

This report contains the following 1 records:

CPOL #	Changes	Title	Inventors	Entered	Updated
1033439		[6 Strategic Pillars] intelligent ChatOps Engine - iCE - DevOps pipeline benchmarking with SkillOps	Srinjoy Chakraborty, Sudipta Debnath	07-Jun-2021	17-Jun-2021

Report generated: Thu, Jun 17 2021, 11:05:11 am

Reviewers need to review this record!

[6 Strategic Pillars] intelligent ChatOps Engine - iCE - DevOps pipeline benchmarking with SkillOps

CPOL #: 1033439

Origin: Cisco

Submission Status: Waiting

Submitted: 07-Jun-2021

Technology, Inc.

Modified: 17-Jun-2021

Submission Portfolio - Patentathon-BID

Portfolio Manager:

Henal Babariya

Address: BGL11/1

Email:

Henal Babariya (hbabariy@cisco.com)

Cessna Business Park, Kadubeesanhalli Varthur
Hobli, Sanjapur Marathalli ORR
Bangalore, Karnataka 560103

Phone: +91 80 4429 4998

BID Details

The contents of this submission and any additions or modifications thereto constitute Cisco confidential information and may be a privileged communication to or from one or more attorneys and/or supporting personnel for purposes of obtaining or facilitating legal advice and/or legal services.

Inventors:

Srinjoy Chakraborty

Type:

Phone: +91 80 4429 8668

Division: EBG

Manager: [Upendra Kale](#)

Site: BANGALORE

Dept: GSP.APJ.ADN.ADC.SW-IND

Info: [Cisco Directory](#)

Sudipta Debnath

Type:

Phone: +91 80 4429 0066

Division: EBG

Manager: [Ajay Srivastava](#)

Site: BANGALORE

Dept: CX GC.ADN.ADC.Proactive.Software-India

Info: [Cisco Directory](#)

Problem to Solve:

Software development is an ever-changing field in a world where every new software product release introduces new features and functionalities that directly affects the existing technical benchmarks. In Agile development to expedite the delivery process DevOps play a crucial role. The advantage of using DevOps while can be shortlisted as follows:

- Check quick performance after the addition of new features to the product
- Prepare environments with appropriate storage, memory etc. accordingly
- Set client expectations upfront to purchase hardware accordingly

In this DevOps wave, there is one piece which is not gaining enough attention:

- Lack of skilled engineers - which attracts cost and quality
- Inter collaboration in a streamlined manner - potentially impacts agile methodology
- Reusability of existing automation in a methodical way

At the heart of DevOps there are orchestration tools which allow us to bring all pieces of automation together and create a specific DevOps model. Usually, there is no benchmarking available for these automated code(s)/pipelines like which stages they should have, which languages can be used, tools they can use, or how much automations should be built. Typically, pipelines are grouped as build pipelines, test pipelines, deployment pipelines, CI pipelines, CD pipelines and nowadays cloud pipelines, too.

In proposed model we have tried to solve the problem, by categorizing code performing similar business objective. We have also tried to benchmark code(s) based on their performance and ease of use.

Possible Prior Art:

In the Proposed model, the concept of chat based devops is not novel. A lot of prior work already performed to bring up the concept of ChatOps Engine. The author(s) have discovered the following Prior Art that is related and provides justification on why we believe our invention is different and novel beyond Prior Art:

[Framework for performing load testing and profiling of services](#) - this solution is intended for load testing and profiling service using CICD pipelines, that analyzes profile data generated by a service and generates profile results associated with the service when the service operates at varying and/or increasing load capacities. This talks about categorizing and correlating data based on load testing. But it is focused on functional testing side of a device and not analyzing or benchmarking the CICD code.

Prior search also able to find [Method and apparatus for differential, bandwidth-efficient and storage-efficient backups](#) - The solution describes that based on the requirement, the process generates one or many content-driven signatures from one or many sources. But the innovation is focused only on differential backup solution and has no relevance to CICD pipeline.

We believe following components are novel in proposed solution:

- Chat based devops implementation with real-time observability in chat window claims to be novel.
- Deriving true objective for code through AI/ML backed analysis and automated code/script generation, increasing reusability of code, reducing learning curve, developmental time & effort also claims novelty.
- Analyzing each stage & task performed in the code and benchmarking it against available code of similar intent across organization, assisting in generating most optimized code solution of high quality and performance. This helps greatly in reducing the overall execution time and complexity for any automation model is also claim for benefits of novelty.
- Proposed model provide a solution which targets to pull code from repository by **username** of developer - being analyzed; that helps in adding score to visualize skill-rating of developer accurately. This idea claims to be novel since it eventually helps identifying areas of improvement and targeted training requirement for the developer directly linked with DevOps model.

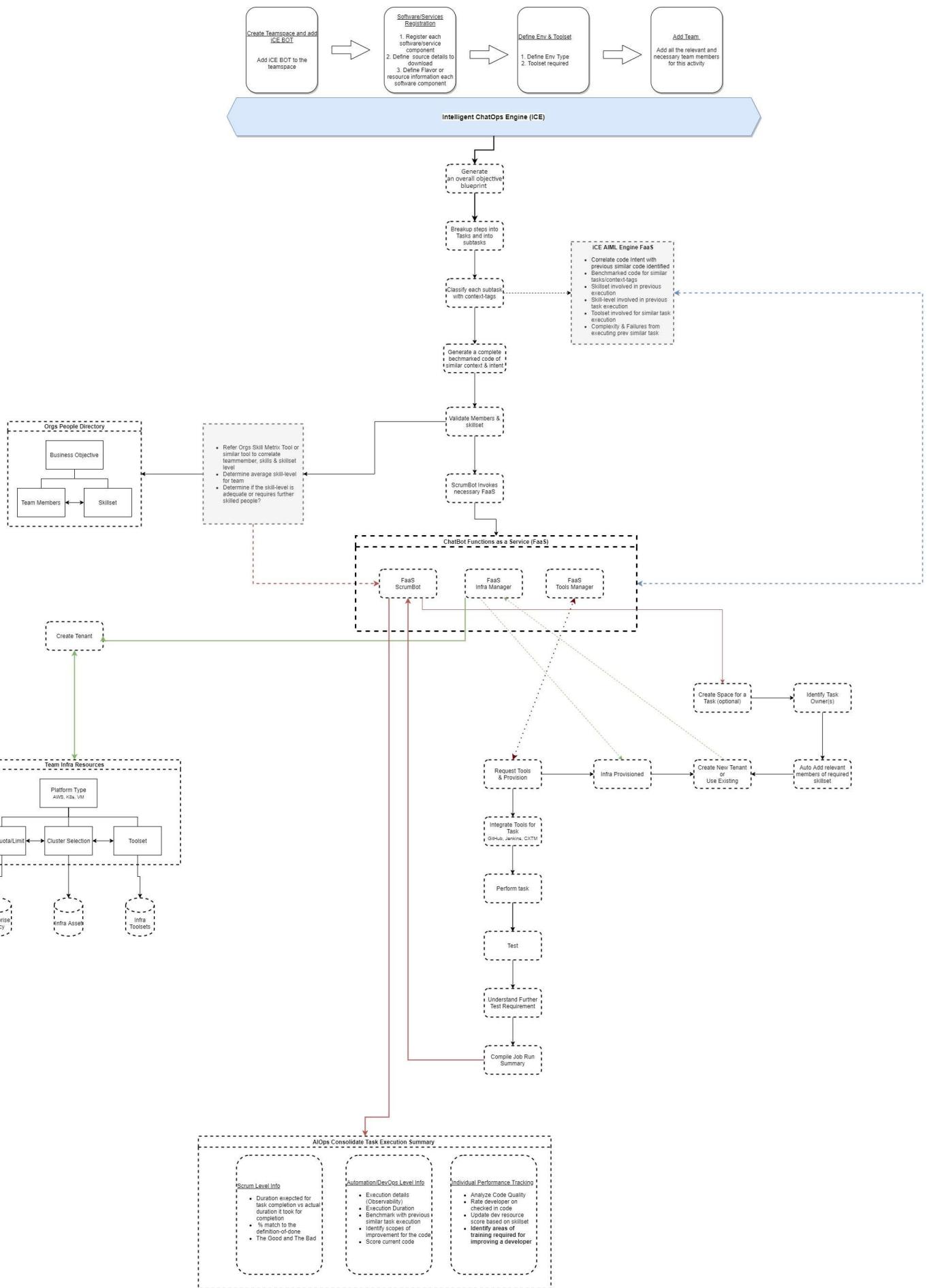
Brief Description:

Embodiments of the invention generally relate to a method and apparatus for CICD pipeline benchmarking, by enabling "ChatOps" based devops capabilities through AI backed Machine Learning. The apparatus is available as a ChatOps software-as-a-service and can be integrated through REST APIs to chat systems such as Webex/MS Teams, Slack etc.

Users can add the apparatus as a chat bot to a team space (group chat service containing more than one user). Once added the BOT activates its following Function-as-a-Service component modules

- Task Assistant & ScrumBot FaaS
- Infra Manager FaaS
- Tools Manager FaaS
- AIML Engine FaaS

High Level Design Workflow:



The functional support of Engine can be summarized as follows:

Infra Manager FaaS

- get resource allocation from the Enterprise Resource Pool (RPool) for the project and project Tenant management.

AIML Engine FaaS

- Analyzes the pipeline objective,
- Identifies all stages involved,
- Breaks down each stage into subtasks ,
- Categorizes each sub task based on task objective
- Identifies the current benchmarked script for that category.
- Uses objective driven correlation mechanism to identify each subtask involved in the CICD operation and generate a complete pipeline script dynamically combining these individual scripts.
- Categorizes user provided new script and tracks its execution time with other relevant parameters for comparison and benchmarking.
- Considers the code quality and re-usability factor during benchmarking
- Determines efficiency of new script w.r.t quality and execution time, Modifies the benchmark script for that specific task objective.

Tools Manager FaaS

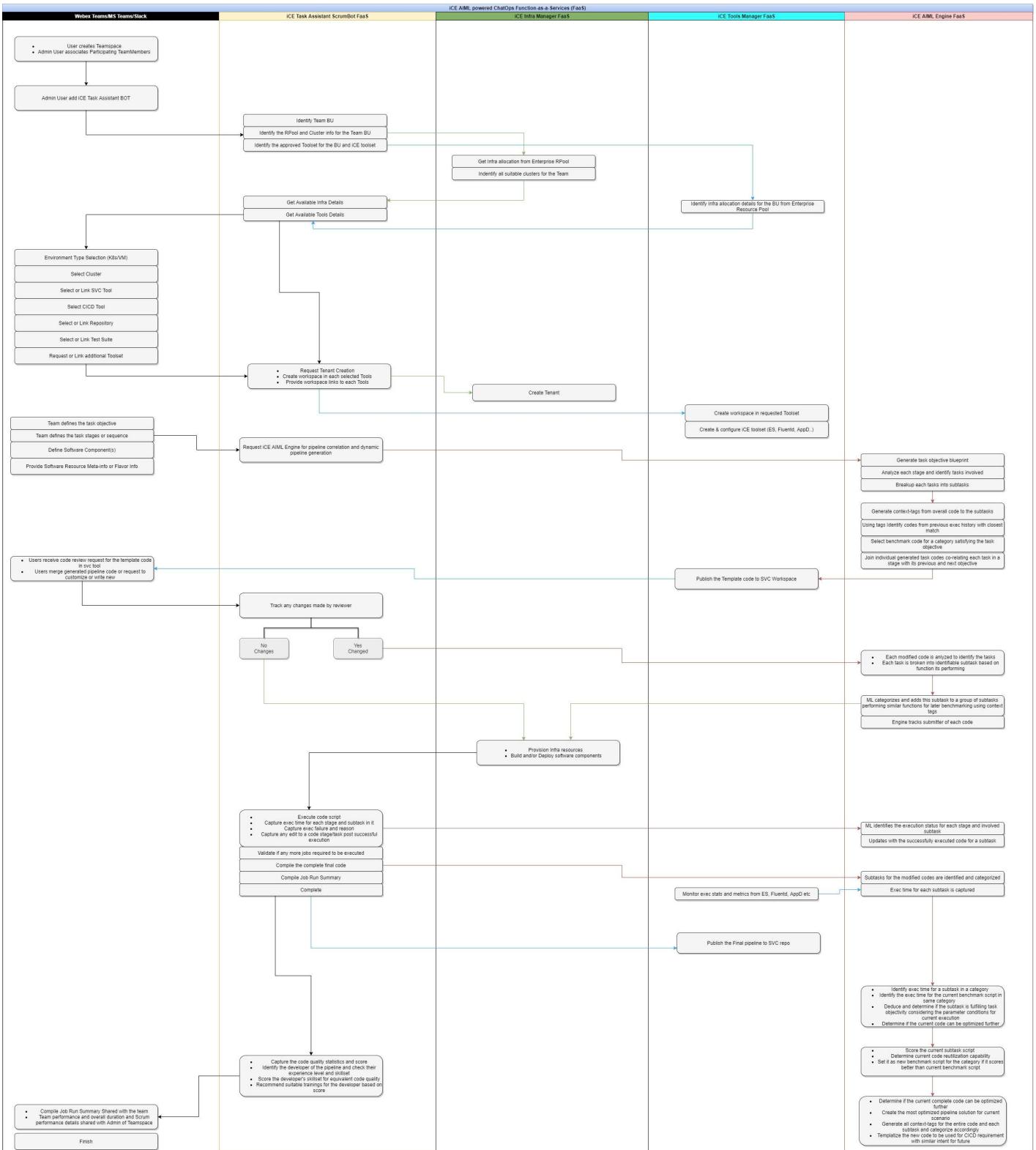
- create workspace in all the involved tools for the team and managing RBAC for access and governance.

Task Assistant & ScrumBot FaaS

- Tracks a requirement from start till its end.
- Identifies all user requests and invoke necessary FaaS subprocess,
- Tracks task progress while providing real-time visibility for each task to the requesters.
- Ensures overall objective for the CICD operation is satisfied.
- Tracks the developer submitting a code
- Tracks the code quality for the submitted code by AIML engine FaaS analysis
- rates the developer's skill-level on the submitted code benchmark score
- Manages overall score assigned depending on the skill-level of the developer from previous work on similar domain
- Assists in targeted training programs identification for the developer.
- Generated complete summary report of performed task upon its completion.

iCE Functionality in a Box – Benefits of Novelty

Based on the above understanding proposed *low level design* can be predicted as follows:



How the proposed model works (in-depth description):

- A group chat is created with multiple users collaborating for a task.
- The group chat admin(s) add the iCE Bot into the chat.
- iCE Bot initializes the **Task Assistant & ScrumBot FaaS**
- Task Assistant FaaS invokes the other sub-component
 - iCE AIML Engine FaaS** is created to connect to the AIML backend of iCE for the task. This is the most important service component of iCE.
 - iCE Infra Manager FaaS** is invoked to run discovery on the available Enterprise resource pools.
 - iCE Tools Manager FaaS** is invoked to run discovery on the available toolset in the Enterprise. It also enables the toolsets used by iCE & available configurations for ElasticSearch, New Relic, Splunk etc. which will be integrated to entire pipeline functions for metrics generation and measurement and general monitoring.
- The users define a task objective. It can be any task or sequence of task(s).
- iCE Task Assistant will provide user choice for automated pipeline script generation or allow a custom code to be defined by the user(s) for the defined task. The user can provide a GitHub repo for the script.
- We will expand further on the steps involved in iCE Auto-generated Scripts vs Custom code developed by users. We believe following are the novelty for this innovation.

For a Custom code developed by User(s) iCE will:

iCE Infra Manager FaaS will be involved in temporary tenant creation and tenant privilege management for the team (or allow defining existing) and provision necessary temporary environment resources for the task.

iCE Tools Manager FaaS will:

- Configure access to the repo and provide link to the script. It will also capture the information of the user(s) submitter of this code.
- Configure the script for execution in Jenkins or similar automation platform available in the Enterprise
- Configure ElasticSearch, Fluentd for capturing the job execution logs, configure AppD, Prometheus with Grafana or similar toolset for generating & capturing performance metrics and behavior, and also for general observability of assigned infra and deployed tools in the environment.
- Execute the script

iCE AIML Engine FaaS will do following tasks:

- Observe execution of the code and identify the steps/stages/modules involved in the code. Identify each stage as a task.
- Co-relate a task with the previous & next task and their nature & relationships and generate task intent
- Generate an overall objective blueprint by combining the task intents generated from each task.
- Help reduce/remove unnecessary steps for attaining the overall objective.
- Parse each task and identify the subtasks in them. Subtasks are functions performed in that task.
- For each subtask understand the input parameters (variables & datatypes) and expected output (datatype details).
- Understand if a subtask function is used once or used in more than one scenario.
- Capture the execution time for each subtask during runtime using iCE Tools Manager FaaS from the toolsets like ES, Fleuntd, AppD for understanding service quality and performance.
- From previous execution history and ML based code classification & categorization, identify codes of similar intent, by matching codes with similar input and output datatype. Once identified tag this subtask code to this identified category. The subtask code can be tagged with more than one category (context-tag)
- A weight factor is assigned to the code block depending on the relevance for the code in this context. Similar weight factor is generated for all other context-tags where this code is applicable.
- Check the present benchmark code for the identified category and execute it with similar dataset and execution scenario. Clock its execution time and compare with current code. A weight factor is assigned based on its relevance and performance.
- Run test for both the current subtask code as well as the current benchmark code in identified category with multiple datasets to determine variation of execution time between these code set.
Refine the weight factor of current code based on its execution time & performance.
- AIML engine will also determine how easily the current code can be reused for more than one usecase, environment etc. It will assign a weight factor same.
- An overall score is generated by the AIML engine combining the performance + reusability weight factors and context applicable.
- This score will be matched against the current benchmark code from that category. If the score is better than the current benchmark code, it will be set as the new benchmark code.
- Alternatively, the current code will be replaced by the benchmark code from that category.
- AIML engine will perform the same exercise with all subtasks and tasks following the guidelines identified from the blueprint generated at the beginning of analysis.
- The latest complete generated code will be executed and compared with the original code provided by the user(s). The performance & execution improvement metrics along with statistics around execution will be shared with the iCE Task Manager for the team(s) to review.

If the latest generated code is approved by the user(s), iCE will store the entire code blueprint as a context and set more than one context-tags. This context-tags will include environment details (VM, k8s etc), services details, operation performed etc. These context-tags help iCE to determine scenario wherever this code can be applicable/reusable. This helps in auto-generation of code for a user with similar intent in a different use case.

For an iCE Auto-generated code

- iCE will understand the specified task(s) and their objective and create an objective blueprint based on intent. The objective blueprint accuracy will depend on how accurately the user defines the tasks and their steps involved.
- It will identify the sequence of stages involved for attaining overall objective blueprint.
- It will classify each subtask as a task and identify subtask necessary per task.
- iCE will generate context-tags by parsing tasks and subtasks.
- It will combine all the context-tags and find out the code from previous execution history which has the closest match.
- If none found, the AIML engine will identify the closest matching code per subtasks through their tags and generate a template code following the guidelines identified from the blueprint.
- The generated template code will be published in a Git repo using iCE Tools Manager FaaS as a pull request.
- The iCE Tools Manager FaaS also updates the AIML engine for any edit or changes from the user to the template code.
- iCE AIML Engine FaaS understands the changes made and ML edits the task objective blueprint based on the changes at appropriate sections.
- It also benchmarks the current edited code from the user and identifies if there is a better benchmark code available for the specific task/subtask which can improve performance of the overall execution time.

Target Users:

DevOps Professionals, Architects, Program Management, Delivery Management - All stakeholders involved in Service Delivery process.

Targeted model is an easy solution which can be integrated with any product module. With a non-UI based approach it allows to make the engine easy to implement. Delivery program where DevOps is mandate to use as part of Agile methodology - insertion of this model is promise to optimize DevOps infrastructure building effort, skill matrix maintenance and make ease Program Management to be much easier.

***** Response to review comments *****

Please address the following comments:

Q1. "It is unclear what the novelty of the submission is, authors need to clean up the submission and identify the specific novelty they are proposing that is beyond plugging ChatBots into a CI/CD pipeline. While there are specific callouts of novelty in their submission they are unreadable and we would expect these to be called out as textual descriptions in the submission."

Novelty of the submission is as follows:

- Chat based devops implementation with real-time devops observability in chat window claims to be novel.

This is ChatDevOps integrated with AI/ML but you can visualize the real status of devOps in chat window.

- Deriving true objective for code through AI & ML backed analysis and automated code/script generation, increasing reusability of code, reducing learning curve, developmental time & effort also claims novelty.
- Analyzing each stage & task performed in the code and benchmarking it against available code of similar intent across organization, assisting in generating most optimized code solution of high quality and performance. This helps greatly in reducing the overall execution time and complexity for any automation model is also

claim for benefits of novelty.

- Proposed model provide a solution which targets to pull code from repository by **username** of developer - being analyzed; that helps in adding score to visualize skill-rating of developer accurately. This idea claims to be novel since it eventually helps identifying areas of improvement and targeted training requirement for the developer directly linked with DevOps model.

Q2. "They need to spell out how to do each of the following:

Q2a. * Derive the true objective of the pipeline (what algorithms are used and how does the ML model do this)

Ans. The solution uses tagging mechanism to identify a code block. The tag can be for one or more objective also identified by a weight factor (score) generated for the code block for that objective depending on relevance and suitability. Perceptron training with weight factor assignment improves tagging for the ML engine.

Note: High Level Design workflow and low level design workflow got elaborated - however reviewer need to understand we have explained how the logic actually works, not how the ML engine suppose to be designed, how ML will handle that logic is out of scope here- still it is textually described in "[How the proposed model works \(in-depth description\)](#)" section. Developing the ML algorithm - need to be considered separately and subject to data science operation.

Q2b. * Analyzing risk of CI/CD steps. Really, there should be known risk since CI/CD typically means automated testing upfront so you know what to expect in production. Rollouts are done using green/blue or other rolling model, so it's easy to backout. What is being done here in this invention that makes this better and how is that being done?

Ans. This solution allows the user to define any task and its sequence of execution. The system can recommend additional tasks based on previous executions of similar task and contextual analysis, but the major objective for the system would be to keep the code in alignment with overall task objective blueprint. What that means the system does not necessarily check if the user has defined a rollback functions while implementing a rolling blue/green update/canary deployment.

But it will identify the most optimized code for a deployment backed with proper stats and analysis metrics. It will also auto generate a most optimized code increasing the code re-usability.

Q2c. * How is code analyzed to determine skill rating?

Ans. The solution can track the username from the git push. AIML analysis of the code will generate a score for the code which would highlight the code quality. Based on that the system can assign a score to the developer and highlight the skills & proficiency he demonstrated for this code. Every time a new code is pushed by the same developer, the system will run fresh analysis and generate a score for that code and adjust the mean score rating of the developer accordingly.

It will also identify the weakness of the developer in a particular domain and suggest suitable trainings.

Q3. Having a bot that can remind people of deliverables, help identify blockers, etc. in and of itself is not new. Jira has a bot. We're using Webex for some CI/CD work in a project I'm working on to help track builds and merge requests. So I want to know more about the AI/ML how to get a good idea of the novelty here."

Ans. Idea is not for a bot implementation, we are trying to define an end-to-end DevOps process which can be controlled through chat window, where one can have complete observability and control of dynamically generated DevOps model through the workspace - this could be JIRA bot, webex, MS Teams, Slack etc. Underneath of this chat mechanism various DevOps tools/frameworks will interact each other to serve the end goal. We are not only proposing for build and merge request - it is beyond that concept where Team skill management to QA certification and Program Management to End Delivery everything can be dynamically controlled without much human intervention.

📁 Portfolio:

[Patentathon-BID](#)

📄 Submission Related Documents:

Type	Document	Date Entered	Size
No Documents To Show.			

Review Information

All notes, comments and other information herein constitute Cisco confidential information and may be privileged communications to or from one or more attorneys and/or supporting personnel for purposes of obtaining or facilitating legal advice and/or legal services.

⌚ Progress:

Review Group	Assigned Reviewers	Status
Patentathon-BID	The Patentathon-BID Committee	<i>Waiting for review</i>

This submission is currently being **reviewed**.

☒ Actions:

Approval action: returned for comments on 15-Jun-2021 13:17:00

Inventors:

Please address the following comments:

"It is unclear what the novelty of the submission is, authors need to clean up the submission and identify the specific novelty they are proposing that is beyond plugging ChatBots into a CI/CD pipeline. While there are specific callouts of novelty in their submission they are unreadable and we would expect these to be called out as textual descriptions in the submission."

"They need to spell out how to do each of the following:

- * Derive the true objective of the pipeline (what algorithms are used and how does the ML model do this)
- * Analyzing risk of CI/CD steps. Really, there should be known risk since CI/CD typically means automated testing upfront so you know what to expect in production. Rollouts are done using green/blue or other rolling model, so it's easy to backout. What is being done here in this invention that makes this better and how is that being done?
- * How is code analyzed to determine skill rating?

Having a bot that can remind people of deliverables, help identify blockers, etc. in and of itself is not new. Jira has a bot. We're using Webex for some CI/CD work in a project I'm working on to help track builds and merge requests. So I want to know more about the AI/ML how to get a good idea of the novelty here."

on 17-Jun-2021 10:31:00

Committee Action History:

Action Date	Action	Committee								
17-Jun-2021	Resubmitted	Patentathon-BID								
15-Jun-2021	Returned for Comments	Patentathon-BID								
<table border="1"><tr><td>Date Assigned to Committee:</td><td>07-Jun-2021</td></tr><tr><td>Days to First Action:</td><td>8</td></tr><tr><td>Days in Committee Awaiting Final Action:</td><td>7</td></tr><tr><td>Inventor Wait Time:</td><td>2</td></tr></table>			Date Assigned to Committee:	07-Jun-2021	Days to First Action:	8	Days in Committee Awaiting Final Action:	7	Inventor Wait Time:	2
Date Assigned to Committee:	07-Jun-2021									
Days to First Action:	8									
Days in Committee Awaiting Final Action:	7									
Inventor Wait Time:	2									

This data was generated on Thu Jun 17 11:05:13 PDT 2021 and is available on-line from **CPOL** at <https://cpol.cisco.com/cpol/patent.cgi>

CISCO CONFIDENTIAL

[CISCO CONFIDENTIAL](#) A printed version of this page is an uncontrolled document.