CHATBOT DEPLOYMENT WITH IBM CLOUD WATSON ASSISTANT

AI&ADS:

Documentation

# AI Chatbot Project Implementation: A Step-by-Step Guide for Managers

## Table of contents

* [Stage 1: Identifying Customer Needs](https://sentione.com/blog/chatbot-project-implementation-guide-for-managers#Stage-1:-Identifying-Customer-Needs)
* [Stage 2: Process Analysis](https://sentione.com/blog/chatbot-project-implementation-guide-for-managers#Stage-2:-Process-Analysis)
* [Stage 3: Chatbot Implementation on the Platform and Tests](https://sentione.com/blog/chatbot-project-implementation-guide-for-managers#Stage-3:-Chatbot-Implementation-on-the-Platform-and-Tests)
* [Stage 4: Pilot Production Deployment](https://sentione.com/blog/chatbot-project-implementation-guide-for-managers#Stage-4:-Pilot-Production-Deployment)
* [Stage 5: Chatbot Maintenance and Development](https://sentione.com/blog/chatbot-project-implementation-guide-for-managers#Stage-5:-Chatbot-Maintenance-and-Development)
* [Work with SentiOne – Your Bot-Building Partner](https://sentione.com/blog/chatbot-project-implementation-guide-for-managers#Work-with-SentiOne-%E2%80%93-Your-Bot-Building-Partner)
* [Article Summary](https://sentione.com/blog/chatbot-project-implementation-guide-for-managers#Article-Summary)

## Identifying Customer Needs

[Your future chatbot has to address a certain use case and bring tangible results](https://sentione.com/blog/maximising-roi-with-ai-chatbots-voice-bots) – it should support your customers in dealing with everyday tasks such as booking an appointment, asking for order status, and more. This stage is essential as it shapes the whole [**chatbot design process**](https://sentione.com/blog/how-to-design-effective-chatbot). That’s why everyone involved in direct work with customers should be included.

### Stakeholders

Identifying customer needs should happen with the participation of:

Process;

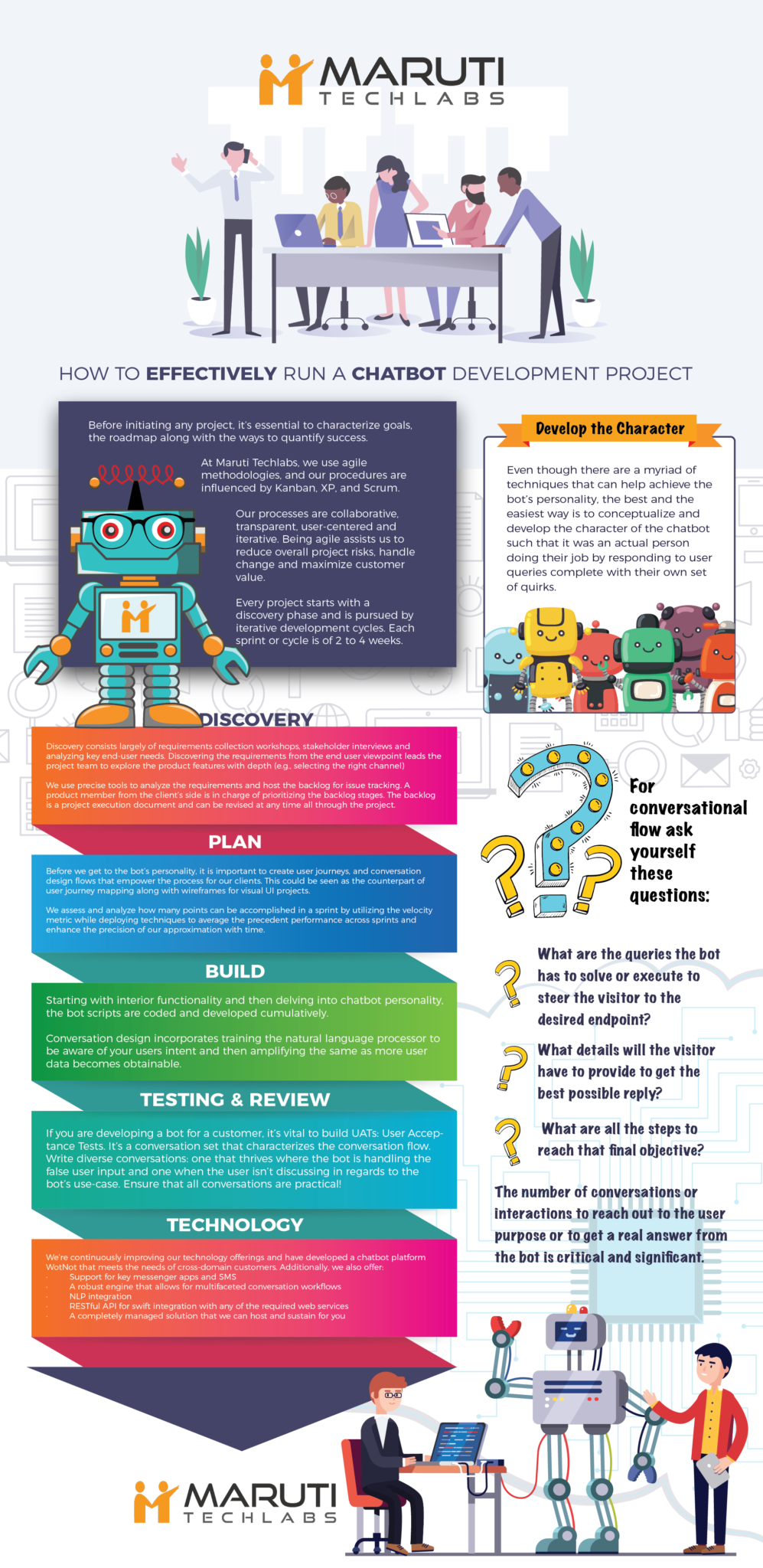
Now, it’s time to dig deeper and get the full picture of how your future bot should work and behave. Above all, your company (together with the bot provider) must assess which part of the e.g., customer service process should be automated. This requires having a closer look .

## Article summary

This article presents a step-by-step guide for managers on implementing AI chatbot projects. The process includes identifying customer needs, analysing processes, deploying and testing the chatbot, piloting production, and maintaining the bot.

Abstract

 Internet of Things (IoT) is fast becoming a disruptive technology business opportunity, with standards emerging primarily for wireless communication between sensors, actuators and gadgets in day-to-day human life, all in general being referred to as “Things”.



IoT Cloud Platform

IoT Cloud based Platforms is an important enabling

technology in many IoT systems today. They deal with various

fragmented technologies in embedded devices from access

protocols (eg. Message Queuing Telemetry Transport (MQTT),

HTTP etc.), wireless protocols (Zigbee, Bluetooth Low Energy

(BLE)) to various services, SDKs and integrations. There has

been positive reports on the established advantages of Cloud

based IoT platforms [25]. Our system design stresses on

accessing and controlling the embedded devices in question,

such as Smart Car, Light, Thermostat etc., through the API

Management/Gateway of the IoT Cloud Platform, regardless of

the standards and protocols of the individual embedded devices.

On the other hand, complex components in Networking and

Computing infrastructure for the Cloud platform have not been

addressed in this paper. Popular IoT platforms today include

Microsoft Azure IoT, IBM IoT, APIGEE Io

Chatbot System

1) Chatbot Channels and Platforms

Chatbot Channels are applications which run Chatbots

on supported Mobile devices (eg. Smartphones, Tablets) or

Terminals (eg. Desktop Applications).They are typically built

on top of the existing instant messaging platforms. Popular

Chatbot channels include Facebook Messenger, Slack,

Telegram, Kik, Skype, Line and Twilio SMS. These channels

are essentially the Chatbot applications in which a user interacts

with the bot. The area of Chatbot development is still in its

infancy and there can be many different architectural

approaches in implementing Chatbots.

In some approaches, the channels are interfaced

separately from the Chatbot Platforms through connectors.

Here, the Chatbot Platforms are hosted on cloud services which

can use Webhooks to communicate with the Channel. In the

context of this paper, we consider the integration of Chatbots as

text based inputs to IoT. By using Software Development Kits

(SDK) it is possible to integrate IoT to voice/speech based

Intelligent Personal Assistants such as Amazon Echo (using

Alexa SDK) and Google Home (using Google Assistant

Once the action has been set and the require parameters have

also been defined, the correct intent can be mapped to an IoT

API endpoint and a HTT

GITHUB LINK

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  | | --- | |  | | |  | | --- | |  |  |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | |  | | --- | |  |   Not able to enter the code? Paste the following link into your browser: <https://github.com/users/Kanmani456/emails/283665712/confirm_verification/82135795?via_launch_code_email=true> | | | | |

CODING

# cleaner.py

2

3import re

4

5def remove\_chat\_metadata(chat\_export\_file):

6 date\_time = r"(\d+\/\d+\/\d+,\s\d+:\d+)" # e.g. "9/16/22, 06:34"

7 dash\_whitespace = r"\s-\s" # " - "

8 username = r"([\w\s]+)" # e.g. "Martin"

9 metadata\_end = r":\s" # ": "

10 pattern = date\_time + dash\_whitespace + username + metadata\_end

11

12 with open(chat\_export\_file, "r") as corpus\_file:

13 content = corpus\_file.read()

14 cleaned\_corpus = re.sub(pattern, "", content)

15 return tuple(cleaned\_corpus.split("\n"))

16

17if \_\_name\_\_ == "\_\_main\_\_":

18 print(remove\_chat\_metadata("chat.txt"))

import sys

import logging

import getpass

import os

from optparse import OptionParser

import sleekxmpp

from sleekxmpp.xmlstream import ET

import subprocess

# Python versions before 3.0 do not use UTF-8 encoding

# by default. To ensure that Unicode is handled properly

# throughout SleekXMPP, we will set the default encoding

# ourselves to UTF-8.

if sys.version\_info < (3, 0):

reload(sys)

sys.setdefaultencoding('utf8')

else:

raw\_input = input

OUTPUT

GeeksForGeeks is the best platform for DSA content

GeeksForGeeks is the best platform for DSA content\*\*Welcome to GFG

File "0b97e8c5-bacf-4e89-9ea3-c5510b916cdb.py", line 1

print(10, 20, sep=' - ', 30)

^

SyntaxError: positional argument follows keyword argument

THANK YOU