Amazon Q Caddie

link to the Quip : Amazon Q Caddie
[May 31 2024]
SkyLab Innovation Program - Idea Submission Questionnaire

- **Project Title**: CaddieAI.
  - **Team Members**: **Venkata Vemuri,** , **Nishanth Athelli**
- **Project Summary**:
  - CaddieAI is generative AI–powered assistant, which is built over Amazon Q, The goal of CaddieAI is to make engineering/project/program managers more productive by taking over the mundane scrum master tasks, the planning, the grooming, the retrospective, the daily stand ups, and summarize the notes of the sprint and communicate to the team.
  - Our AI sends out personalized messages to each team member asking about the status of the sprint and tasks their are assigned over slack to keep them updated and notified every day.
  - Forget tedious stand-up meetings! Our AI schedules all the meetings, runs the standup, It asks insightful questions, automatically sends clear, concise notes to engineering managers after stand ups and summarize the retrospective at the end of the sprint to the entire team.
  - The AI will also take a look at all the blocked tickets and prompts every ticket owner to discuss with the team at the end of the standup. [as parking lot discussions]
  - Managers/Engineers can just chat with Amazon Q, ask leading questions such as,
    - Create a sprint. [will create the sprint [taskei/sim] with all the team members, creates a retrospective the sprint]
    - Status of the sprint? [summarizes all tasks in progress vs blocked per team members visually chart/ table.]
    - are there any blockers for the team?
    - Create a story/ task / issue.
    - punt all tickets from the previous sprint into the current sprint.
    - summarize the retrospective with action items for the next!
    - what were the action items from yesterday!
    - what are the actions items for today/tomorrow.
    - Summarizes the standup meeting/scrums.
  - This frees up Engineering managers to become strategic coaches, unlocking team brilliance. Faster sprints, happier teams, and products that dominate the market. Making everyone in the team more productive and focus on the job at hand. - that's the future of Agile with AI!
  - ScrumAI will also be able to give real status updates of the sprint.
- **Objective**:
  - If there is one thing money can't buy is time. ScrumAI offers time to Engineering/Project managers by taking over the mundane tasks of the sprint. Making everyones lives more productive.
  - Managers spend about 10-12 hours every sprint to schedule and run the mundane meetings. Our AI will automate those tasks and give back managers those 10-12 hours back every sprint. That is about 30 days in a calendar year. Imagine the impact when every manager has an extra month in a calendar year.
  - Engineers don't have spend more time trying to create and update stories/tasks. Our AI will prompt engineers to update the tasks every day. Add blockers, dependencies on the tickets and send those updates to the managers.

- Everyone in the team gets daily updates on how far the sprint goals are achieved.
- **Innovation and Relevance**:
  - In the current Tech industry, Most of the engineering teams follow some form of agile software development. Our scrumAI aims to automate significant amount of the mundane tasks in software development.
- **Target Audience**:
  - All Managers/Engineers who want to have max impact on their teams and build software.
  - All teams in Amazon.
  - Eventually, scrumAI will be the bot every team would use to manage their team processes.
- **Technical Feasibility**:
  - Detains below..
- **Potential Challenges**:
  - Need an expert mentor to navigate us through training the models.
- **Estimated Timeline**:
  - **POC - Phase 1.**
    - We would integrate SIM ticketing with Amazon Q, so users can ask questions about their sprint boards,
      - such as summarize the current sprint.
      - pending tickets per engineer.
    - We would integrate Amazon Q with SIM ticket syetem. API
      - users should be able to create/update/delete/ assign a sprint/story/task from Amazon Q
    - We would integrate Amazon Q with Meetings.amazon.com
      - user should be able to schdule meetings for the sprint cermonies.
  - **Integrate with Chime + Amazon transcribe + customer feedback Phase II**
    - While we integrate chime and summarize the meeting notes, action items, etc.
    - We would reach out to early subscribers of our product for customer feedback to improve the product incrementally.
    - Existing DEMO for Amazon Chime SDK Meeting Summarizer.
  - **Integrate with slack/email + customer feedback phase III**
    - we improve and grow our product with customers and their feedback.
    - Improve # of customers.
  - **Phase IV**
    - Integrate with third party alternatives to chime/meetings/sim ticketing systems.
    - Add new persona's   like Product Manager, Director of engineering etc; Imagine how can we summarize the entire org information into a single ask.
    - Integrate the Amazon Q caddie ability inside visual code or other IDE's for engineers using Amazon Q developer.
    - Scheduling of OOTO, 1:1 with other team members one off team meetings should be done over Amazon Q caddie.
    - Ability to create the story, description, acceptance criteria, create sub tasks from just the title.
    - Customer feedback!!
    - Improve # of customers.
- **Expected Impact**:
  - Agile software development was built in 2000. In 24 years there hasn't been a significant change on the way teams

build software. Over time, The role of managing the team processes have been run by project managers, scrum masters, program managers, engineering managers, Now, its time for a computer to save time and speed up software development.

○ No engineer or team member deserves to spend 10-12 hours on sprint cermonies.

○ We expect to automate about 80% of the mundane tasks of scrum process which would enable engineers, mangers and leaders to focus more on product than process.

○ Team processes will be automated and teams become lighter and free from micro management.

○ Planning, grooming and retrospective of sprints should be hassle free.

Details:

We have 2 target persona's **Engineers** and **Manager,**

Manager : Katrin is a Engineering Manager, Lives in Seattle, managing a Engineering team of #10

- As a manager, I would to chat with Amazon Q asking leading questions such as
  - I would like to create a new sprint board.
    - creates the new sprint board,
    - pulls in tickets from the backlog.
    - schedules all the meetings for the sprint. standups, retrospective, demo's etc;
  - give me the summary my sprint.
    - returns the counts of open, resolved, blocked, and in progress tickets ni the sprint.
  - what is the summary of the standup/scurm today?
    - summaries the updates from each engineer.
    - lists any action items.
    - lists any blockers.
  - what percent of the sprint is completed ?
    - based of the #of engineers, days left and burn rate of the sprint, AI would give an analysis.
  - what are blockers in the current sprint ?
    - highlights all the tickets which are marked blocked.
  - who is not available for the sprint.
    - check for OOTO updates in the calender, lists the members.

Engineer: Ravi is a SDE II, lives in San Francisco, full stack engineer,

- As a engineer, I would to chat with Amazon Q asking leading questions such as
  - create a new story in the curent sprint.
    - this should prompt the user to give a title description and estimated story points.
  - create sub tasks for the story/task.

- this should prompt the user to give a title description and estimated story points.
- ask to add more sub tasks.
  - show the tickets assigned to me in the current sprint.
    - should list all the tickets with links to them.
      - update #1,#2, #3 in the above list
      - should promt the user about the type of upate, mark it resolved, or blocked or move it to the input queue.
      - burn points on the ticket.
    - add comments to #1, #2 or #3
- As a engineer, I would like to add comments or notes to the retorspective board.
- As a engineer, I would like eta tickets in tnput queue.