My Title

Github release notes process  
  
Rewrite release notes from a github repo for Program Managers, Engineers, and C-suite so that it's understandable to every audience.   
It needs to access each commit, understand the issues resolved, changes to the code, changes to documentation, and if there are any breaking changes. There's example documentation for what is a good response for each audiences. The process is triggered by a question from user on slack.  
  
  
Process Definition: GitHub Release Notes Generation and Distribution  
  
Process Overview and Objectives:  
The GitHub Release Notes Generation and Distribution process aims to create tailored release notes from a GitHub repository for different audience groups: Program Managers, Engineers, and C-suite executives. The process involves analyzing repository commits, understanding resolved issues, code changes, documentation updates, and identifying breaking changes. The ultimate goal is to produce clear, audience-specific release notes that effectively communicate updates and changes.  
Scope and Boundaries:  
  
  
In-scope: GitHub repository commits, issues, code changes, documentation changes, breaking changes  
Out-of-scope: Changes not recorded in the GitHub repository, future planned features  
  
Detailed Step-by-Step Description:  
  
a. Process Trigger:  
  
User asks a question about release notes in Slack  
Slack bot recognizes the request and initiates the process  
  
b. Repository Analysis:  
  
Access the GitHub repository  
Retrieve all commits since the last release  
Analyze commit messages and associated data  
  
c. Issue Resolution Identification:  
  
Cross-reference commits with closed issues  
Compile a list of resolved issues  
  
d. Code Change Analysis:  
  
Review code diffs in commits  
Identify significant code changes and improvements  
  
e. Documentation Update Review:  
  
Scan for changes in documentation files  
Compile a list of documentation updates  
  
f. Breaking Change Detection:  
  
Analyze commits and code changes for potential breaking changes  
Flag any identified breaking changes  
  
g. Audience-Specific Content Generation:  
  
For Program Managers: Focus on feature updates, resolved issues, and high-level changes  
For Engineers: Emphasize code changes, breaking changes, and technical details  
For C-suite: Highlight strategic improvements, major feature additions, and business impact  
  
h. Release Notes Compilation:  
  
Organize gathered information into structured release notes for each audience  
Apply appropriate language and detail level for each audience  
  
i. Quality Check:  
  
Review generated release notes for accuracy and clarity  
Make necessary adjustments  
  
j. Distribution:  
  
Post audience-specific release notes to designated channels (e.g., Slack, email)  
  
  
Roles and Responsibilities:  
  
  
Slack Bot: Recognize release notes request and initiate process  
GitHub Integration: Access and analyze repository data  
Content Generation AI: Create audience-specific release notes  
Quality Assurance AI: Review and refine generated content  
  
  
Inputs and Outputs:  
  
Inputs:  
  
User request via Slack  
GitHub repository data (commits, issues, code changes, documentation)  
  
Outputs:  
  
Program Manager-focused release notes  
Engineer-focused release notes  
C-suite-focused release notes  
  
  
Key Performance Indicators (KPIs):  
  
  
Accuracy of release notes content  
Time from request to release notes distribution  
User satisfaction with audience-specific content  
  
  
Required Resources and Systems:  
  
  
Slack integration  
GitHub API access  
Natural Language Processing (NLP) AI for content analysis and generation  
Content distribution system (e.g., email, Slack channels)  
  
  
Exception Handling and Error Scenarios:  
  
  
GitHub API access failure: Retry connection or notify admin  
Insufficient data for meaningful release notes: Notify user and suggest manual review  
Ambiguous user request: Prompt user for clarification  
  
  
Integration Points:  
  
  
Slack: For receiving user requests and potentially distributing release notes  
GitHub: For accessing repository data  
Email system: For distributing release notes (if applicable)  
  
This process definition provides a comprehensive overview of the GitHub Release Notes Generation and Distribution process, detailing each step from the initial trigger to the final distribution of audience-specific release notes. The process leverages AI and automation to efficiently analyze repository data and generate tailored content, ensuring that Program Managers, Engineers, and C-suite executives receive relevant and understandable release information.