Cómo Implementar Scrum en GitHub

Introducción

Scrum es una metodología ágil utilizada para gestionar proyectos de desarrollo de software. GitHub, una plataforma de desarrollo colaborativo, ofrece diversas herramientas que facilitan la implementación de Scrum. A continuación, se detalla cómo llevar a cabo esta metodología en GitHub.

1. Crear un Nuevo Repositorio

Inicia sesión en tu cuenta de GitHub.

Haz clic en el botón "New" para crear un nuevo repositorio.

Asigna un nombre al repositorio y agrega una descripción opcional.

Selecciona las opciones "Public" o "Private" según tus necesidades.

Haz clic en "Create repository".

2. Configurar un Proyecto (Project)

Dentro del repositorio, ve a la pestaña "Projects".

Haz clic en "New project" y asigna un nombre al proyecto.

Selecciona "Kanban (default)" para un tablero Kanban básico.

Haz clic en "Create project".

3. Crear Columnas del Tablero Scrum

Dentro del proyecto, verás tres columnas predeterminadas: "To do", "In progress" y "Done".

Puedes agregar nuevas columnas o modificar las existentes según tu flujo de trabajo Scrum, como "Backlog", "Sprint", "Review", etc.

4. Crear Issues (Historias de Usuario)

Ve a la pestaña "Issues" del repositorio.

Haz clic en "New issue" para crear una nueva historia de usuario.

Asigna un título descriptivo y agrega una descripción detallada.

Etiqueta la issue según corresponda (por ejemplo, "enhancement", "bug", "story", etc.).

Asigna la issue a un miembro del equipo si es necesario.

Haz clic en "Submit new issue".

5. Planificación del Sprint

Selecciona las issues que formarán parte del sprint y muévelas a la columna correspondiente en el tablero Kanban.

Define el tiempo del sprint (generalmente de 1 a 4 semanas).

Realiza una reunión de planificación del sprint con todo el equipo para discutir las tareas.

6. Seguimiento del Progreso

Durante el sprint, los miembros del equipo moverán las issues entre las columnas "To do", "In progress", "Review", y "Done" según el estado de cada tarea.

Realiza reuniones diarias (daily stand-ups) para discutir el progreso y los obstáculos.

7. Revisión y Retrospectiva del Sprint

Al finalizar el sprint, realiza una reunión de revisión para evaluar lo que se ha completado.

Discute las tareas pendientes y replanifícalas si es necesario.

Realiza una retrospectiva del sprint para discutir qué se puede mejorar en el próximo ciclo.

8. Uso de GitHub Actions (Opcional)

Configura GitHub Actions para automatizar tareas como despliegues, pruebas y otros procesos repetitivos.

Crea un archivo .yml en la carpeta .github/workflows y define los workflows según tus necesidades.