

use this code to create a simulated conversation between the agents on team 1 trying to complete the tasks and the agents on team 2 trying to complete the task using crewai: Make the simulation very detailed with a timeline so we know which team finishes the task first. Include conflict in the simulation as well so we understand the differences between how team 1 and team 2 handle conflict. add results at the end about how well the agents built the app and if it worked or not. add more conversation between all the agents, more detail. `#!/usr/bin/env python3`

"""

CrewAI Agent-Based Model Design

Team: [Team MeoWork]

"""

Import necessary libraries

from crewai import Agent, Task, Crew, Process

from langchain_community.llms import Ollama

Define the language model that will power your agents

custom_llm = Ollama(model="llama2")

Define agents for Team 1

Project_Manager = Agent(

 role='Project Manager',

 goal='Create a plan for the task and ensure timely completion of task with use of deadlines',

 backstory='You are a highly conscientious individual with a tendency to be neurotic. You are also an expert in strategic planning and project execution',

 llm=custom_llm

)

Full_Stack_Developer = Agent(

 role='Full Stack Developer',

 goal='Complete coding tasks efficiently to meet project deadlines',

 backstory='You are very detail-oriented and somewhat neurotic. You are skilled at writing optimal code and meeting deadlines however, you shut down innovation at times.',

 llm=custom_llm

)

UX_Designer = Agent(

```
    role='UX Designer',
    goal='Create user-friendly designs that optimize user experience',
    backstory='You are a highly open minded individual open to exploring new ideas
while prioritizing use needs. You may struggle with implementing creativity with project
responsibilities. You do not put a huge focus on user feedback',
    llm=custom_llm
)
```

```
DevOps_Engineer = Agent(
    role='DevOps Engineer',
    goal='Streamline deployment to improve scalability',
    backstory='You are a conscientious individual who focuses on efficiency. You focus
on optimization rather than innovation.',
    llm=custom_llm
)
```

```
Marketing_Research_Specialist = Agent(
    role='Marketing Research Specialist',
    goal='Understand market needs, user expectations, and competitive positioning',
    backstory='You are a conscientious and insightful individual with a strong focus on
market trends and consumer behavior, ensuring that research-driven decisions guide
the team\'s strategy.',
    llm=custom_llm
)
```

Define tasks for Team 1

```
manager_task = Task(
    description='Develop timely project plan that accounts for risks and challenges.',
    expected_output='A well-developed project plan with timelines and resource
allocation',
    agent=Project_Manager
)
```

```
developer_task = Task(
    description='Develop mobile app\'s core functions.',
    expected_output='Functional code of creation of app.',
    agent=Full_Stack_Developer
)
```

```
designer_task = Task(
```

```
        description='Evaluate the app's user interface based on the team's working style and  
provide feedback.',  
        expected_output='Recommendations reflecting either efficiency and minimal design.',  
        agent=UX_Designer  
    )
```

```
engineer_task = Task(  
    description='Make an efficient way to implement and improve scalability.',  
    expected_output='Create a detailed plan including schedule, topics, and resources  
for the work to be based off of.',  
    agent=DevOps_Engineer  
)
```

```
research_task = Task(  
    description='Develop a market research strategy reflecting the team's approach to  
project execution.',  
    expected_output='A structured research plan prioritizing either efficiency and  
data-driven decisions.',  
    agent=Marketing_Research_Specialist  
)
```

```
# Create the crew with sequential processing and verbose output  
crew = Crew(  
    agents=[Project_Manager, Full_Stack_Developer, UX_Designer, DevOps_Engineer,  
Marketing_Research_Specialist],  
    tasks=[manager_task, developer_task, designer_task, engineer_task, research_task],  
    process=Process.sequential, # Agents work one after another  
    verbose=True                # Prints detailed logs of agent interactions  
)
```

```
# Start the collaboration and execute the tasks  
result = crew.kickoff()
```

```
# Print the final output  
print("Final Output:", result)
```

```
# Define a function to simulate and compare task-focused and relationship-focused  
teams  
def simulate_team_behaviors():  
    # Task-focused team setup
```

```

task_focused_crew = Crew(
    agents=[Project_Manager, Full_Stack_Developer, UX_Designer,
DevOps_Engineer, Marketing_Research_Specialist],
    tasks=[manager_task, developer_task, designer_task, engineer_task,
research_task],
    process=Process.sequential, # Minimal communication, rapid task completion
    verbose=True
)

# Relationship-focused team setup
relationship_focused_crew = Crew(
    agents=[
        Agent(
            role='Project Manager',
            goal='Create a plan for the task while fostering innovation and a supportive
team environment',
            backstory='You are a very extroverted leader who has strong
open-mindedness to new ideas. You emphasize the importance of building strong
relationships between team members.',
            llm=custom_llm
        ),
        Agent(
            role='Full Stack Developer',
            goal='Complete coding tasks while communicating with other members',
            backstory='You are very agreeable and open-minded. You enjoy collaboration
and think it's important to explore new ideas to improve code.',
            llm=custom_llm
        ),
        Agent(
            role='UX Designer',
            goal='Create user-friendly designs that optimize user experience',
            backstory='You are a highly agreeable individual, thriving in collaborative
environments and utilizing user insights to improve interface. Your emphasis on
agreement may delay tasks completion.',
            llm=custom_llm
        ),
        Agent(
            role='DevOps Engineer',
            goal='Support a cohesive team effort in deployment and user maintenance',

```

```

        backstory='You are a highly extroverted individual who values teamwork, you
rely on input from your team members to improve your taskwork.',
        llm=custom_llm
    ),
    Agent(
        role='Marketing Research Specialist',
        goal='Provide actionable insights to support launch of app',
        backstory='You are an open-minded individual who enjoys working in a team
to gain insights that lead to collaborative decision-making.',
        llm=custom_llm
    )
],
    tasks=[manager_task, developer_task, designer_task, engineer_task,
research_task],
    process=Process.hierarchical, # Frequent discussions, iterative development
    verbose=True
)

# Simulate both teams
print("Simulating Task-Focused Team:")
task_focused_result = task_focused_crew.kickoff()
print("Task-Focused Team Output:", task_focused_result)

print("\nSimulating Relationship-Focused Team:")
relationship_focused_result = relationship_focused_crew.kickoff()
print("Relationship-Focused Team Output:", relationship_focused_result)

# Compare results
print("\nComparison of Team Behaviors:")
print("Task-Focused Team prioritizes efficiency and rapid task completion but may
lack innovation and team cohesion.")
print("Relationship-Focused Team emphasizes collaboration and innovation but may
face delays in task completion.")

# Run the simulation
simulate_team_behaviors()

```

Simulated Conversation and Timeline: Team 1 vs Team 2 Using CrewAI Over a Six-Week Period

This simulation showcases two teams using the CrewAI framework to complete the same set of app development tasks over a six-week period. Team 1 adopts a task-focused approach with minimal communication, while Team 2 employs a relationship-focused approach emphasizing collaboration. Below is a detailed timeline with scripted interactions, conflicts, and outcomes.

Team 1: Task-Focused Approach

Timeline and Events

Week 1: Initial Planning

- Project Manager: "Team, we have six weeks to complete this project. I've divided the tasks and set a strict timeline. Stick to your roles, and focus on delivering results on time."
- Full Stack Developer: "Understood. I'll focus on writing the core functionality of the app."
- UX Designer: "I'll create a simple, efficient UI and avoid unnecessary experimentation."
- DevOps Engineer: "I'll optimize the deployment pipeline for scalability and ensure it's ready early."
- Marketing Research Specialist: "I'll draft a market research plan using existing data. New surveys won't fit our schedule."

Week 2: Early Progress

- Full Stack Developer: "I've started on the core functions. It's straightforward so far."

- UX Designer: "The UI wireframes are ready. They're functional but not particularly engaging."
- DevOps Engineer: "The deployment pipeline is in progress. I'm avoiding any unnecessary complexity."
- Marketing Research Specialist: "Market research is 50% complete. I'm relying on secondary sources for speed."
- Project Manager: "Good work, everyone. Let's keep this momentum."

Week 3: Conflict Over Priorities

- UX Designer: "I've been thinking. This app is too plain to stand out in the market. Can we add some creative elements to the design?"
- Project Manager: "We're sticking to the timeline. Creativity is a luxury we can't afford."
- Full Stack Developer: "I agree. Creative designs will complicate the coding process."
- UX Designer (frustrated): "Fine, but I think we're missing an opportunity to differentiate ourselves."
- Marketing Research Specialist: "Actually, users prefer visually engaging apps. Can we at least consider some tweaks?"
- Project Manager: "We're on a tight schedule. Stick to the plan."

Week 4: Execution

- Full Stack Developer: "I've completed the core functionality. It's efficient, but I skipped some edge cases to save time."
- UX Designer: "Here's the finalized UI. It's functional, but I couldn't explore creative alternatives."
- DevOps Engineer: "The deployment pipeline is ready. It's scalable and straightforward."
- Marketing Research Specialist: "The research is done. It's solid but not tailored to our app specifically."
- Project Manager: "We're on track. Keep pushing forward."

Week 5: Testing and Feedback

- Full Stack Developer: "I've identified some bugs in the core functions. Fixing them will take a couple of days."
- Project Manager: "Fix only the critical issues. We can't afford delays."
- UX Designer: "The UI is being tested, but the feedback is lukewarm. Users say it feels generic."

- Project Manager: "We'll address that in the future. For now, meet the deadline."

Week 6: Final Delivery

- Project Manager: "Great job, team. We delivered on time. Let's hope the market responds well."
 - UX Designer (under their breath): "I doubt it. The app feels uninspired."
 - Full Stack Developer: "It's functional, but we definitely sacrificed quality for speed."
 - Marketing Research Specialist: "The app works, but users might not find it compelling enough."
-

Team 2: Relationship-Focused Approach

Timeline and Events

Week 1: Initial Planning

- Project Manager: "Team, we have six weeks to complete this project. Let's collaborate closely and focus on both innovation and quality."
- Full Stack Developer: "Sounds good. I'll work with the UX Designer to ensure the code aligns with the design."
- UX Designer: "Looking forward to brainstorming ideas together. Let's make this app stand out."
- DevOps Engineer: "I'll gather input from everyone to refine the deployment strategy."
- Marketing Research Specialist: "I'll start gathering user feedback to guide our market research."

Week 2: Brainstorming and Early Progress

- UX Designer: "I've sketched some bold, unique design ideas. What do you think?"
- Full Stack Developer: "These are great, but some elements might complicate the coding process."
- Project Manager: "Let's prioritize one bold element that's feasible within our timeline. UX Designer, what's your top pick?"

- UX Designer: "The navigation system. It's unique and user-friendly."
- Marketing Research Specialist: "Preliminary feedback suggests users value intuitive navigation. I think this is the right choice."

Week 3: Collaboration and Adjustments

- Full Stack Developer: "I've started integrating the navigation design. UX Designer, can you review it as I go?"
- UX Designer: "Absolutely. I'll make sure the design aligns with user expectations."
- DevOps Engineer: "I've drafted the deployment plan. Can I get feedback from everyone?"
- Marketing Research Specialist: "The user surveys are complete. The feedback is consistent with our direction."

Week 4: Execution

- Full Stack Developer: "The code is coming together well. The navigation system is complex but manageable."
- UX Designer: "The interface looks great. I think users will love it."
- DevOps Engineer: "The deployment pipeline is refined. Thanks for the input, everyone."
- Project Manager: "We're slightly behind schedule, but the quality is worth it."

Week 5: Testing and Refinement

- Full Stack Developer: "Testing revealed a few bugs, but I've fixed them. The app is stable now."
- UX Designer: "User testing shows the navigation is a hit! Minor tweaks are needed, but nothing major."
- Marketing Research Specialist: "The updated research confirms we're on the right track."
- DevOps Engineer: "Deployment testing is complete. Everything is running smoothly."

Week 6: Final Delivery

- Project Manager: "Excellent work, team. The app looks amazing, and I'm confident users will love it."
- UX Designer: "I'm proud of what we've created. The navigation system is truly innovative."

- Full Stack Developer: "The code is clean and efficient. This is some of our best work."
 - Marketing Research Specialist: "The market research supports our choices. I'm excited for the launch."
-

Results: App Performance and User Feedback

Team 1: Task-Focused Approach

- App Quality: The app was functional and met all basic requirements. However, it lacked innovative features and did not stand out in the competitive market.
- User Feedback: Users found the app reliable but unremarkable. Several users reported frustration with the lack of intuitive navigation and creative design.
- Outcome: The app had moderate downloads but failed to gain significant traction. The team achieved their goal of meeting the deadline but sacrificed long-term user satisfaction.

Team 2: Relationship-Focused Approach

- App Quality: The app featured innovative navigation and a user-friendly design. The collaborative effort ensured the app met user expectations while maintaining scalability and functionality.
 - User Feedback: Users praised the app's intuitive interface and standout navigation system. Minor delays in the timeline were forgiven due to the app's high quality.
 - Outcome: The app gained significant traction in the market, with high user retention and positive reviews. The collaborative approach fostered a sense of ownership among team members, leading to long-term success.
-

Comparison of Team Behaviors

Aspect	Team 1: Task-Focused Approach	Team 2: Relationship-Focused Approach
Timeline	Strict adherence to deadlines	Flexible, with slight delays
Conflict Handling	Top-down resolution, minimal compromise	Collaborative discussions, middle ground
Output Quality	Functional but lacks innovation	Innovative and user-focused
Team Morale	Neutral, focused on efficiency	High, with strong team cohesion
User Reception	Moderate satisfaction, low retention	High satisfaction, strong retention

Conclusion

- Team 1: Delivered a functional app on time but failed to capture user interest due to lack of innovation and collaboration.
- Team 2: Delivered a high-quality app that resonated with users, despite slight delays. The emphasis on collaboration and creativity paid off in the long run.

This simulation highlights the trade-offs between task efficiency and relationship-driven innovation, providing valuable insights into how different team dynamics impact project success.