Tecnicatura Universitaria en Ciberseguridad

CONCEPTOS DE DESARROLLO DE SOFTWARE (V.TUCS.2.16.1)

Trabajo Práctico N° 3

"Descripción breve y diferenciada de las fases del ciclo de vida de desarrollo de software"

Agustin E. Guida

6 de Junio del 2025

Contenido

Descripción breve y diferenciada de las fases del ciclo de vida de desarrollo de software

Análisis de requerimientos:
En esta fase se recopilan y documentan las necesidades y expectativas del cliente o usuario final, definiendo qué funciones debe cumplir el software y bajo qué condiciones operará. Se identifican requerimientos funcionales, que describen las acciones del sistema, y no funcionales, que establecen criterios como rendimiento, seguridad y usabilidad.

Diseño:
Aquí se traduce lo especificado en los requerimientos en una estructura técnica que guiará la construcción del software. Se define la arquitectura general, los módulos, las interfaces y los datos que conformarán el sistema, creando modelos y diagramas que faciliten la implementación posterior.

Implementación:
En esta etapa los desarrolladores codifican el software siguiendo el diseño definido. El objetivo es transformar los modelos y especificaciones en código funcional utilizando uno o más lenguajes de programación, aplicando buenas prácticas para asegurar calidad y mantenimiento.

Pruebas:
Se realizan verificaciones para detectar y corregir errores, asegurando que el software cumpla con los requerimientos y funcione correctamente bajo diferentes condiciones. Se ejecutan pruebas unitarias, de integración, funcionales y de aceptación, garantizando la calidad antes de la entrega.

Despliegue:
Consiste en instalar y poner en operación el software en el entorno real, configurando los sistemas necesarios y capacitando a los usuarios. Esta fase implica que el sistema esté disponible y operativo para los usuarios finales, listo para su uso productivo.

Mantenimiento:
Una vez en producción, el software requiere correcciones, mejoras o adaptaciones a nuevas necesidades o cambios en el entorno. El mantenimiento asegura que el sistema siga funcionando correctamente, solventando fallos detectados y actualizando funcionalidades para prolongar su vida útil.

Cuadro de seguridad en el ciclo de vida de desarrollo de software

Fase	Riesgos de seguridad comunes	Actividades concretas del experto en ciberseguridad
Análisis	• Omitir requisitos de seguridad • Malentender necesidades regulatorias (como protección de datos personales) • Falta de trazabilidad de los requisitos	• Identificar requisitos de seguridad funcionales y no funcionales • Verificar cumplimiento legal (ej. LPD, GDPR) • Asegurar trazabilidad y documentación de seguridad
Diseño	• Arquitectura sin separación de componentes críticos • Falta de cifrado en tránsito y reposo • Diseño sin validación de entradas o sin control de acceso	• Realizar análisis de amenazas (threat modeling) • Definir políticas de cifrado, autenticación y control de acceso • Revisar el diseño desde una perspectiva de defensa en profundidad
Implementación	• Inclusión de vulnerabilidades como XSS, inyección SQL, CSRF • Uso de librerías inseguras o desactualizadas • Manejo inseguro de errores y datos sensibles	• Realizar revisiones de código (code review) • Aplicar prácticas de codificación segura (OWASP SAMM o ASVS) • Validar dependencias externas y versiones seguras
Pruebas	• Casos de prueba que no contemplan ataques comunes • Falsos negativos por pruebas incompletas • Herramientas de testing mal configuradas	• Ejecutar pruebas de seguridad como análisis estático (SAST), dinámico (DAST) y fuzzing • Simular ataques con pruebas de penetración • Validar cobertura de pruebas de seguridad
Despliegue y mantenimiento	• Configuraciones inseguras (puertos abiertos innecesarios, servicios sin protección) • Parcheo tardío o inexistente • Exposición de credenciales o secretos	• Realizar hardening del sistema • Configurar monitoreo de seguridad y alertas • Aplicar parches y actualizaciones de forma controlada y documentada

Referencias y Fuentes

1. *Software development life cycle (SDLC)*. (2020, febrero 26). GeeksforGeeks. https://www.geeksforgeeks.org/software-development-life-cycle-sdlc/