Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 3 (2024)

An Improved After-Deployment of Quality Assurance Method Using Auto Sync and Chatbot Modules to Enhance Business Customer Relations

Benjamin Asubam Weyori[1], Samuel Tetteh[2]

[1]*Department of Computer and Electrical Engineering, University of Energy and Natural Resources, Sunyani, Ghana*

[2]*D Jarvis College of Computing and Digital Media, DePaul University, Chicago, USA*

Abstract

Technology has recently been applied in various spheres of life to ensure that daily activities and obstacles are easily tackled regardless of impediments such as time, location, etc. Responding to the growing needs and high demand of individuals and organizations to solve problems and make work more efficient, software developers are developing software applications to meet these demands. Existing software applications must be modified regularly to meet customers' constantly changing needs; this act of service seeks to improve and enhance business customer relations. This study selected, analyzed, and implemented five (5) Software quality assurance methods: Capability Maturity Model Integration, Total Quality Management, Six Sigma, Kaizen, and Perpetual Beta on a software product from the development stage to the deployment stage. Because the implementation of each technique in related research focused on respective companies and partially on their customers without ensuring effective quality assurance services to customers after software deployment, we adopted the Perpetual Beta method; we improved it by incorporating generic modules such as Chatbot and Auto Sync to extend Software Quality Assurance processes to customers after deployment. The study makes recommendations for future usage and further research.

Keywords: Software, Quality Assurance, Chatbot, Auto Sync, Software Quality Assurance Methods

1. Introduction

The idea of technological evolution began in the first century and gained popularity when the Industrial Revolution got underway in the eighteenth century. Businesses with multiple offices in different locations found it difficult for people to interact with one another in the middle of the 20th century when there was no internet. However, since the internet's invention, people have been able to engage in person via Skype, Facebook, live streaming, and other platforms, which have solved many issues for businesses (Nation, 2018). Technology can be defined as the use of abilities, know-how, procedures, and methods to achieve a specific objective. It integrates methods, assets, and information exchange to advance development for the benefit of humanity. In addition to assisting in problem-solving, saving time, and saving lives, it encourages interaction and information sharing among people to deal with everyday dynamic situations. New technologies are advancing society and improving the effectiveness and quality of life. Businesses and industries are expanding globally due to technology-reduced limitations on our daily lives in the twenty-first century (Digitaledenzindia, 2019; Slidingmotion, 2021).

Engineers must have certain abilities and beliefs that act as laid-down guidelines to create high-quality software products that meet business needs. A framework that specifies the responsibilities that must be fulfilled at each stage of the software development process is called the software development life cycle (SDLC). The life cycle in software engineering defines a method for enhancing software quality and the overall development process.

The SDLC aims to deliver high-quality software that meets or exceeds client expectations, adheres to timeframes, and is economically priced. It contains multiple phases, but the basic phases are five: planning, design, development, testing, and maintenance (Goran, 2019). The cycle flow shown in Figure 1.1 corresponds to the SDLC.

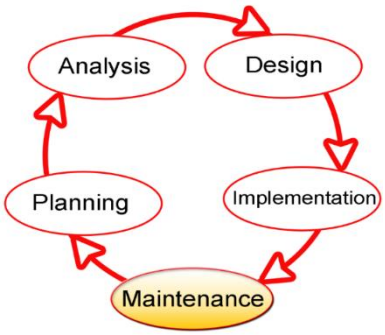

Fig. 1. 1: SDLC Phases – Source: Dzonatas 2008

The framework's guiding premise is that there should be some assurance certificate showing the project team has followed the appropriate best practices before moving from one phase to the next. Many team leads are designated within the software development processes to verify that each phase complies with the quality assurance practices used by the company and, as a result, ensure that all stakeholders approve the project phase sign-off form.

1.2 Case Study

Because of quality checks by professional software engineering businesses and other corporate in-house software engineers, every software product is always packaged as a high-percentage, defect-free product. This inspection is accomplished by ensuring that every Software Quality Assurance (SQA) technique is successfully used at every level of the development phases. In addition, some of the widely used SQA techniques are auditing, reviewing, code and design inspection, simulation, functional testing, walkthroughs, and stress testing. Before deploying the software application to its respective owners, the SQA team ensures that the to-be systems are of higher quality. However, they often forget that the environment in which the applications will be used is governed by other known and unexpected factors, such as economic stress, electric power fluctuations, dynamic user behavior, changes in organizational organograms, system runtime errors, etc.

Such challenges would attract customer complaints. Emails, phone calls, and frequently asked questions (FAQs) are the conventional customer feedback/communication channels accessible between the contract source (Institutions) and software service providers, as shown in Figure 1.2.

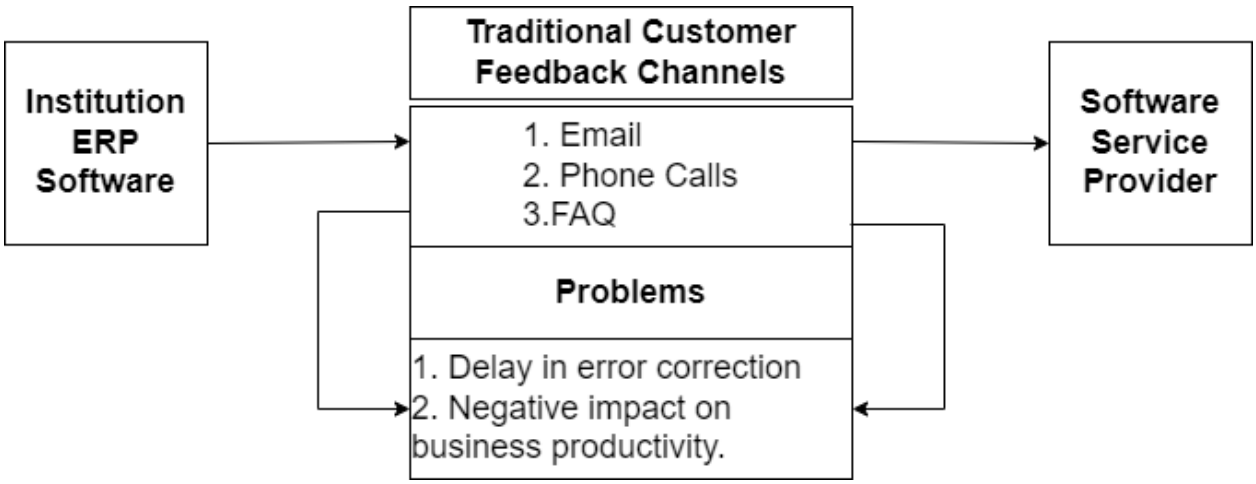

Fig. 1. 2: Communication channels between software providers and contract source

This study raises two significant issues with these communication channels as follows:

- **Delay in error correction:** Because multiple users access the software from various nodes, whenever an error is discovered at one of the company's remote offices, the user must fill out a complaint form and email it to the head of the information systems department (ISD) before system-issues-compilations can start. The head of the ISD contacts the vendor by phone or email after compiling all consumer complaints. The vendor has to start a fresh round of system upgrades to solve the compiled system flaws. Thus, the steps in the method delay the processes for fixing errors.
- **Negative effects on business productivity:** When suppliers are slow to respond to user-system problems due to several steps required, businesses are likely to see a drop in productivity as one of their main sources of revenue is affected.

The ideal situation would have been if the system had included an automated error detection module that could send alerts to software providers as soon as issues were discovered. By implementing this Module, vendors will be able to update systems more proactively, ensuring that organizations continue to function efficiently every day without experiencing a drop in productivity.

2.0 Related Works

Many software quality assurance techniques exist, such as Capability Maturity Model Integration (CMMI), Total Quality Management (TQM), Six Sigma, Kaizen, and Perpetual Beta (PB). This study adopts the Perpetual Beta and enhances it by adding Chatbot and Auto Sync modules.

2.1 Perpetual Beta

Perpetual Beta refers to creating a system or software application that remains in the beta stage for an extended or limitless period (Morris, 2006; O'Reilly, 2005; Wikipedia, 2022). Developers typically use PB when constantly delivering new features that might not have been completely tested. PB has also been applied to knowledge management, business, entrepreneurship, and software development (Levy, 2009). In these situations, "perpetual beta" often refers to a continuous improvement process where the final output would never be satisfactory. Based on the concepts of the everlasting beta concept, as well as technological improvements and an open-source approach, we adopt the unfinished notion of city-making (Fredericks et al., 2019)

2.2 Chatbots

Information systems (IS) and marketing research serve as the two key theoretical pillars for evaluating chatbots in the context of customer services. The study of virtual assistants is an important field of IS research. A virtual assistant is a software agent who works on a user's behalf to carry out activities or offer services. Businesses increasingly use virtual assistants to enhance client interaction and manage conversations about their services or products (Balakrishnan & Sankar, 2018; Przegalinska et al., 2019). Virtual assistants commonly assist users by providing information, acting as mentors or guides, pointing users to relevant web pages, and offering buying advice. Also, they give users the impression of chatting with a real person by keeping up a conversation while mimicking human behavior.

One of the popular technologies used to offer virtual help is chatbots. Natural language inputs can trigger responses from chatbots, which also try to mimic human conversational behavior. Chatbots can communicate with people using a variety of frameworks, such as simple text interfaces or speech recognition software (Cooper et al., 2020). These virtual assistants are receiving a lot of attention from businesses because they can enhance customer care services and reduce costs in customer service centers by automating and handling multiple clients. According to Trivedi (2019), chatbot success in a customer service environment often depends on the quality of the information, the technology, and the service. When users employ the service, their perception of its quality is influenced by Chabot's objective quality (Sugathan et al., 2018).

2.2.1 Satisfaction, Trust, and Commitment as the Three Drivers of Customer Loyalty

Customer satisfaction, trust, and commitment are the key factors influencing consumer loyalty (Pratminingsih et al., 2013; Vuuren, 2012). Customer happiness is key to customer retention (Bryant & Colledge, 2002). Indeed, customer commitment is simultaneously affected by customer satisfaction and trust (Vuuren, 2012). Moreover, customer loyalty is crucial for every organization because it directly correlates with profitability. A positive customer experience is essential to ensuring customer satisfaction and increasing the organization's total customer satisfaction (Bryant & Colledge, 2002). Therefore, it is believed that customer experience can anticipate consumer satisfaction.

2.2.2 System Quality, Service Quality, and Information Quality: Chatbots' Role in the Customer Experience

By implementing chatbots, which offer a new layer of quick service and important information to the internet anytime from any location, businesses want to provide their customers with an outstanding customer experience (Bryant & Colledge, 2002). Studies show that adding a chatbot to a website enhances user experience (Hoyer et al., 2020; Trivedi, 2019). According to Trivedi (2019), citing DeLone and McLean (1992), chatbots need high-quality systems, services, and information to produce a positive customer experience that results in customer satisfaction. So, Trivedi investigated how users utilize information technology systems, such as chatbots, to do jobs successfully and effectively by concentrating on the crucial five factors of system quality. A chatbot must be available 24/7 and deliver accurate information at the right time and place to meet consumer expectations.

The system's quality determines the technological success of the chatbot. Users anticipate that all internet connections will result in the flawless operation of information systems. The same is true of a chatbot's adaptability; therefore, chatbot creators must maintain their systems up-to-date and ready to react to significant environmental changes. The same Trivedi study shows that giving a quick answer enhances the customer experience while delaying has a detrimental impact. However, Gnewuch et al. (2018) claimed that being quicker is not necessarily better. Yet, the study showed that chatbots with quick responses raise customer satisfaction.

This outcome results from the perception that dynamically delayed chatbot messages were more human-like and natural than messages sent right away (Kvale et al., 2020). The goal of chatbots is to deliver fantastic user experiences while assisting users in real-time with comments and queries on a website or app (Akhtar et al., 2019). Chatbots are supposed to make life easier for customers and enterprises due to their accessibility, availability, and prompt responses (Ashfaq et al., 2021). That is consistent with studies that found a website's usability supports an even better user experience (Ba & Johansson, 2008; van der Goot & Pilgrim, 2020).

2.2.3 Privacy Paradox and Personalization

According to Brandtzaeg and Folstad's (2017) study, clients utilize chatbots most frequently because they are quick, convenient, and simple. That way, productivity seems to be a major argument in favor of adopting a chatbot. Additionally, according to Brandtzaeg and Folstad's investigation results, utilizing a chatbot was justifiable because of its assistance and educational resources. Researchers discovered that consumer acceptance of chatbots is less widespread than expected, despite projections that they will play a big role in customer support both now and in the future (Nordheim et al., 2019).

The lack of chatbot adoption may make people think chatbot talks are too informal, impersonal, and unnatural. Additionally, the trial by (Nordheim et al., 2019) demonstrated that most participants would employ the creative chatbot. However, only one-third of their participants used the chatbot, primarily because of worries that doing so might cost people their jobs. Additionally, chatbot resistance is influenced by privacy concerns, which lowers usage frequency (Liebrecht et al., 2021).

Chatbots may bring a new level of support by providing personalized care that addresses customers' needs whenever and wherever they arise (Chung et al., 2020). This addition is because multiple studies have shown that customers prefer personalized dialogues with chatbots, which is why chatbot personalization is becoming an increasingly important issue (Akhtar et al., 2019; Elsholz et al., 2019; Youn et al., 2021). A chatbot's language style must be altered to make it more distinctive and human-like (Go & Sundar, 2019; Hwang et al., 2019;

Liebrecht et al., 2021; Moriuchi, 2021). Chatbots are generally human-like, and their tailored approach greatly impacts the customer experience, which can result in client loyalty (Carvajal, 2011; Sheehan et al., 2020).

Considering the nature of chatbots in customer service, the study adapted the use of the chatbot framework for software developers to get closer to their customers by providing them with first-hand information once new updates are released as well as a means through which customers can interact with the software developing company in real-time through a robot (chatbot). The Module also allows the deployed software to communicate with the software developing company in real-time through the Auto-Sync Application Programming Interface (API).

3. Methodology

The study engineered the implementation processes using conceptual frameworks and flowcharts for Auto-Sync and Chatbots. API documentation on how to implement the Auto-Sync module was also engineered.

The architectural framework of the Auto-Sync module, as presented in Figure 3.2, is divided into three levels:

i. Local Storage Level (LSL): This is where the system error messages and system user comments are first stored.

ii. API Level: This is where the stored records in the LSL are prepared to be sent over a secured network protocol to the third level.

iii. Service Provider Data Center Level (SPDCL): This is where all records submitted from LSL through the API Level are received for further consideration.

The Auto-Sync module functions like a system plugin that software developers adapt to and implement as an added system feature for them to be able to ensure after-deployment software quality assurance practices to their customers. Implementing the LSL, the software developer has to create two tables in addition to the primary tables of the software; these tables, according to Figure 3.2, are the auto_sync_tbl and temp_auto_sync_tbl.

The Auto-Sync plugin is configured to actively listen to system user comments and system runtime error messages and save them into the auto_sync_tbl with or without internet connectivity. In situations with no internet connectivity, the gathered records are saved into the auto_sync_tbl and temp_auto_sync_tbl tables concurrently. The Auto-Sync API is only activated when internet connectivity exists (internet = true). Once the records stored in the temp_auto_sync_tbl table are posted to the service provider through the Auto-Sync API, all such records are deleted from the temp_auto_sync_tbl table. Still, on the side of the auto_sync_tbl table, all records are maintained intact for historical data purposes. When the service provider's endpoint receives the posted requests from their respective customers, many system notifications are triggered in beep sounds, blink indicators, SMS alerts, and emails. These notifications repeat until the posted requests have respectively been resolved.

Figure 3.4 is the process flowchart representation of Figure 3.2. It can be observed that there exist two loops in Figure 3.4; the first one has to do with the listening port that constantly checks for any system error and user comments so that they can be processed for submission as soon as possible. The second loop is configured on the temp_auto_sync_tbl table. The Module constantly checks the emptiness of the table to make sure that there are no pending tasks that have been left unattended. The loops concept in this Module makes it efficient, dependable, and reliable to adopt because it can project how proactive the service providers are.

A. Auto-Sync Conceptual Framework

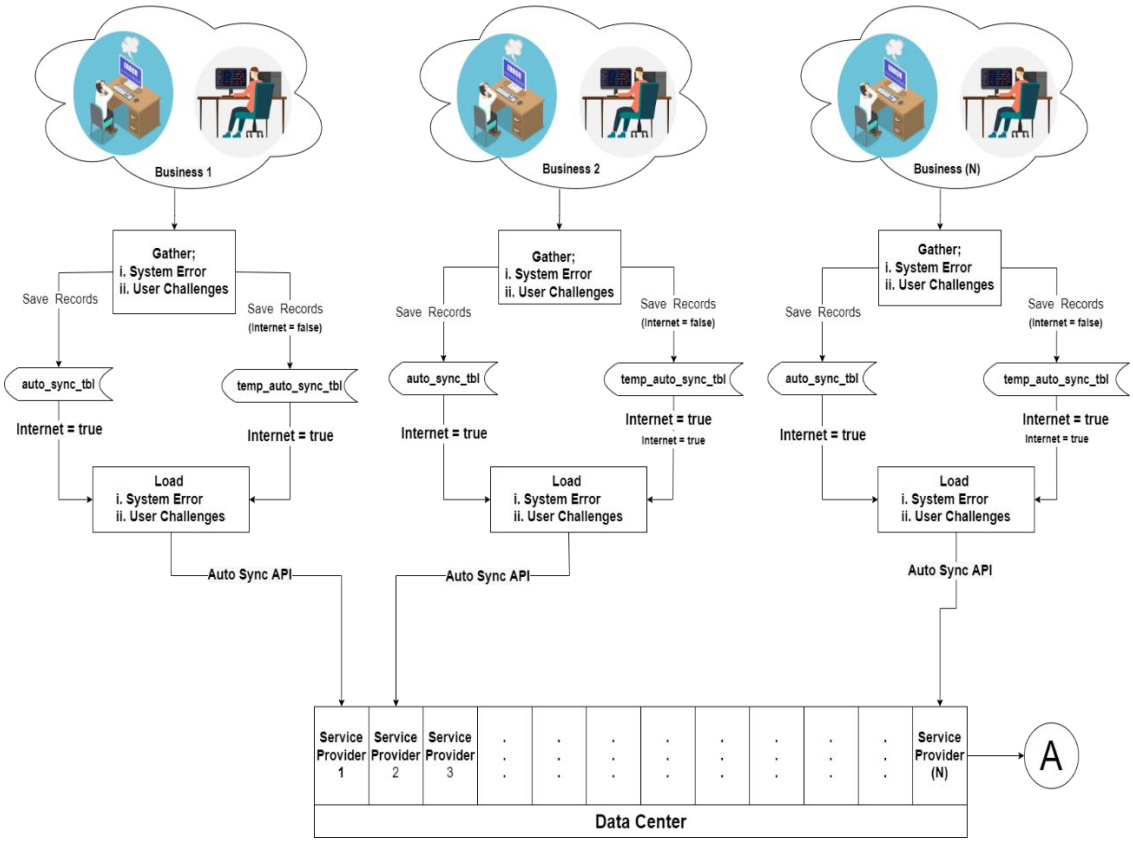

Fig. 3.1: Conceptual Framework of the Auto-Sync Module.

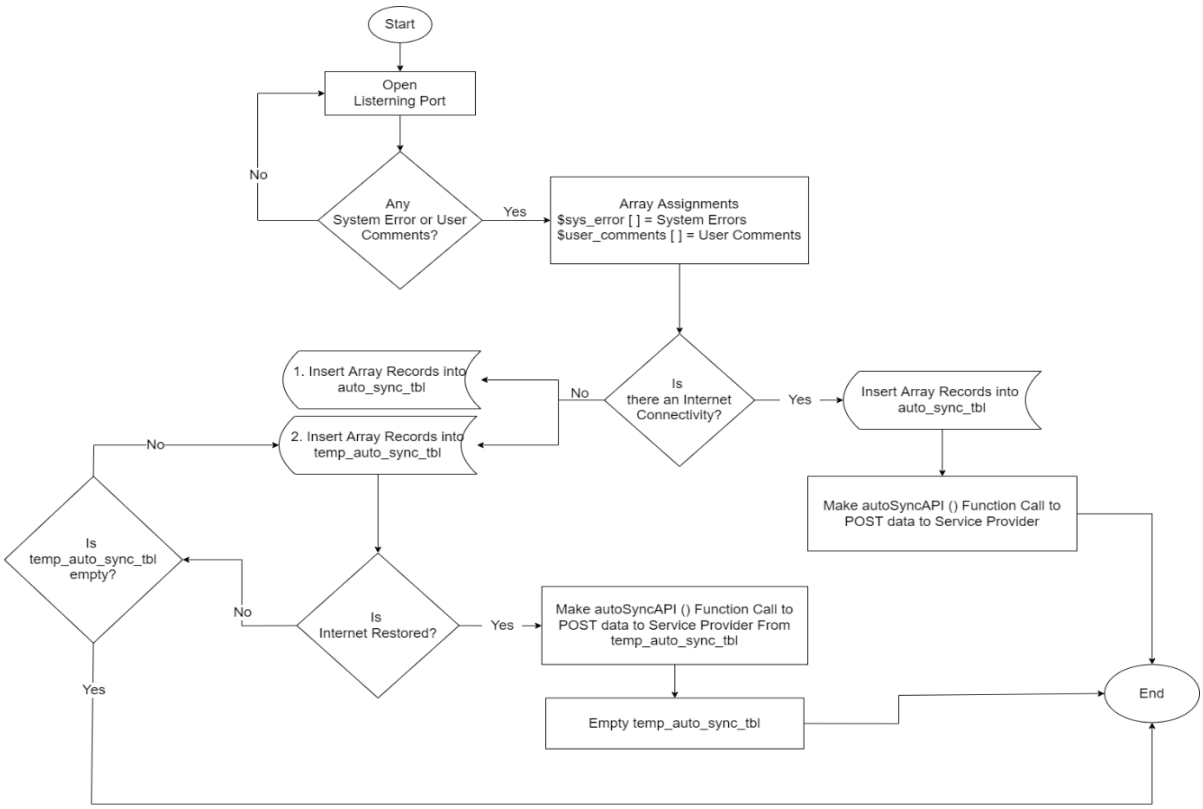

Fig. 3.2: Auto-Sync Module Implementation Flowchart Diagram

1. AutoSyncAPI Algorithm

i. Start

ii. Open listening port

iii. Are there any system error messages or user comments?

iv. If Yes then,

v. Assign them to an array

a. $sys_error_msg[ ] = System_error_messages;

b. $user_comments [] = User_comments;

vi. Else

vii. Go to step ii.

viii. Is the server connected to the internet?

ix. If Yes then,

x. Insert array records into auto_sync_tbl and

xi. Make an autoSyncAPI function call to POST records to the service provider.

xii. Else

xiii. Insert array records into both the auto_sync_tbl table and temp_auto_sync_tbl table.

xiv. Is the internet restored?

xv. If Yes then,

xvi. Make an autoSyncAPI function call to POST records from the temp_auto_sync_tbl table to the service provider and

xvii. Empty temp_auto_sync_tbl table

xviii. End.

2. AutoSyncAPI Documentation

This API documentation describes how Figure 3.2 POST records to the service provider. It is open for developers' modifications to suit their respective projects.

Record Insertion End-Point: Any developer-defined path.

Request Type: POST

Content-Type: application/JSON

Command Line: Client URL (CURL)

Table 3. 1: AutoSyncAPI Documentation Parameters

Parameter	**Description**
bus_name	The name of the registered business
bus_branch_name	The branch name of the business
bus_branch_location	The location of the business
bus_secret_key	This secret key is an encrypted token generated by the service provider and assigned to every customer.
sys_error_msg	The system identified an error message
sys_user_comment	The comments submitted by the system user
trans_time	The time that the item is recorded
trans_date	The date that the item is recorded
opt	The opt should have defaulted to **onlInvInsert**

3.1.3 CURL POST API Sample Code

function **autoSyncAPI (**$company, $busSecretKey, $sysErrorMsg, $sysUserComment, $time, $date**) {**

$param = array (

‘bus_name’ => $company,

‘bus_secret_key’ => $busSecretKey,

‘sys_error_msg’ => $sysErrorMsg,

‘sys_user_comment’ => $sysUserComment,

‘trans_time’ => $time,

‘trans_date’ => $date,

'opt' => **'onlInvInsert'**

);

foreach ($param as $key=>$value) {

```
$param[$key] = $value;
}
$abena_curl = curl_init ();
curl_setopt ($abena_curl, CURLOPT_URL,"developers preferred url");
curl_setopt ($abena_curl, CURLOPT_POST, 1);
curl_setopt ($abena_curl, CURLOPT_POSTFIELDS, $param);
curl_setopt ($abena_curl, CURLOPT_RETURNTRANSFER, TRUE);
curl_setopt ($abena_curl, CURLOPT_SSL_VERIFYPEER, FALSE);
$result = curl_exec($abena_curl);
 if (curl_errno ($abena_curl)) {
return "Error: ". curl_error ($abena _curl);
} else {
return $result;
  }
curl_close ($abena_curl);
}
```

B. Chatbot Conceptual Framework

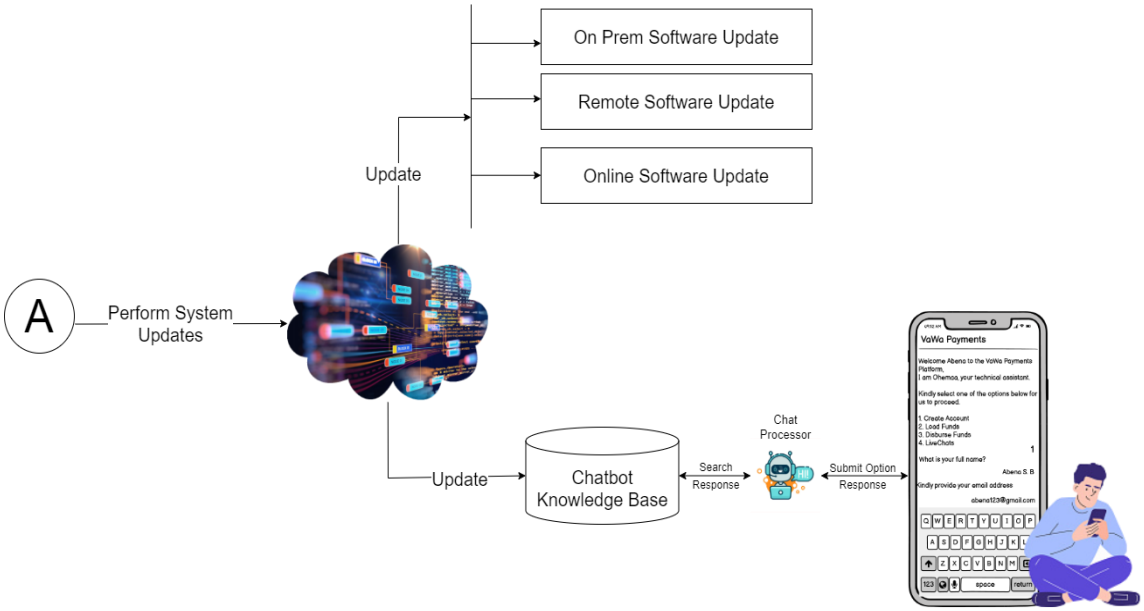

Fig 3.3: Chatbot Module Conceptual Framework

Figure 3.3 is the continuation of Figure 3.1; that is, as soon as the service providers resolve the posted system error messages and user comments, the software developers initiate the system updates processes. The update procedures are separated into two main areas. The first major section deals with the customer's software application patches, which are carried out in three different ways:

i. On-Prem Software Update: This is where the service provider moves to the customer's premises to effect the necessary updates.

ii. Remote Software Update: The service provider updates customers' software applications using a Virtual Private Network (VPN) to connect to the customer's server through a Remote Desktop Connection, TeamViewer, Microsoft Teams, and any other application that is capable of performing such activities.

iii. Online Software Update: The service provider directly updates the customers' software application by accessing the online web servers when the customer's software is deployed on an online hosting server with a dedicated domain name.

The chatbot's knowledge base is updated in the second major portion. An online knowledge base is a collection of documentation that often includes troubleshooting instructions, how-to manuals, and how-to guides. Its goal is to make it simple for people to resolve their problems on their own, without the aid of others (Ben, 2021; Mathew, 2022).

Customers can find speedier solutions to their problems thanks to the chatbot's often updated knowledge base. Thus, it is the first place they should turn to if they run into any issues or want to learn more about the installed software.

Service providers must regularly update the chatbot's knowledge base by storing a series of questions and answers concerning the software and any other updates relevant to other versions of the software. After updating the knowledge base, the customer accesses the chatbot from the service provider's website, WhatsApp number, or Telegram ID. When a customer asks a question of the chatbot, the robot accepts the request, compares the keywords in the question to the specified contents, searches the chatbot's knowledge base, and then replies to the consumer via the robot. Because no human is involved in these exchanges, the Module is incredibly reliable and effective because it can service customers continuously without fatigue.

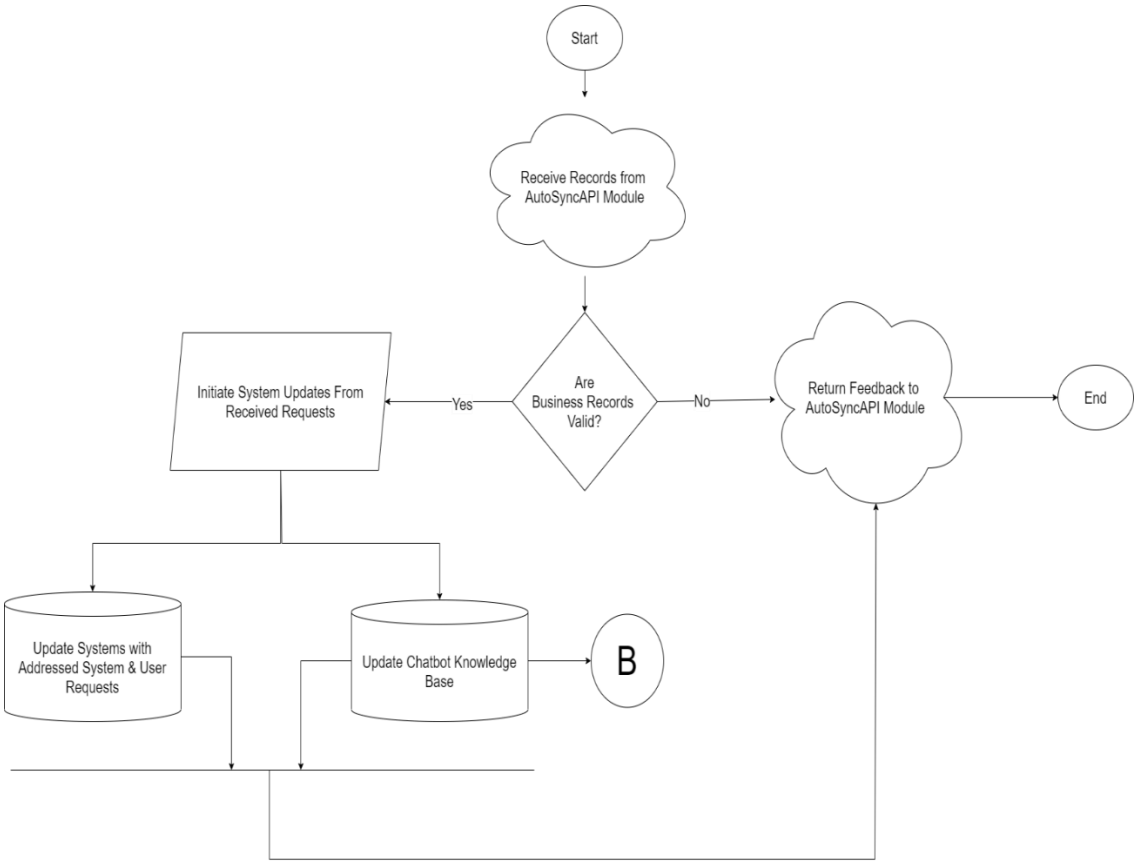

Fig. 3.4: Service Provider's API Endpoint Implementation Flowchart

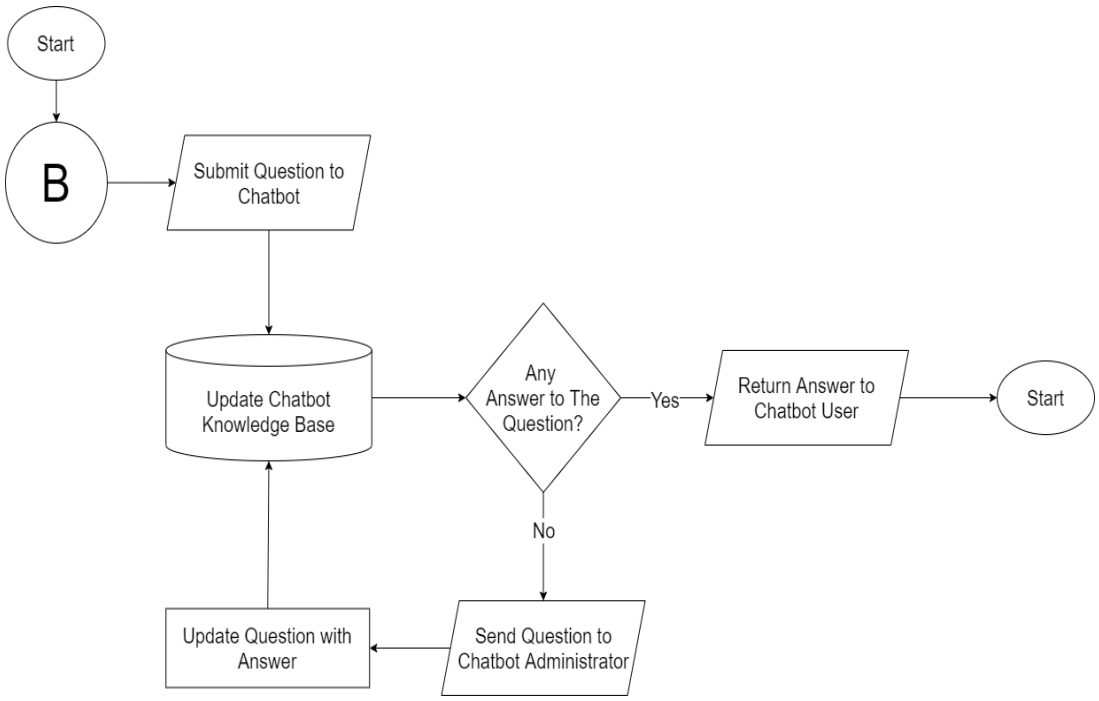

Fig 3.5: Chatbot Module Implementation Flowchart Diagram

In Figure 3.4, the Module verifies the source of the request to make sure it is truly coming from a registered customer before working on the system and user requests that are received through the Auto Sync API and making the necessary updates to both the general software application and the Chatbot knowledge base. The Module becomes immediately available to service its present and potential customers after updating the knowledge base.

When the chatbot's knowledge base is continuously updated, it is always prepared to assist its users. Figure 3.5 shows how users can connect with the chatbot directly. The chatbot then checks its knowledge base to see if the user's request can be fulfilled. The chatbot responds to the user with the response if there is feedback on their request. When the chatbot's knowledge base does not include the answer to a user's query, the chatbot sends the query to the chatbot administrator. The administrator responds to the query and updates the chatbot's knowledge base.

4. Results and Analysis

The study contrasts and explores the enhanced Perpetual Beta employing Chatbot and Auto Sync modules with the current quality assurance approaches such as Capability Maturity Model Integration (CMMI), Kaizen, Six Sigma, and Total Quality Management (TQM).

The Chatbot and Auto Sync Module ensure double packet encryptions. The first one concerns the encryption of the data to be transferred to the service provider. The second encryption involves data transfer, packets from the customers' software that uses the Hypertext Transfer Protocol Secure (HTTPS). The Hypertext Transfer Protocol (HTTP) is extended with the HTTPS protocol, which uses Transport Layer Security (TLS), formerly known as the Secure Sockets Layer, to encrypt communication (SSL). These security features make the Module secure for businesses compared to the CMMI.

Table 4. 1: CMMI Compared to the Chatbot and Auto Sync Module

CMMI	**Chatbot and Auto Sync Module**
CMMI may not meet security standards for some businesses.	The Module executes on encrypted and secured protocols.

CMMI is not universally relevant to all organizations.	The Module is relevant to all organizations using software applications to run their business.
CMMI may not operate well with other business frameworks and procedures.	The Module is easily adaptable to other business frameworks.
Putting it into practice could take a lot of time and work.	The Module is implemented during the software installation period.
It demands a significant change in the company culture and attitude.	The Module does not affect organizational culture and attitude.

Due to the simple end-to-end communication flow between businesses and their service providers, the Chatbot and Auto Sync Module becomes relevant when considering the companies that outsource their core business software applications. This communication flow ensures customer service satisfaction as opposed to the CMMI. In addition, the Module performs the function of an error detection plugin built into the software application deployed during the installation because the Module is easily adaptable and compatible with business frameworks. Unlike the CMMI, the Module does not call for a change in the business process flow.

Another disadvantage of the CMMI is that it requires a lot of time and work to see benefits. In contrast, the Chatbot and Auto Sync Module generates real-time results since its actions are automatically done because of the Module's looping nature. Kaizen may occasionally fall short of expectations or not function as planned in specific scenarios. Some people might also be averse to change or unwilling to participate in Kaizen projects. However, the departmental heads of each company work on the system user comment submitted to the service provider using the Auto Sync API. Hence, participation from all personnel is not necessary.

Table 4. 2: Kaizen Compared to Chatbot and Auto Sync Module

Kaizen	**Chatbot and Auto Sync Module**
Takes time and effort.	The Module does not take time to implement.
Staff resistance to change.	The Module does not require all staff participation.
Costly in some cases.	One-time deployment cost.
Management involvement.	The Module requires one-time management involvement.
An additional burden for employees.	Minimum employee involvement.

It becomes difficult to convince employees to use the Kaizen process if management is not on board. Consequently, the Kaizen process may fail if management does not provide enough resources, such as time and training. Management engagement in the Chatbot and Auto Sync Module is far less than it is in the Kaizen model since it only occurs when the choice is made to purchase new core business software.

In Six Sigma, increasing quality is prioritized over cutting expenses. Improving quality and eliminating flaws could be expensive because of increased administrative costs and more machinery purchases. On the other hand, the Chatbot - Auto Sync Module is relatively affordable because it is part of the program's overall cost.

To achieve Six Sigma, it is typically necessary to enhance testing processes, increase the number of quality checks, and decrease tolerances. Resources might be swiftly exhausted with this technique. Additionally, spending money on resources is essential for the statistical analysis needed for Six Sigma. On the other hand, the Chatbot - Auto Sync Module does not need any physical resources because it runs in the system's background.

Table 4.3: Six Sigma Compared to Chatbot and Auto Sync Module

Six Sigma	**Chatbot and Auto Sync Module**

Recurring implementation cost	One-time implementation cost
It consumes a lot of physical resources	It requires no physical resources
It focuses on a strict and rigid process	It focuses on flexibility and speed
It requires skillful and motivated staff to manage it.	It does not require any skillful staff to manage it.

The innovative method strongly emphasizes repetition, unorthodox solutions, and production deviations, which are flagrantly in conflict with the Six Sigma principles. Following strict procedures is heavily emphasized in Six Sigma, which conflicts with contemporary tendencies that prize creativity and innovation. The Chatbot and Auto Sync Module focuses more on communication flexibility and speed between deployed software applications and service providers.

Another negative aspect of Six Sigma is that it is no longer regarded as a methodology in and of itself. Firms are converting to alternative methodologies or strategies that demand the outsourcing of work that poses severe concerns with responsibility. This methodology is merely a continuation of the improvement techniques used by Toyota. Given this, it is clear that Six Sigma also requires a sizable number of trained personnel who must be quickly motivated and undergo extensive training over a long period. The Chatbot and Auto Sync Module does not have this flaw because using it doesn't require specialized knowledge.

A company's success is based on how effectively its operations work. Comparatively speaking, the Chatbot and Auto Sync Module's implementation cost is low because it is a one-time expense, and the price of the program covers it. However, striving to improve the effectiveness of the operation of a company always entails costs, some of which may be above average.

A corporation can utilize the TQM tools to improve its operations, but this move does not guarantee that customers would value them higher. The deployment of TQM may dissatisfy many customers due to service delays. Customers are satisfied with the Chatbot section of the Chatbot and Auto Sync Module because they can obtain first-hand information from the chatbot even before contacting the service provider.

Table 4.4: Total Quality Management Compared to Chatbot and Auto Sync Module

Total Quality Management	**Chatbot and Auto Sync Module**
Increased costs	Reduced costs
Customer dissatisfaction	Customer satisfaction
It is difficult and time-consuming	It is flexible and works at a speed
It necessitates a company-wide commitment to quality improvement, which is challenging.	It requires no special commitments.

Implementing TQM is a challenging endeavor because it can be time-consuming and challenging. Both human and financial resources in large quantities are needed. It is challenging because putting TQM into practice forces an organization to switch from top-down to bottom-up management of its operations. Reviewing, assessing, and adjusting operational processes and procedures necessitate a large resource commitment. At the same time, the Chatbot - Auto Sync Module instead emphasizes speed and the adaptability of deployed software applications' communication with service providers.

The requirement for and difficulty in achieving company-wide commitment to quality improvement is one of the key drawbacks of TQM. All management levels must support the program for it to be genuinely effective. A lack of resources or effort can undermine a TQM program's execution, which would be bad for the firm. The organization's management would not make explicit pledges, as with the Chatbot and Auto Sync Module.

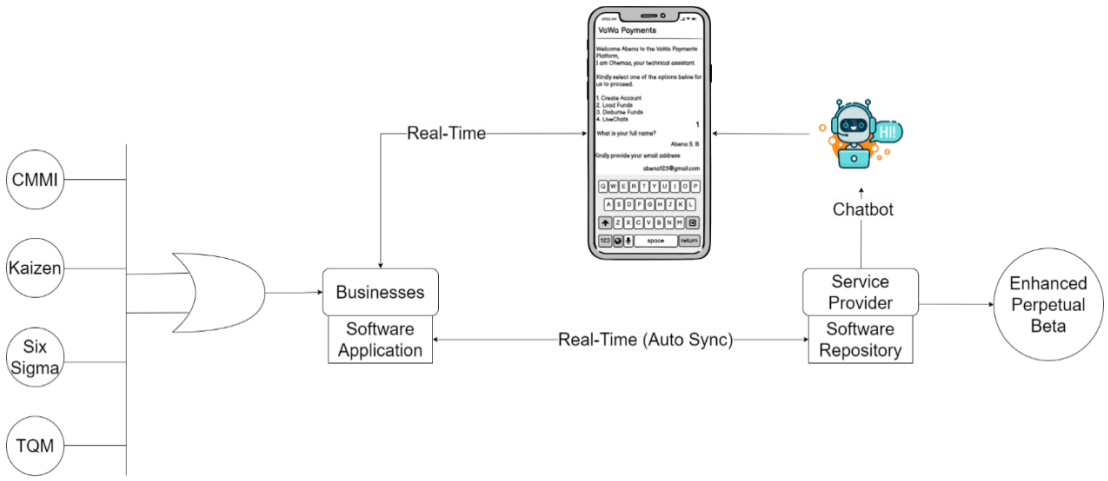

Fig 4.1: Enhanced perpetual beta model using Chatbot and Auto Sync module

4. Conclusion

The organizational structure and chosen quality assurance methodology used by a company are unaffected by the summary shown in Figure 4.1. Companies can apply CMMI, Kaizen, Six Sigma, TQM, or Perpetual Beta. The study's main goals are improving communication response times and offering flexible, dependable channels that customers or enterprises can use when they require technical support.

The Perpetual Beta, often known as the agile technique, is improved by the Chatbot and Auto Sync Module implementation. Instead of creating software that has every component as intended by the designer, the main goal of every software project is to satisfy the customer's needs with a condensed but still useful version. The agile process is similar to the adage "perpetual beta" because the software improves with each iteration until all desirable features are available.

The principles of the perpetual beta are used in the creation of the software and other updates that follow. Still, in the enhanced perpetual beta described in the study, all the requirements for the upcoming system iteration are sent to the software development company in real-time via the Auto Sync API. Every change made to the program is promptly updated on the Chatbot knowledge base, which is easily accessible to other companies and clients.

5.1 Contributions of the Study

The reviewed literature indicated that even though software development companies and their customers are practicing several quality assurance methods. No plugin linked software development companies directly to their customers. At the end of the research, the study contributed to the knowledge by the following:

i. Implementing a module known as the Chatbot and Auto Sync plugin based on the existing perpetual beta model.

ii. This plugin enables software development companies to ensure quality assurance service to their customers even after deployment.

iii. The plugin directly connects business institutions to their service providers, making information sharing easier.

The study's goal, which was to create an improved perpetual beta model to support post-deployment software quality assurance techniques, utilizing chatbot and auto-sync modules to improve business customer connections, was effectively accomplished. Software developers can now offer their customers additional software quality assurance services even after they have sent their goods to market because of this Module's implementation, which produces an automated error detection plugin.

Additionally, the Chatbot component of the Module can be installed on a company's website and messaging apps like WhatsApp and Telegram. As a result, software developers can provide real-time information to their clients.

Since the Module is a plugin included in the primary deployed program rather than standing alone, it does not require staff training or incentive, saving many business investments.

References

[1] Akhtar, S., Irfan, M., Sarwar, A., Asma, & Rashid, Q. U. A. (2019). Factors influencing individuals' intention to adopt mobile banking in China and Pakistan: The moderating role of cultural values. *Journal of Public Affairs*, *19*(1). https://doi.org/10.1002/PA.1884

[2] Ashfaq, M., Cavazos, T., Reboita, M. S., Torres-Alavez, J. A., Im, E. S., Olusegun, C. F., Alves, L., Key, K., Adeniyi, M. O., Tall, M., Sylla, M. B., Mehmood, S., Zafar, Q., Das, S., Diallo, I., Coppola, E., & Giorgi, F. (2021). Robust late twenty-first century shift in the regional monsoons in RegCM-CORDEX simulations. *Climate Dynamics*, *57*(5–6), 1463–1488. https://doi.org/10.1007/s00382-020-05306-2

[3] Ba, S., & Johansson, W. C. (2008). An exploratory study of the impact of e-service process on online customer satisfaction. *Production and Operations Management*, *17*(1), 107–119. https://doi.org/10.3401/POMS.1070.0006

[4] Balakrishnan, K., & Sankar, R. (2018). Empowering Chatbots with business intelligence by big data integration. *International Journal of Advanced Research in Computer Science*, *9*(1). https://doi.org/10.26483/ijarcs.v9i1.5398

[5] Ben, L. (2021). *What is a Knowledge Base? Definition, Examples and Guide*. https://www.techtarget.com/whatis/definition/knowledge-base

[6] Bryant, A., & Colledge, B. (2002). Trust in electronic commerce business relationships. *Journal of Electronic Commerce Research*. https://www.academia.edu/18135539/Trust_in_electronic_commerce_business_relationships

[7] Cooper, B. J., Vijayaraghavan, V., & Rian Leevinson, R. L. (2020). Algorithm inspection for Chatbot performance evaluation. *Procedia Computer Science*, *171*, 2267–2274. https://doi.org/10.1016/J.PROCS.2020.04.245

[8] Digitaledenzindia. (2019, October 18). Why technology is important in our daily life? https://digitaledenz.com/why-technology-is-important-in-our-daily-life/

[9] Fredericks, J., Caldwell, G. A., Foth, M., & Tomitsch, M. (2019). The city as Perpetual Beta: fostering systemic urban acupuncture. *The Hackable City*, 67–92. https://doi.org/10.1007/978-981-13-2694-3_4

[10] Gnewuch, U., Morana, S., Adam, M. T. P., & Maedche, A. (2018). Faster is not always better: Understanding the effect of dynamic response delays in human-chatbot interaction. *Proceedings of the 26th European Conference in Information Systems (ECIS)*, Portsmouth, United Kingdom, June 23-28.

[11] Goran, J. (2019, May 15). *What is SDLC? Phases of Software Development & Models*. PhoenixNAP Global IT Services. https://phoenixnap.com/blog/software-development-life-cycle

[12] Hoyer, W. D., Kroschke, M., Schmitt, B., Kraume, K., & Shankar, V. (2020). Transforming the customer experience through new technologies. *Journal of Interactive Marketing*, *51*, 57–71. https://doi.org/10.1016/J.INTMAR.2020.04.001

[13] Kvale, K., Prowe, A. E. F., Chien, C. T., Landolfi, A., & Oschlies, A. (2020). The global biological microplastic particle sink. *Scientific Reports 2020 10:1*, *10*(1), 1–12. https://doi.org/10.1038/s41598-020-72898-4

[14] Levy, M. (2009). WEB 2.0 implications on knowledge management. *Journal of Knowledge Management*, *13*(1), 120–134. https://doi.org/10.1108/13673270910931215/FULL/XML

[15] Patterson, M. (2022, June 7). The ultimate guide to using a knowledge base for self-service support. *Help Scout*. https://www.helpscout.com/playlists/knowledge-base/

[16] Morris, J. H. (2006). Software product management and the endless beta. Carnegie Mellon University. http://jimmorris.blogspot.com/2006_08_01_jimmorris_archive.html

[17] O'Reilly, T. (2005). *What Is Web 2.0*. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html?page=4

[18] Pratminingsih, S. A., Lipuringtyas, C., & Rimenta, T. (2013). Factors influencing customer loyalty toward online shopping. *International Journal of Trade, Economics and Finance*, 104–110.

https://doi.org/10.7763/IJTEF.2013.V4.268

[19] Przegalinska, A., Ciechanowski, L., Stroz, A., Gloor, P., & Mazurek, G. (2019). In bot we trust: A new methodology of chatbot performance measures. *Business Horizons*, *62*(6), 785–797. https://doi.org/10.1016/J.BUSHOR.2019.08.005

[20] Swapnil, N. (2021). *Why* technology is important? *Slidingmotion* https://slidingmotion.com/why-technology-is-important/

[21] Sugathan, P., Rossmann, A., & Ranjan, K. R. (2018). Toward a conceptualization of perceived complaint handling quality in social media and traditional service channels. *European Journal of Marketing*, *52*(5–6), 973–1006. https://doi.org/10.1108/EJM-04-2016-0228

[22] Trivedi, J. (2019). Examining the customer experience of using banking chatbots and its impact on brand love: The Moderating Role of Perceived Risk. *Journal of Internet Commerce*, *18*(1), 91–111. https://doi.org/10.1080/15332861.2019.1567188

[23] van der Goot, M. J., Pilgrim, T. (2020). Exploring age differences in motivations for and acceptance of chatbot communication in a customer service context. In: , *et al.* Chatbot Research and Design. *Conversations 2019. Lecture Notes in Computer Science*, 11970. Springer, Cham. https://doi.org/10.1007/978-3-030-39540-7_12

[24] Vuuren, T. Van. (2012). Customer satisfaction, trust and commitment as predictors of customer loyalty within an optometric practice environment. *Southern African Business Review*, *16*(3), 81–96.

[25] Wikipedia. (2022, March 21). *Perpetual beta - Wikipedia*. https://en.wikipedia.org/wiki/Perpetual_beta