**SonarQube Tool**

**Prework**

* Identificar criterios de calidad en el software.
* Analizar los criterios de análisis estático del código.
* Ejecutar un análisis estático con SonarQube.
* Analizar los 10 riesgos de seguridad más frecuentes en aplicaciones web (OWASP Top Ten).

**Criterios de calidad de software:**

* **Productividad:** procesamiento rápido de los datos.
* **Toma de decisiones:** Los PMs y los CEOs pueden ordenar los objetivos, las prioridades y evitar resoluciones impulsivas.
* **Clasificación de datos:** Puedes utilizar las métricas para reducir los malentendidos y las ambigüedades en proyectos complejos.
* **Prioridades:** Con las métricas, los gestores ya no tendrán dificultades a la hora de seguir, identificar o priorizar los problemas del proyecto.
* **Gestión del progreso:** ¿Se está cumpliendo el calendario del proyecto? ¿Va todo bien?
* **Estrategia de gestión:** Hay algunos riesgos que hay que estimar, controlar y priorizar al instante. Las métricas ayudan a gestionar estos problemas y a evitar futuras soluciones costosas.

**Aspectos de la calidad del software**

* Fiabilidad
* Capacidad de mantenimiento
* Testabilidad
* Portabilidad
* Reutilización

**Análisis Estático de Código**

Análisis de software que se realiza sin ejecutar el programa. Es un proceso de detección de errores y defectos en el código fuente de un software.

**Tareas que resuelve el software de análisis estático de código**

* Detección de errores en los programas.
* Recomendaciones sobre el formato del código.
* Cálculo de métricas.

Las herramientas de análisis estático permiten detectar rápidamente muchos errores en la fase de codificación, lo que reduce significativamente el coste de desarrollo de todo el proyecto.

**Ventajas del análisis estático de código**

* Cobertura total del código. Los analizadores estáticos comprueban incluso fragmentos de código que se controlan muy raramente.
* El análisis estático no depende del compilador que esté utilizando ni del entorno en el que se ejecutará el programa compilado. Le permite encontrar errores ocultos que pueden revelarse sólo unos años después de su creación.
* Puede detectar fácil y rápidamente los errores de impresión y las consecuencias del uso de Copy-Paste.

**Desventajas del análisis estático de código**

* El análisis estático suele ser deficiente en cuanto al diagnóstico de fugas de memoria y errores de concurrencia.
* Una herramienta de análisis estático le advierte de los fragmentos extraños. Esto significa que el código puede ser en realidad bastante correcto; a estos informes los llamamos "falsos positivos".

**Los 10 Principales Riesgos de Seguridad de las Aplicaciones Web**

**Control de acceso roto:** El control de acceso no hace cumplir la política de manera que los usuarios no puedan actuar fuera de los permisos previstos.

**Fallos criptográficos:** No hay protección de los datos en tránsito y en reposo.

**Inyección:** el usuario aplica código malicioso.

**Diseño inseguro:** los controles de seguridad necesarios nunca se crearon para defenderse de ataques específicos.

**Desconfiguración de la seguridad:** no hay un proceso concertado y repetible de configuración de la seguridad de las aplicaciones, los sistemas corren un mayor riesgo.

**Componentes vulnerables y obsoletos:**

Es probable que nuestro software sea vulnerable:

* Si no conoce las versiones de todos los componentes que
* Si el software es vulnerable, no está soportado o no está actualizado.
* Si no escanea las vulnerabilidades regularmente y se suscribe a los boletines de seguridad relacionados con los componentes que utiliza.
* Si no corrige o actualiza la plataforma subyacente, los marcos de trabajo y las dependencias de manera oportuna y basada en el riesgo.
* Si los desarrolladores de software no comprueban la compatibilidad de las librerías actualizadas o parcheadas.
* Si no se aseguran las configuraciones de los componentes.

**Fallos de identificación y autenticación:** La confirmación de la identidad del usuario, la autenticación y la gestión de la sesión no son suficientes para protegerse de los ataques relacionados con la autenticación.

**Fallos de integridad del software y de los datos:** Los fallos de integridad del software y de los datos están relacionados con el código y la infraestructura que no protegen contra las violaciones de la integridad.

**Fallos en el registro y la supervisión de la seguridad:** El registro, la detección, la supervisión y la respuesta activa no son suficientes. Es vulnerable a la fuga de información al hacer que los eventos de registro y alerta sean visibles para un usuario o un atacante.

**Falsificación de solicitudes del lado del servidor (SSRF):** Se producen cuando una aplicación web obtiene un recurso remoto sin validar la URL proporcionada por el usuario. Permite a un atacante coaccionar a la aplicación para que envíe una solicitud manipulada a un destino inesperado, incluso cuando está protegida por un cortafuegos, una VPN u otro tipo de lista de control de acceso a la red (ACL).

**SonarQube:** Ejecuta diferentes escáneres que realizan análisis de código, una base de datos para almacenar los resultados y un panel de control web para ver los resultados.

**Work**

* Identificar criterios de calidad en el software
* Analizar los criterios de análisis estático del código
* Ejecutar un análisis estático con SonarQube
* Analizar los 10 riesgos de seguridad más frecuentes en aplicaciones web (OWASP Top Ten)

**Postwork**

* Desarrollar interfaz gráfica para añadir entrevistadores.
* Conectar la interfaz gráfica con nuestra API.
* Ejecutar un análisis con SonarQube.