

## Application of Generative AI for Business Analyst Role

Nilesh D Kulkarni<sup>1\*</sup> and Saurav Bansal<sup>2</sup>

<sup>1</sup>Sr. Director - Enterprise Architecture, Fortune Brands Home & Security, USA

<sup>2</sup>Sr. Manager - Digital Applications, Fortune Brands Home & Security, USA

### ABSTRACT

This paper presents a comprehensive study on the application of Generative AI, specifically focusing on its role in enhancing the business analyst role within the realm of software development. It delves into the nuances of requirements engineering, a crucial phase in the software development lifecycle, and the transformation of user requirements into user stories. The paper explores the use of natural languages for crafting user requirements and emphasizes the revolutionary impact of Generative AI, particularly Open AI's Chat GPT, in automating and refining the process of generating user stories, acceptance criteria, and the definition of done. It speaks about the prompt-compression method with "LLM Lingua," and conversation chaining using "Lang chain" aimed at optimizing the efficiency of large language models in handling extensive requirements, thereby signifying a significant stride in the field of software development and requirements engineering.

### \*Corresponding author

Nilesh D Kulkarni, Sr. Director – Enterprise Architecture, Fortune Brands Home & Security, USA.

**Received:** December 11, 2023; **Accepted:** December 19, 2023; **Published:** December 29, 2023

**Keywords:** User Story, Generative AI, User Requirements, Requirements Engineering

### Introduction

Requirements engineering (RE) is a discipline that defines a common vision and understanding of socio-technical systems among the involved stakeholders and throughout their life cycle [1,2].

A system requirements specification, often referred to simply as a "requirements specification," is a technical document that delineates and structures the aspects and considerations of these systems from the perspective of requirements engineering (RE). An effective requirements specification brings forth numerous advantages, as documented in the literature [2,3,4]. These include –

- Facilitating the creation of an agreement and a business contract between customers and suppliers.
- Establishing a shared foundation for aiding in project budgeting, scheduling, and planning.
- Assisting in the validation and verification of the project's scope.
- Potentially supporting deployment and future maintenance activities.

The software development process commences with requirements engineering, a phase of paramount significance. In the seamless progression of software development, the effective gathering of requirements assumes a pivotal role, as emphasized in reference [5]. Efficient requirements not only lead to the development of a streamlined system but also contribute to cost-effectiveness in the final product.

Requirement elicitation marks the inaugural phase of requirements engineering, wherein all relevant users and stakeholders of the system convene to extract fundamental system requirements [6]. This process of requirement elicitation encompasses another pivotal aspect of requirements engineering, which is requirement gathering. Requirement gathering consists of specific steps outlined in reference, including requirement elicitation, requirement analysis, requirement documentation, requirement validation, and requirement management [5].

Requirements gathering serve the dual purpose of catering to both technical and business stakeholders, and as a result, they are typically crafted in natural languages. Natural languages are indeed chosen for this purpose because they offer a high level of communicative flexibility and universality. Humans are proficient in employing natural languages for communication, which makes them resistant to adoption issues as a technique for documenting requirements.

### Methodology

The data for this research was sourced from well-regarded academic databases, such as Google Scholar, IEEE Xplore, journals, and studies. We performed thorough searches using keywords like 'User Story', 'Use Case', 'Agile', 'Generative AI', and 'Generative AI User Story'. This method enabled us to uncover a wide array of sources that could potentially contribute to our study.

### Agile

Agile is an iterative and flexible approach to software development and project management that emphasizes collaboration, customer feedback, and the ability to adapt to changing requirements.

Agile is the ability to create and respond to change. It is a way of dealing with, and ultimately succeeding in, an uncertain and turbulent environment. The authors of the Agile Manifesto chose "Agile" as the label for this whole idea because that word represented the adaptiveness and response to change which was so important to their approach [2,7].

### User Story

A user story, is a way of expressing a software requirement from the perspective of an end-user or customer [8]. User stories are short, simple descriptions of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system. Kent Beck, creator of extreme programming (a software development methodology) developed the concept of stories. Kent's simple idea was to stop – stop working so hard on writing perfect document, and to get together to tell stories. In early 2000, Rachel Davies at Connextra build a story telling template after multiple experiments 'AS a <Role>, I want <Goal>, so that <Benefit>', which was later named as User Story Template.

### Role

A user story describes a specific role or persona who interacts with the software. This helps in understanding who will benefit from the feature.

### Goal

Each user story outlines a goal or objective that the user wants to achieve with the software. It focuses on the "what" and "why" of a feature rather than the "how."

### Benefit

User stories also highlight the value or benefit that the user will gain from the feature. This helps in prioritizing and understanding the importance of the user story.

Jeff Patton emphasizes the importance of ongoing conversations and collaboration between development teams and stakeholders to clarify and refine user stories [8]. This iterative process ensures a shared understanding of the business requirements.

Characteristics of a good user story are explained with an acronym INVEST which represents a set of criteria used to evaluate and write effective user stories in Agile software development.

### Independent

User stories should be independent of each other. This means that they should be self-contained and not rely on the completion of other stories. Independence allows for flexibility in prioritizing and sequencing stories.

### Negotiable

User stories should be negotiable, meaning that they are open to discussion and can be refined through collaboration between the development team and stakeholders. They should not be overly prescriptive or rigid.

### Valuable

Each user story should deliver value to the end-users or customers. It should focus on solving a real problem or meeting a specific need. Value helps prioritize stories based on their impact.

### Estimable

User stories should be estimable, meaning that the development team can reasonably estimate the effort required to implement

them. This helps with planning and resource allocation.

### Small

User stories should be small or appropriately sized. They should not be too large or complex. Small stories are easier to understand, implement, and test. They also allow for more frequent delivery of functionality.

### Testable

User stories should be testable, which means that there should be clear and measurable acceptance criteria associated with each story. These criteria define when the story is considered complete and working as intended.

### Generative AI

The field of AI was reinaugurated in 2000s, driven by the major three forces. First was Moore's law in action - the rapid movement of computer of computational power. By the 2000s computer scientists could leverage dramatic improvement in processing power, reduction in the form factor of computing with mainframe computers, minicomputers, personal computers, laptop computers, and the emergence of mobile computing devices, and the steady decline in computing costs [9].

AI has long been predicted as one of the prominent technologies capable of allowing communication among devices and machines as well as AI can simplify processes by solving problems at higher levels of speed and accuracy while at the same time managing large volumes of data [10,11,12,13].

A significant catalyst for this renewed fervor surrounding AI is the advent of Open AI's Chat GPT in November 2022 [14]. Chat GPT, an acronym for Generative Pre-trained Transformer, introduced the public not to AI but to a specific facet of AI – generative AI [15].

In earlier times, artificial intelligence (AI) was commonly linked to its effectiveness in handling analytical tasks [16,17]. Conversely, creative, and generative tasks were traditionally reserved for human capabilities. A survey conducted among innovation managers prior to the notable rise of GPT-3 in the last quarter of 2022 indicated that, during that period, the application of AI in activities like idea generation, idea evaluation, and prototyping was considered relatively less important in the innovation process [18].

However, within a short span, the emergence of transformer language models and generative AI has transformed this perspective. The utilization of AI in creative tasks, including idea generation, has now become a focal point of interest for both practitioners and researchers [19, 20, 21].

### Gen AI and User Story

#### Building Requirement Gathering

A Business Requirements is a documented record that offers an extensive and detailed account of the business requirements, goals, and functional prerequisites pertaining to a particular module or section within a software application. This information is typically gathered during business user interview sessions and is commonly facilitated by a business analyst or functional analyst.

The requirements are crafted with the user in focus, and these personas subsequently assume the role of actors during the design phase. User requirement should be composed in a manner that mirrors natural speech to ensure the user journey maps are distinctly and easily identifiable.

As an example (figure 1) – “Jane a B2B user from ABC Company accesses the Customer portal, where they promptly locate the "Orders" tab and view the list of active orders. To inspect a specific order, they simply click on its hyperlink, leading them to a comprehensive overview of the order's details”

Jane had a faucet that was old and finally broke, she heard about the connected faucet. She goes to Faucet Store's website and looks for the right faucet that meets her need but also aesthetically matches her kitchen theme. She takes assistance from the interactive product search which provides her with exactly what she is looking for."

**Figure 1:** Apply façade to phone system

### User Requirement to User Story

After the user requirements are created, product owners proceed to construct user stories, aiming to cover every aspect of the requirements. This process is quite labor-intensive and time-consuming as it involves meticulously defining all the elements of a comprehensive user story, including the acceptance criteria and the definition of done (figure 2).

**User Story 1:** As a customer, I want to be able to easily find a faucet that meets my needs and matches my kitchen theme on the Faucet Store's website, so that I can make an informed purchase decision..  
**Acceptance Criteria:**  
 - The website should have an interactive product search feature.  
**Definition of Done:**  
 - The interactive product search feature has been implemented and tested to ensure accurate and relevant results.

**Figure 2:** Apply façade to phone system

### Gen AI – User Story Builder

A Gen AI – Chat GPT can be used by providing the correct prompt like, Write an Epic, User Story, Acceptance Criteria and Definition of done for the given user requirement and Chat GPT can provide you with the right response shown in figure 3.

**ChatGPT**

**Epic**

**Title:** Enhanced Online Faucet Purchase Experience

**Description:** Develop an interactive, user-friendly e-commerce platform for Faucet Store, enabling customers to find and purchase faucets that not only meet their functional needs but also align with their kitchen aesthetics.

---

**User Story**

As a homeowner,

I want to use an interactive product search on Faucet Store's website,

So that I can find a faucet that meets my functional requirements and matches my kitchen theme easily.

---

**Acceptance Criteria**

1. Interactive Product Search Functionality:

**Figure 3:** Phone Application

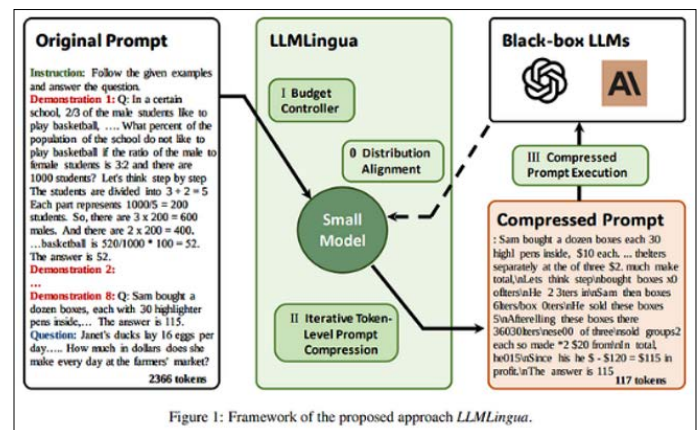
Sometime this might be a good idea, but we have experimented with this and found that keeping a chain of conversation or

requirement context is very difficult. With the limitations on the number of tokens returned and processed by Chat GPT can also create challenges with bigger requirements and projects.

As large language models (LLMs) models advance and their potential becomes increasingly apparent, an understanding is emerging that the quality of their output is directly related to the nature of the prompt that is given to them. This has resulted in the rise of prompting technologies, such as chain-of-thought (CoT) and in-context-learning (ICL), which facilitate an increase in prompt length. In some instances, prompts now extend to tens of thousands of tokens, or units of text, and beyond.

While longer prompts hold considerable potential, they also introduce a host of issues, such as the need to exceed the chat window's maximum limit, a reduced capacity for retaining contextual information, and an increase in API costs, both in monetary terms and computational resources. Chat GPT-3.5's token limit is 4,096 tokens, including tokens used for the prompt and Chat GPT's response. The updated GPT-3.5 Turbo language model has 16,385 tokens and a response output with a maximum of 4,096 tokens.

To address these challenges, we introduce a prompt-compression method in our paper, “LLMLingua: Compressing Prompts for Accelerated Inference of Large Language Models,” presented at EMNLP 2023 [22].



**Figure 1:** Framework of the proposed approach LLMLingua.

**Figure 4:** Apply façade to phone system

### Use of the Langchain Chain the Conversation

To chain the multiple requirements and to get a consolidated response, we have used langchain framework. The sample Python code for the implementation is as under –

```
import os
os.environ["OPENAI_API_KEY"] = "<key>"
from langchain.chat_models import ChatOpenAI
from langchain.prompts import ChatPromptTemplate
from langchain.schema import StrOutputParser
from langchain.memory import ConversationSummaryMemory,
ChatMessageHistory, ConversationBufferWindowMemory
from langchain.chains import ConversationChain

model = ChatOpenAI(model="gpt-3.5-turbo", temperature=0)

conversation_with_summary = ConversationChain(
    llm=model,
    # We set a low k=2, to only keep the last 2 interactions in memory
```

```
memory=ConversationBufferWindowMemory(k=10),
verbose=True
)

prompt1 = ChatPromptTemplate.from_messages([
    ("system", "As a business analyst, skilled in identifying persona,
    epic, user stories, acceptance criteria, definition of done for the
    project based on a given chat conversation."),
    ("human", "I have a chat conversation, and I need you to identify
    persona and create epic. The chat involves {context}")
])
prompt2 = ChatPromptTemplate.from_messages([
    ("system", "As a business analyst, skilled in identifying persona,
    epic, user stories, acceptance criteria, definition of done for the
    project based on a given chat conversation."),
    ("human", "I have this context {context}, and I need you to
    identify and create user stories, acceptance criteria and add it
    into context to create new description.")
])
prompt3 = ChatPromptTemplate.from_messages([
    ("system", "As a business analyst, skilled in identifying persona,
    epic, user stories, acceptance criteria, definition of done for the
    project based on a given chat conversation."),
    ("human", "I have this context {context}, and I need you to
    generate definition of done based on the acceptance criteria and
    add it into context to create new description.")
])

chain = (
    {"context": prompt1 | model | StrOutputParser()}
    | {"context": prompt2 | model | StrOutputParser()} | prompt3 |
    model | StrOutputParser()
)

context=new.readText()

for chunk in chain.stream({"context": context}):
    print(chunk, end="", flush=True)

from langchain.memory import
ConversationBufferWindowMemory
memory = ConversationBufferWindowMemory( k=1)

memory.save_context({"input": "hi"}, {"output": "whats up"})
memory.save_context({"input": "not much you"}, {"output":
"not much"})

print(memory.load_memory_variables({}))
```

## Conclusion

The paper concludes that the integration of Generative AI, especially Open AI's Chat GPT, into the business analyst's toolkit revolutionizes the domain of requirements engineering. It highlights the transformative capability of Generative AI in automating the conversion of user requirements into structured user stories thereby streamlining the software development process. The paper acknowledges the complexities involved in managing extensive requirements and the context continuity issues in large language models. The introduction of "LLM Lingua," a prompt-compression method, and "Langchain" for the chaining the conversations. The paper emphasizes that while the path to integrating AI into software development is fraught with challenges, the potential benefits in terms of efficiency, accuracy, and scalability are immense. It suggests that the future of software

development and requirements engineering is intrinsically linked with the advancements in AI and machine learning technologies [23].

## References

1. Pohl K (2010) Requirements Engineering: Fundamentals, Principles, and Techniques. Berlin, Germany: Springer-Verlag <https://link.springer.com/book/9783642125775>.
2. Robertson S, Robertson J (2006) Mastering the Requirements Process 2nd ed. Addison-Wesley <https://www.amazon.com/Mastering-Requirements-Process-Suzanne-Robertson/dp/0321419499>.
3. Kovitz B (1998) Practical Software Requirements: Manual of Content and Style. Manning <https://www.manning.com/books/practical-software-requirements?origin=product-look-inside>.
4. Withall S (2007) Software Requirements Patterns. Unterschleißheim, Germany: Microsoft Press <https://ptgmedia.pearsoncmg.com/images/9780735623989/samplepages/9780735623989.pdf>.
5. Malik MU, Chaudhry NM, Malik KS (2013) Evaluation of efficient requirement engineering techniques in agile software development. Int J Comput Appl 83: 24-29.
6. Rasool HF, Saher N, Iqbal Z, Ajmal MR, Arshad S (2014) Requirements engineering and its role in mobile telephone industry development. J Adv Comput Netw 2: 218-221.
7. Siebel T (2019) Digital Transformation: Survive and Thrive in an Era of Mass Extinction. Amazon <https://www.amazon.in/Digital-Transformation-Survive-Thrive-Extinction/dp/1948122480>.
8. Patton J, Economy P, Fowler M, Cooper A, Cagan M (2014) User Story Mapping: Discover the Whole Story, Build the Right Product. Amazon <https://www.amazon.in/User-Story-Mapping-Jeff-Patton/dp/1491904909>.
9. Davenport T, Mittal N (2023) All-in On AI: How Smart Companies Win Big with Artificial Intelligence. Amazon <https://www.amazon.in/All-AI-Companies-Artificial-Intelligence/dp/1647824699>.
10. Guzman AL, Lewis SC (2020) Artificial Intelligence and Communication: A Human-Machine Communication Research Agenda. New Media & Society 22: 70-86.
11. Dwivedi YK, Hughes L, Ismagilova E, Aarts G, Coombs C, et al. (2019) Artificial Intelligence (AI): Multidisciplinary Perspectives on Emerging Challenges, Opportunities, and Agenda for Research, Practice and Policy. International Journal of Information Management 57.
12. Schutzer D (1990) Business Expert Systems: the Competitive Edge. Expert Systems With Applications 1: 17-21.
13. Schniederjans DG, Curado C, Khalajhedayati M (2020) Supply Chain Digitization Trends: An Integration of Knowledge Management. International Journal of Production Economics 220.
14. Models GPT-3 (2023) OpenAI, San Francisco, CA, USA.
15. Brown TB, Benjamin M, Nick R, Melanie S, Jared K, et al. (2020) Language models are few-shot learners. Proc Adv Neural Inf Process Sys 33: 1877-1901.
16. Kakatkar C, Bilgram V, Fuller J (2020) Innovation analytics: Leveraging artificial intelligence in the innovation process. Business Horizons 63: 171-181.
17. Marchuk A, Biel S, Bilgram V, Jensen S (2021) The best of both worlds: Methodological insights on combining human and AI labor in netnography. Netnography Unlimited: Understanding Technoculture Using Qualitative Social Media Research [https://ebrary.net/216792/business\\_finance/](https://ebrary.net/216792/business_finance/)

- best\_worlds\_methodological\_insights\_combining\_human\_labor\_netnography.
18. Fuller J, Hutter K, Wahl J, Bilgram V, Tekic Z (2022) How AI revolutionizes innovation management-Perceptions and implementation preferences of AIbased innovators. Technological Forecasting and Social Change 178.
  19. Bouschery SG, Blazevec V, Piller FT (2023) Augmenting human innovation teams with artificial intelligence: Exploring transformer-based language models. Journal of Product Innovation Management 40: 139-153.
  20. Brem A, Giones F, Werle M (2023) The AI digital revolution in innovation: A conceptual framework of artificial intelligence technologies for the management of innovation. IEEE Transactions of Engineering Management 70: 770-776.
  21. Bilgram V, Canadas Link D, Lang-Koetz C (2023) Generative KIs in Kreativprozessen: Praxiserfahrungen aus den ersten Monaten mit ChatGPT & Co. Ideen- und Innovationsmanagement 1: 18-22.
  22. LLMLingua: Innovating LLM efficiency with prompt compression. Microsoft Research Blog <https://www.microsoft.com/en-us/research/blog/llmlingua-innovating-llm-efficiency-with-prompt-compression>.
  23. Boone T, Ganeshan R (2007) The Frontiers of E Business Technology and Supply Chains. Journal of Operations Management 25: 1195-1198.