

(SWEEBOK V3 INTERSECCIÓN SWEEBOK V4 INTERSECCIÓN ACM- IEEE 2014) MÁS ACM-IEEE 2020

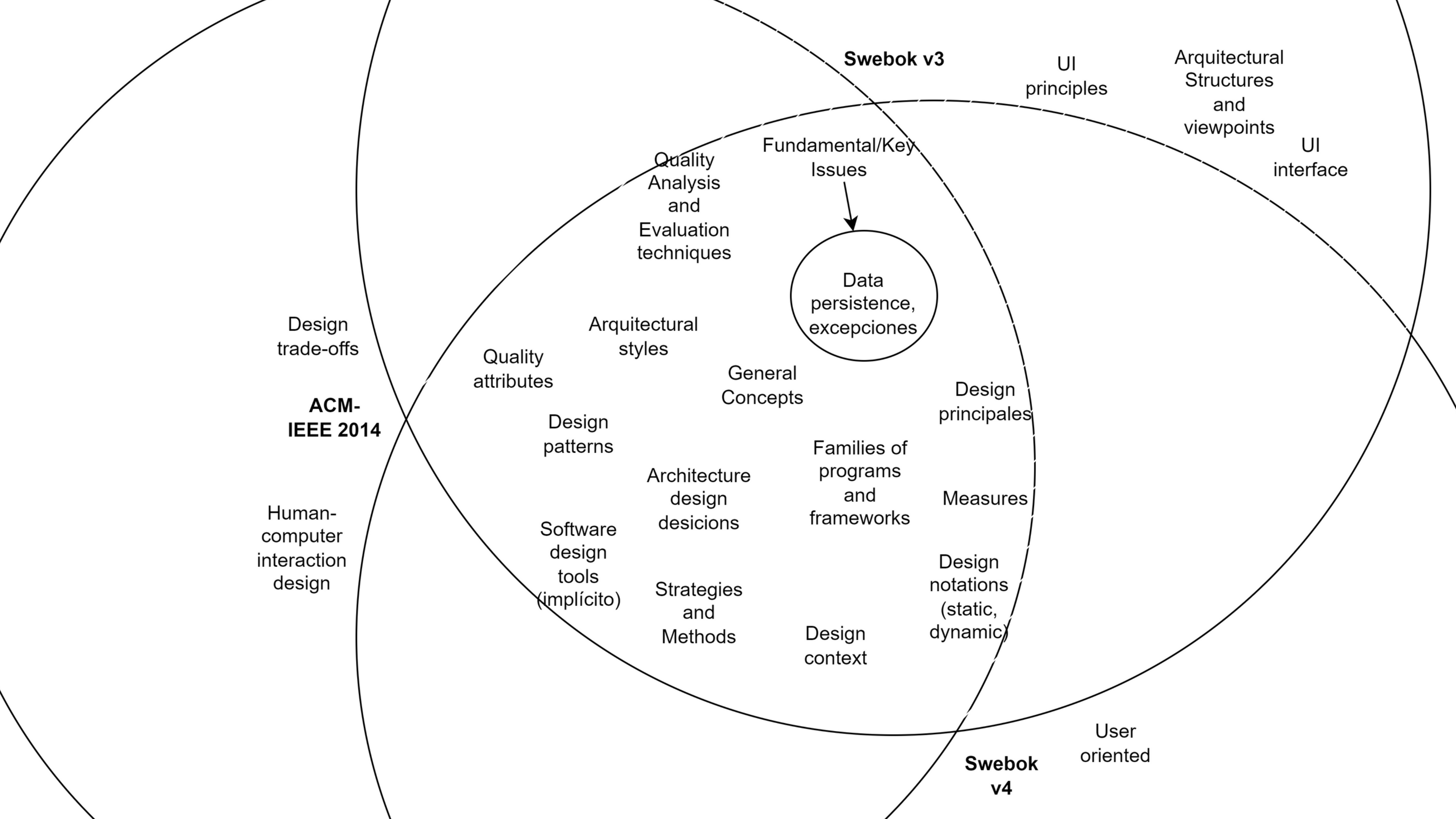
AUTORES:

BONILLA PECH RUSSEL ADRIAN

ORTIZ CHAY JESUS MATEO

PACAB CANUL RODRIGO JOAQUÍN

RODRIGUEZ COUOH ORLANDO ISAÍAS



GENERAL CONCEPTS

DESIGN THINKING, GENERAL DESIGN CONCEPTS, ETC.

DESIGN CONTEXT (CONTEXT OF SOFTWARE DESIGN)

MENCIÓN A LOS CICLOS DE VIDA DEL SOFTWARE.

SOFTWARE DESIGN PRINCIPLES

CONCEPTOS ESTRECHAMENTE RELACIONADOS CON POO

FUNDAMENTAL/KEY ISSUES IN SOFTWARE DESIGN

SE SEÑALAN DATA PERSISTENCE Y EXCEPTIONS COMO LAS MÁS COMUNES

ARCHITECTURE DESIGN DECISIONS

SE ENCONTRÓ CONTENIDO SIMILAR EN ACM-IEEE 2014 Y SWEBOK 3, SIN EMBARGO, EN SWEBOK 4, ESTE TIENE EL NOMBRE DE DESIGN RATIONALE QUE HABLA DE LA TOMA DE DECISIONES IMPORTANTES.

FAMILIES OF PROGRAMS AND FRAMEWORKS

SWEBOK 4 ESTE TIENE NOMBRE DE DETAILED DESIGN.

DESIGN PATTERNS

TANTO EN POO COMO EN ARQUITECTURA

SOFTWARE DESIGN NOTATIONS (STATIC VIEW) (DYNAMIC VIEW)

CURRÍCULA DE ACM-IEEE 2014, SE DEDUCE CUAL ES SV Y DV

ARCHITECTURAL STYLES

QUALITY ATTRIBUTES

QUALITY ANALYSIS AND EVALUATION TECHNIQUES

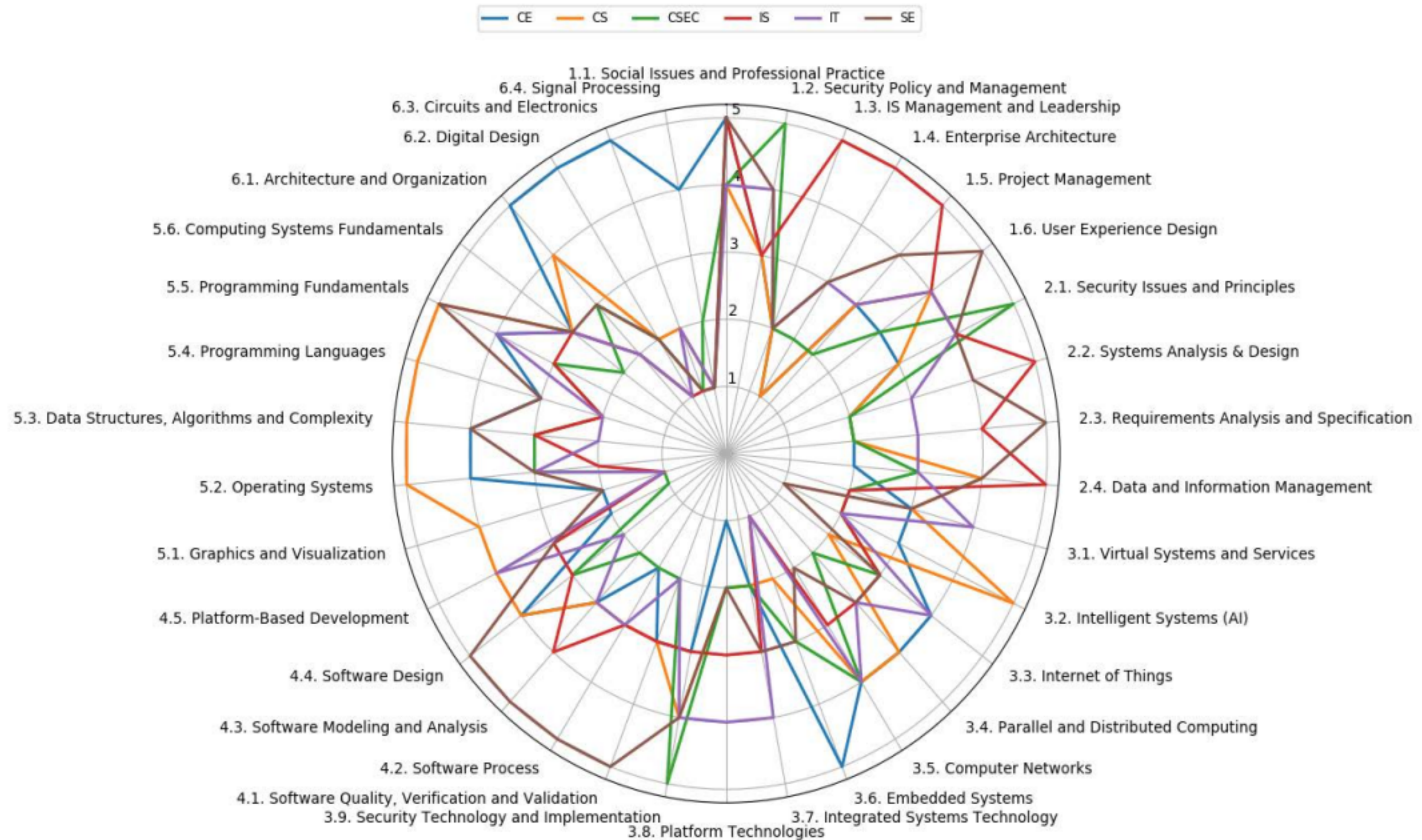
MEASURES

SE ENCONTRÓ CONTENIDO SIMILAR EN LOS TRES DOCUMENTOS.

STRATEGIES AND METHODS

SWEBOK 3 ES LA ÚNICA QUE TIENE UN BREVE APARTADO

COMPUTING CURRICULA 2020



COMPETENCIAS QUE UN INGENIERO DE SOFTWARE DEBERÍA TENER EN CUANTO AL DISEÑO DE SOFTWARE:

1. **Presenta** a los tomadores de decisiones empresariales requisitos arquitectónicamente significativos a partir de un documento de especificación de requisitos de software.
2. **Evaluar** y **comparar** ventajas y desventajas de posibilidades de diseño alternativas para satisfacer requisitos funcionales y no funcionales y **escribir** una breve propuesta que **resuma** las conclusiones clave para un cliente.
3. Producir un diseño de alto nivel de subsistemas específicos que sea **presentable para una audiencia no informática** considerando patrones arquitectónicos y de diseño.
4. **Producir** diseños **detallados** para un cliente para diseños de alto nivel de subsistemas específicos utilizando principios de diseño y aspectos transversales para satisfacer requisitos funcionales y no funcionales.
5. **Evaluar** las pruebas de software considerando los atributos de calidad en el diseño de subsistemas y módulos para un desarrollador/fabricante.
6. **Crear** documentos de diseño de software que permitan **comunicar de manera efectiva** a los clientes el diseño de software, así como a los analistas, implementadores, planificadores de pruebas, etc.

AQUÍ PODEMOS NOTAR PALABRAS CLAVE QUE HACEN REFERENCIA A SOFT SKILLS TALES COMO COMUNICACIÓN EFECTIVA, ANÁLISIS, ESCRITURA, ETC. HABILIDADES MUY RELEVANTES PARA PODER APROVECHAR LAS HABILIDADES TÉCNICAS QUE NOS PROPORCIONARÁ DISEÑO DE SOFTWARE.