Gestión de components de terceros como elemento clave del **SSDLC**

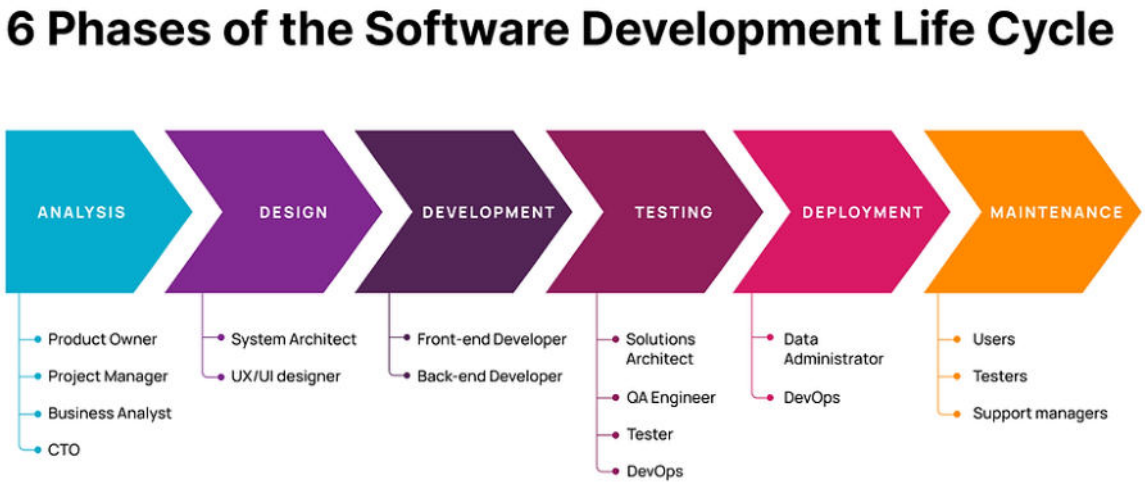
Glenda Suarez Cabrera (CISA, CISM, CISSP)

Director IT Quality, Risk & Compliance (QRC) and Security at Pitcher A.G.

Agenda

1. Gestión Clásica del Ciclo de Vida del Desarrollo de Software (SDLC)
2. ¿Cómo dar el salto de SDLC a **S**SDLC?
3. Secure Software Development Framework (SSDF) Version 1.1 (**NIST SP 800-218)**
4. Estadísticas-Riesgos de componentes
5. ¿Cómo gestionar de manera efectiva los riesgos asociados a los componentes de terceros como parte del proceso SSDLC?
6. Consideraciones Importantes

1. Gestión Clásica del Ciclo de Vida del Desarrollo de Software (SDLC)



Productos de alta calidad funcional

1.1. ¿Pero, **Calidad** no debería incluir el tema **Seguridad**?

Funcionalidad

Eficiencia

Confiabilidad

Mantenibilidad

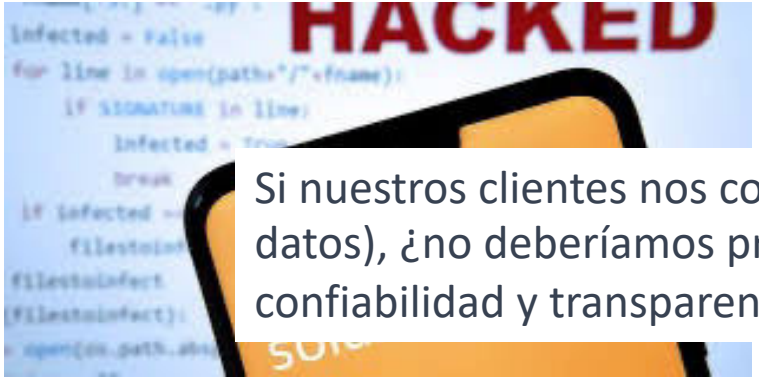
Usabilidad

Portabilidad

Escalabilidad

Seguridad

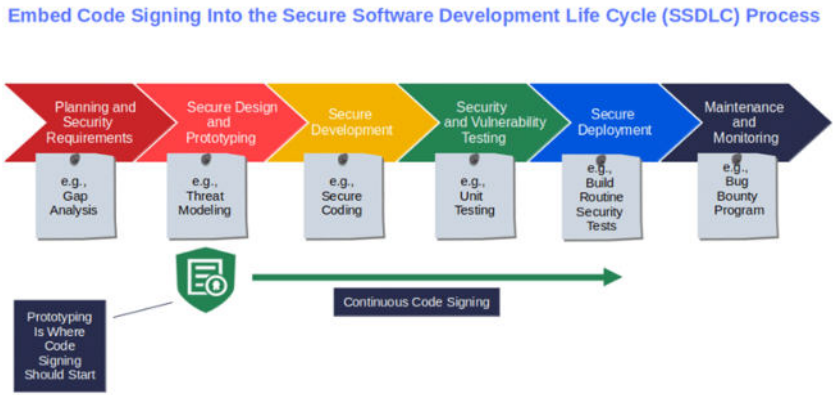
1.2. La Seguridad pasa de ser un “Add-on” a un “ Must-have”



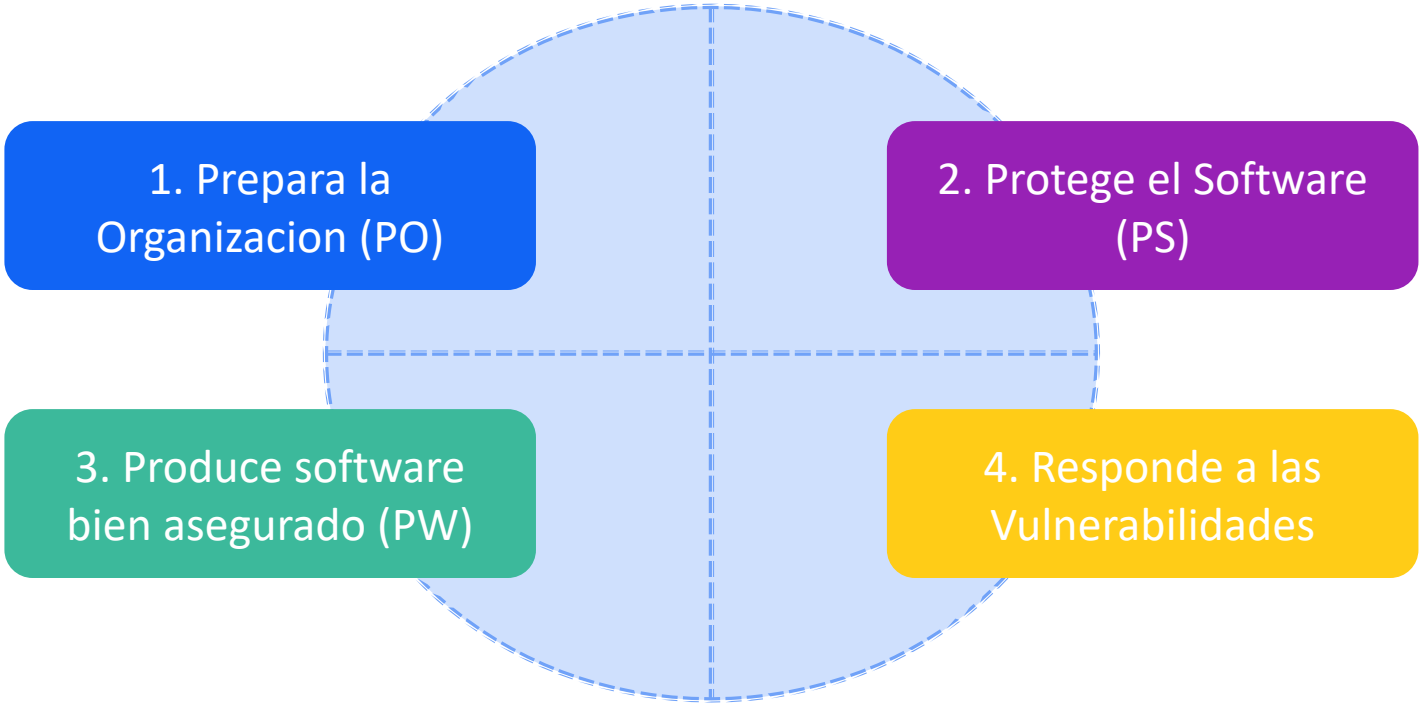
Si nuestros clientes nos confían su activo más importante (sus datos), ¿no deberíamos proporcionar un nivel igual de seguridad, confiabilidad y transparencia en los productos que les ofrecemos?

¿2. Cómo dar el salto de SDLC a SSDLC?

- ¿Como empresa, estamos realmente listos?
- ¿Tenemos el compromiso de nuestros líderes?
- ¿Conocemos nuestos objetivos de seguridad? ¿Conocemos nuestros requisitos para el producto?
- ¿Sabemos qué prácticas adoptar a lo largo del SDLC?
- ¿Sabemos cómo medir el nivel de seguridad según nuestro apetito de riesgo y compromisos e.g. OKR, KRI?



3. Secure Software Development Framework (SSDF) Version 1.1 (NIST **NIST SP 800-218)**

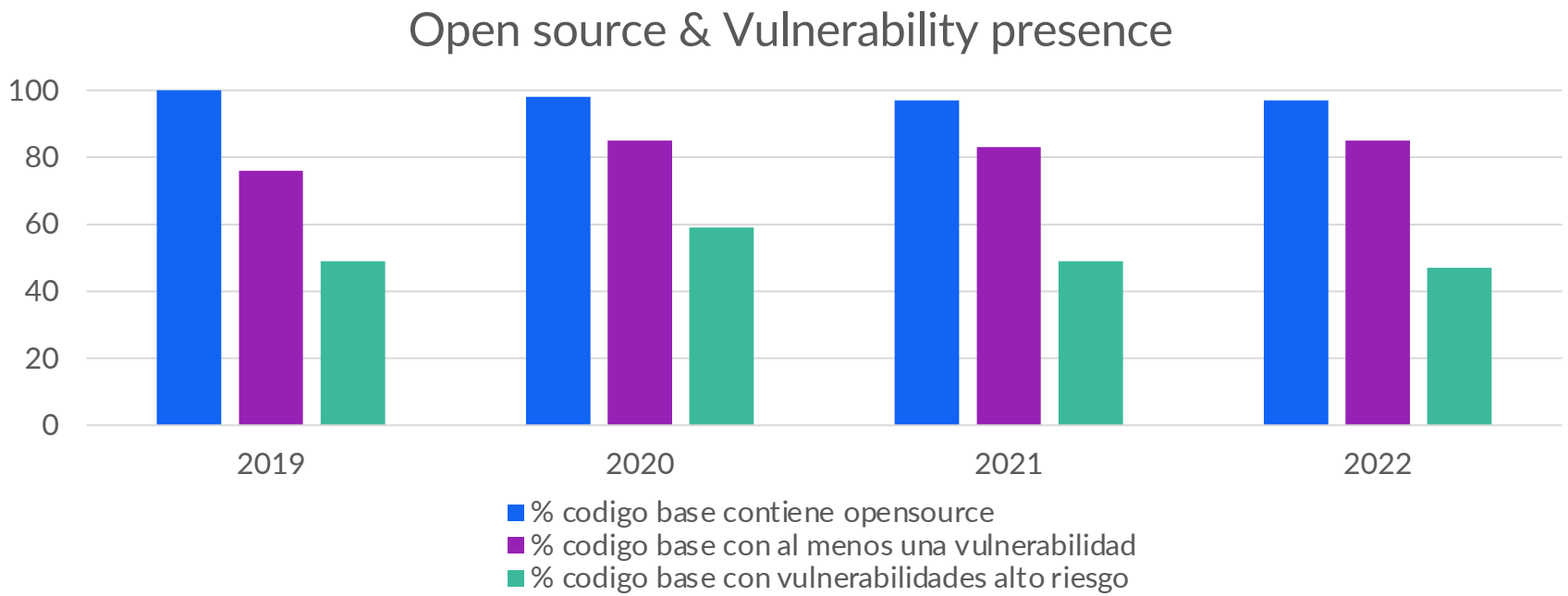


4. Estadísticas – Riesgos de componentes

De los 1,703 conjuntos de códigos escaneados en 2022

76% del código total es OSS

4.1 Estadísticas – Riesgos de componentes



4.2. Estísticas ISACA

¿Cómo gestionan los riesgos de componentes de terceros las empresas?

5. ¿Cómo gestionar de manera efectiva los riesgos asociados a los componentes de terceros como parte del proceso SSDLC?

1. Realiza un estudio de Propósito, Viabilidad y Mantenibilidad antes de introducir un componente de terceros.

2. Somete los componentes de software de terceros a una Evaluación de Riesgos de Terceros (BIA y TPRM).

3. Realiza un Análisis de Composición de Software (SCA) y SBOM para fortalecer la seguridad.

1. Realiza un estudio de Propósito, Viabilidad y Mantenibilidad antes de introducir un componente de terceros.

- **Definir el Propósito:** Evaluar múltiples componentes para determinar cuál satisface mejor la funcionalidad requerida. Documentar los pros y contras de al menos tres componentes preferidos.
- **Determinar la Viabilidad:** Evaluar el costo de desarrollar su propia funcionalidad frente a un componente de terceros. Evaluar el costo de los esfuerzos de integración, así como los recursos para mantenerlo.
- **Analysar la Mantenibilidad:** ¿Hay suficiente documentación y soporte sobre el componente?¿Existe una hoja de ruta del producto? ¿Se actualiza regularmente? ¿Podemos prevenir components obsoletos?

2. Los componentes de software de terceros también deben someterse a una Evaluación de Riesgos de Terceros.

- **Evaluaciones de Impacto en el Negocio (BIA):** Analizar el impacto en la Continuidad, el CIA, y la Privacidad.
- **Las Evaluaciones de Riesgo de Terceros (TPRA):** Evaluar criterios como el cifrado de datos, la gestión de parches y el programa de divulgación de vulnerabilidades de terceros, las capacidades de respuesta a incidentes de seguridad. Revisar informes SOC 2, ISO 27001 (SOAs) y revisión de políticas de protección de datos, etc.
- **Evaluacion de Riesgos IA:** Ética, transparencia, protección de datos.

3. Examina la Composición de Software (SCA) y SBOM para fortalecer la seguridad.

- Dependiendo del formato, ya sea SPDX, CycloneDX o SWID, los SBOMs nos proporcionan información de procedencia de los componentes de producto, ej. el nombre del fabricante, la versión del paquete, la información de licencias, vulnerabilidades, relaciones de paquetes, servicios, dependencias etc.
- Identificar conflictos de licencias, vulnerabilidades, VEX (Intercambio de Explotabilidad de Vulnerabilidades), integridad de archivos y potencialmente identificar componentes "de alto riesgo", como abandonware (sin actualizaciones en dos años o más), paquetes vacíos (sin archivos fuente) y código binario nativo.

6. Consideraciones Importantes

1. Designación formal de Propietarios de Producto (POs).

2. Inventario actualizado de todos los productos y componentes subyacentes.

3. ”Trust but verify”

Preguntas?