activities, which lowers the cost of output, and helped them focus on complex tasks that require creativity. The tools also enabled them to better engage with their junior counterparts, review their juniors' work, and guide them in the overall product development process. This resulted in higher engagement levels for senior engineers.

Generative AI tools' implementation has emerged as a strategic measure to reduce attrition among engineers. This reduction in attrition is attributed to the tools' multifaceted benefits, including the automation of repetitive coding tasks, real-time learning assistance, testing support, and adherence to coding style. By minimizing mundane activities, enhancing learning opportunities, and promoting a collaborative coding environment, Generative AI tools contribute significantly to increased job satisfaction, and consequently, lower attrition rates within engineering teams.

ATTRITION-RELATED CHALLENGES ARE MITIGATED WITH...

Limited career growth opportunities
(Limited understanding of the role in the project, limited opportunities to showcase creativity and problem-solving)

Poor work-life balance
(Stress, low collaboration among teams, no time for personal interests)

Limited learning opportunities
(No guidance from seniors, very little time to upskill)

Unsatisfactory project assignments
(Less challenging, repetitive work on outdated technologies)

→

BENEFITS FROM GENERATIVE AI IMPLEMENTATION

Toil reduction

Better mentorship from senior engineers

Improved quality of work

Increased ability to focus on complex problem-solving

Assistance in learning new programming languages, coding patterns

03

IMPLICATIONS OF GENERATIVE AI ON TRANSFORMING ENGINEERING PRODUCTIVITY

Our findings indicate that Generative AI significantly impacts the traditional product development lifecycle and improves engineering productivity. These engineering productivity improvements could be used to optimize costs as well as accelerate time-to-market of software products, depending on the overall business goals and objectives. However, to capture the full productivity gains, we believe that Generative AI needs to be leveraged as part of a broader engineering productivity framework and enabled by a holistic transformation program that takes into account the existing distribution of development teams, physically and virtually.

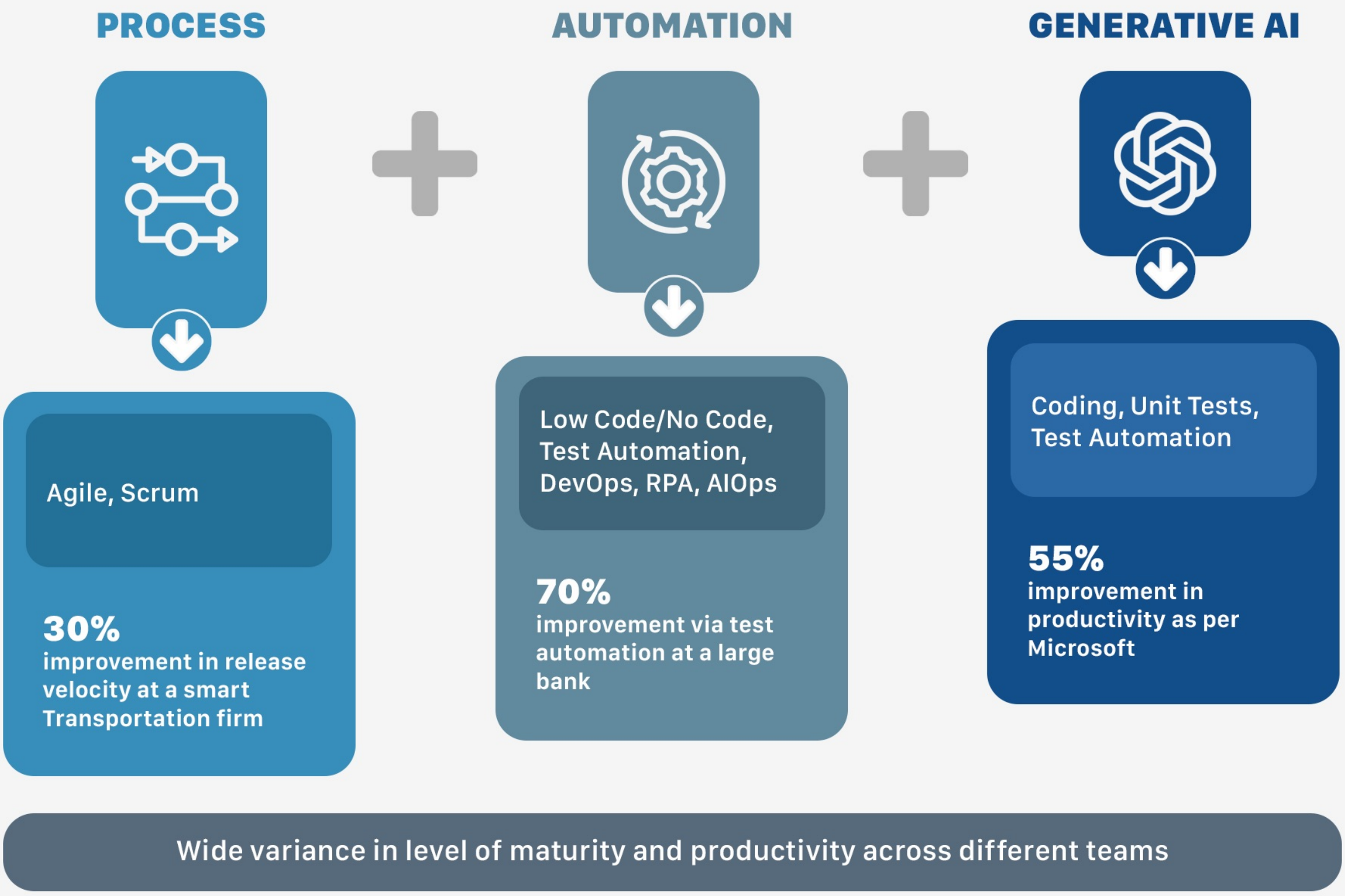
FRAMEWORK FOR ENGINEERING PRODUCTIVITY TRANSFORMATION

Our assessment of Generative AI use cases indicates numerous areas within engineering process automation where Generative AI has the potential to transform software engineering.

EXAMPLE USE CASES OF GENERATIVE AI	
AIOps	Automated RCA written based on Incident Logs, Predictive Maintenance, and Self Heal
Infra As Code	Creating IAC code based on the request from development teams
Observability	Event Correlation, Anomaly Detection, Predictive Analytics, RCA Analysis
Continuous Deployment	Generative AI implementation can enhance automation and the decision-making processes within the deployment pipeline, contributing to more efficient and effective continuous deployment practices
Release Management	Automating the process of generating release notes
Configuration Management	Generative AI can significantly contribute to configuration management by introducing automation, optimization, and intelligence into the process. A few examples include: • **Automated Configuration Discovery:** Generative AI can analyze and automatically discover configurations within a system, identifying settings, parameters, and dependencies. • **Dynamic Configuration Adjustment:** Generative AI can dynamically adjust configurations based on changing requirements, workload variations, or environmental factors. • **Predictive Configuration Changes:** Utilizing Machine Learning capabilities, Generative AI can predict future configuration changes based on historical data and trends. This foresight allows proactive adjustments to prevent potential issues.
Security	Integrating AI responsibly and ethically into security frameworks. Infra Security is normally achieved by the 3Rs (Repair, Repave, Rotate). Generative AI can create and recommend new architecture to repave the old architecture based on the security threat model.

Test Automation	Creating manual and automation test cases based on the acceptance criteria on the stories
Continuous Build	Generative AI in DevOps combines the power of AI with the principles of DevOps, enabling teams to automate various stages of the software development and deployment process. From code generation to testing, monitoring, and even troubleshooting, Generative AI brings a new level of speed, accuracy, and scalability to DevOps practices.
Continuous Integration	The whole pipeline can be intuitively created by Generative AI based on code in the repository
Test-driven Development	Test Scenario Generation, Data Generation for Tests
Version Management	Writing code using Copilot, CodeWhisperer
Agile Development	Writing Epics and Stories leveraging Generative AI

Our findings also suggest that there is a need for selective Generative AI adoption to maximize impact for software engineering organizations. We believe that many organizations have yet to tap into the full potential of this technology to improve productivity through process maturity and automation. A fundamental transformation of the product lifecycle process will require a holistic approach including changes to engineering processes and automation tools before deploying Generative AI. In our experience, process improvement and automation alone can deliver 30-70% improvement in productivity. Generative AI has the potential to take it to the next level.



PROCESS

Agile, Scrum

30% improvement in release velocity at a smart Transportation firm

AUTOMATION

Low Code/No Code, Test Automation, DevOps, RPA, AIOps

70% improvement via test automation at a large bank

GENERATIVE AI

Coding, Unit Tests, Test Automation

55% improvement in productivity as per Microsoft

Wide variance in level of maturity and productivity across different teams

LAUNCHING AN ENGINEERING PRODUCTIVITY TRANSFORMATION JOURNEY

Organizations may be at varying levels of maturity in their IT portfolio and operating model, Engineering Processes, and Automation. These need to be assessed to identify the highest opportunities for Generative AI application. Assessing the maturity of software engineering organizations needs a two-pronged approach –

> **Distributed Development Assessment:** This assessment includes reviewing the technology portfolio, distribution of talent, and business priorities for product/application development.
>
> **Engineering Methods and Tools Assessment:** This includes the assessment of key engineering processes, methods, and tools across application environments.

While Ness brings proven expertise in optimizing engineering processes for productivity as a specialist in product engineering, Zinnov brings an established track record of consulting with large enterprises in assessing their product portfolios for globalizing their development footprint. Given the dual impact of Generative AI, influencing both the globalization of product development and enhancing engineering productivity, harnessing its full potential necessitates an integrated approach. Blending Ness and Zinnov methodologies and expertise enables enterprises to transform engineering producitivity and facilitate distributed development through Generative AI.

The outcome of this assessment helps enterprises to identity specific applications/products and processes that are most amenable to transformative impact through Generative AI.

GENERATIVE AI USE CASES FOR MAXIMUM ROI

ZINNOV DISTRIBUTED DEVELOPMENT REVIEW

- Technology Assessment
- Talent Requirements
- Business Roadmap/Criticality
- Sourcing Mix
- Infrastructure Complexity

NESS ENGINEERING MATURITY ASSESSMENT

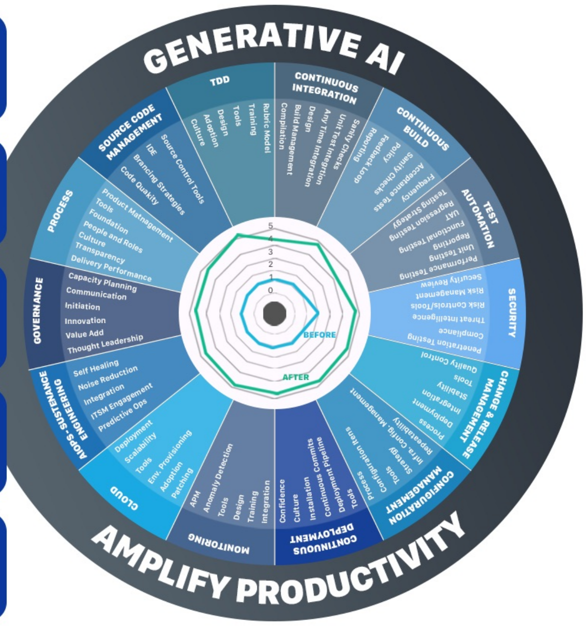
LEVEL 5 High Performer

LEVEL 4 Performer

LEVEL 3 Practitioner

LEVEL 2 Ready To Start

LEVEL 1 Needs Attention



In conclusion, the collaboration between **Zinnov** and **Ness Digital Engineering** charts a strategic course for enterprises leveraging Generative AI to revolutionize global product development. From Zinnov's Global Excellence assessments and strategic roadmaps to Ness' hands-on implementation, including Engineering Maturity Models and Transformation Accelerators, this partnership can help enterprises accelerate innovation, optimize costs, and streamline global expansion. Together, Zinnov and Ness will help enterprises navigate the evolving landscape by showcasing how Generative AI transforms product development, making it agile, efficient, and globally distributed.

Please feel free to reach out to the authors of this whitepaper for more information regarding driving productivity and distributed product development through Generative AI implementation.

ABOUT ZINNOV

Founded in 2002, Zinnov is a global management and strategy consulting firm, with presence in New York, Santa Clara, Houston, Seattle, Bangalore, Gurgaon, Pune, Hyderabad, and Paris. Over the past 21 years, Zinnov has successfully consulted with over 250+ Fortune 500 enterprises to develop actionable insights to help them accelerate their technology journeys to create value – across dimensions of revenue, transformation, and optimization. With core expertise in Digital Engineering Talent, Digital Transformation, Innovation, and Outsourcing Advisory, Zinnov assists clients by:

- Enabling global companies to develop and optimize a global engineering talent footprint through center setups and accelerators - in an as-a-service model, as well as optimizing their global portfolios, to achieve higher R&D efficiencies, innovation, and productivity;
- Advising global PE firms in asset shortlisting and target evaluation, commercial due diligence, and value creation;
- Growing revenue for companies' products and services in newer markets through account intelligence, market entry, and market expansion advisory;
- Helping global companies outline and drive their open innovation programs, design and operate accelerator programs, and enable collaboration with start-ups across specific use cases and predefined outcomes;
- Structuring and implementing Digital Transformation levers enabled by technologies like AI/ML, Intelligent Automation, Cloud, IOT, etc.

With their team of experienced consultants, subject matter experts, and research professionals, Zinnov serves clients from across multiple industry verticals including Enterprise Software, BFSI, Healthcare, Automotive, Retail, and Telecom in the US, Europe, Japan, and India.

For more information, visit www.zinnov.com.

ABOUT NESS

Ness Digital Engineering, which was acquired by global investment firm KKR in 2022, is a full life-cycle digital engineering firm offering digital advisory through scaled engineering services. The talent resource pool comprises 5000+ engineers across 11 innovation hubs in the US, Eastern Europe, and India. Combining our core competence in engineering with the latest in digital strategy and technology, we seamlessly manage Digital Transformation journeys from strategy through execution to help businesses thrive in the digital economy. For more information, visit www.ness.com

AUTHORS

NIKHIL KULKARNI
PARTNER
ZINNOV

VIKAS BASRA
GLOBAL HEAD - INTELLIGENT ENGINEERING PRACTICE
NESS

CHARY MLN
PRINCIPAL
ZINNOV

ADITI WADIA
SENIOR DIRECTOR – ENGINEERING EXCELLENCE GROUP
NESS

DIVYA SHETTIGAR
PROJECT LEAD
ZINNOV

ZOLTAN SIPKAI
SENIOR ENGINEERING DIRECTOR
NESS

SMRITI B
PROJECT LEAD
ZINNOV