# Workshop 1 – AI-Generated Release Notes

Contents

[Workshop 1 – AI-Generated Release Notes 1](#_Toc190037171)

[Workshop Overview 1](#_Toc190037172)

[Why AI for Release Notes? 1](#_Toc190037173)

[Bad vs. Good Prompt Comparison 1](#_Toc190037174)

[❌ Bad Prompt (Generic & lacks structure) 1](#_Toc190037175)

[✅ Good Prompt (Well-structured & detailed) 2](#_Toc190037176)

[Exercise: Transform Jira Tickets into Release Notes 3](#_Toc190037177)

[Instructions 3](#_Toc190037178)

[Sample Jira Tickets 3](#_Toc190037179)

[Questions and Reflection 4](#_Toc190037180)

## Workshop Overview

**Objective:** Learn how to leverage AI (ChatGPT) to transform raw Jira ticket content into professional, user-friendly release notes.

### Why AI for Release Notes?

* Saves time by automating content structuring.
* Ensures consistency across product updates.
* Makes release notes more engaging and user-friendly.

## Bad vs. Good Prompt Comparison

### ❌ Bad Prompt (Generic & lacks structure)

**------------**

*Write release notes for a new feature.*

**------------**

🔴 **Why this is ineffective?**

* No context about the audience.
* No formatting instructions.
* AI guesses the structure, leading to inconsistent output.

### ✅ Good Prompt (Well-structured & detailed)

**Prompt:**

**------------**

*You are an expert in generating structured and engaging release notes. I will provide raw Jira ticket content, and you will transform it into a professional, user-friendly format similar to the "What's new in Zonda iPad 10.5" page found here:* [*https://zondahome.com/whats-new-in-zonda-ipad-10-5/*](https://zondahome.com/whats-new-in-zonda-ipad-10-5/)*.*

*Format each feature as:*

* **Feature Title** (clear and concise)
* **Description** (engaging, explaining purpose and benefits)
* **Details** (summarizing functionality)
* **Image Placeholder** [Insert image here]

*Ensure the tone is user-friendly and clear.*

*Next I will paste in a group of Jira tickets.*

*Do you understand?***------------**

**Why this works?**

✅ **Specifies a reference style (RAG technique)** – Ensures consistency by mimicking an existing release note format.  
✅ **Breaks down formatting requirements** – AI knows exactly how to structure the output.  
✅ **Clarifies audience tone** – Ensures AI writes in a clear and engaging way.

## Exercise: Transform Jira Tickets into Release Notes

### Instructions

1. Copy the “Good Prompt” from above, paste into Chat GPT and press “Enter”
2. Copy the three Jira tickets below, paste into Chat GPT and press “Enter”
3. Observe the AI-generated output.
4. Refine the response by tweaking the prompt (e.g., adjusting tone, length, or style).

### Sample Jira Tickets

#### Ticket 1: Advanced Search Filters

* **Summary:** Enhance search functionality with advanced filters.
* **Description:** Implement new search filters allowing users to refine results by date, category, and relevance. These filters will be available in the search panel and dynamically update the results.
* **Requirements:**
  + Users can filter by:
    - Date range (custom & predefined)
    - Category selection
    - Relevance sorting
  + Filters should persist across sessions.
  + UX should match the existing design system.
* **Acceptance Criteria:**
  + Search filters update results dynamically.
  + Filters persist across sessions.
  + Performance impact is minimal.

#### Ticket 2: MacOS Support

* **Summary:** Enable MacOS native application support.
* **Description:** Extend application support to MacOS devices, ensuring compatibility and performance optimization. Users should have a native experience when running the application.
* **Requirements:**
  + Seamless installation on MacOS.
  + Support native MacOS gestures & UI interactions.
  + Optimize performance for M1/M2 chips.
* **Acceptance Criteria:**
  + Application installs successfully on MacOS.
  + Native UI components are used.
  + No critical performance issues.

#### Ticket 3: Dark Mode UI Enhancement

* **Summary:** Improve dark mode UI for better readability.
* **Description:** Adjust the dark mode color scheme to improve readability and reduce eye strain. Ensure all UI elements adhere to WCAG accessibility standards.
* **Requirements:**
  + Optimize contrast ratios for text & background.
  + Ensure buttons & links remain visible.
  + Apply dark mode settings consistently.
* **Acceptance Criteria:**
  + Improved contrast in dark mode.
  + No UI elements are unreadable.
  + Dark mode is consistent across all views.

## Questions and Reflection

🔹 **How did the AI-generated release notes compare to manual writing?**  
🔹 **What adjustments improved the AI’s output?**  
🔹 **Where else could structured AI prompts help in your workflow?**