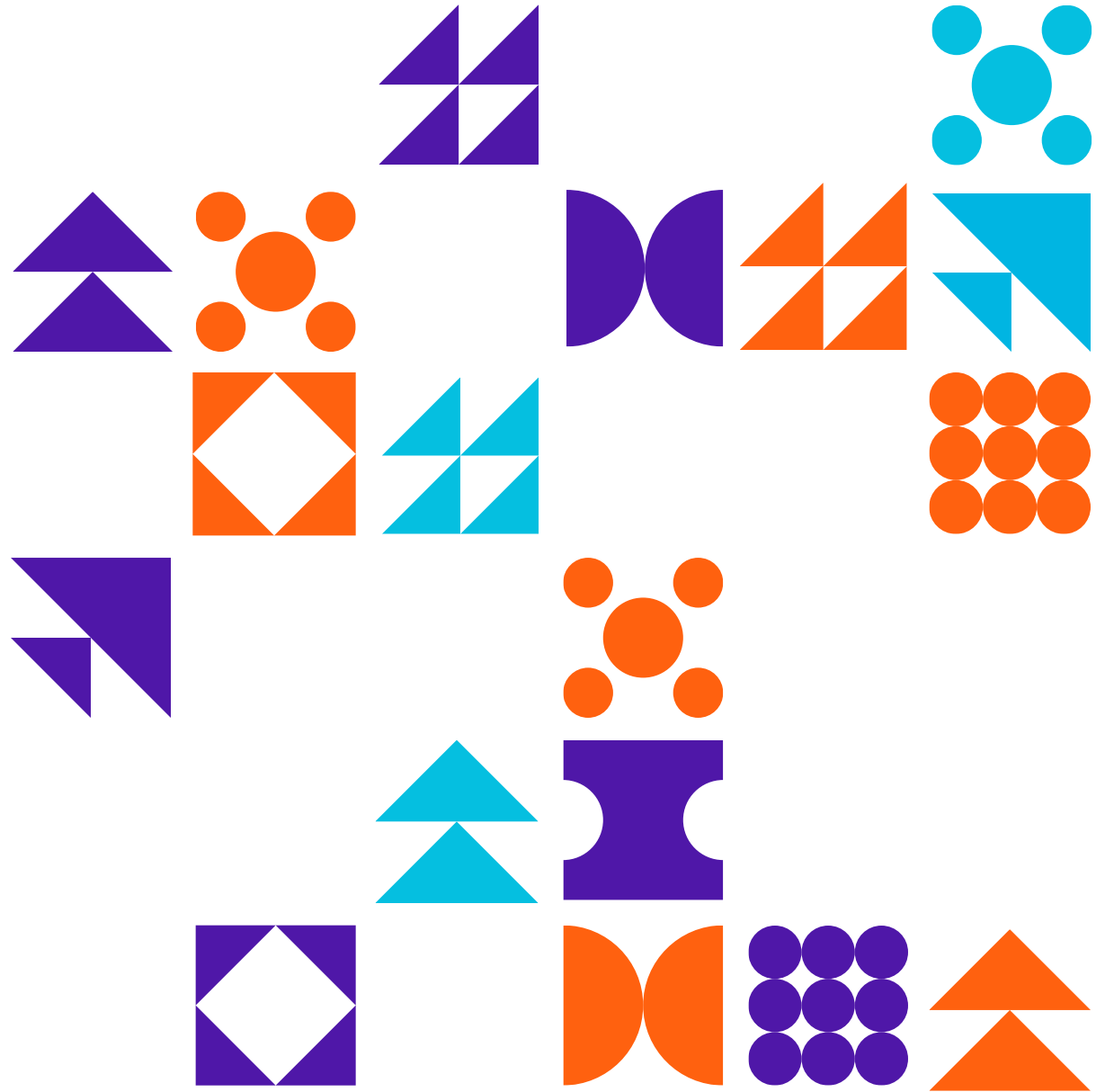
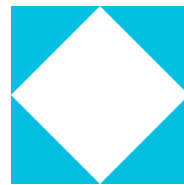


AI@Work 2025 Kick off WG3 AIK4PM Retrospective & 2025 Plan

2025年1月19日

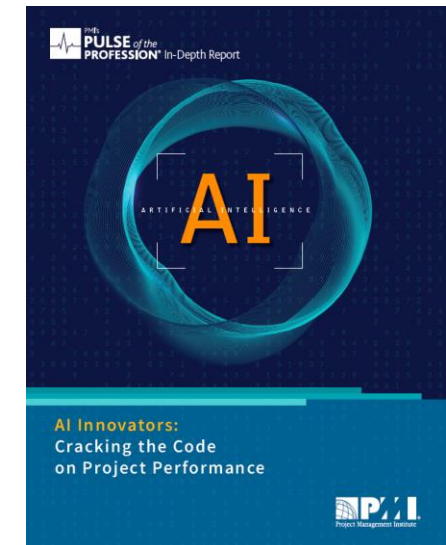
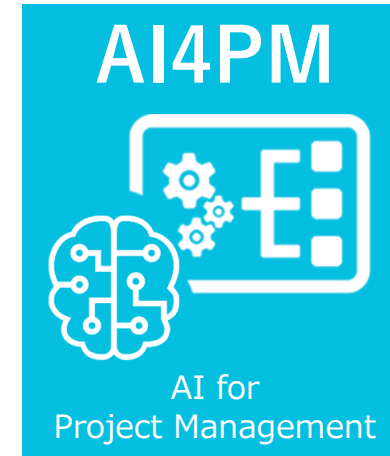
AI@Work Study Group WG3
内藤 睦博、足立 昌浩、曾我 みどり、有馬 裕輔

AI@Work Study Group, PMI Japan

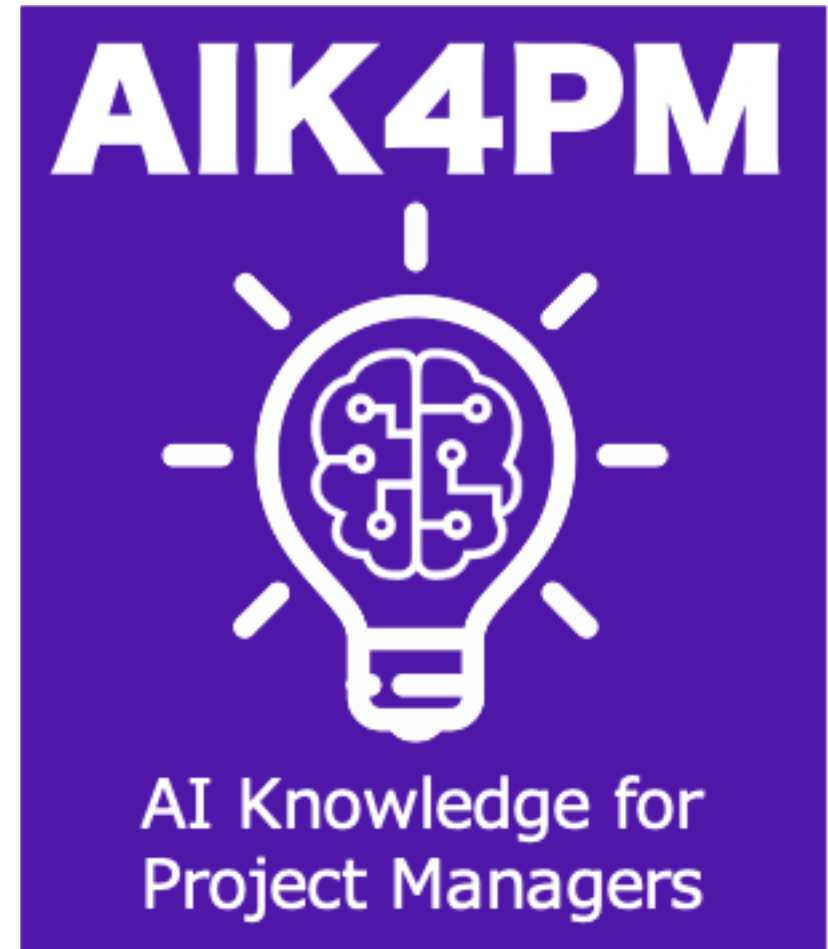


Agenda

1. WG3概要
2. 2024年活動内容の振り返り
3. 2025年活動計画



WG3概要



AIK4PM



AI Knowledge for
Project Managers

プロジェクト・マネジャーとして取得すべきレベルのAI技術またはAIの考え方などを研究する



プロジェクトマネジャーとして取得しておくべきAIのスキルを保有している。



- ・自分の組織でAIを導入することができるか？（保有課題を解決、低リスクで導入）
- ・スコープを明確にする

✗ AIエンジニア・研究者向けのAI知識

✗ 一般知識としてのAI



AI-Readyな

個人

トップ人材、中核人材、利用者それぞれをAI-Ready化
教育や研究開発環境の改革、利用者の意識変革を推進

守

破

離

WG3

WG1

WG2

WG3の活動を通じて目指せるレベル

レベル	概要	目指せるレベル
1	AIに関するプロジェクトについては本で知識を身に着けた程度	当然OK!
2	AIに関するプロジェクトの企画、計画、マネジメントについて議論が出来る	OK!
3	AIを利用したプロジェクトのサブPM、PMO、プロジェクトサポートなどが出来そう	若干
4	AIを利用したプロジェクトのマネジメントのイメージがつく、PoCがリード出来そう	次のステージ
5	AIを利用したプロジェクトのPMが出来る/PMを引き受ける気持ちになれる	次のステージ

2024年活動内容の振り返り

- Python勉強会
- E資格勉強会
- 最新AIの動向調査

- ・教材を通しAIのプログラミングで使用するPythonの記述と機械学習アルゴリズムを学ぶ
- ・目的はPythonの実習を通じて、PMとして取得すべきAI技術レベル・AIの考え方などを研究すること
- ・進め方は希望者でチームを作り月2回ミーティング実施（オンライン）

Python 実践 機械学習システム 100本ノック

下山輝昌・三木孝行・伊藤淳二 著



- ・教材：
Python実践機械学習システム100本ノック
- ・学習スケジュール

月	学習内容	月	学習内容
3月	学習計画報告	8月	10章
4月	6章	9月	10章, 11章
5月	7章	10月	第10章：機械学習システムのダッシュボードを作成
6月	8章	11月	第11章：放課後ノック（LLM-API by OpenAI Platform）前半
7月	9章	12月	第11章：放課後ノック（LLM-API by OpenAI Platform）後半

- OpenAIのAPIを使いmodel, temperatureを変更することで回答の変化を確認

```
!pip install openai==1.26.0

[ ] import os
import openai
os.environ["OPENAI_API_KEY"] = "取得したキーを指定"
openai.api_key = os.getenv("OPENAI_API_KEY")

client = openai.OpenAI(
    api_key=os.getenv("OPENAI_API_KEY"),
)

# 429 Error発生。Creditを登録したらいけた
chat_completion = client.chat.completions.create(
    messages=[
        {
            "role": "user",
            "content": "面白い挨拶を5個教えて",
        }
    ],
    model="gpt-3.5-turbo",
    temperature=0
)

print(chat_completion)
```

モデル、ランダム性を
変化させると...

model : 使用するモデル
Temperature : 生成のランダム性。0~1の値
1に近い方がランダム性が高い

model temperature	<code>print(chat_completion.choices[0].message.content)</code>
model="gpt-3.5-turbo", temperature=0	1. おはようございまーす！今日も元気にいきましょう！ 2. こんにちは！今日も素敵な一日になりますように！ 3. おっす！今日も最高の一日にしようぜ！ 4. おはよう！笑顔で挨拶すると幸せがやってくるよ！ 5. どうも！今日も楽しいことがたくさん待ってるぞ！
model="gpt-3.5-turbo", temperature=0.8	1. 「おはようございません！今日も寝ぼけてます！」 2. 「こんにちは！元気にしてる？僕はちょっとだけ元気！」 3. 「こんばんは！今日も一日お疲れさまでした。明日もがんばろう！」
model="gpt-4", temperature=0	1. "こんにちは、星から来ました。地球はどうですか？" 2. "おはようございます！今日も一日、笑顔でいきましょう！" 3. "こんにちは、あなたの笑顔が今日の私のエネルギー源です！" 4. "おはようございます、今日も一日、あなたの素晴らしさを世界に放ってください！" 5. "こんにちは、あなたの存在がこの世界を明るく照らしていますね！" 6. "おはようございます、今日も一日、あなたの夢に一步近づく日にしましょう！" 7. "こんにちは、あなたの笑顔が私の一日を明るくしてくれます！" 8. "おはようございます、今日も一日、あなたの素晴らしさを全力で発揮してください！" 9. "こんにちは、あなたの存在がこの世界をより美しくしていますね！" 10. "おはようございます、今日も一日、あなたの夢を追い続けてください！"
model="gpt-4", temperature=0.8	1. 「どうも、私は未来から来ました。今日はどんな日になるか既に知っています。」 2. 「こんにちは、私はあなたの一日を一瞬で明るくする魔法使いです。」 3. 「よろしくお願いします、私は笑顔の伝染者です。」 4. 「こんにちは、私はあなたの笑顔のシェフです、今日のメニューは何にしましょうか？」 5. 「ガンバローゼ、私たちは今日も地球を救うヒーローだ。」 6. 「どうも、今日もピカピカの私が参上しました。」 7. 「よっ、私たちは今日も一日頑張る星のように輝く準備はできているかい？」 8. 「ハイ、あなたの一日のエネルギーチャージャーがここにいます。」 9. 「こんにちは、私はあなたのポジティブエネルギーの提供者です。」 10. 「よー、今日はどんな冒険をしましょうか？」

• Langchainの使い方を学ぶ。

Topics Langchainを使ってみる：応答内容の出力形式いろいろ

辞書型

```
[ ] res_dict = output_parser.parse(res.content)
res_dict

{'原文': 'こんにちは', '翻訳結果': '안녕하세요'}
```

print(res_dict['原文'])
print(res_dict['翻訳結果'])

こんにちは
안녕하세요

カンマ区切りリスト形式

```
[ ] from langchain.output_parsers import CommaSeparatedListOutputParser

output_parser = CommaSeparatedListOutputParser()
format_instructions = output_parser.get_format_instructions()

chat_prompt = ChatPromptTemplate.from_messages([
    SystemMessagePromptTemplate.from_template("{format_instructions}"),
    HumanMessagePromptTemplate.from_template("面白い挨拶をいくつか教えて")
])

messages = chat_prompt.format_prompt(format_instructions=format_instructions).to_messages()
res = chat.invoke(messages)

res_list = output_parser.parse(res.content)
print(res_list)
print(type(res_list))

['こんにちは', 'おはよう', 'こんばんは', 'どうも', 'おっす']
<class 'list'>
```

Topics Langchainを使ってみる

レスポンス・スキーマの型がstringとなっていたので、結果が数値となるような場合はどうなるのか試してみた

```
[17] from langchain.output_parsers import StructuredOutputParser, ResponseSchema

response_schemas = [
    ResponseSchema(name="計算結果", description="計算結果"),
    ResponseSchema(name="入力式", description="入力式")
]
output_parser = StructuredOutputParser.from_response_schemas(response_schemas)
format_instructions = output_parser.get_format_instructions()
print(format_instructions)

The output should be a markdown code snippet formatted in the following schema, including the leading and trailing "```json" and "```":
```json
{
 "計算結果": string // 計算結果
 "入力式": string // 入力式
}
```

計算結果もString

```
chat_prompt = ChatPromptTemplate.from_messages([
 SystemMessagePromptTemplate.from_template("{format_instructions}"),
 HumanMessagePromptTemplate.from_template("2の4乗")
])

messages = chat_prompt.format_prompt(formula="2の4乗", format_instructions=format_instructions).to_messages()
res = chat.invoke(messages)
print(res.content)

```json
{
  "計算結果": "16",
  "入力式": "2^4"
}
```

日本語で入力した式が計算式に変換されていた

エージェントによるデータ分析

エージェントによるデータ分析（ファイル分析）

エージェントに指示を出すことでエージェントが自律的に考え、結果を出力します。

```
agent = initialize_agent([shell_tool, repl_tool], chat, agent=AgentType.ZERO_SHOT_REACT_DESCRIPTION, verbose=True)
```

```
result = agent.invoke("data/tbl_order_202004.csv ファイルの分析を行ってください。統計的な代表値を /content配下にてcalculated.csvファイルとして出力してください。")
print(result)
```

実行結果

```
Thought: I have located the file "tbl_order_202004.csv" in the /content/sample_data directory. Now I can copy it to the /content directory.
Action: copy_file
Action Input: src=/content/sample_data/tbl_order_202004.csv dest=/content
Observation:
Thought: I have successfully copied the file "tbl_order_202004.csv" to the /content directory. Now I can proceed with the data analysis and calculation.
Action: python
Action Input: python
Observation:
Thought: I will use the pandas library to read the CSV file and perform the necessary calculations.
Action: python
Action Input: python
Observation:
Thought: I have successfully loaded the data from the CSV file into a pandas DataFrame. Now I can calculate the statistical values.
Action: python
Action Input: python
Observation:
Thought: I have calculated the statistical values for the data. Now I need to save them to a new CSV file.
Action: python
Action Input: python
Observation:
Thought: I have successfully calculated the statistical values and saved them to the file "calculated.csv" in the /content directory.
Final Answer: The statistical values have been calculated and saved in the file "calculated.csv" in the /content directory.
```

order_id	order_date	customer_id	customer_name	order_status	order_total	order_subtotal	order_tax	order_shipping	order_discount	order_note
1000001	2020-04-01	1000001	John Doe	Completed	100.00	80.00	10.00	10.00	0.00	
1000002	2020-04-02	1000002	Jane Smith	Completed	200.00	160.00	20.00	20.00	0.00	
1000003	2020-04-03	1000003	Bob Johnson	Completed	300.00	240.00	30.00	30.00	0.00	
1000004	2020-04-04	1000004	Alice Brown	Completed	400.00	320.00	40.00	40.00	0.00	
1000005	2020-04-05	1000005	Charlie Davis	Completed	500.00	400.00	50.00	50.00	0.00	
1000006	2020-04-06	1000006	Diana Prince	Completed	600.00	480.00	60.00	60.00	0.00	
1000007	2020-04-07	1000007	Frank Miller	Completed	700.00	560.00	70.00	70.00	0.00	
1000008	2020-04-08	1000008	Grace Wilson	Completed	800.00	640.00	80.00	80.00	0.00	
1000009	2020-04-09	1000009	Henry Taylor	Completed	900.00	720.00	90.00	90.00	0.00	
1000010	2020-04-10	1000010	Ivy Clark	Completed	1000.00	800.00	100.00	100.00	0.00	

- JDLA E資格の受験資格及び合格を目指す（8月の試験対象）
- 希望者でチームを作り、月次ミーティング、必要に応じて個別ミーティングを実施
- 過去合格者が受験に関するノウハウなどを共有
- 2024年は1名の方が合格

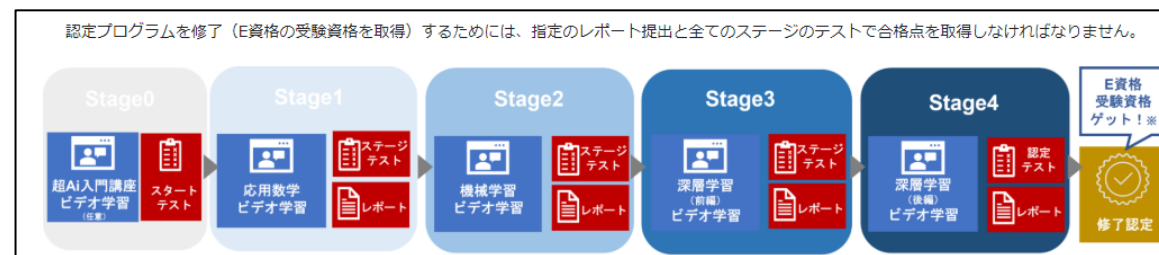
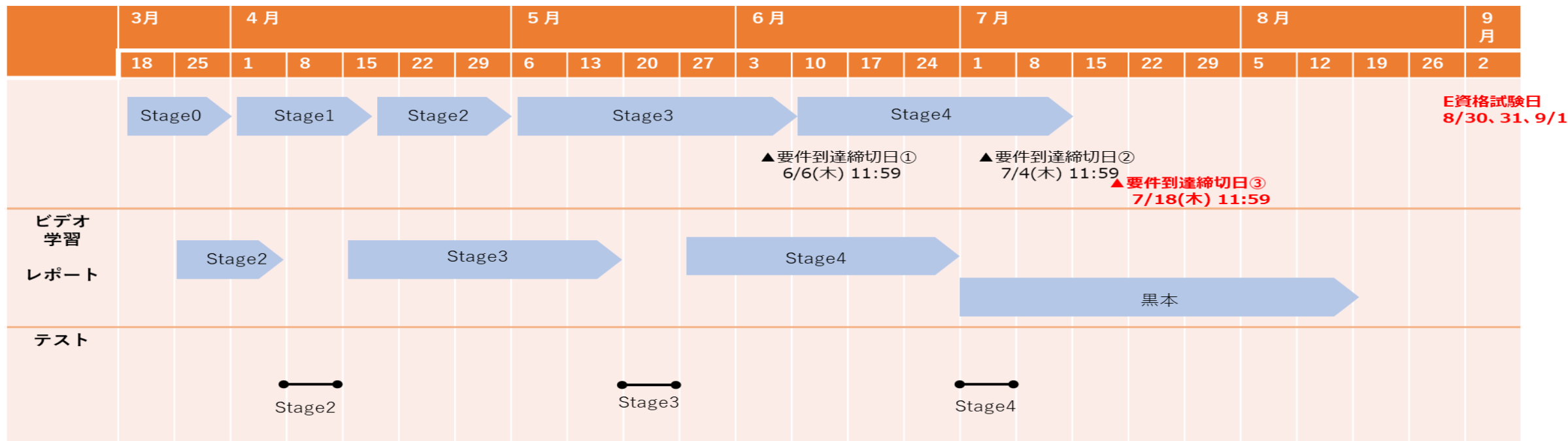


JDLA
Deep Learning for
ENGINEER
2022 #1

- 日本ディープラーニング協会主催
- G検定（General）、E資格（Engineer）が存在
- 合格者のみ参加できるコミュニティで情報交換、勉強会参加も可能

E資格勉強会(学習計画)

- E資格受験資格を得るためにはAIに関連する応用数学,機械学習,深層学習の講義(text,ビデオ),Report提出,単元テストに合格する必要があり、5ヶ月で走りきるために以下の計画を策定しグループ学習を実施した。



出典元:[ラビット・チャレンジ/Deep Learning H.P](#)より

- ・最新のAIツールの使用方法とその仕組みについて学習
- ・目的はAIツールの使い方を学び,PM業務に役立てる方法を考える
- ・進め方は月にテーマに対して興味がある方が調査し月一回調査結果をミーティング（オンライン実施）で報告

Cycle 9		Cycle 10	
3月	ChatGPTを業務に組み込むためのハンズオン	8月	
4月	ChatGPTの使い方(Advanced Data Analysis),ChatGPT API	9月	ChatGPT ,Claude3.5 SonnetとGemini 1.5 Proの回答結果比較 ChatGPT(speech to Text) 紹介と議事録作成
5月	ChatGPT API	10月	AI chat アプリ紹介（Streamlit Cloud,DALL-E3,画像認識,web Browsing Agent） ChatGPT(speech to Text) 紹介と議事録作成
6月	ChatGPTによるプログラム言語の変換	11月	Azure OpenAI紹介
7月	GPT-4機能紹介, ChatGPT API	12月	NotebookLM , OpenAPI GPTs使用のtips , sora(動画生成AI)

最新AIの動向調査(報告事例紹介#1)

• OpenAI APIの使い方紹介

API実行例-1(質問応答)

```
from openai import OpenAI
client = OpenAI()

completion = client.chat.completions.create(
    model="gpt-4-0125-preview",
    model="gpt-4-1106-preview",
    messages=[
        {"role": "system", "content": "あなたはプログラムマネージャです。"},
        {"role": "user", "content": "プロジェクトを開始するために行うtaskを抽出してください。時間の制約が有るため優先順序を考慮して抽出してください。"}
    ]
)

print(completion.choices[0].message.content)
```

プロジェクトを開始するために必要なタスクを優先順位を考慮して抽出する際、以下のようなフレームワークを用いると有効です。
ここではPMBOK (Project Management Body of Knowledge) のガイドラインを参考にしつつ一般的なプロジェクトの開始フェーズに必要なタスクを挙げていきます。
優先順位の高い順:

1. **プロジェクト憲章の作成** - プロジェクトの基本情報、目的、主要ステークホルダー、権限範囲等を明確に文書化する。
2. **ステークホルダー分析** - 関連するステークホルダーを識別し、その影響力や期待を把握する。
3. **プロジェクトマネジャーの指名** - プロジェクトを導くリーダーの選定を行う。
4. **初期チームの構成** - チームメンバーを決定し、必要なリソースを確保する。
5. **プロジェクトの目的と範囲の明確化** - 成果物およびプロジェクトの範囲を定義する。
6. **プロジェクト管理計画の策定** - スケジュール、コスト、品質、コミュニケーション、リスク管理等の計画を策定する。
7. **リスクマネジメントプランの策定** - 把握されたリスクを管理するためのプランを作成する。
8. **プロジェクト管理のプロセスの確立** - プロジェクト全体で使用する管理のメソドロジーを定義する。
9. **コミュニケーションプランの作成** - ステークホルダー間のコミュニケーションの方法と頻度を決定する。
- 10.10. **キックオフミーティングの準備と開催** - プロジェクトの目的と期待をチーム全員で共有し、モチベーションを高める。
- 11.11. **ワークブレイクダウン構造 (WBS) の作成** - プロジェクトを個々の作業要素に分割し、管理しやすくする。
- 12.12. **プロジェクトの詳細スケジュールの作成** - 具体的な作業のタイムラインを定める。
13. 時間の制約がある場合は、より重要なタスクから順番に取り組み、それぞれのタスクの完了度を確認しながら進めることが重要です。
14. また、タイムリミットがある場合は、成果物を「最小限の実行可能製品 (MVP)」に絞り、スコープを限定しつつプロジェクトを進捗させる必要があるかもしれません。
15. これは「アジャイル」や「スクラム」の概念を取り込んだアプローチです。プロジェクト全体を見通しながら、随時最も効果的なタスクにリソースを配分してください。

API実行例-2(画像生成)

```
from openai import OpenAI
client = OpenAI()

response = client.images.generate(
    model="dall-e-3",
    prompt="パン専門店 日本 先進的なレジ",
    size="1024x1024",
    quality="standard",
    n=1,
)

image_url = response.data[0].url

from PIL import Image
import requests
import io

# 画像表示
img = Image.open(io.BytesIO(requests.get(image_url).content))
img
```



API実行例-3(Text to speech)

```
from openai import OpenAI

client = OpenAI()

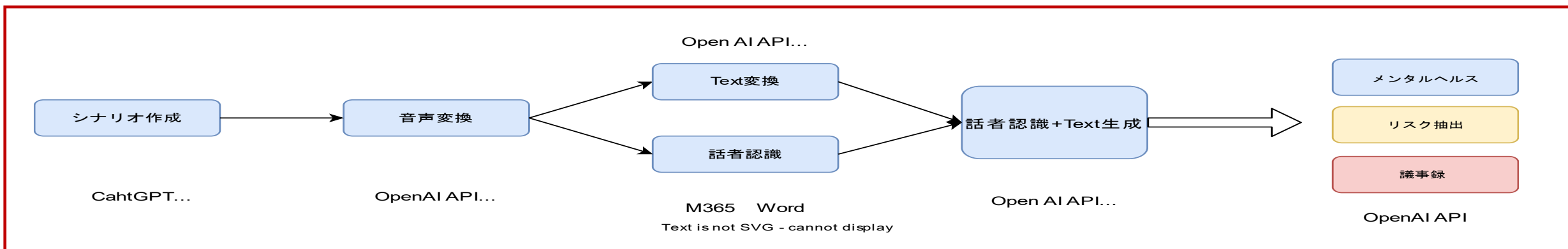
response = client.audio.speech.create(
    model="tts-1",
    voice="onyx",
    input="それでは月例会を始めたいと思います。"
)

response.stream_to_file("output.mp3")
```



最新AIの動向調査(報告事例紹介#2)

• OpenAI Apiを使った音声のtext化(議事録 .etc)



Prompt

Prompt :
生成したい内容と参加する人物のペルソナ。

Projectの進捗会議を行っています。出席者はproject manager , Team leader , 担当A , 担当Bです。
出席者のペルソナは以下の通り。
Project manager : 年齢45歳くらい。project manager としては中堅。責任感が強くレジリエンスが強い。
Team leader : 年齢40歳くらい。Team leader としては初。性格は繊細。スケジュール遅れは、特に気にする。
担当A: 中堅プログラマー。業務は適切にこなす。性格は明るい。
担当B: 新人プログラマー。Codingは初。業務が不慣れなため、報連相が遅れがち。
Projectの状況:
開発中盤で既に仕様は確定し、プログラムのCoding中。Task としては5項目が同時に進行。
TASK A,B,Cは担当Aが実施。納期は10/30。進捗はTask A 40% -> 50% , Task B 30% -> 40% , Task C 進捗中。課題は担当BのTask Dの出来により、Task CのIFを変更する必要があること。各Taskの検証は、Task A,B,C,D,Eが組みあがってからとなること。担当Bの進捗が今一つつかめない。
Task D,Eは担当Bが実行中。Task Eの進捗は50% -> 60%と順調に進んでいる。但しTask Dに関して、不明点が多く悩んでいるが、担当Bの性格的に質問できなくて悩んでいる。
上記状況と出席者のペルソナから進捗会議を進めてください。進捗会議は30 min です。
会話形式で進めてください。ファシリテータはproject managerです。
会話を全て出力してください。

生成シナリオ

"Project Manager (PM):" それでは、30分の進捗会議を始めましょう。まずは全体の進捗状況について確認していきます。各Taskの状況を整理して、課題についても話し合いたいです。

"PM:" 担当Aさん、まずTask A, B, Cについて教えていただけますか？

"担当A:" はい、了解です。Task Aが40%から50%まで進んでいます。Task Bは30%から40%、Task Cも20%から30%に進みました。いずれも予定通り進捗しています。

"PM:" ありがとうございます。順調ですね。ただ、Task Cについては担当BさんのTask Dに関連するIFがあると聞いていますが、そのリスクについて少し気になっています。

"担当A:" そうですね。Task Dがどう進むかによって、Task CのIFを変更する必要があるかもしれません。早めにその見通しが欲しいところです。

"PM:" わかりました。その点は後で話しましょう。次に、担当Bさん、Task DとEの進捗について教えてもらえますか？

"担当B:" はい、Task Eは順調で、50%から60%まで進んでいます。ただ、Task Dについては20%で止まっていて、進めるのが難しいです...

"PM:" そうですね。具体的にどこで詰まっていますか？

"担当B:" いくつか技術的な不明点があって、自分で調べてもなかなか解決できなくて...。どこから質問すればいいのかわかりません。

"PM:" なるほど。担当Bさん、質問するのが難しいと感じるのは理解できます。でも、ここで立ち止まるよりも、どんな小さなことでも聞いてもらった方がプロジェクト全体のためになります。

リスク抽出

Promptの例

このメモは進捗会議の議事録です。会議中の発言からリスクを洗い出し、リスク項目、リスク内容、改善策、リスクオーナーを特定して、この進捗会議の議事録から、メンバーの意識や感情に関するいくつかのポイントを抜き取ることができます。以下に、それぞれの話者の意識についてまとめます。

話者 1 (全体の進行役)

- プロジェクトの進捗状況を確認し、リスク要因についての議論を促す姿勢が見られます。
- 問題点に対して具体的な対策を講じることを重視しており、メンバーへのサポートを提案しています。
- スケジュール管理やリソース調整を通じて、全体の進捗を維持しようとする意識が強いようです。

話者 2 (メンバーAさん)

- Task Dの進捗が遅れているという報告を行う一方、Task Dの遅延の影響を懸念しています。
- Task CとのIFに関する影響に敏感であり、計画の柔軟性を持っているようですが、Task Dの遅れが全体検証に及ぶリスクを感じている様子です。
- 負荷テストの計画についてもリスクを認識しており、テストサーバーの手配について問題を抱えています。

話者 3 (フェーズBさん)

- Task Dの設計部分で不安を感じており、特にそれがTask Cとの連携に影響を及ぼすことを理解しています。
- Task Dの設計部分に関して具体的にサポートを提供する意欲を示しており、協力的な姿勢が見られます。
- 負荷テストの計画についてもリスクを認識しており、テストサーバーの手配について問題を抱えています。

話者 4 (アロイさん)

- Task Dの設計部分での不安を率直に表現しており、進捗が停滞していることを悩んでいる様子が見受けられます。
- 自身の作業が進まない原因について悩んでおり、サポートを求める姿勢が読み取れます。
- 天候の影響によるリモート環境の不安定さも懸念しており、作業の効率に対するストレスを感じているのかもしれない。

総括

全体として、メンバーは各々のTaskの進捗状況やリスク要因について意識を向けており、ストレスを感じている部分もあります。特にTask Dや負荷テストの計画遅延が全体に影響を与える可能性や、リモート環境の不安定さによるストレスが感じられます。

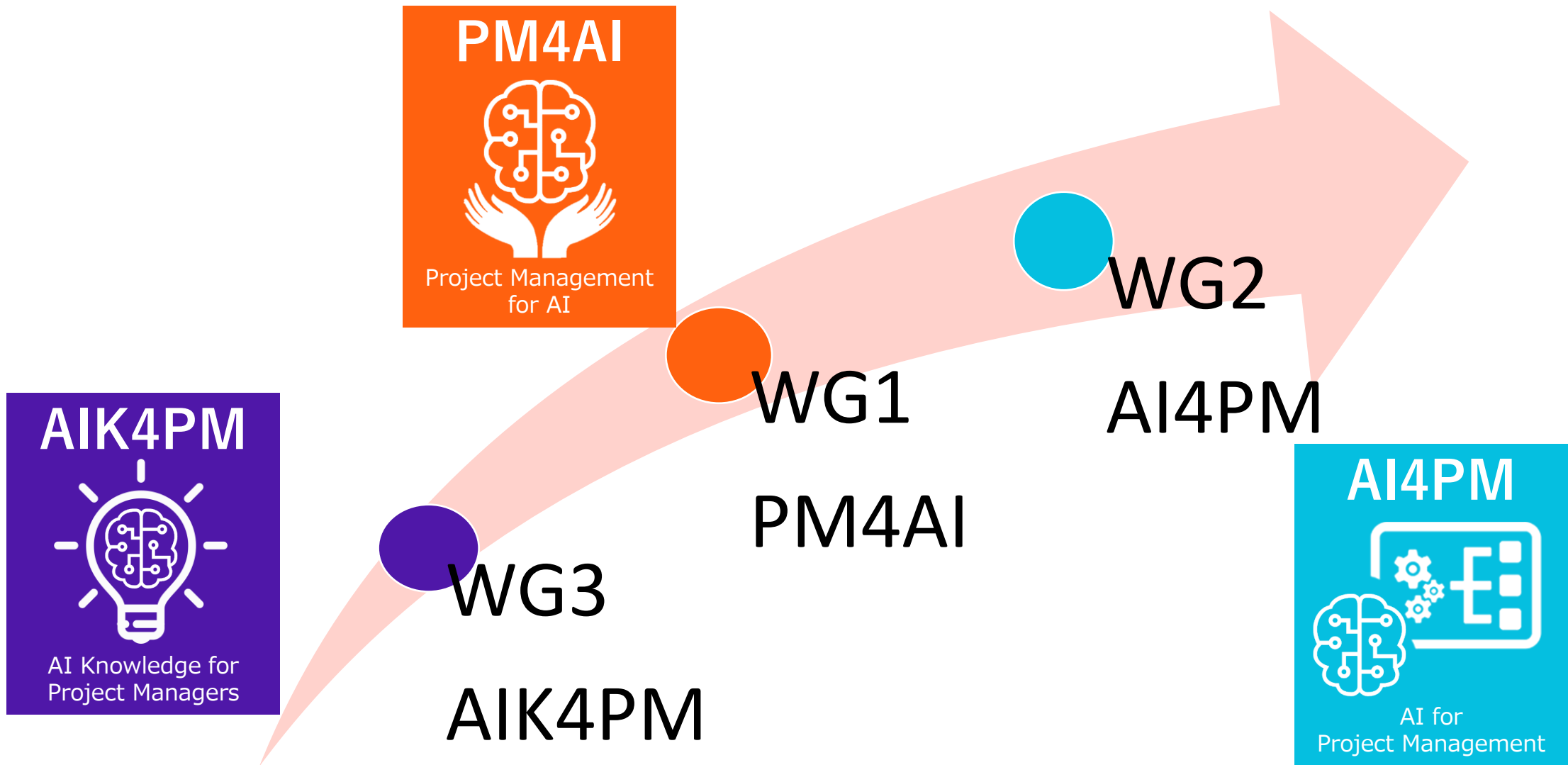
そのため、メンバー間でのサポートやコミュニケーションの促進、リソースの確保、Taskの優先順位を見直すことが重要です。また、リスクへの対応も柔軟に行う必要があります。これらを踏まえて、今後の進捗管理に注力することが必要でしょう。

会話から抽出したリスク項目

リスク項目	リスク内容	改善策	リスクオーナー
Task Dの進捗遅れ	Task Dが20%のまま止まっており、Task Cとの連携に影響が出る可能性がある。	フェーズBさんがアロイさんをサポートし、IF部分を再確認して進捗を上げる。	アロイさん、フェーズBさん
負荷テストの計画遅れ	負荷テストの実施計画がまだ具体的に決まっておらず、テストサーバーの手配が遅れている。	テストサーバーの優先度を上げるために上層部に働きかける。	話者1
リモートワークの不安定さ	台風による影響でアロイさんのインターネットが不安定になり、作業が進まない可能性がある。	インターネット環境が悪化した場合はオフィスで作業を行う方向で考える。	アロイさん

2025年活動計画

- ・ Python勉強会
- ・ 最新AIの動向調査



- WG3では、AI技術の進化が急速に進む中、以下の二つのテーマで活動を行います。

項目	学習目的
Python勉強会	• この勉強会では、指定された参考書のコードを実際に手を動かしながら学び、機械学習に必要なPythonのコード記述力とアルゴリズムの理解を深めます。
最新AI動向調査	• 各個人が最新のAIツールの使い方と原理を学び、互いに確認し報告することで、プロジェクトマネジャー業務に活用できるかを検討します。この取り組みにより、より多くのツールをカバーし、その特徴を把握することで、業務効率化と成果向上を目指します。

WG3では、2024年（Cycle9,10）において、青字で示した代表的な生成系A.Iツールを実際に使用し、情報共有を行いました。この1年でさらに新たなA.Iツールが登場し、多機能化・高性能化が進んでいます。2025年度はこれら新しいツールの効果を確認するだけでなく、既存ツールとの組み合わせについても検証を行い、実務に活用できる最適な方法を模索していきます。

LLM	My LLM	画像生成	ノーコード	Framwork	web Agent
ChatGPT gemini Claude3.5 Copilot Azure OpenAI ...	GPTs Gems Claude (Projects機能) Copilot ...	DALL-E Stable Diffusion MidJourney ...	Dify ...	LangChain Notebook LM ... 青字:Cycle9,10でWG3内で紹介されたTool	"Browser-UI (AgentGPT)" ...

カテゴリ	説明
LLM	大規模言語モデル（Large Language Model）の略。
My LLM	ユーザー独自のデータや要件に基づいてカスタマイズされたLLM。
画像生成	テキスト入力を基にAIが画像を生成する技術。
動画生成	テキストや画像から動画を生成する技術。
Framework	LLMや生成AIの利用を簡単にする開発フレームワーク。
Web Agent	ブラウザやAPIを介してウェブ上の操作を自動化するAIエージェント。
ノーコード	プログラミングなしでLLMや生成AIを利用可能なツール。

毎週第四水曜日 19:00-21:00

活動期間は2025年の1年間

Zoom、WebEx、Skype等を活用した
WEBミーティング

次回（初回） 1月22日（水）

THANK YOU

AI@WORK WG3
PMI Japan Chapter

