

Next-gen Sitecore development

Leveraging LLMs for code generation

Presented by:

Muktesh Mehta, Khushboo Sorthiya

Horizontal Digital

October 17, 2024





Agenda

1. Introduction
2. Understanding LLMs
3. Role of LLMs in code generation
4. Improving quality of LLM responses
5. Potential benefits: LLMs for Sitecore development
6. Future trends

Speakers



Muktesh Mehta

Sr. Technology Lead, Horizontal

- Over 13 years of expertise crafting digital experiences for clients across diverse industries using Sitecore and related technologies
- Deep learning enthusiast with certificate from Indian Institute of Science

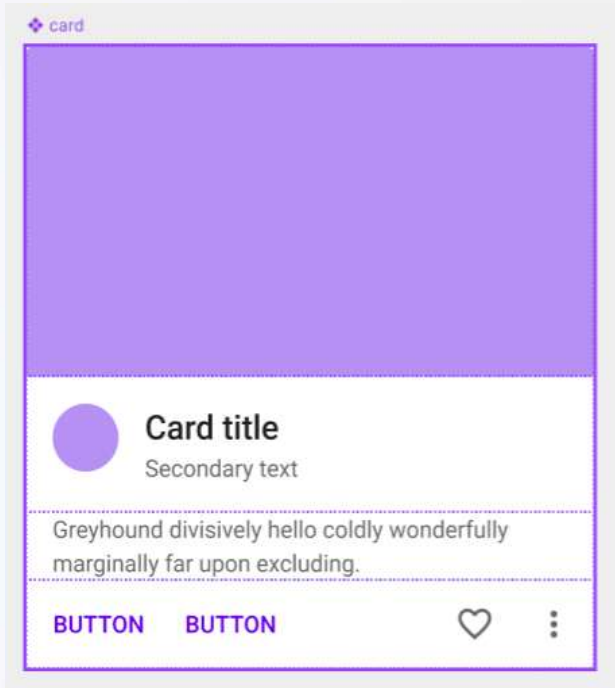


Khushboo Sorthiya

Lead, Horizontal

- Sitecore Technology MVP (3x)
- Over 10 years of experience in developing and implementing a wide range of enterprise solutions on Sitecore platform

Sprint Story



Story Title: Build card component for Sitecore XMCloud website.

Sprint Planning

How could I automate development of these components?

I've done this component dozens of times.

Where do I start?

And a while later..

Pull Request

Commit: Added card component for Sitecore XMCloud website.

```
// components/Card.tsx

import React from 'react';
import { Field, Text, Image } from '@sitecore-jss/sitecore-jss-nextjs';

type CardProps = {
  fields: {
    image: Field<string>;
    title: Field<string>;
    subtitle: Field<string>;
    description: Field<string>;
    button1: Field<string>;
    button2: Field<string>;
  };
};

const Card = (props: CardProps): JSX.Element => {
  return (
    <div className="card">
      <div className="card-image">
        <Image field={props.fields.image} alt="Card Image" />
      </div>
      <div className="card-content">
        <Text tag="h2" field={props.fields.title} className="card-title" />
        <Text tag="p" field={props.fields.subtitle} className="card-subtitle" />
        <Text tag="p" field={props.fields.description} className="card-description" />
      </div>
    </div>
  );
};
```

AI Generated Image



“A large language model (LLM) is a specialized type of artificial intelligence (AI) that has been trained on vast amounts of text to understand existing content and generate original content.”

— Gartner

Capabilities of LLM

- Text generation
- Chatbots
- Sentiment analysis
- Language translation
- Code generation, etc.

Limitations & challenges

- Bias
- Inconsistent or hallucinated responses
- Cost

Role of LLMs in code generation



Automated code generation



Bug detection and fixing



Code completion



Documentation and comments



Code refactoring



Understanding code

Improving Quality of LLM Responses

Code responses can be improved using various techniques based on knowledge, cost, and time constraints.



Prompt
engineering



Embeddings



Human-in-the-
loop



Retrieval-
augmented
generation

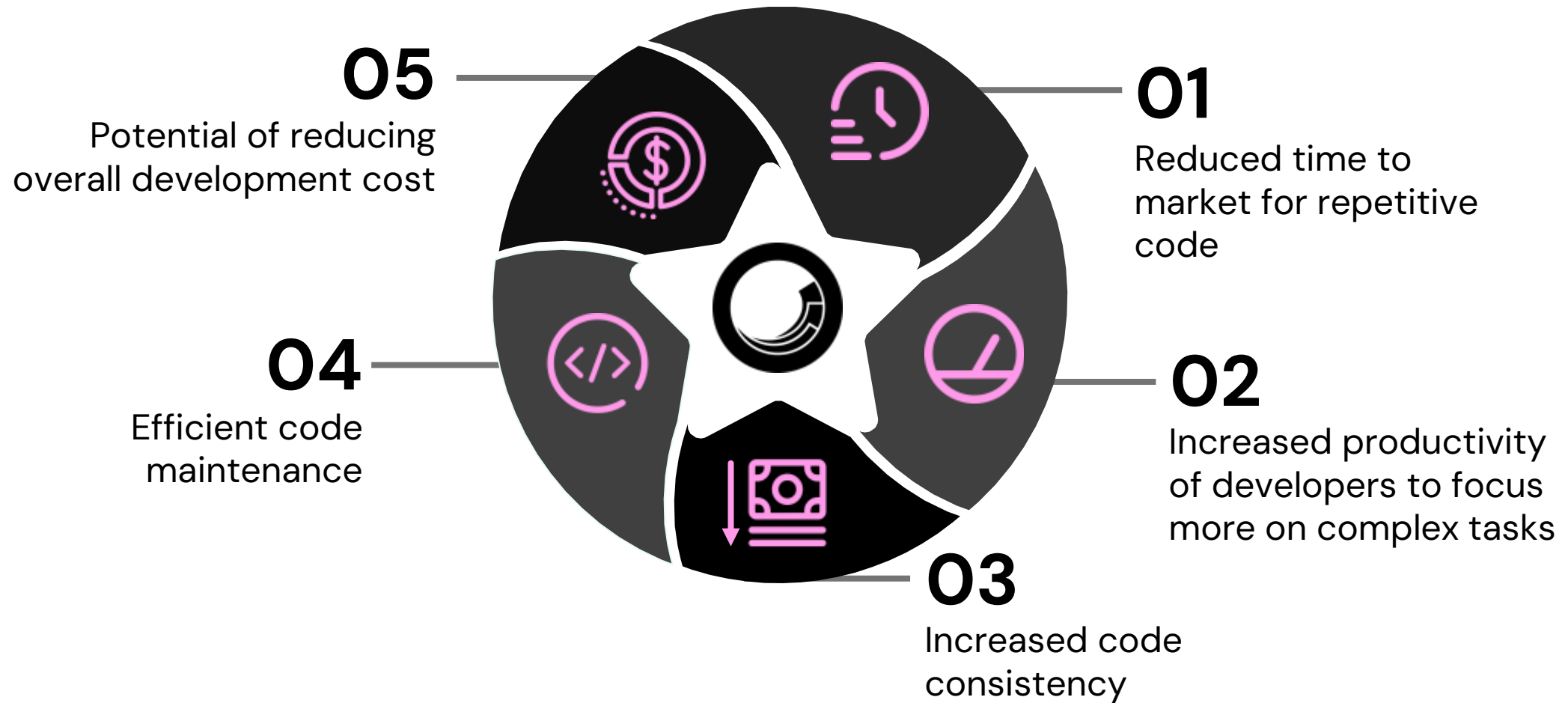


Reinforcement
learning



Fine tuning

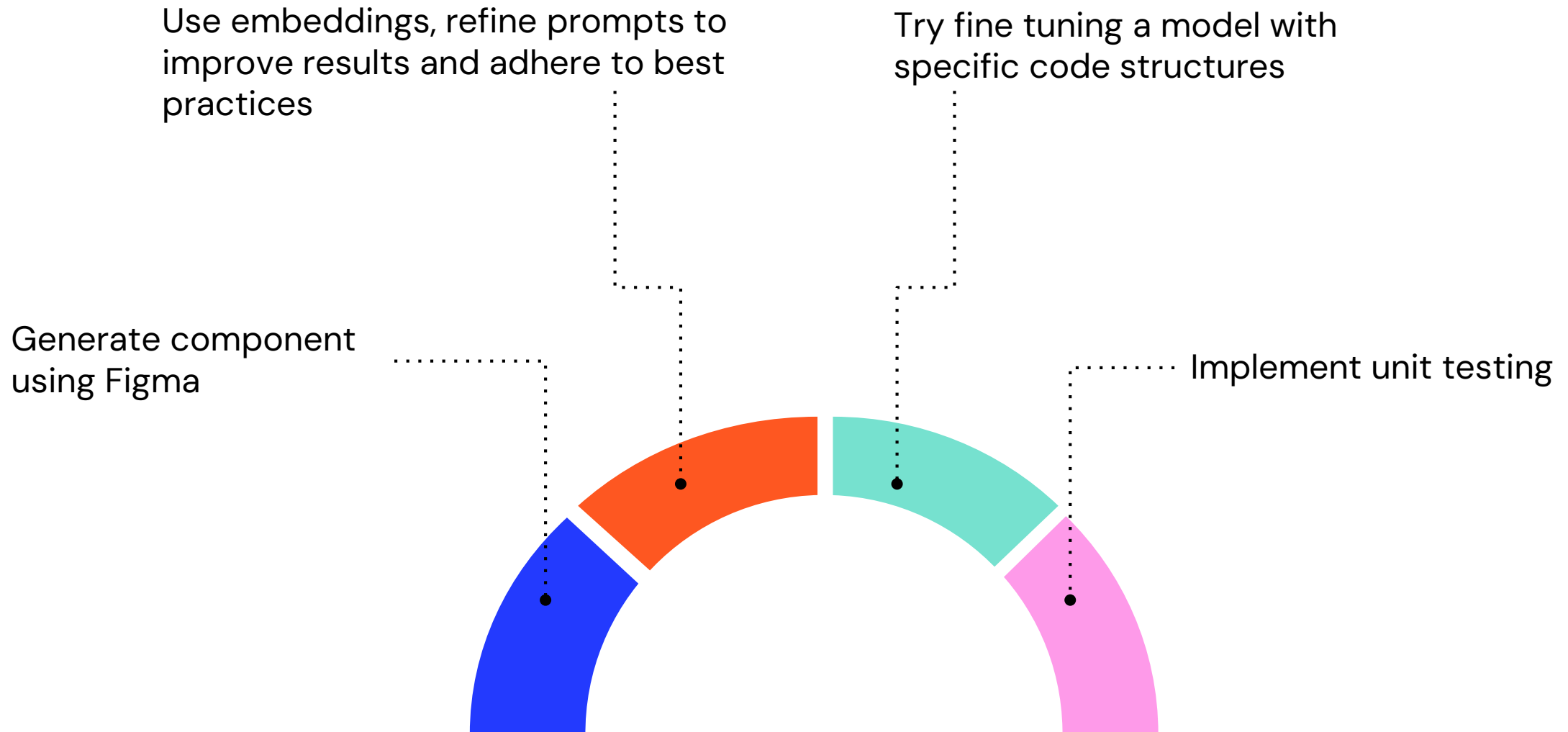
Potential benefits of using LLMs for Sitecore development



Path for developers to learn and use LLMs effectively



Future trends





Thank you

Contact us:

mmehta@horizontal.com 

ksorthiya@horizontal.com 

Appendix

LLMs for code generation

Model	How to use
OpenAI GPT	API-driven or integration in VS Code using available extensions. Use via Azure AI Studio. Ref
LLama 3	Use Olama to run LLama3 model locally and integrate using API. Additionally, you can integrate Llama model in VS Code using CodeGPT: Chat & AI Agents extension. Ref
Github Copilot (GPT4o)	Visual studio extension available. Ref
Mistral Large 2	Use Amazon Bedrock to deploy the model. Ref