

UNIVERSIDAD ESTATAL DE BOLÍVAR

**FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN
EMPRESARIAL E INFORMÁTICA**

CARRERA:



TEMA:

Tabla Comparativa de Metodologías de desarrollo de software

AUTOR:

Antony Alexander Colina Espinoza

DOCENTE:

Ing. Mónica Bonilla

CICLO ACADEMICO:

Enero – mayo 2025

TABLA COMPARATIVA DE METODOLOGIAS DE DESARROLLO DE SOFTWARE

Metodología	Ventaja	Desventaja	Funcionalidad	Aplicación	Seguridad
DevSecOps	Integra seguridad desde el inicio, lanzamiento rápido de software seguro.	Requiere cambio cultural y resistencia a la inclusión de expertos en seguridad.	Automatización de pruebas de seguridad, integración continua, monitorización.	Desarrollo de software moderno con ciclos rápidos y alta demanda de seguridad.	Seguridad integrada en todo el ciclo de vida del softwar
FDD	Enfoque en características específicas funcionales, fácil de seguir y medir el progreso de manera organizada.	No es adecuado para proyectos de equipos pequeños.	Desarrollo basado en características, iteraciones cortas y grandes.	Proyectos grandes con equipos grandes y bien definidos.	Seguridad no integrada, requiere enfoque adicional.
Modelo V	Fases bien definidas, fácil de entender y seguir como la validación y verificación.	Poco flexible, cambios costosos en etapas avanzadas.	Desarrollo secuencial con pruebas en cada fase.	Valiosa en proyectos donde la calidad y la conformidad son imprescindibles,	Seguridad implementada en fases de desarrollo, no integrada desde el inicio.

				como sistemas críticos.	
RUP	Flexible, adaptable a diferentes tipos de proyectos donde se prioriza la gestión del riesgo.	Complejo de implementar en proyectos pequeños, requiere experiencia y documentación extensa.	Desarrollo iterativo e incremental, fases bien definidas.	Proyectos grandes y complejos con requisitos con una arquitectura bien definida y documentada.	Seguridad no integrada, requiere enfoque adicional en cada iteración.
SCRUM	Flexibilidad, adaptabilidad a cambios para los equipos, entrega incremental.	Requiere compromiso del equipo y experiencia en SCRUM.	Desarrollo iterativo e incremental, sprints cortos, reuniones diarias.	Proyectos complejos y ágiles con requisitos cambiantes y equipos pequeños.	Seguridad no integrada, requiere enfoque adicional en cada sprint.
Kanban	Visualización del flujo de trabajo, mejora continua y flexibilidad para cambios.	No define roles requiere disciplina y seguimiento constante, y no establece plazos fijos de entrega .	Gestión visual del flujo de trabajo, limitación de trabajo en progreso.	Proyectos ágiles con flujo continuo y necesidad de mejora continua.	Seguridad no integrada, requiere enfoque adicional en el flujo de trabajo.
Modelo Espiral	Flexibilidad y genérico,	Complejo, requiere	Desarrollo iterativo con	Proyectos grandes con alto	Seguridad gestionada en

	gestión de riesgos, iterativo. Máximo control.	experiencia en gestión de riesgos. No es apropiado para pequeños proyectos.	enfoque en riesgos y prototipos. Con máximo control de costes, recursos y la calidad.	riesgo y requisitos cambiantes es decir poco claro y complejos.	cada iteración, pero no integrada desde el inicio.
XP	Entrega rápida, enfoque en calidad del código. Flexible y fomenta el trabajo en equipo.	Requiere alta disciplina y compromiso del equipo para el desarrollo.	Desarrollo iterativo, programación en parejas, integración continua.	Proyectos pequeños con requisitos cambiantes y equipos altamente colaborativos.	Pruebas automatizadas para la funcionalidad, pero requiere enfoque adicional en cada iteración.
Modelo Cascada	Fases claras y bien definidas, fácil de entender.	Poco flexible, cambios costosos en etapas avanzadas.	Desarrollo secuencial con fases bien definidas.	Proyectos con requisitos bien definidos y poco cambiantes.	Seguridad implementada en fases de prueba, no integrada desde el inicio.
Scrumban	Combina flexibilidad de SCRUM con visualización de Kanban.	Puede ser complejo de implementar correctamente y necesita una	Desarrollo iterativo con gestión visual del flujo de trabajo. Asigna	Proyectos que requieren flexibilidad y mejora continua.	Seguridad no integrada desde el inicio, requiere enfoque

	Gestión de riesgos y control y seguimiento del proyecto.	app online para comunicarte con todo el equipo. Falta de control en cada actividad.	actividades a cada miembro según corresponda.		adicional en cada iteración.
AUP (Agile proceso unificado)	Versión simplificada de RUP, enfoque ágil y flexible.	Menos documentación que RUP, puede ser demasiado corta para proyectos grandes.	Desarrollo iterativo e incremental, fases simplificadas.	Proyectos ágiles con los requisitos cambiantes y equipos pequeños a medianos.	Seguridad no aplicada, requiere enfoque adicional en cada iteración.