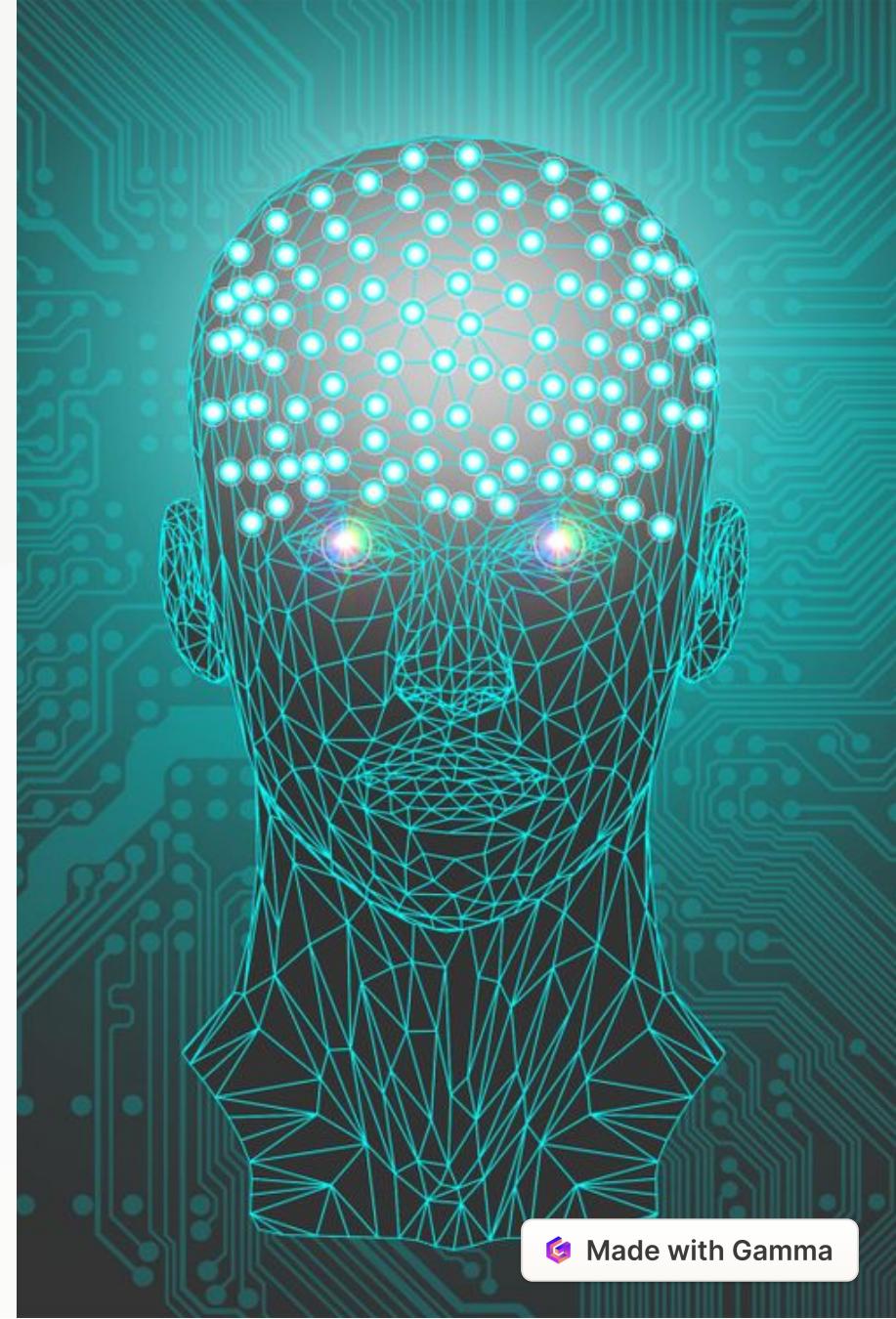


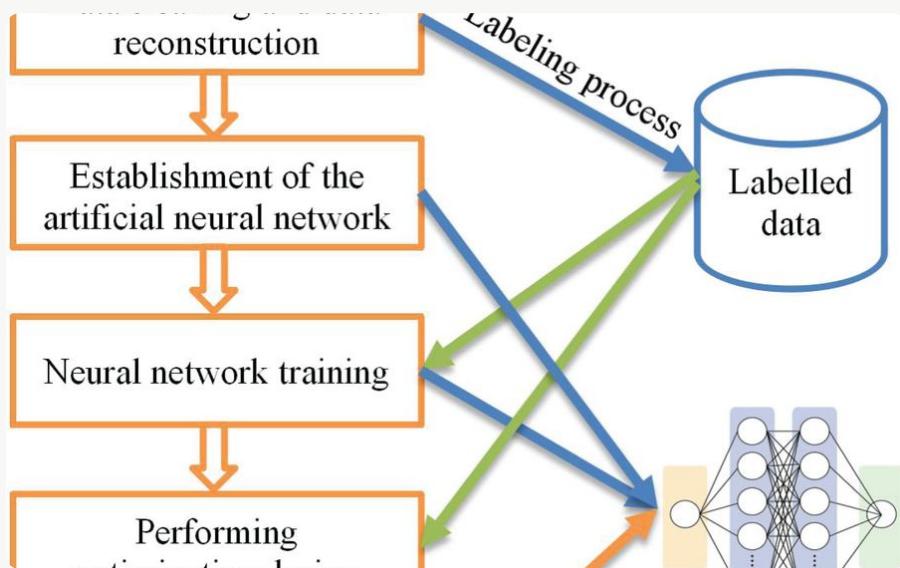
# Generative AI in Software Development: Revolutionizing Software Development

Discover the transformative potential of Generative AI in software development. Explore how it goes beyond conventional coding to guide innovation and reshape project delivery and developer roles.

 by Jiang Zhao



# The Reality of AI in SDLC: Challenges and Potential



## Ideal Vision

A graphical representation or diagram portrays the idealized vision of a fully automated SDLC process with AI agents, setting high expectations for the future of software development.

### 1 Challenge 1: Accuracy in Auto Coding

Current LLMs struggle with generating highly accurate and complete code, especially for complex or unique business logic. The nuanced understanding required for specific business contexts is not fully achievable yet, leading to gaps in automated coding.

### 2 Challenge 2: Complexity of Real- World Projects

Real projects often involve intricate requirements and multifaceted problem-solving, beyond the current scope of LLMs. Fully automating SDLC with AI, with minimal human intervention, remains a challenge due to the complex nature of software development.

### 3 The Reality Check

While AI has made significant strides, the vision of complete SDLC automation is still in its infancy. Current AI technologies, including LLMs, are tools to augment human capabilities, not replace them entirely.

The slide honestly addresses the limitations while acknowledging the progress and potential of AI in software development, using contrasting visuals for clarity.

# Efficient Path: From User Requirements to Automated Test Scripts with LLM



## User Requirements to User Stories

Utilizing LLM's natural language understanding to interpret and translate user requirements into detailed user stories.

## User Stories to Test Cases

Transforming user stories into structured test cases through LLM's text transformation capabilities.

## Test Cases to Test Scripts

Generating executable test scripts from test cases, guided by predefined rules and sample scripts.

### 1 Natural Language Proficiency

LLMs are adept at processing and rephrasing natural language, making them ideal for creating user stories and test cases.

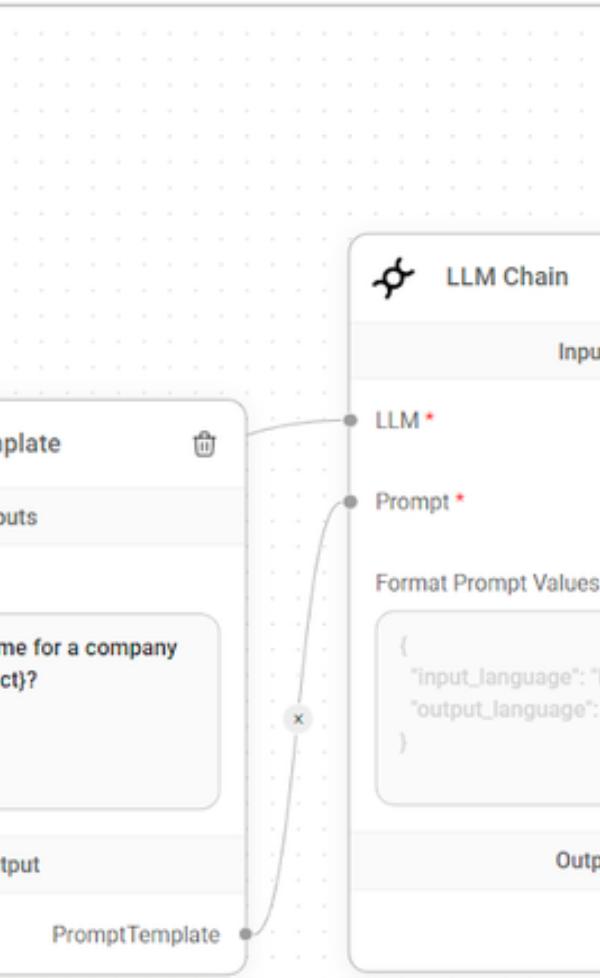
### 2 Learning from Examples

LLMs can efficiently learn to convert test cases into scripts using provided examples, similar to training new testers.

### 3 Integration with Existing Frameworks

Generated test scripts are designed for easy integration with existing automation frameworks.

This concise slide emphasizes the advantages of leveraging LLM in software testing and streamlining the path from initial requirements to ready-to-use test scripts.



# Introduction to Enhanced LLM Integration Application

## Advanced Web-Based Application

Introducing an advanced web-based application for integrating LLM into the SDLC, automating user story, test scenario, and script generation for automated testing, targeting BAs and QA teams.

# Functional Requirements

## User Story Generation

Enables the creation of one requirement per session for clear, structured stories.

## Test Scenario Creation

Automatically generates test scenarios from user stories.

## Test Case Formulation

Converts scenarios into detailed test cases.

## Test Script Generation

Produces executable test scripts for automation frameworks.

## Data Persistence

Stores all entities (stories, scenarios, cases, scripts) for future use.



# Non-Functional Requirements

## LLM Model Flexibility

Supports various models (ChatGPT, LLMA) for diverse needs.

## Scalable Architecture

Handles growing data and complexity efficiently.

## Secure and Compliant

Adheres to data privacy laws and software security standards.

## Integration-Ready

Seamlessly integrates with existing automation frameworks.



# High-Level Design: Comprehensive and Modular Test Generation Process



## User Story Generator

Transforms user requirements into actionable user stories. Operates independently for direct story generation or integrates seamlessly into the next stage of test scenario development.

## Test Scenarios Generator

Converts user stories into detailed test scenarios. Functions as a standalone module for scenario creation or as a precursor to test case formulation.



## Test Case Generator

Crafts comprehensive test cases from the developed scenarios. Capable of independent operation or serving as input for the Automated Test Script Builder.

## Automated Test Script Builder

Efficiently generates executable test scripts from the prepared test cases, ready for deployment in automation frameworks. Integrates smoothly with the preceding modules for a cohesive workflow.

This modular design offers versatile, efficient, and comprehensive test generation, aligning with modern software development practices, illustrated through a visual diagram of interconnected modules.