SKILLS FRAMEWORK FOR INFOCOMM TECHNOLOGY
TECHNICAL SKILLS & COMPETENCIES (TSC) REFERENCE DOCUMENT

TSC Category	Design and Architecture					
TSC Title	Software Design					
TSC Description	Create and refine the overall plan for the design of software, including the design of functional specifications starting from the defined business requirements as well as the consideration and incorporation of various controls, functionality and interoperability of different elements into a design blueprint or model which describes the overall architecture in hardware, software, databases, and third party frameworks that the software will use or interact with					
TSC Proficiency Description	**Level 1**	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Level 6**
			ICT-DES-3005-1.1	**ICT-DES-4005-1.1**	**ICT-DES-5005-1.1**	**ICT-DES-6005-1.1**
			Design simple software components, assessing functionality of different elements, and produce design documentation	Create a software design blueprint based on a broad design concept, and business and user requirements	Translate complex software ideas and concepts into a design blueprint and establish key design principles and methodologies	Inspire new and innovative software design ideas, and align design principles and parameters with current and future needs
Knowledge			• Design requirements for simple, basic software components • Basic software design tools and techniques • Types of controls, elements and features in software • Indicators of software functionality and interoperability • Documentation of design details	• Components and requirements of a software design blueprint • Software design standards, methods and tools - and their pros, cons and applications • Requirements of functional specifications of software • Impact of different software design elements on overall software operations and usability	• Software design principles • New and emerging methodologies and tools for software design • Pros, cons and trade-offs of different software design options	• New and emerging trends in software design ideas • Best practices and external regulations in software design standards and practices • Process to determine software design principles

Abilities			• Design a simple software component or interface according to functional specifications and business requirements • Utilise appropriate software design methods and tools, in line with the organisation's software design practice and principles • Identify relevant controls, elements and features to be included in the software to meet its design objectives • Assess functionality and interoperability of different elements or components in the software design • Produce detailed design documentation mapped to user specifications	• Create a software design blueprint based on a broad design concept, and business and user requirements • Recommend appropriate standards, methods and tools for the design of software, in line with the organisation's practice and design principles • Design functional specifications of software systems to address business and user needs • Evaluate trade offs from the incorporation of different elements into the design, and their impact on overall functionality, interoperability, efficiency and costs of the software • Produce design documentation for complex software • Review design documentations produced	• Translate complex software ideas and concepts into a design blueprint and plan • Establish key design principles to guide the further definition and detailing of a software blueprint • Introduce new methods and tools for the design of software • Lead the design of highly complex software and systems • Evaluate multiple software design options, so as to select the one which best meets business, user and functional requirements • Justify design elements to the end user	• Inspire new and innovative software design ideas • Establish organisation-wide software design standards, guidelines and methodologies, in line with emerging trends, industry best practices and external regulations • Anticipate future business and user requirements, and their implications on software design, features and capabilities • Guide the setting of design principles, ensuring alignment with current and future needs • Chart a future-focused direction for the design of multiple software systems
Range of Application	Types of Software Applications may include but are not limited to: • Mobile/Native Applications • Augmented Reality / Virtual Reality Applications • Web Applications • Hybrid Applications • Cloud Applications Types of methodologies may include but not limited to: • Agile Software Development • Design Pattern • Extreme Programming • Object-Oriented					