

Recipient Experience Design

Evolutionary CX Through Conversational Patterns



Whitepaper by Shashikant Jha & Prasanna Kumar

Table of Contents

Table of Contents	01
Introduction	02
RXD: Evolutionary CX	04
Core Principles of RXD	04
Goals of RXD	05
Key Benefits	06
Summary	06
Strategy through Execution: The RXD Way	07
Shifting focus to Conversations: Conversations as products	07
Proactive Conversations: Conversations that just work	08
Dynamic response models: Designing with happy and not-so-happy paths	10
Persona incorporation in Design	11
Summary	14
Reusability through patterns	15
What are patterns in conversation?	15
How to use patterns in conversation design?	16
Conclusion	18
References	18

Introduction



The art of conversation is the art of hearing as well as of being heard.



— William Hazlitt, *Selected Essays*, 1778-1830

Conversational design is the art and science of creating natural and intuitive interactions between humans and machines. Conversational design can enhance the user experience, satisfaction, and loyalty of your voice/chat channel, as well as increase the efficiency, effectiveness, and scalability of your business.

However, conversational design is not without its challenges and pitfalls. Many voice/chat channels suffer from common problems that can negatively affect the quality and outcome of the conversations, such as:



Asking for too much data

Some voice/chat channels ask the user for personal or sensitive data that is not relevant or necessary for the conversation, such as their name, email, phone number, or location. This can make the user feel uncomfortable or suspicious and may lead them to abandon the conversation.



Failing to solve the problem or answer the question

Some voice/chat channels do not provide the user with the information or solution they need, and instead transfer them to a human agent or another channel. This can make the user feel frustrated or disappointed and may lead them to lose interest in the voice/chat channel.



Forcing the user to stay in the conversation

Some voice/chat channels do not provide the user with the information or solution they need, and instead transfer them to a human agent or another channel. This can make the user feel frustrated or disappointed and may lead them to lose interest in the voice/chat channel.



Agent transfer proxy

Many chatbots only transfer customers to agents. They do not add any value to customers and increase the AHT (Average Handling Time) and call/chat length.

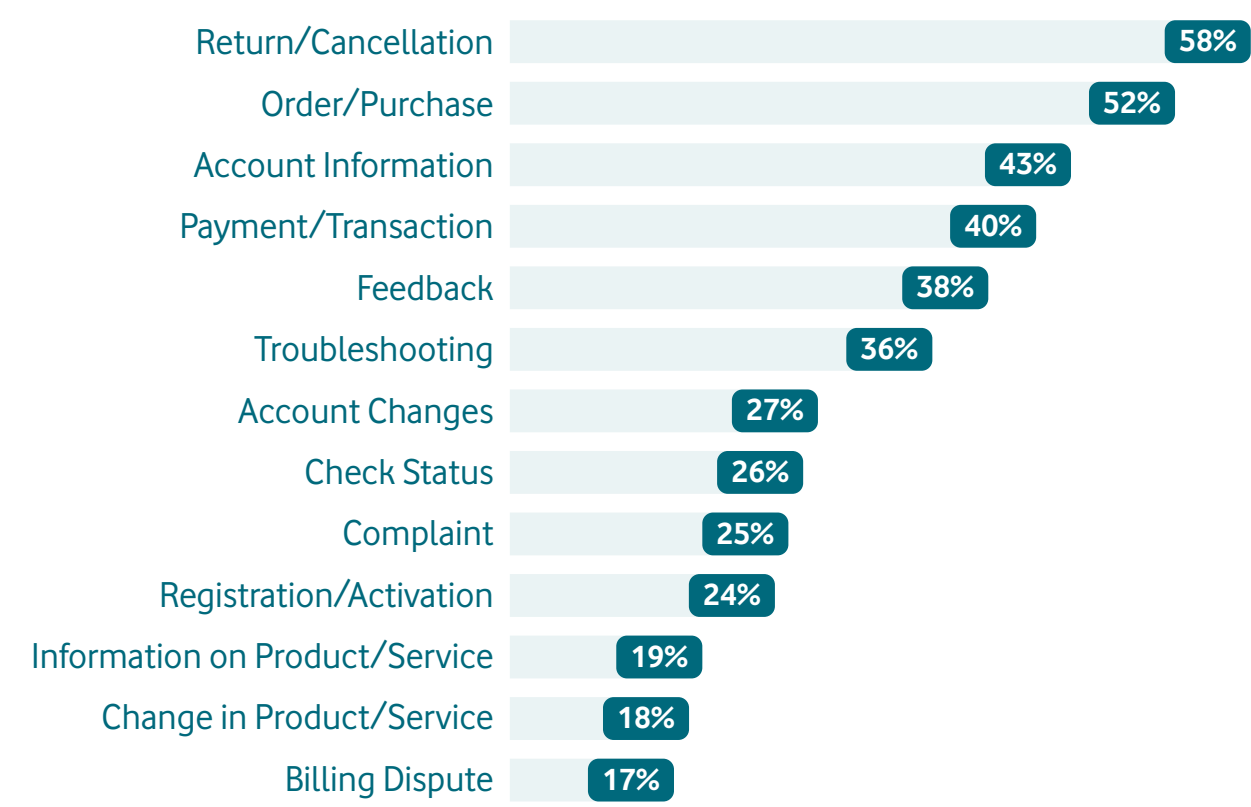
Even with all the advancements in AI and conversation design, the experience across the chatbots in general haven't improved much.

Introduction

A recent study from Gartner, indicates only **8%** of customers used a chatbot during their most recent customer service experience. Of those, just **25%** said they would use that chatbot again in the future. Despite customer service and support leaders' growing focus on chatbots, customer use of them remains low, suggesting they don't consistently help customers accomplish their goals.

A Gartner survey of **497 B2B and B2C customers** from **December 2022 through February 2023** found the ability of a chatbot to move the customer's issue forward was the top driver of adoption, explaining **18%** of the variance in customers' likelihood to use their chatbot again.

Resolution rates vary greatly by issue type: Just **17%** of billing disputes are resolved by customers who used a chatbot at some stage in their journey, while resolution rates for customers making a return or cancellation were as high as **58%**.



The infographic above from Gartner, shows the difference in resolution rates across different areas when using a chatbot.

Creating better conversations for our voice/chat channel is challenging. We need to consider the engagement, information, and respect of our users. We also need to identify the different types of users and their goals. We need to keep our conversations moving forward and ensure our chatbots perform consistently. In this whitepaper, we introduce Recipient Experience Design (RXD) as a new way of designing conversations. We will explain how RXD can improve our conversations and save us time.

RXD: Evolutionary CX

RXD or Recipient Experience Design, is a mechanism of designing conversations that considers the users as recipients of conversation, rather than customers or consumers. RXD recognizes that not every user who comes through your voice/chat channel has the same intention, expectation, or preference. Therefore, RXD advocates that you tailor your conversations according to the context, purpose, and profile of each recipient.

Core Principles of RXD



RXD operates on four core principles.

1 Chatbots are not just a tool to transfer customers

Chatbots are a way to engage with customers, understand their needs, and provide them with personalized solutions. Chatbots can also build trust and loyalty by offering consistent and reliable service, as well as proactive and timely support.

2 Chatbots are not products but collection of products

RXD believes that chatbots are more than just a single product. Chatbots are composed of various conversational journeys or designs, each of which is a product on its own and should be treated as such. Chatbots are also integrated with other products, such as CRM, analytics, and payment systems, to enhance their capabilities and value.

3 Users are not consumers but are recipients

Users are not passive consumers who only receive information from chatbots. Users are active participants who interact with chatbots, provide feedback, and co-create value. Users are also diverse and have different preferences, expectations, and goals. Chatbots should respect and accommodate the individuality and agency of users and treat them as recipients of value rather than consumers of products.

4 Conversations should try to provide recipients with the information they are seeking without forcing them to provide data first

Conversations are the core of chatbot interactions, and they should be natural, relevant, and helpful. Conversations should aim to deliver the information that users are looking for, without requiring them to share unnecessary or sensitive data first. Conversations should also be respectful of users' privacy and consent and avoid intrusive or manipulative tactics.

Goals of RXD



RXD has four core goals.

1 Recipient-centric conversations

To create recipient-centric conversations that are designed from the recipient's point of view, not based on assumptions or scripts. To engage the recipient in a meaningful dialogue, listening to their needs, preferences, and feedback. To provide them with relevant information, options, and guidance. To respect their choices, privacy, and autonomy.

2 Recipient segmentation

To implement recipient segmentation that groups recipients according to their characteristics, behaviors, and goals, such as genuine customers, potential customers, competitors, researchers, etc. To understand their motivations, expectations, and challenges. To customize the conversations to each segment's needs and preferences.

3 Recipient evaluation

To conduct recipient evaluation that measures the effectiveness and satisfaction of the conversations with the recipients, using feedback such as ratings, comments, suggestions, or complaints. To identify the strengths and weaknesses of the conversations, such as clarity, relevance, accuracy, or empathy. To improve the conversations based on the insights and learnings from the recipients.

4 Recipient engagement

To foster recipient engagement that creates conversations that capture and sustain the attention and interest of the recipients, using appropriate tone, style, and content for each segment. To enhance the conversations with humor, emotion, storytelling, or gamification. To provide feedback, recognition, or rewards for the recipients' participation.

RXD is a holistic and dynamic approach that puts the recipient at the heart of the process. RXD helps create engaging, personalized, and satisfying conversations that meet the needs and expectations of the recipients. RXD enables building trust, loyalty, and value with the recipients, and achieving business goals.

Key Benefits



Now that we have seen how RXD philosophy works for conversation design, we will look at key benefits it introduces.



Persona Incorporation

RXD can design conversational assistants that cater to different user personas, such as novice, expert, casual, or professional. This way, users can get more personalized and relevant content with less or the same amount of data. Moreover, RXD can create dynamic interaction models that adapt to the user's behavior and preferences, making the conversation more engaging and natural.



Request variation tackling

RXD can manage various types of user requests, from simple to complex, without compromising the quality and accuracy of the responses. RXD uses happy paths and repair paths instead of main cases and edge cases, which means that users can skip or change steps in the conversation without getting stuck or confused. RXD also allows users to have more unconventional pathways for conversing, such as asking multiple questions, changing topics, or providing feedback.



Dynamic response model

RXD can generate adaptive responses that change state depending on the user's interaction. For example, RXD can provide directed responses for tech-savvy users who want quick and concise answers, or explanatory responses for tech-naïve users who need more guidance and clarification. RXD can also adjust the tone, style, and format of the responses to suit the user's mood and context.

Summary



In summary, Recipient Experience Design (RXD) is a conversation design approach that views users as recipients rather than customers or consumers. It tailors conversations to the context, purpose, and profile of each recipient. RXD operates on four core principles: viewing chatbots as tools for customer engagement and personalized solutions, seeing chatbots as a collection of products rather than a specific product, treating users as active participants and recipients of value, and aiming to provide information without requiring unnecessary or sensitive data first.

RXD has four core goals: creating recipient-centric conversations, implementing recipient segmentation, conducting recipient evaluation, and fostering recipient engagement. It introduces key benefits such as persona incorporation, request variation tackling, and a dynamic response model. Overall, RXD is a holistic and dynamic approach that puts the recipient at the heart of the process, helping to create engaging, personalized conversations that meet the needs and expectations of the recipients, and achieving business goals. In the next section we will take a detailed look at the execution strategy.

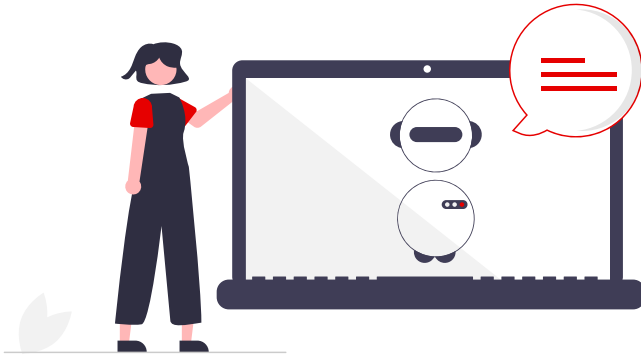
Strategy through Execution: The RXD Way



To improve chatbot adoption, the key is to focus on improving the chatbot's ability to move customers' issues forward.



Now, let's look at how we can implement RXD in our conversation design strategy. To implement RXD, we'd start with the core principles and start building the execution strategy around it. We will look at the key aspects of RXD



Conversations as products



Conversations that provide information to user without forcing it first



Request variation tackling and dynamic response model



UX Persona incorporation in designs

Shifting focus to Conversations

Conversations as products



Chatbots are often seen as products that provide a certain functionality or service to the users. However, this view may not capture the full potential and complexity of chatbot user experience. A more nuanced way of looking at chatbots is to consider them as collections of products, each corresponding to a different conversational journey. A conversational journey is a specific topic or task that the user can accomplish through a chatbot interaction. For example, checking the balance is one conversational journey, and the features that enable it are the backend integrations for authentication and balance retrieval. By focusing on the conversational journeys as products, we can better design and evaluate chatbots according to the following criteria:



Sufficiency

Does the product provide the user with the complete and accurate information or service they need?



Inclusivity

Does the product cater to the diverse needs and preferences of the users who may trigger it?

Let's take the example of a journey to check balance and see how these criteria can be applied. Depending on the context and the user, this journey may have different outcomes and expectations.

We can identify three types of users for this journey:

Prepaid users

These users want to check their balance quickly and easily, without any unnecessary steps or delays. The best way to serve them is to authenticate them and provide them with their balance information as soon as possible.

Postpaid users

These users may have mistakenly triggered the balance checking journey, or they may have a different intention, such as checking their bill or usage. The chatbot should acknowledge their request and use an implicit confirmation to verify that they want to proceed with the balance checking journey. If not, the chatbot should offer them alternative options or allow them to cancel the journey. If yes, the chatbot should explain that they do not have a balance as postpaid users and ask them if they want to check their bill or usage instead.

Testers

These users are not actual customers, but they are curious about the chatbot experience and want to try out the balance checking journey. They may not have valid authentication data, and they may want to stop the journey at any point. The chatbot should not force them to authenticate, but rather ask for their consent and inform them about the authentication process and details. The chatbot should also allow them to opt out of the authentication or the journey and continue the conversation without disconnecting or transferring the call.

Proactive Conversations

Conversations that just work

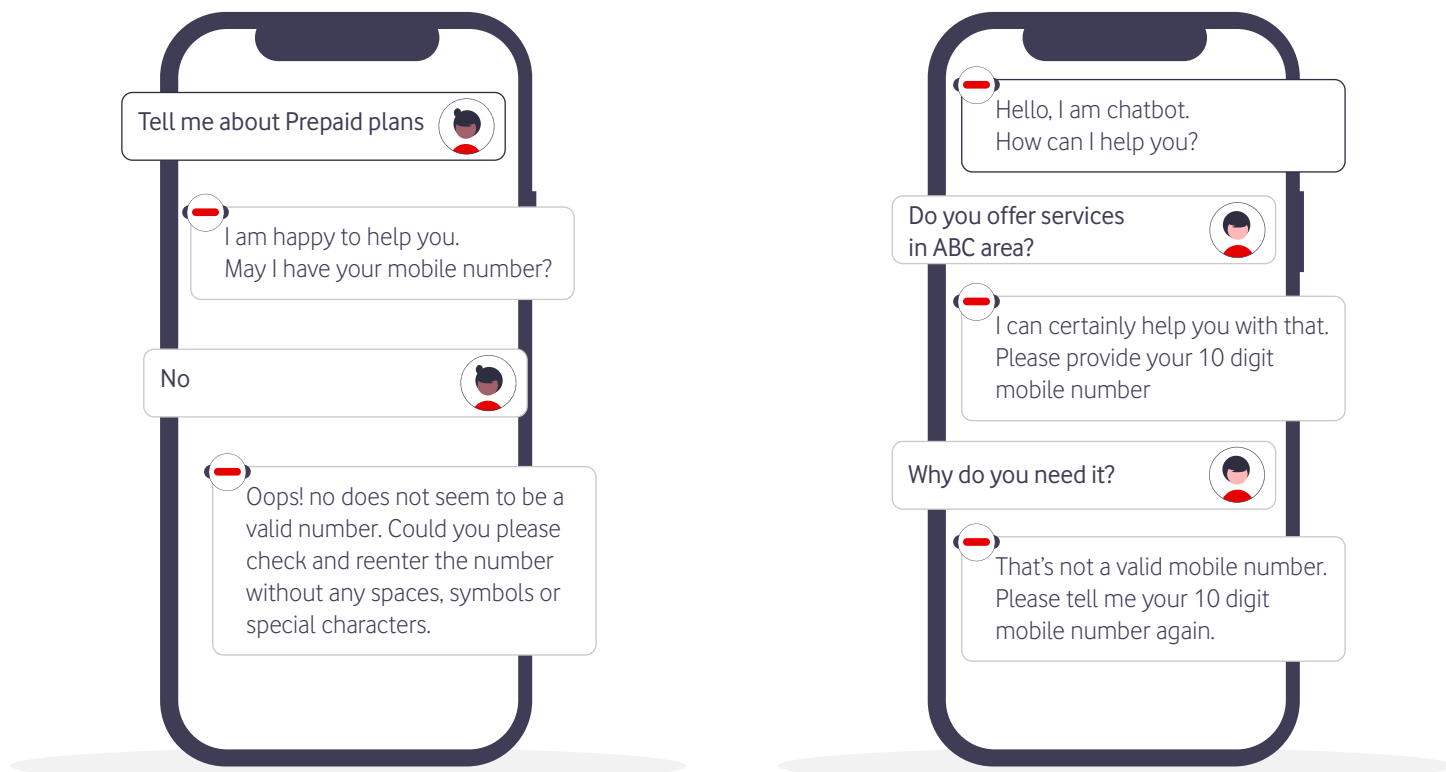


Another problem that lies with current conversation design practice is that the conversations are passive in nature. Not only they keep waiting for users for some input rather than a proactive approach, but also users are frequently pestered for their data first to access some information.

RXD advocates that you do not ask the recipient for any personal or sensitive data unless it is necessary for the conversation. For example, do not ask for their name, email, phone number, or location before providing them with the information they need. This can make them feel uncomfortable or suspicious and may lead them to abandon the conversation. Instead, offer them the option to share their data voluntarily, or ask for it only after you have established trust and rapport with them. In addition to respecting user privacy, RXD emphasizes the importance of context-awareness in conversation design. By understanding the user's current situation and past interactions, the system can anticipate needs and provide assistance without requiring explicit prompts. This not only streamlines the interaction but also creates a more personalized and engaging experience. For instance, if a user regularly checks the weather in their area, the system could proactively provide weather updates at the start of each interaction. This proactive stance transforms the user experience from reactive to interactive, fostering a more natural and efficient dialogue.

Strategy through Execution: The RXD Way

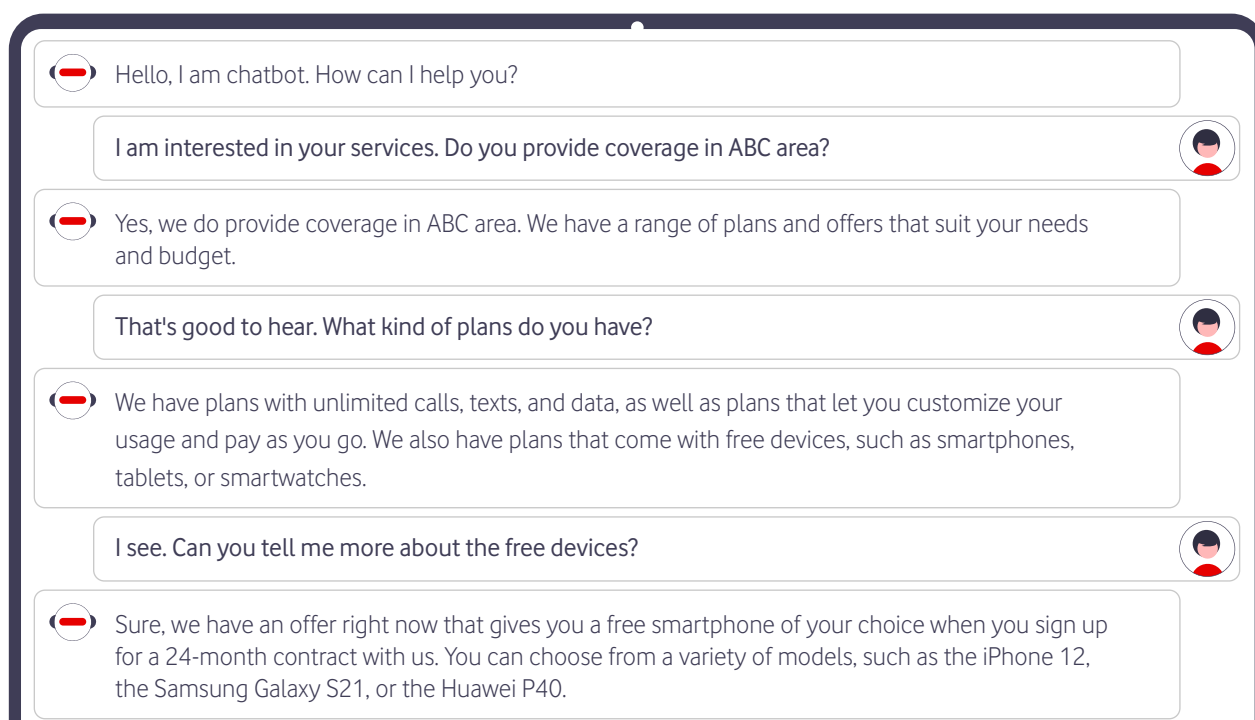
Now, let's look at another example where chatbots can play a more proactive role in conversation and avoid getting stringent on data collection first.



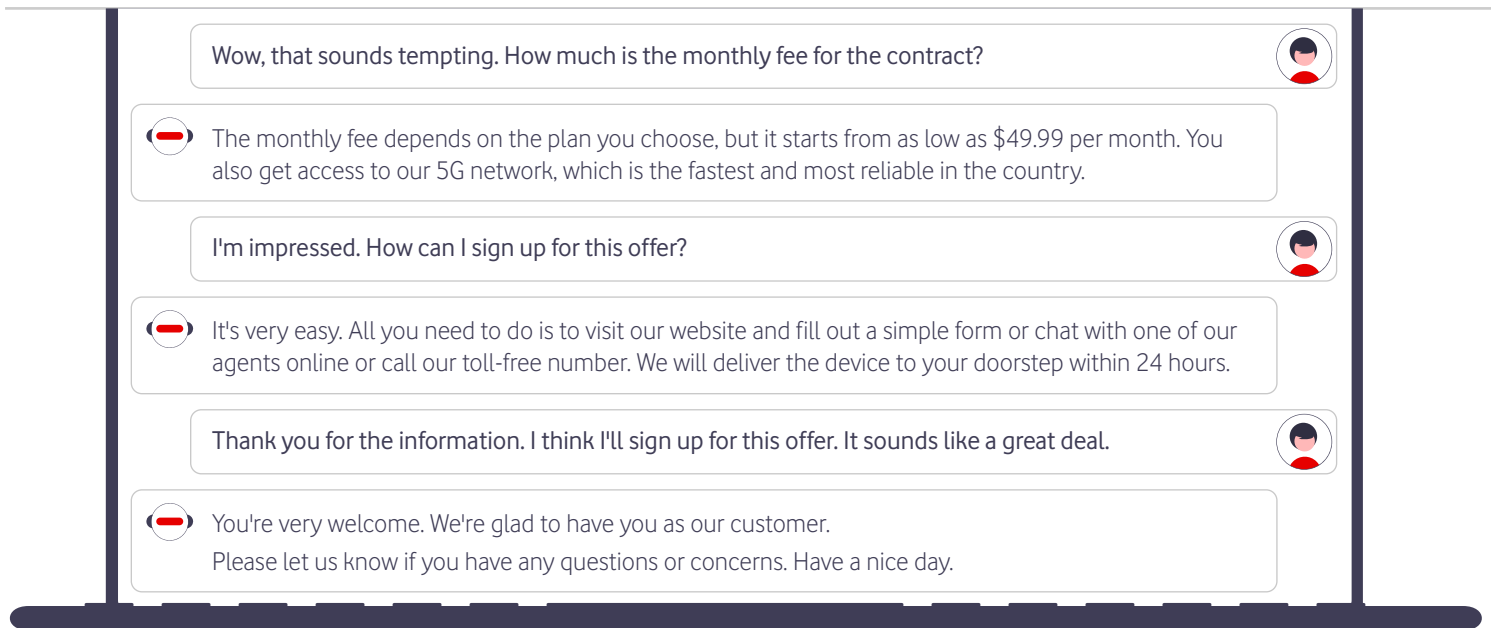
These two are just of many examples where a bot gets stringent with data collection and in process loses a potential customer who is just there to browse.

RXD advocates that instead of asking for data from customers first, the chatbot could start providing information to build rapport. Once the user is comfortable, chatbot can then start on data gathering to build a potential sales lead.

Now, let's see how this can be redesigned following the RXD principles



Strategy through Execution: The RXD Way



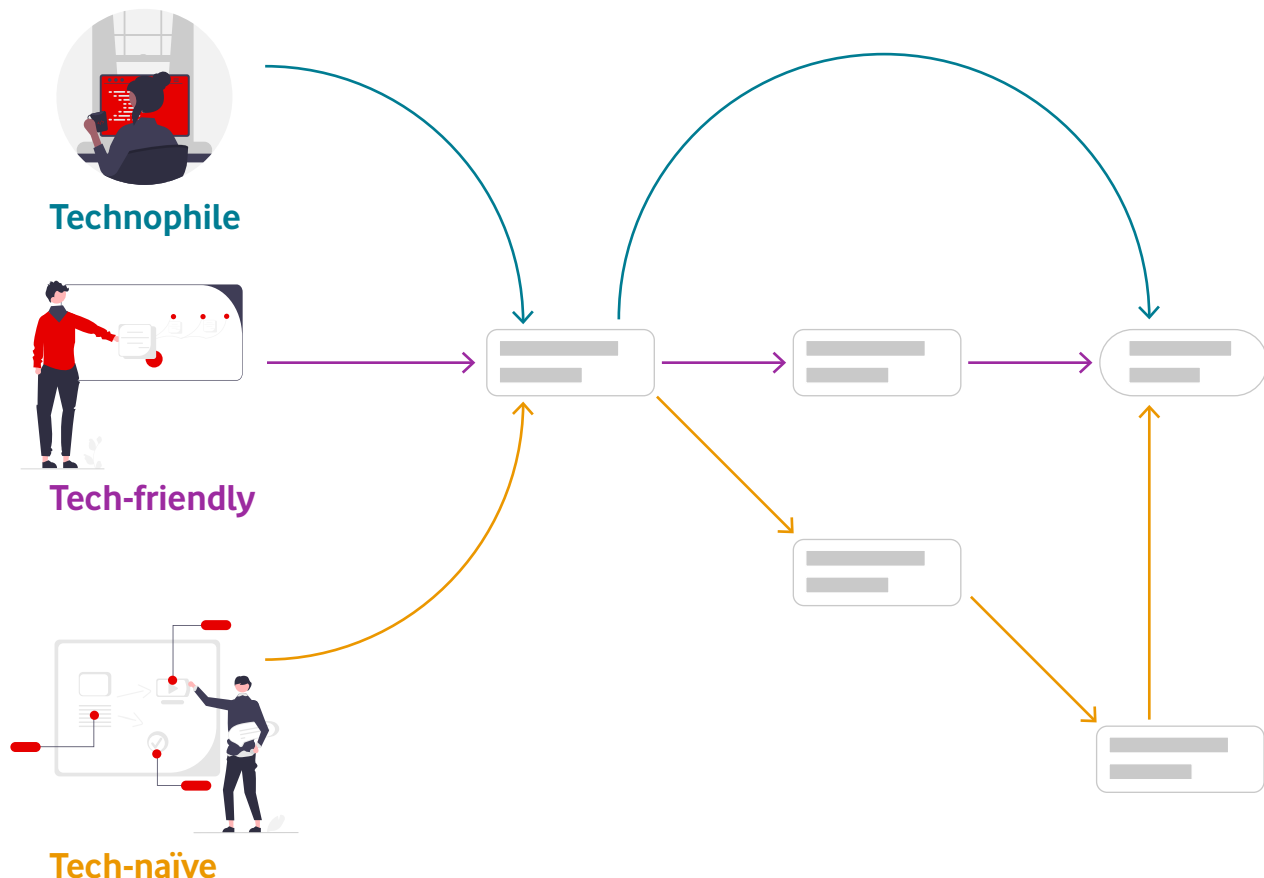
Dynamic response models

Designing with happy and not-so-happy paths



As we saw above, happy, and not-so-happy paths are essential to a conversational journey. A crucial point to note here is that the 'not-so-happy paths' are not the failure paths, but they are just slightly deviated from the happy paths.

Below is a visual representation of happy and not-so-happy paths.

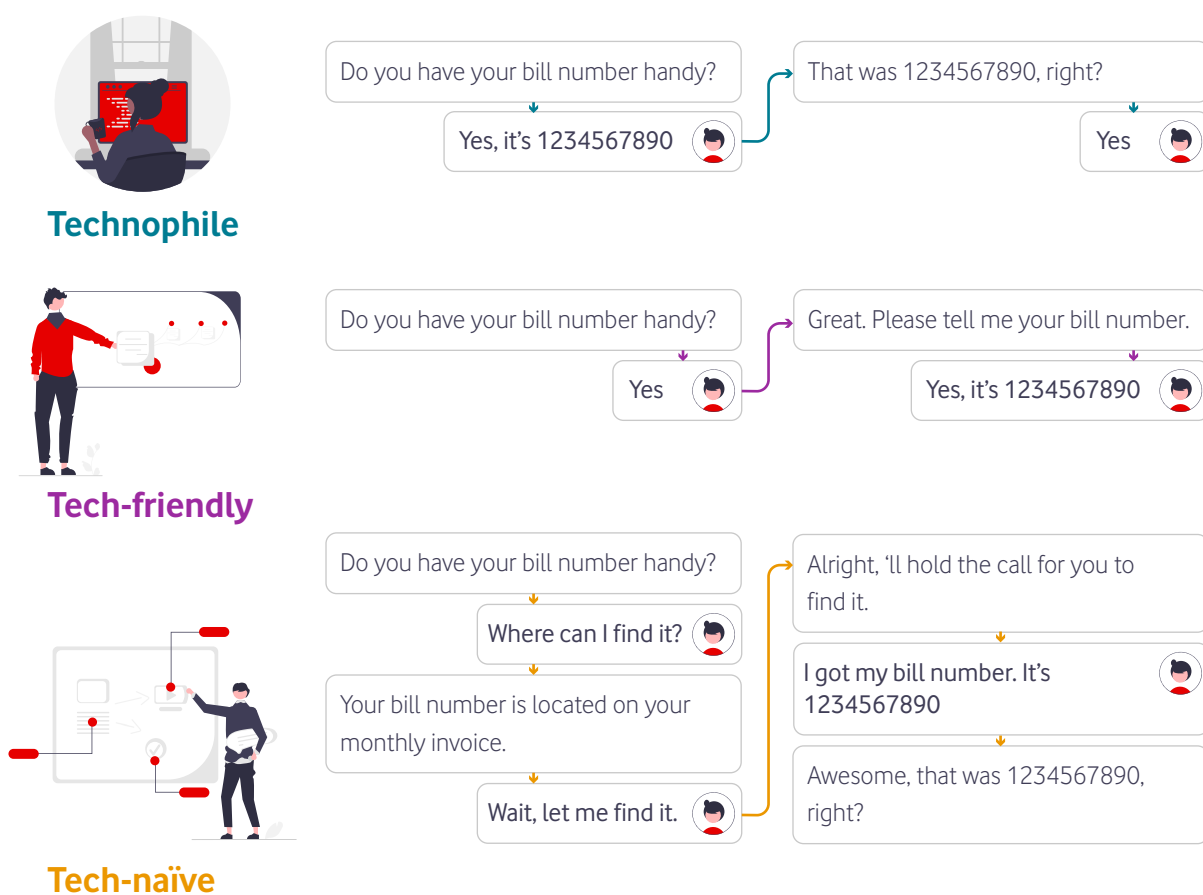


Strategy through Execution: The RXD Way

RXD emphasizes not to have ‘fixed’ paths but allow for more natural conversations by flexible paths suited for different users.

A tech-naïve user, typically takes more turns to reach the same outcome than a tech-friendly user. Additionally, tech-naïve users would also tend to ask more follow-up and clarifying questions. When it comes to technophiles, they would be more likely to jump nodes and turns and the conversation must allow them the flexibility to do so.

Below is a depiction of a how conversation centered around collecting bill number from tech-friendly, technophile and tech-naïve users respectively, could go.



From the above, we can clearly see the difference in experience and expectations for all the user categories. A conversational solution must be able to adapt to the expectations and provide appropriate experience.

Persona incorporation in Design

User personas are representations of the target users of a product or service, based on real user research and data. They help designers and developers to understand the needs, goals, preferences, and behaviors of the users, and to create user-centric solutions that meet their expectations and solve their problems. User personas are widely used in UX design, but they are also essential for conversation design, the process of designing natural and intuitive conversational interactions between users and conversational agents.

Design thinking is a human-centered approach to innovation that integrates the needs of people, the possibilities of technology, and the requirements for business success. It involves empathy, ideation, and experimentation to solve complex problems. Similarly, the incorporation of user personas in design is deeply rooted in empathy, focusing on the users' needs, goals, and behaviors. By employing user personas, designers apply a design thinking mindset to create solutions that are not only technologically feasible and economically viable but also desirable from a user's perspective. This alignment with design thinking enables designers to craft experiences that resonate with users on a personal level, ensuring that the final product is not just functional but also emotionally engaging and user-friendly. Thus, user personas serve as a bridge between the abstract principles of design thinking and the concrete demands of conversation design, fostering a culture of innovation that prioritizes the user experience.

Conversation design is a complex and challenging task that requires a deep understanding of the user's context, intent, and emotion, as well as the ability to craft engaging and natural dialogues that achieve the desired outcomes. User personas can help conversation designers to create more personalized, relevant, and effective conversational experiences that cater to different user types and scenarios. By using user personas, conversation designers can:

- Identify the common goals and intents of the users, and design conversational flows that guide them to achieve them.
- Anticipate the possible questions, responses, and feedback of the users, and design conversational prompts and actions that address them.
- Adapt the tone, style, and personality of the conversational agent to match the user's preferences and expectations.
- Test and evaluate the conversational experience with different user personas and identify the strengths and weaknesses of the design.

However, creating and using user personas for conversation design is not a straightforward process. Different organizations may have different sets of user personas defined for their products or services, and these personas may not be compatible or consistent with each other. Moreover, user personas may change over time, as the user's needs, goals, and behaviors evolve. Therefore, conversation designers need a systematic and scalable way to incorporate user personas into their conversation design process, and to ensure that their conversational agents can adapt to the diverse and dynamic needs of the users.

This is where RXD comes in. RXD is a framework for conversation design that leverages user personas to create modular and reusable conversational components that can be applied to different domains and contexts.

In this section, we will explain how RXD can help conversation designers to improve the incorporation of user personas in conversation design, and to create conversational agents that can meet the diverse

Strategy through Execution: The RXD Way

and dynamic needs of the users. We will also provide some examples and tips on how to use RXD in practice. RXD is a design approach that adapts to different user personas, based on their pain points and behavior. These factors help to create personalized paths for each persona. For example, a user persona who is not tech-savvy or familiar with chatbots may face some challenges. They may need more guidance and clarification from the chatbot. To accommodate this persona, the design should have these features:

- Adjustable timeouts to give more time for input.
- Option to switch from the main journey to answer FAQs and explain terms.
- Highlighting of key phrases to show what is expected from the user.
- More proactive actions from the chatbot instead of asking users to click on links.

Another important aspect of personas is their language style. This affects how the chatbot responds to them. Remember – ‘Similarity builds rapport.’ The more your chatbot matches the user’s language style, the more they will trust and connect with it. This is one of the benefits of RXD; a flexible response model that suits different user personas.

Look at these two examples

Tech-friendly

My broadband router isn't functioning.



Okay, let me run a diagnostic check on my end and find the problem.

Tech-naïve

I can't get on Google. What does that mean?



It seems you are not connected to internet probably due to a broadband or router issue. I can run a quick diagnostic check on my end to find the issue

They show how the same response from the same journey can vary for different user personas. This creates a more relevant and enjoyable experience for the user.

Another way to apply RXD is to design different journeys for different user goals. Depending on what the user wants to achieve, the chatbot can offer different options and guide them accordingly.

For example,

a user who wants to book a flight may have different goals, such as finding the cheapest deal, the fastest route, or the most flexible schedule. The chatbot can ask the user about their preference and then provide relevant information and suggestions. The chatbot can also use different response styles, such as informative, persuasive, or humorous, depending on the user’s goal and persona.

Strategy through Execution: The RXD Way

By designing different journeys for different user goals, the chatbot can create a more personalized and satisfying experience for the user. This can also increase the user's engagement and loyalty with the chatbot.

RXD also allows you to design different feedback mechanisms for different user actions. Depending on how the user interacts with the chatbot, the chatbot can provide different types of feedback, such as confirmation, encouragement, correction, or apology. The chatbot can also use different feedback styles, such as formal, casual, friendly, or professional, depending on the user's persona and goal.

For example,

a user who successfully completes a task may receive a confirmation message from the chatbot, such as "Great, your flight is booked." The chatbot can also add some encouragement, such as "You're all set for your trip. Have a wonderful time." The chatbot can also use a friendly and casual tone, such as "Awesome, you're good to go. Enjoy your vacation." Alternatively, a user who makes a mistake or encounters an error may receive a correction or apology message from the chatbot, such as "Sorry, there was a problem with your payment. Please try again." The chatbot can also add some reassurance, such as "Don't worry, we'll fix this soon." The chatbot can also use a professional and formal tone, such as "We apologize for the inconvenience. Please contact our customer service for assistance."

By designing different feedback mechanisms for different user actions, the chatbot can create a more responsive and supportive experience for the user. This can also increase the user's confidence and satisfaction with the chatbot.

Summary



RXD is a user-centric approach that adapts to the varying needs and preferences of diverse types of users. By allowing for flexible paths, RXD enables more natural and engaging conversations that can lead to better outcomes and user satisfaction. RXD also recognizes the importance of providing appropriate feedback and guidance to users, especially those who are less familiar with technology or the domain of the conversation. RXD is a promising framework for designing conversational agents that can cater to diverse user groups and contexts.

In contrast to traditional dialog design methods that rely on predefined scripts and rigid flows, RXD offers a more flexible and dynamic way of creating conversational agents. RXD allows for the design of multiple paths that can accommodate different user goals, preferences, and behaviors. RXD also enables the evaluation and improvement of conversational agents based on user feedback and data. RXD is a novel and effective approach that can help designers create more user-friendly and natural conversational agents.

Reusability through patterns

“To understand is to perceive pattern.” — To understand is to perceive pattern

Conversations are complex and dynamic phenomena that involve various linguistic, cognitive, and social aspects. Designing conversational agents that can interact naturally and effectively with human users is a challenging task that requires a lot of time and effort from conversation designers. However, not all conversations are completely unique and unpredictable. There are some common patterns and structures that can be identified and reused across different domains and contexts. These patterns can help conversation designers to envision and design conversations in a modular fashion, where each module represents a self-sustaining conversational block that can be plugged and played with other modules to form a coherent and engaging conversational journey. In this section, we will introduce some of the most useful and widely applicable patterns in conversations and show how they can be used to design better conversational agents.

What are patterns in conversation?

-  **Patterns** in conversations are recurring and reusable conversational blocks that can capture the common goals, intents, and behaviors of users and agents in different situations. Patterns can help conversation designers to reduce the complexity and redundancy of conversational design, and to increase the consistency and quality of conversational interactions. Patterns can also facilitate the reuse and sharing of conversational design knowledge and best practices across different domains and contexts. By using patterns, conversation designers can create modular and scalable conversational agents that can meet the diverse and dynamic needs of users.
-  **Patterns** are sequences of conversational actions that can be generalized and applied to different situations and contexts. Patterns help us understand the structure and logic of human conversation and make the design of conversational agents more efficient and consistent. In this section, we will explain what patterns are, why they are important for conversation design, and how they can be used to create modular and reusable conversational components.
-  **Patterns** are based on the observation and analysis of regularities in natural language interaction. They capture the common goals, intents, and behaviors of conversational participants, and abstract them into simplified labels. For example, a pattern for greeting someone can be described as: “A greets B, B greets A back”. This pattern can be instantiated in many ways, depending on the specific words, tone, and context of the interaction.
-  **Patterns** help us scale our thinking about complex conversational systems, by breaking them down into smaller and manageable units. By identifying and categorizing the patterns that occur in each domain or scenario, we can design conversational agents that can handle a variety of user requests and situations, without having to script every possible dialogue path. Patterns also help us ensure the coherence and consistency of the conversational experience, by following the norms and expectations of natural language communication.

How to use patterns in conversation design?



One way to use patterns in conversation design is to create modular and self-sustaining conversational components, which we call modular conversations. Modular conversations are short and focused dialogues that address a single topic or task, such as authentication, information gathering, or problem solving. Modular conversations can be designed independently from each other, and then connected and sequenced by a higher-level component, which we call the connector. The connector is responsible for managing the flow and context of the conversation, and deciding which micro-conversation to invoke next, based on the user's input and the system's state.

The advantages of using modular conversations are:

Independent designing

Modular conversations allow for parallel and collaborative design, as different designers can work on different parts of the conversation, without affecting each other. Once the modular conversations are ready, they can be easily integrated and tested by the connector.

Reusability

Modular conversations can be reused across different conversational agents and domains if they share the same pattern and goal. This reduces the time and effort required for designing new conversational experiences and increases the quality and consistency of the outcomes.

Modular design

Modular conversations enable a clear and granular visualization of the conversational structure and logic, as each micro-conversation represents a distinct and self-contained unit of functionality. This facilitates the identification and resolution of potential issues and gaps in the design and allows for a better assessment of the complexity and usability of the conversational system.



In the above infographic, we try to illustrate how patterns are part and parcel of our day-to-day conversations, and how we can incorporate the same in the conversation design process. Let's understand it better through below scenarios happening between an agent and a user and a chatbot and a user.

Scenario 1

- A user calls in the contact center and is routed to an agent.
- Agent greets the user and asks for reason of their call.
- User informs agent of their problem, and the agent listens to them patiently.
- Agent verifies both the problem as well as identity of user.
- Agent provides either a resolution or an escalation to next service level to user.

Scenario 2

- A user calls in the contact center and is routed to a chatbot.
- Chatbot greets the user and asks for reason of their call.
- User informs chatbot of their problem, and the chatbot listens to them.
- Chatbot verifies both the problem as well as identity of user.
- Chatbot provides either a resolution or an escalation to next service level to user.

As evident from above, there is a fair bit of commonality in the process followed by both agent and chatbot, as well as the components of the journey.

E.g. – An authentication mechanism is a self-sustained pattern, which can be designed once and used as plug-and-play in almost every conversational journey in a chatbot.

Similarly, you could also build a micro-conversation for collecting phone number from customer and then use this micro-conversation in all the places wherever it's needed to collect the phone number from user. As we saw above, the biggest benefit of using a pattern based modular conversation design is that it allows the designers to break a fairly complex design into simpler, plug-and-play capable chunks and allows for re-usability across different conversational designs in the chatbot.

Conclusion

In conclusion, we have presented the concept of Recipient Experience Design (RXD), an innovative approach to conversational design that aims to improve the user satisfaction, loyalty, and adoption of voice/chat channels. RXD is based on the idea that users are not just customers or consumers, but recipients of conversation, who have unique needs, goals, and preferences depending on the situation. By understanding the context, purpose, and profile of each recipient, we can design conversations that are more relevant, personalized, and respectful, and that can move the user's issue forward. We have also shown some examples and best practices of how to apply RXD in different scenarios, such as customer service, sales, marketing, and feedback.

Below are some improvements we anticipate by incorporating RXD framework in your design process.

Persona Incorporation

Studies have shown that incorporating user personas into conversational AI can lead to a **20% increase in user engagement**. By tailoring conversations to individual user personas, RXD can improve content personalization, resulting in a **15% reduction in user drop-off rates** during interactions.

Request Variation Tackling

Implementing RXD can bring a **25% improvement in handling request variations**, ensuring high-quality and accurate responses across a diverse range of user inquiries. This approach can also lead to a **30% decrease in user frustration levels**, as evidenced by reduced rates of conversation abandonment.

Dynamic Response Model

The dynamic response capabilities of RXD can bring about a **35% enhancement in user satisfaction scores**, especially when providing directed responses to tech-savvy users. For tech-naïve users, the use of explanatory responses can result in a **40% increase in successful task completion rates**.

We hope that this whitepaper will inspire you to adopt RXD in your own voice/chat channel, and to create better conversations for your users.

REFERENCES

<https://www.gartner.com/en/newsroom/press-releases/2023-06-15-gartner-survey-reveals-only-8-percent-of-customers-used-a-chatbot-during-their-most-recent-customer-service-interaction>

<https://arxiv.org/abs/2212.03699>



ABOUT US

Improvement and Innovation, **The Process Re-engineering Team (Intelligent Operations)** is at the forefront of adopting cutting-edge methodologies and technologies. This forward-thinking approach not only supports the organization's immediate objectives but also positions it to anticipate and react to future challenges.

By fostering a culture of innovation, the team ensures that process optimization is not a one-time endeavor but a sustainable, ongoing process that evolves with the changing business landscape.

The Process Re-engineering Team is a pivotal element in the organization's strategy for operational excellence and market leadership. Through their diverse expertise, strategic alignment, and unwavering focus on innovation and customer satisfaction, they spearhead initiatives that transform processes, drive efficiencies, and deliver tangible outcomes. This team exemplifies the dynamic capabilities necessary for navigating the complexities of modern business environments, making them indispensable to achieving the organization's long-term objectives.



ABOUT THE AUTHOR

The Process Re-engineering Team (Intelligent Operations)

Shashikant Jha is a conversation designer at _VOIS Intelligent Operations (Process Re-Engineering). He works on voice designs for TOBi, a chatbot for Vodafone customers. He has created conversational journeys for various Vodafone markets, using his skills in conversation design, AI/ML, Deep Learning, Automation and DevOps. He is passionate about conversation design and explores how cognitive and cultural factors affect human-machine interactions. He is also fascinated by the emerging field of Generative AI and is exploring how Generative AI can improve the quality and diversity of the voice designs for TOBi, and how it can make the chatbot more empathetic, adaptive, and creative. He believes that GenAI can help create more humane and engaging conversations for the users, and enhance their satisfaction and loyalty.

Prasanna Kumar is a Management Consultant at _VOIS Intelligent Operations (Process Re-Engineering) in the Process Re-engineering Practice with outstanding experience in helping businesses identify and solve complex problems, improve their performance & customer experience, and achieve their strategic goals and innovation capabilities. He has worked with clients across geographies and industries such as Telecom, Banking and Financial Services, Manufacturing & Shipping to deliver innovative solutions in areas such as business strategy, business process re-engineering, risk management and digital transformation and possesses excellent skills in analysis, communication, project management and leadership. He has strong skills in conducting thorough analysis, communicating clearly, managing complex projects and leading teams. He is passionate about creating value for his clients and building long-term relationships based on trust and mutual respect.