# Use of ChatGPT as an Assistant in the End-to-End Test Script Generation for Android Apps

A-TEST 2024

15th ACM International Workshop on
Automating Test Case Design, Selection and Evaluation
19 September 2024

Boni García
Universidad Carlos III de Madrid, Spain
boni.garcia@uc3m.es

# Introduction

- Context:



- This work provides an empirical study that aims to evaluate the impact of the use of **ChatGPT** as an assistant in developing automated E2E test scripts for **Android** applications

# Experiment - Objective

**Objective**: To analyze the use of ChatGPT to understand if there is an impact concerning the development efforts for creating automated tests for Android apps in the context of junior developers.

**RQ1. Test development effort.** Does the effort required to develop automated tests for existing Android apps vary when using ChatGPT as a code assistant?

**RQ2. Test code reliability**. Are the test scripts created with the aid of ChatGPT more reliable than those developed manually?

# Experiment - Treatments

(1) **Manual**: Without the assistance of any GenAI tool.

(2) **ChatGPT-assisted**: Using the free tier of ChatGPT at the moment of the experiment (i.e., GPT-3.5 in April 2024) to get help coding E2E tests for Android.

# Experiment - Participants

- The experiment was conducted in a university laboratory under controlled conditions
- Participants were 8 students from a **Mobile Applications** course in the second semester of the third year of the Bachelor in Data Science and Engineering Computer Science at Universidad Carlos III de Madrid (UC3M), Spain, in April 2024

uc3m | Universidad Carlos III de Madrid

# Experiment - Objects

- We asked to implement four different E2E tests using **JUnit** and **Espresso** for each participant
- Each test was designed to test a different UI feature explained during the course, namely:
  - Test 1. Interaction with an activity with fragments containing a bottom navigation view.
  - Test 2. Interaction with a UI containing an app bar.
  - Test 3. Interaction with a spinner (dropdown list with different values).
  - Test 4. Interaction with a recycler view (group view designed to display large data sets efficiently).

https://github.com/bonigarcia/android-examples

# Experiment - Objects

- A specification in the **Gherkin** language for each of these tests was provided to the participants
  - For example:

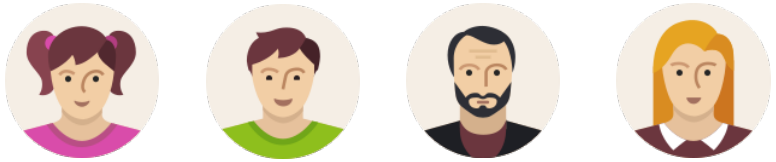
```
Feature: Android Spinner

  Scenario: Interaction with an UI spinner
    Given the user is the main screen
    When the user opens the first spinner displayed on the screen
    And the user selects the second element in this spinner
    Then the text of the selected element should be "Venus"
```

Gherkin Syntax

# Experiment - Design

## Treatment 1 – Manual



android studio

JUnit

ANDROID

UI TESTING FOR ANDROID
espresso

## Treatment 2 – ChatGPT-assisted

android studio

ChatGPT

JUnit

ANDROID

UI TESTING FOR ANDROID
espresso

# Experiment - Dependent Variables

- Our experiment has two dependent variables:
  - **Time** required to develop each test. This variable is a proxy for measuring the development effort for creating the requested tests.
  - **Score** obtained in each test, assigned by the Professors. This variable is a proxy for measuring the implementation reliability of the developed tests.
- Regarding the *Time*, we measured the time the participants took to develop four E2E tests for different existing Android demo apps using two treatments
- Regarding the *Score*, we asked the participant to upload each Android project containing the developed tests. We evaluated each test using the following evaluation rubric

# Experiment - Dependent Variables

| Criteria | Excellent (3) | Good (2) | Fair (1) | Poor (0) |
|---|---|---|---|---|
| **Test implementation** | The test meets all the Gherkin specifications using the Espresso API | The test accurately translates most of the Gherkin scenarios, but there are some minor issues (e.g., use of deprecated APIs) | The test translates some of the Gherkin scenarios into Espresso test cases but with noticeable omissions or inaccuracies (e.g., exercise or verify is not correct) | The test fails to translate all the Gherkin scenarios accurately into Espresso |
| **Test structure** | The test is well-structured and organized and can be executed as a JUnit test | The test is generally well-structured and organized, but there are minor redundancy or inefficiency (e.g., use of not logical method names) | The test lacks structure and organization (e.g., setup and teardown methods are not used or are used improperly) | The test cannot be executed as a JUnit test |
| **Code readability** | The code is easy to read and understand | The code has minor problems (e.g., un-used code) | The code has relevant issues (e.g., overcomplicated or very inefficient logic) | Code is very difficult to read and understand |
| **Code quality** | The code is clean, efficient, and free of errors | The code has minor issues (e.g., formatting problems, bad indentation) that do not affect the functionality | The code is functional but has noticeable inefficiencies and some errors | The code contains significant errors (e.g., it cannot be compiled) |
| **Project configuration** | The project structure and configuration are solid according to the Gradle best practices | The project configuration is correct but contains minor issues (e.g., use of old dependencies versions) | The project configuration is functional but has noticeable issues (e.g., missing dependencies) | The project is not recognized as a valid Gradle project |

# Experiment - Hyphothesis formulation

- Since we could not find any previous empirical evidence that points out a clear advantage of one treatment vs. the other, we formulated the null hypotheses as non-directional

**$H_{0A}$**. The use of a ChatGPT as a code assistant does not reduce the total effort required to create E2E tests for Android apps.

**$H_{0B}$**. The use of a ChatGPT as a code assistant does not improve the overall reliability of the created E2E tests for Android apps.

# Experiment - Material, procedure and execution

- A pilot experiment with a Bachelor Thesis to assess the experimental material and estimate the time needed to accomplish the tasks
  - The student finished the overall development of both treatments in 110 minutes and feedback for improving the experimental material
- Each participant was provided a link to an online form (based on Google Forms) to gather their responses
  - This form included questions about the time required to implement each test and the source code generated (exporting the Android project as a ZIP file)

# Experiment - Material, procedure and execution

- For the ChatGPT-assistant treatment, the following extra instructions were provided:
  - Open a new ChatGPT session per test
  - Use ChatGPT 3.5 (i.e., the free model available at that time)
  - Ask as many prompts to ChatGPT as you want to get help developing the test
  - Copy the code provided by ChatGPT as a UI test in your Android Project
  - Manually fix the possible problems you find until you believe the test fits the specifications
  - Before you close the ChatGPT session, get the chat URL using the following option: ChatGPT 3.5 → Share chat → Copy link
  - Report this link in this form

# Experiment - Analysis procedure

**Descriptive statistics** (mean, median, and standard deviation) for both treatments (manual and ChatGPT-assisted) for each dependant variable (*Time* and *Score*)

**Inferential statistics**. Mann-Whitney U test to compare the effects of the two treatments on each participant

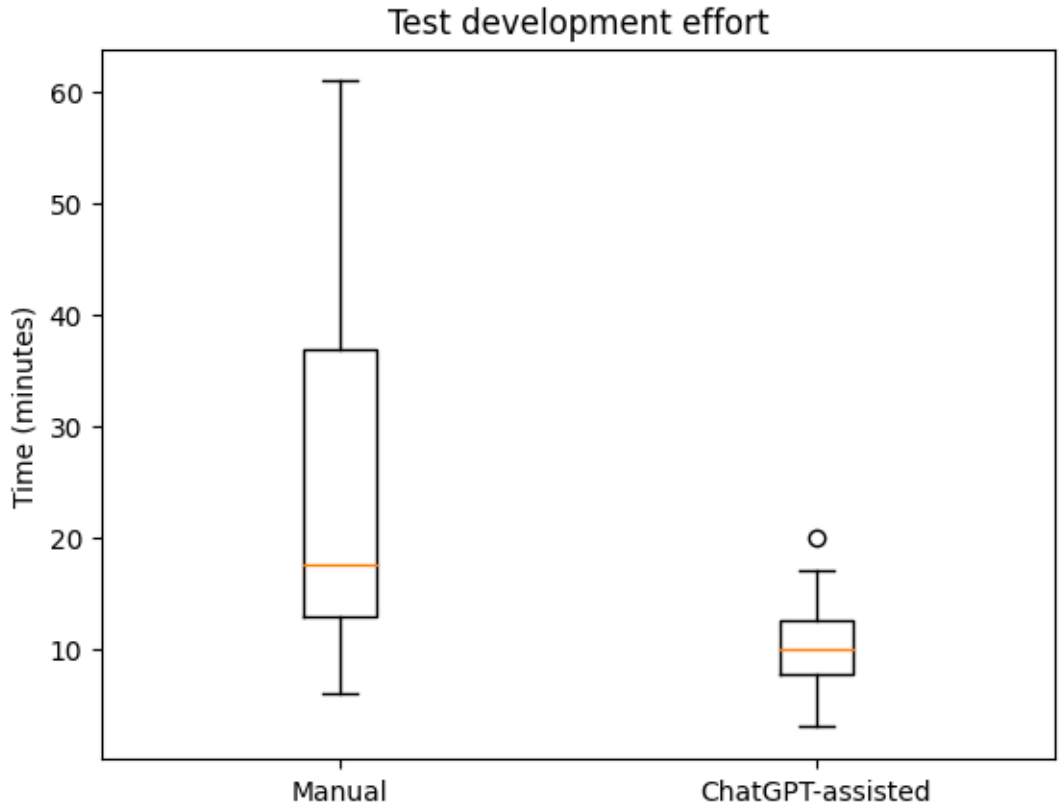
- Significance level of 5% ($\alpha$ = 0.05) to reject the null hypothesis

- Effect size is considered small using Cohen's delta 0.2 ≤ |$d$ | < 0.5, medium for 0.5 ≤ |$d$ | < 0.8 and large for |$d$ | ≥ 0.8

**Pearson correlation** analysis to examine the relationship between development time and the scores within each treatment. This analysis can help understand if quicker test generation impacts the reliability of the tests.

# Results - RQ1

- RQ1: Test Development Effort: Manual vs. ChatGPT-assisted



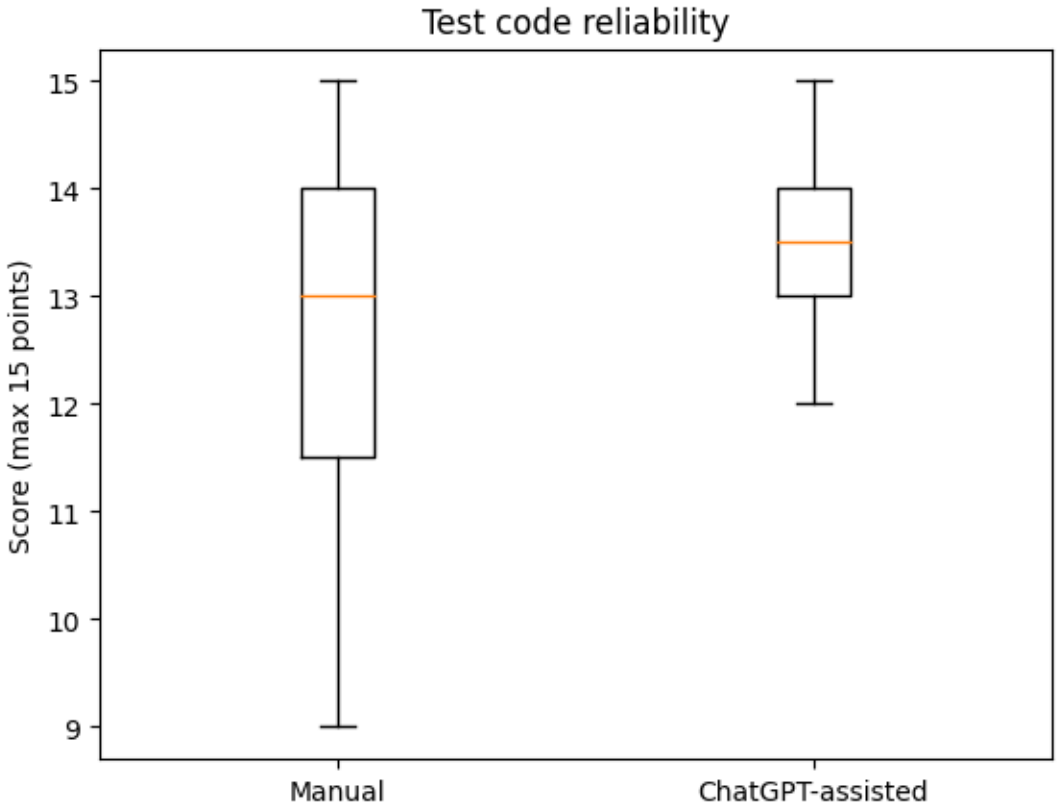

| Treatment | Mean | Median | SD |
|---|---|---|---|
| Manual | 25.625 | 17.5 | 17.389 |
| ChatGPT-assisted | 10.562 | 13 | 4.704 |

Using the Mann-Whitney U test (p = 0.002), we found a significant difference in time between the two groups (large effect size, d = 0.866). Null hypothesis $H_{0A}$ was rejected

**RQ1 Summary**: The use of ChatGPT as a code assistant significantly reduces the time required to develop E2E tests for Android apps (around -58%).

# Results - RQ2

- RQ2: Test Code Reliability: Manual vs. ChatGPT-assisted




| Treatment | Mean | Median | SD |
|---|---|---|---|
| Manual | 12.5 | 13 | 2.0 |
| ChatGPT-assisted | 13.437 | 13.6 | 0.963 |

Using the Mann-Whitney U test (p = 0.274), we found no significant difference in test code reliability between the two treatments (medium effect size, d = -0.468). Null hypothesis $H_{0B}$ was not rejected.

**RQ2 Summary**: The use of ChatGPT as a code assistant does not impact the resulting reliability in the development of E2E tests for Android apps.

# Results - Correlation Analysis

- The correlation coefficient of 0.094 suggests a very weak positive relationship between variables *Time* and *Score*.
- However, the p-value of 0.609 is significantly higher than the conventional threshold of 0.05, indicating that the observed correlation is not statistically significant.

**Correlation Summary**: There is no relationship between the time used to develop the test scripts and their reliability scores

# Discussion

- The results of the data analysis indicate that using ChatGPT significantly reduces the time required to develop automated test scripts for Android applications
- This fact suggests that ChatGPT can be an effective tool to assist junior developers in generating efficient and reliable test scripts
- On the other hand, the results indicate no relevant difference between the reliability of E2E developed manually, and the same tests developed using ChatGPT as an assistant
- This fact suggests that ChatGPT provides accurate coding assistance, similar to human performance

# Discussion

- Prompt example used by the experiments' participants:

I have an Android app developed whose main activity is as follows:

`[Java source code]`

I want to implement a JUnit test using Espresso to evaluate the previous app following this Gherkin specification:

`[Gherkin feature]`

Using the SUT source code in the ChatGPT prompt poses privacy and confidentiality concerns

Gherkin is a convenient language for interacting with LLMs like ChatGPT since it is both human and LLM-readable

# Threats to Validity

- Internal Validity
  - Selection Bias. Students' skills and prior experiences with automated testing and Android development might vary. Randomly assigning participants to the manual and ChatGPT-assisted groups helped to mitigate this threat.
  - Testing Effect. To minimize this effect, we asked the participants to test four different Android features against four distinct SUTs
  - Maturation. Participants might become fatigued during the experiment. To mitigate this threat, we kept the total duration of the experiment within reasonable limits.
  - Instrumentation. Two Professors evaluated the resulting E2E tests using a standardized rubric with specific criteria to ensure consistent and objective evaluation.

# Threats to Validity

- External Validity
  - Generalizability. Further studies with more diverse samples are needed to validate the findings.
  - Environment. The experiment was conducted in a controlled university laboratory setting. Future studies should consider evaluating the impact of ChatGPT assistance in more varied and practical settings.
- Conclusion Validity
  - Statistical Power. The small sample size limits the statistical power of our analysis. Increasing the sample size in future studies would help to achieve more reliable and generalizable results.
  - Effect Size. While we observed significant differences in the dependant variable *Time*, the effect size for the dependant variable *Score* of the test scripts was medium

# Conclusions and Future Work

- ChatGPT can significantly reduce the effort required to generate automated E2E tests for Android apps, making it a valuable tool for junior developers
- The Gherkin is effective in describing E2E test scenarios for Android apps since it is human and LLM-readable
- Source code can be effectively used to obtain assistance from ChatGPT but it might present potential problems, like privacy and confidentiality
- Future research could explore the impact of using LLMs like ChatGPT or Copilot with a more extensive and diverse test complexity, involving more participants in heterogeneous environments and different types of applications and testing frameworks