

El objetivo de esta actividad es resolver la pregunta: ¿Cuáles son las competencias mínimas, desde la perspectiva de la industria, en el área de arquitectura de software para un recién egresado de pregrado? Se sugiere que sea una lista pequeña de tal forma que puedan ser abordadas en una asignatura de pregrado.

Entradas: Lista completa de competencias de arquitectura de software, resultados de la encuesta.

Salidas: Lista reducida de competencias clasificadas en obligatorias y opcionales.

Pasos:

1. Clasificar las competencias de los stickies en los tres grupos: obligatorias, opcionales, y fuera de alcance para un recién egresado. Cada uno de los cinco integrantes puede tomar la decisión de ubicar el sticky en la columna correspondiente. En caso de dudas puede apoyarse en las opiniones de sus colegas. También, se pueden apoyar en los resultados de la encuesta. Tiempo máximo 10 minutos
2. Revisar en grupo la clasificación realizada y hacer ajustes que se consideren necesarios. Es un espacio para refinar el trabajo de grupo. Tiempo máximo 15 minutos
3. Socializar los resultados. Es una breve justificación de la clasificación realizada hacia el investigador. La puede hacer un integrante con apoyo de los demás. Tiempo máximo 5 minutos

Obligatorias

Opcionales

Fuera de alcance

1. Identifica claramente los atributos de calidad relevantes del software que conducirán la arquitectura de un sistema de software a construir.

5. Evalúa independientemente una arquitectura de software para determinar la satisfacción de los requisitos funcionales y no funcionales.

18. Analiza críticamente los requisitos de software funcionales y de atributos de calidad.

23. Desarrolla componentes de software reutilizables.

22. Realiza periódicamente revisiones del código fuente escrito por el equipo de desarrollo.

28. Diseña e implementa procedimientos de prueba considerando aspectos de la arquitectura (tipos de componentes/servicios, integración).

11. Mantiene los sistemas existentes y su arquitectura para lograr la evolución de los sistemas software.

8. Realiza imparcialmente un análisis de compensación (trade-off) para evaluar arquitecturas.

12. Rediseña las arquitecturas existentes para la migración a nuevas tecnologías y plataformas.

2. Diseña consistentemente la arquitectura de software definiendo cómo los componentes interactúan entre sí.

4. Expande cuidadosamente los detalles del diseño perfeccionándolo para converger en el diseño final.

9. Prepara organizadamente los documentos arquitectónicos y presentaciones útiles para las partes interesadas (stakeholders).

21. Guía especificaciones de software claras a partir de los requisitos de negocio (dependiendo del estilo de trabajo de la organización)

14. Lidera con entusiasmo actividades de mejora de la arquitectura en una organización desarrolladora de software.

15. Participa activamente en la definición y mejora de procesos de software en una organización.

20. Captura sistemáticamente los requisitos del cliente, de la organización y del negocio en la arquitectura [relacionada: 20]

3. Toma decisiones de diseño relevantes sobre cómo debe construirse un sistema involucrando las elecciones que enfrenta un arquitecto al diseñar un sistema de software.

26. Recomienda metodologías de desarrollo para el equipo de desarrollo.

24. Desarrolla soluciones basadas en componentes reutilizables existentes.

19. Comprende rápidamente las necesidades del negocio y del cliente para asegurar que los requisitos satisfagan estas necesidades.

7. Aplica sistemáticamente técnicas de arquitectura basadas en valor para evaluar decisiones arquitectónicas.

16. Define reflexivamente la filosofía y los principios para la arquitectura global.

27. Participa en el trabajo de consultores y proveedores externos.

10. Produce estándares de documentación que incluyan la variabilidad y el comportamiento dinámico.

17. Proporciona colaborativamente apoyo a la supervisión de la arquitectura de los proyectos de desarrollo de software.

6. Revisa frecuentemente los diseños de los componentes propuestos por ingenieros junior verificando el cumplimiento de la arquitectura.

13. Proporciona pro-activamente pautas arquitectónicas para las actividades de diseño de software.

29. Construye el producto facilitando la identificación y corrección de fallos.

25. Sugerir lineamientos de codificación por parte del equipo de desarrollo, y mecanismos para verificarlos automáticamente

