**Análisis de requerimientos:**

a) Claridad y especificidad de los requisitos: Evaluar cómo se han definido los requerimientos, si son comprensibles, detallados y específicos para facilitar su implementación y evitar ambigüedades.

b) Cobertura de requerimientos: Analizar si se han identificado y considerado todos los requisitos necesarios para satisfacer las necesidades del cliente y del proyecto en general.

**Gestión de proyectos:**

a) Cumplimiento de plazos y presupuesto: Evaluar si el proyecto se ha llevado a cabo dentro de los plazos y recursos previamente establecidos, lo que indica una buena gestión del tiempo y el presupuesto.

b) Gestión de riesgos: Analizar cómo se han identificado, evaluado y mitigado los riesgos a lo largo del proyecto, lo que refleja una gestión adecuada de posibles problemas o desafíos.

**Arquitectura y diseño detallado:**

a) Escalabilidad y modularidad: Evaluar si la arquitectura y el diseño permiten que el sistema sea escalable para acomodar futuros cambios y si los componentes son independientes y fácilmente reutilizables.

b) Adherencia a los estándares y buenas prácticas: Analizar si la arquitectura y el diseño siguen estándares de la industria y buenas prácticas, lo que facilita el mantenimiento y la comprensión del sistema.

**Verificación y validación:**

a) Cobertura de pruebas: Evaluar si las pruebas realizadas abarcan todos los aspectos críticos del sistema y si se han considerado diferentes escenarios para asegurar su correcto funcionamiento.

b) Cumplimiento de estándares de calidad: Analizar si los resultados de las pruebas cumplen con los criterios de calidad establecidos, lo que asegura que el producto esté libre de errores importantes.

**Construcción y pruebas unitarias:**

a) Código limpio y mantenible: Evaluar si el código fuente se ha construido siguiendo buenas prácticas de programación para facilitar su mantenimiento a lo largo del tiempo.

b) Cobertura de pruebas unitarias: Analizar si todas las partes individuales del código han sido probadas exhaustivamente para detectar errores a nivel de componente.

**Integración y pruebas:**

a) Interoperabilidad entre módulos: Evaluar si los diferentes módulos del sistema se integran de manera adecuada y si pueden trabajar juntos sin conflictos.

b) Ejecución y resultados de pruebas de integración: Analizar si las pruebas de integración se realizaron con éxito y si los resultados muestran que el sistema completo funciona correctamente.

**Despliegue del producto:**

a) Éxito del despliegue: Evaluar si el producto se ha implementado con éxito en el entorno de producción y si está disponible para su uso según lo previsto.

b) Tiempo de inactividad y problemas durante el despliegue: Analizar si hubo tiempos de inactividad significativos o problemas graves durante el proceso de despliegue y cómo se manejaron.

**Autoevaluación:**

a) Evaluación objetiva: Evaluar si la autoevaluación ha sido realizada de manera imparcial y objetiva, considerando tanto logros como áreas de mejora.

b) Planes de mejora y corrección de errores: Analizar si se han establecido planes de acción para mejorar aspectos identificados en la autoevaluación y cómo se abordarán los errores encontrados.

**Versionamiento:**

a) Control de versiones efectivo: Evaluar si el sistema de control de versiones ha sido utilizado de manera efectiva para rastrear y gestionar los cambios en el código y la documentación.

b) Gestión de conflictos y ramas: Analizar cómo se han manejado los conflictos y las ramas de desarrollo, asegurando una integración adecuada y sin errores entre las diferentes versiones del software.