

# The Impact of Generative AI on Software Engineering Activities

*Arlene Minkiewicz*

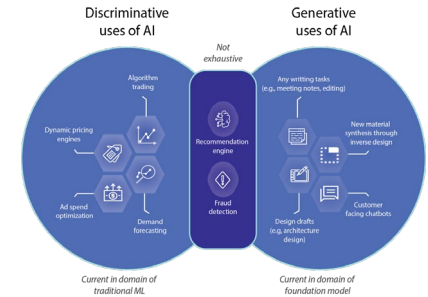
*2024 Joint IT and Software Cost Forum*





# Introduction

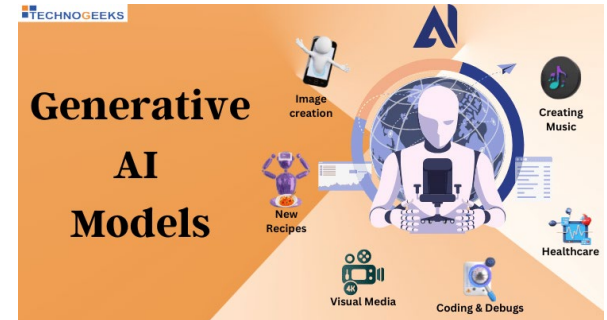
- AI is not about “lower cost of prediction” but rather about “enable vastly more productive products, services and organizational design”
- AI is defined as the ability of software to perform tasks that traditionally require human intelligence
- Significant shift in automation in software engineering
  - DevOps Platforms
  - Continuous Integration/Continuous Delivery
  - Low-Code/No-Code
- Generative AI is the latest (and potentially greatest) evolution to date
- ChatGPT went from obscurity to over a million users in 5 days and over 100 millions in 60 days
- “ I feel like, for the first time, I can communicate with a computer , and it could interpret what I meant. We can now translate language into something a machine can understand. I can’t think of anything that has been more powerful since the desktop computer.”





# Generative Artificial Intelligence (GenAI)

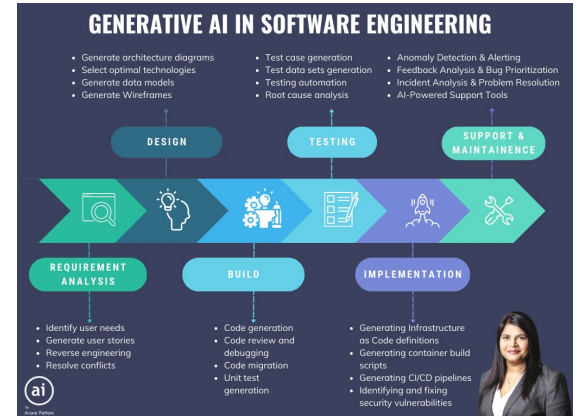
- Generative AI leverages Large Language Models (LLMs) and Generative Adaptive Networks (GANs) in order to understand language, imagery and code
- Typically built on foundation models that contain expansive artificial neural networks
  - Inspired by the billions of neural networks that make up the human brain
- These models are trained on hundreds of thousand data points
- Gen AI has undergone significant advancements in the last decade – going from rule based systems to sophisticated models that can generate human like text, images and software code





# Application of GenAI on Software Engineering

- At present, GenAI can fill gaps and accelerate solutions within the software development lifecycle
- While this makes some nervous, the current opinion is that software engineering is far from losing the need for the human touch.
- Currently the interaction between GenAI and the human software engineer is a collaboration
- Some software engineering activities where GenAI can be a good partner include
  - Inception and Planning
  - Code Generation
  - Code Translation and Migration
  - Test Case Generation
  - Documentation
  - Code Standard Enforcement
  - Architecture and Design Analysis
  - Maintenance Activities





# Generative AI Impacting Coding Activity Productivity

- GenAI can increase coding productivity by:
  - Enhancing efficiency
  - Reducing manual effort
  - Freeing up developers to do more creative (and fun) activities
- Four areas of productivity potential
  - Expediting manual and repetitive work through autocompletion and standardized documentation
  - Jump starting the first draft of code
  - Accelerating updates to existing code through reviews and modification suggestions
  - Increasing developer's ability to tackle new challenges



ChatGPT

Certainly! You can calculate the average of an array of numbers in Python using the following code:

```
python Copy code  
  
def calculate_average(numbers):  
    if not numbers:  
        return 0 # Return 0 for an empty array to avoid division by zero  
    else:  
        return sum(numbers) / len(numbers)  
  
# Example usage:  
numbers_array = [1, 2, 3, 4, 5]  
average_result = calculate_average(numbers_array)  
print(f"The average of the array (numbers_array) is: {average_result}")
```

This code defines a function `calculate_average` that takes an array of numbers as input and returns the average. The `sum` function is used to add up all the elements in the array, and then the average is calculated by dividing the sum by the length of the array.



# Generative AI Impacting Software Design Productivity

- Engineers can use GenAI to create multiple architectures and designs and then iterate to optimize
- Some ways GenAI facilitates the design process include:
  - Early Research Analysis
  - Rapid Prototypes
  - Enhanced Collaboration
  - Virtual Design
  - Virtual Simulations
  - Design Automation
  - Design Review Regulations
- While there are Design tasks facilitated by GenAI, the evidence and extent is less conclusive than with coding and test tasks



You

Draft design for user input for my cost estimation model



ChatGPT

Creating a user-friendly input form is crucial for a cost estimation model. Here's a basic HTML and CSS template for a simplified user input form. This example assumes a web-based interface:

```
html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Aircraft Cost Estimation</title>
</head>
<body>
  <div style="font-family: 'Arial', sans-serif; background-color: #f5f5f5; margin: 0; padding: 0; display: flex; align-items: center; justify-content: center; height: 100vh;">
    <div style="background-color: #ffffff; padding: 20px; width: 100%; height: 100%; border-radius: 10px; display: flex; flex-direction: column; align-items: center; justify-content: center;">
      <input type="text" style="width: 100%; height: 40px; margin-bottom: 10px;" value="Aircraft Type" />
      <input type="text" style="width: 100%; height: 40px; margin-bottom: 10px;" value="Flight Hours per Year" />
      <input type="text" style="width: 100%; height: 40px; margin-bottom: 10px;" value="Cost per Hour" />
      <input type="text" style="width: 100%; height: 40px; margin-bottom: 10px;" value="Annual Maintenance Cost" />
      <button style="width: 100%; height: 40px; background-color: #2196f3; color: white; border: none; border-radius: 10px; margin-top: 10px;" type="button" value="Estimate Cost" />
    </div>
  </div>
</body>
</html>
```

This example includes basic input fields for the aircraft type, flight hours per year, fuel cost per hour, and annual maintenance cost. The "Estimate Cost" button is a placeholder for the function that you would implement to calculate the cost based on the user's inputs.



# Generative AI Impacting Software Testing and Quality Assurance Productivity

- GenAI clearly creates artifacts that facilitate bug detection and test planning
- Areas where Test and Business Analysis teams can benefit from GenAI
  - Data Set Generation
  - Generation of Automated Tests
  - Test Plan Creation
  - Unit Test Generation
  - GenAI Augmented Test Creation
  - Test Scripts based on the application's code
  - There has been a documented 20-40% increase in overall code quality for Greenfield projects

**You**  
User

Generate a test data set for a cost estimation model for the manufacturing of aircraft

**ChatGPT**

Certainly, here's an example of a test data set for a cost estimation model for the manufacturing of aircraft:

```
plaintext Copy code
```

Aircraft Model	Number of Units	Raw Material Cost (USD)	Labor Cost (USD)
Boeing 737	20	150,000,000	50,000,000
Airbus A320	15	120,000,000	45,000,000
Cessna Citation	30	25,000,000	20,000,000
Bombardier Global 7500	10	180,000,000	60,000,000
Embraer Phenom 300	25	40,000,000	30,000,000

In this example:

- **Aircraft Model:** The model or type of the aircraft being manufactured.
- **Number of Units:** The quantity of aircraft units to be manufactured.
- **Raw Material Cost (USD):** The cost of raw materials required for manufacturing.
- **Labor Cost (USD):** The labor cost associated with manufacturing.
- **Engineering and Design Cost (USD):** The cost of engineering and design processes.
- **Testing and Quality Assurance Cost (USD):** The cost associated with testing and quality assurance.
- **Total Manufacturing Cost (USD):** The expected total cost of manufacturing, to be





# Generative AI Impacting Collaboration and Team Dynamics



- GenAI Tools and technologies are being increasingly integrated in software development workflows to enhance team collaborations
  - Code generation and auto-completion
  - Pair Programming Assistance
  - Code Review and Quality Assurance
  - Natural Language Interfaces
  - Knowledge Sharing and Transfer
  - Continuous Integration and Delivery
  - Team Coordination and Project Management

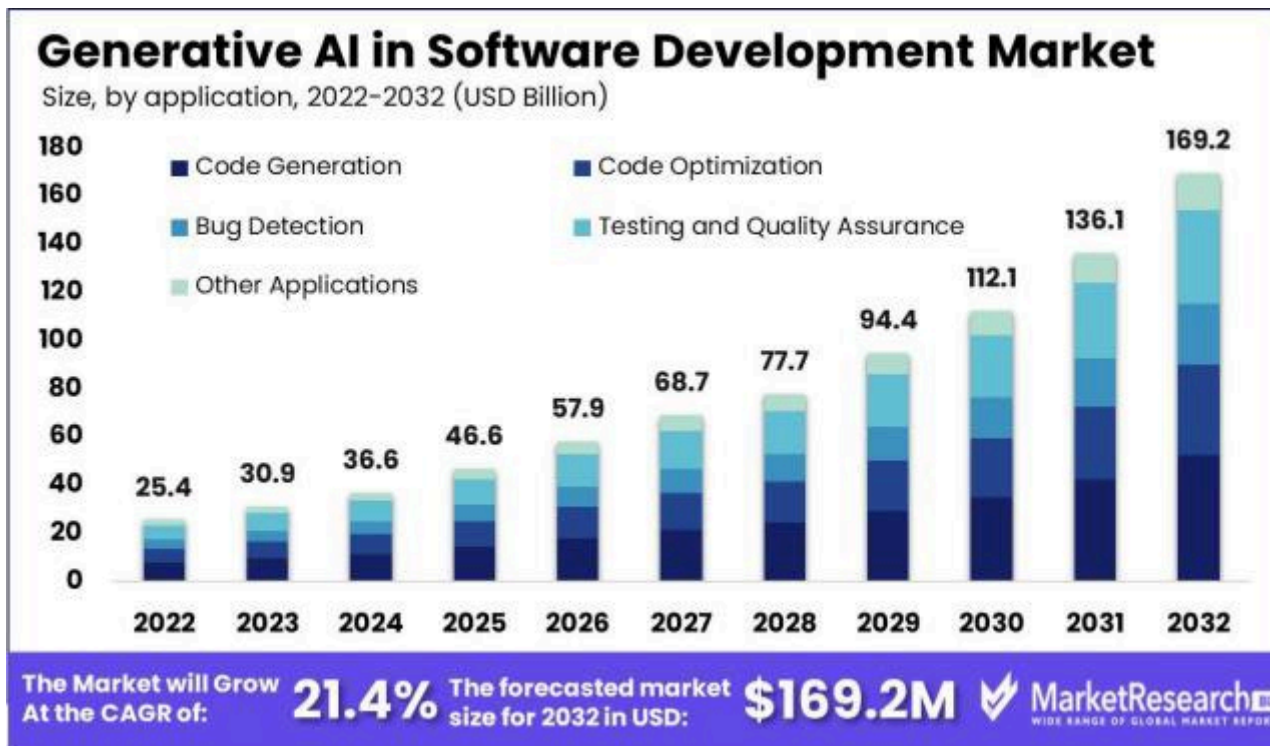


# Productivity and Cost Implications with Generative AI

- While it is still too early to have definitive proof – there are various studies that indicate we will see productivity improvements as we integrate GenAI into our software engineering projects
  - McKinsey study found significant increase in developer productivity for certain specific tasks:
    - Code Documentation ( 45-50% faster)
    - Code Generation ( 35-45% faster)
    - Code Refactoring (20-30% faster)
  - The study also found that these productivity improvement were magnified if multiple GenAI tools (with different Focii) were used
- A different McKinsey report found that GenAI could impact current spending on Software Engineering function by 20-45%
- Joe Walsh of Launch Consulting reported that incorporating GitHub Copilot into Visual Studio, programmers were able to reduce ten minute tasks, such as writing a small function into the 30 seconds it took to write the prompt.
- GenAI offers the possibility for improved productivity through the automation of repetitive and tedious tasks
- We should be cautious to use this information wisely – some software engineering activities will still require human thought and context – so productivity improvements aren't going to be project wide!!



# Productivity and Cost Implications with Generative AI





## Case Studies

### ■ Case Study at Cornell University

- 95 programmers were tasked to implement an HTTP Server in Javascript as quickly as possible
- 45 had access to GitHub Copilot and a brief training video
- 50 (the control group) had no access to GitHub but did have access to Internet and self discoverable open-source code
- Participants were between 25-34 years old with high education (4 year degree or more)
- The test group (with GitHub) were 55.8% faster than the control group





## Case Studies

- McKinsey did an internal study focused on coding, refactoring and documentation
  - Conducted over a period of weeks
  - Each task was performed by a test group with access to two GenAI tools and a control group with no such access
  - Each participant was in the test group for half of the tasks and in the control group for half of the tasks
- Areas where GenAI shined
  - Expediting manual and repetitive tasks
  - Jump starting the first draft of new code
  - Accelerating updates to existing code
  - Increasing developers ability to tackling new challenges





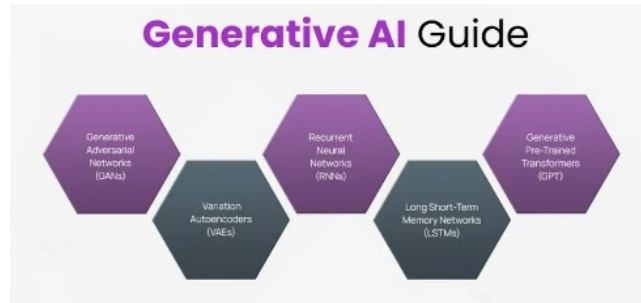
## Future Trends and Challenges

- Predictions for GenAI going forward
  - GenAI will not replace the need for software developers – it will free developers to do more of the fun software engineering activities
  - GenAI will democratize software – making it possible to develop more software using natural language
  - GenAI will speed up digital transformation in traditional sectors by increasing access to organizations lagging in digitization
- GenAI tools can be great assets for software engineers
- Important to remember that commercial GenAI tools have been trained on open source (generic) code bases
- More and more larger organizations are using their own code based to train Gen AI tools.
- Organizations interested in AI endeavors are hiring more data engineers, machine learning engineers, AI data scientists and prompt engineers
- McKinsey suggests that half of today's knowledge work activities could be automated between 2030 and 2060



# Future Trends and Challenges

- Some challenges that GenAI Adaptors are facing
  - Intellectual Property (IP) Concerns - training data comes from many open sources – creating potential for copyright infringement, trademarks, patents
  - Privacy
  - Explainability
  - Security
  - Concerns of knowledge workers regarding their continued relevance in the work place





## Wrap Up

- The advent of GenAI has ushered in a new era of possibilities for software engineering, driving increased efficiency, collaboration, and innovation.
- GenAI models generate human like text based on natural language prompts – democratizing software development
- By automating repetitive and tedious tasks, GenAI frees up software engineers to focus on the more creative (and more fun) parts of the software engineering process.
- There is much evidence that GenAI can increase productivity in software development projects
- Software cost and effort estimators should:
  - Be aware of this possibility and take it into account in estimating
  - Be aware that it will be (at least in present times) the repetitive and tedious tasks not the more complicated tasks that benefit from these gains
  - Be aware that productivity increases will slowly trend toward more intense software engineering tasks, but that productivity gain is likely to develop over time





Thank You

[arlene.minkiewicz@unisonglobal.com](mailto:arlene.minkiewicz@unisonglobal.com)