El Estándar IEEE 730

Guía para la planificación y ejecución de un plan de aseguramiento de la calidad del software (SQA)

Autores: GRUPO N

June 19, 2024

Integrantes

- Casquino Ticona Jorge Romel
- Condori García Paul Wenceslao
- García Jiménez Angel Raul
- Phocco Soncco Yeni Rosmirian

ESTANDAR IEEE 730

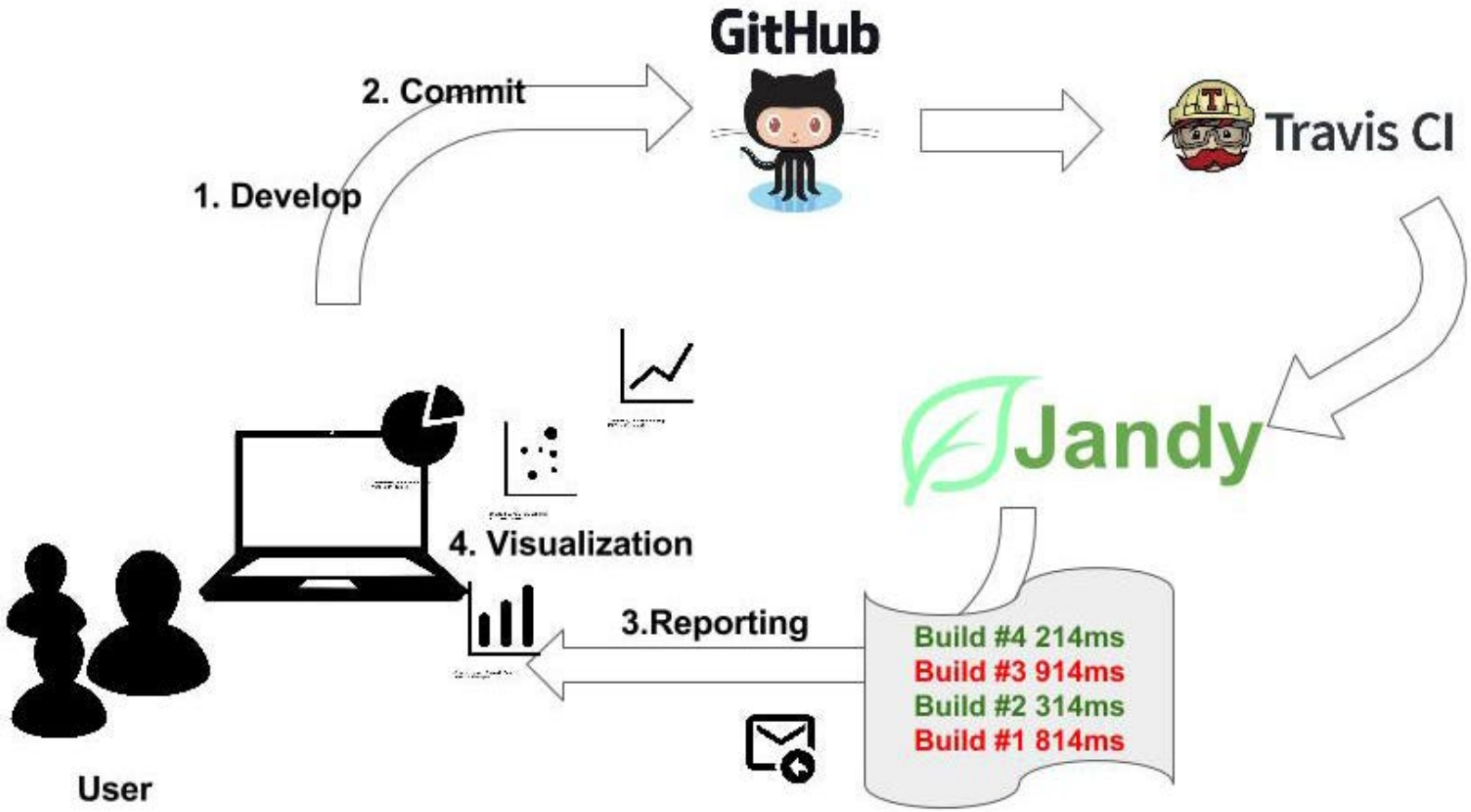
proporciona directrices para la planificación y ejecución de un plan de aseguramiento de la calidad del software (SQA).

- Planificación del SQA: Define los objetivos de calidad, responsabilidades, y recursos necesarios para asegurar la calidad del software.
- Revisión y Auditoría: Realización de revisiones y auditorías independientes para verificar que los procesos y productos cumplen con los estándares establecidos.
- Pruebas y Validación: Implementación de pruebas de software para asegurar que el producto final cumple con los requisitos y especificaciones.
- Informes de Calidad: Documentación y reporte del estado de calidad del software durante todo el ciclo de vida del desarrollo

Cómo Travis CI y GitHub Ayudan al Cumplimiento del IEEE 730

- Travis CI permite configurar pipelines de CI que ejecutan automáticamente pruebas unitarias, de integración y funcionales cada vez que se realiza un commit o se crea una pull request en GitHub.
- Cumplimiento IEEE 730: Automatizar las pruebas garantiza que los cambios en el código son evaluados consistentemente, reduciendo la probabilidad de introducir defectos y mejorando la fiabilidad del software.

Cómo Travis CI y GitHub Ayudan al Cumplimiento del IEEE 730



Travis CI

- Activar Travis CI en el Repositorio: En la interfaz de Travis CI, activa Travis CI para el repositorio que creaste en GitHub.
- Configurar el Archivo .travis.yml: Crea un archivo llamado .travis.yml en la raíz de tu repositorio. Este archivo define cómo Travis CI debe construir y probar tu proyecto.

Ejemplo de .travis.yml para un Proyecto de Python

```
language: python
python:
  - "3.8"
install:
  - pip install -r requirements.txt
  - pip install flake8
  - pip install pytest
  - pip install coverage
  - pip install pylint
  - pip install mypy
  - pip install safety
script:
  - flake8 .  # Análisis de estilo de código
  - pylint my_module/  # Análisis de calidad de código
  - mypy my_module/  # Comprobación de tipos
  - coverage run -m pytest  # Ejecución de pruebas unitarias con cobertura
  - coverage report  # Informe de cobertura
  - pytest integration_tests/  # Ejecución de pruebas de integración
  - safety check  # Análisis de dependencias para vulnerabilidades
notifications:
  email:
    recipients:
      - tu_email@example.com
    on_success: never
    on_failure: always
```

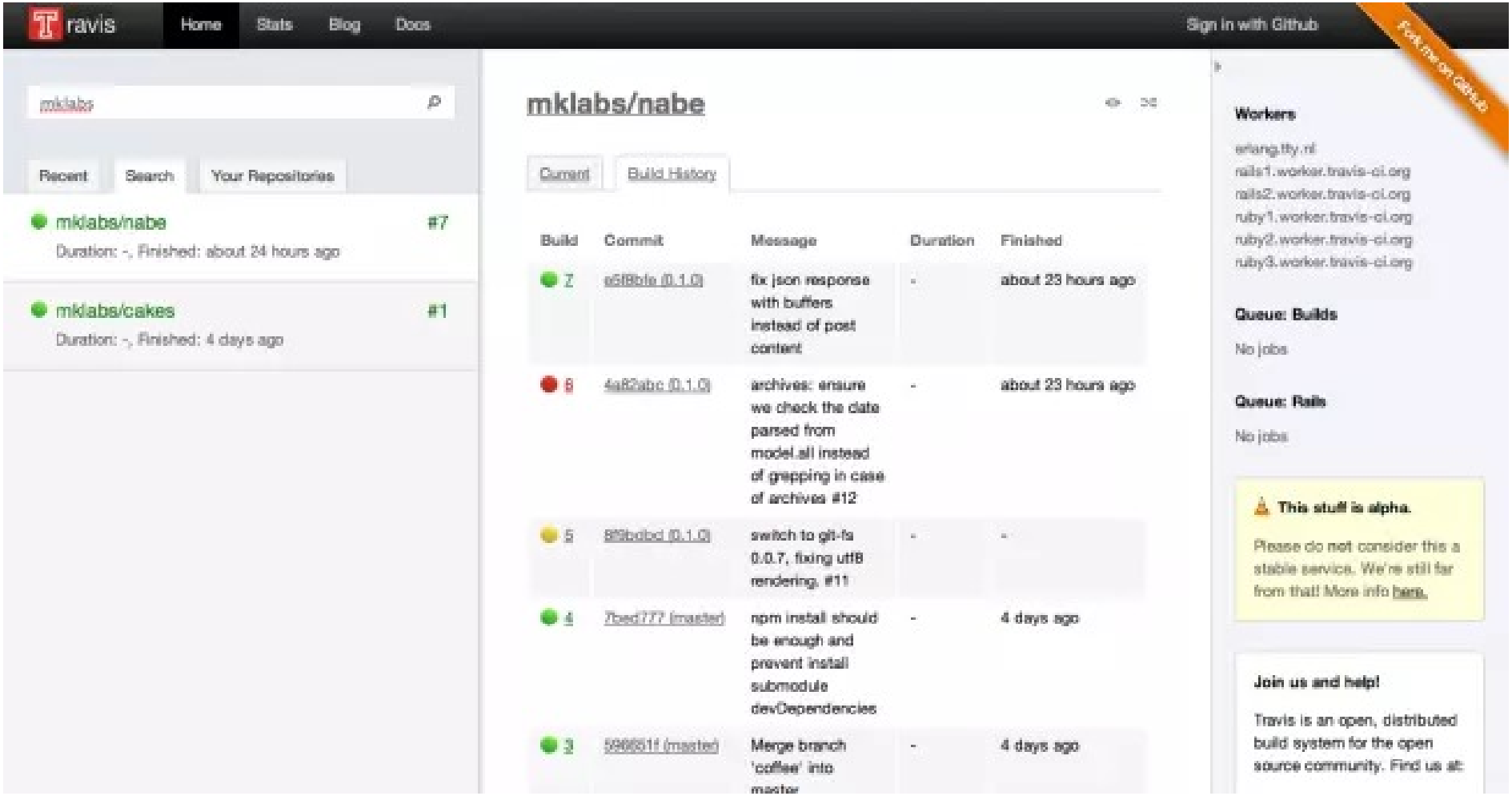
Generación de Informes de Pruebas

- Compilación del Proyecto: Travis CI automáticamente clona el repositorio, instala las dependencias necesarias y compila el proyecto según las instrucciones definidas en el archivo .travis.yml.
- Ejecución de Pruebas: Travis CI ejecuta las pruebas automatizadas que has definido. Esto incluye pruebas unitarias, de integración y funcionales. Los resultados de las pruebas se registran y se muestran en la interfaz de Travis CI.

Automatizar el Proceso de Evaluación

- Generación de Informes de Pruebas: Travis CI genera informes sobre el estado de las pruebas, indicando si han pasado o fallado. Estos informes son esenciales para el aseguramiento de la calidad según IEEE 730.
- Ejecuta pruebas automatizadas y genera informes.

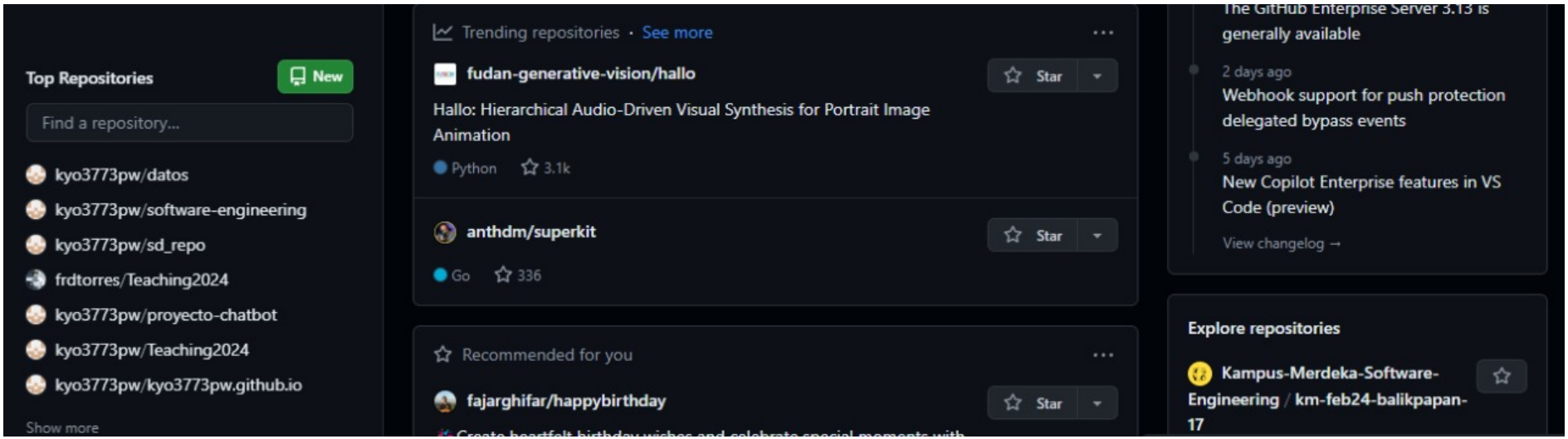
Automatizar el Proceso de Evaluación



Evaluar Resultados y Realizar Revisión de Código en GitHub

- Revisión de Resultados de Pruebas en Travis CI: Revisa los resultados de las pruebas en la interfaz de Travis CI. Si alguna prueba falla, se puede ver un registro detallado del error para diagnosticar y solucionar el problema.
- Revisión de Código en GitHub: Utiliza pull requests (PR) para realizar revisiones de código colaborativas. En cada PR, Travis CI ejecuta las pruebas automatizadas y muestra el estado de la construcción (build status) directamente en GitHub. Los revisores de código pueden ver los resultados de Travis CI y aprobar o solicitar cambios basados en estos resultados.

Evaluar Resultados y Realizar Revisión de Código en GitHub



Despliegue Continuo (Opcional)

- Configura despliegue automático si todas las pruebas pasan.
- Añade pasos de despliegue en el archivo .travis.yml.