COMPETENCY ASSESSMENT

CATEGORY	COMPETENCIES (10)	SOFTWARE INDICATORS (guidance on example content that will demonstrate the competency)
1. Technical Competence (10 competencies)	1.1 Demonstrate your knowledge and awareness of Canadian regulations, codes and standards. This includes local engineering procedures and practices as applicable.	1. For design and project activities, identify legal and regulatory requirements for information control, auditing and security (e.g. Bill C198 CSOX) – information access and protection of privacy legislation, and non-disclosure or intellectual property agreements 2. Incorporate knowledge of standards (such as: coding standards, interface standards for hardware and software operating systems, external standards for programming languages and tools, technical interfaces, internal standards, IEEE/ISO/IEC Standards) in design materials 3. Prepare reports assessing project compliance with codes, standards, and legal/regulatory requirements 4. Ensure that appropriate codes, standards, and legal/regulatory requirements are complied with in the design and implementation process
	1.2 Demonstrate knowledge of materials, or operations as appropriate, project and design constraints, design to best fit the purpose or service intended and address inter-disciplinary impacts.	1. Demonstrate knowledge of business or industrial processes, operations, project and design constraints, e.g. timeline, cost, physical, operational availability, reliability, user competence, failure rate, maintenance downtime and support 2. Demonstrate knowledge of appropriate existing software components, package, frameworks, APIs, libraries, development stacks , software patterns, both internal (to the development organization) and external (open-source or proprietary) 3. Demonstrate understanding of and coordination with other engineering and professional disciplines 4. Demonstrate use of software development best practices (e.g. code reviews, revision management, and automated deployment) to ensure process integrity
	1.3 Analyze technical risks and offer solutions to mitigate the risks.	1. Demonstrate familiarity with system protection, security and/or damage/hazard mitigation (e.g. from unexpected loads, unintended uses, cyberattacks, catastrophic failures and losses, etc.) 2. Identify risk and threat areas including causes of risks and their impacts to the client and public 3. Perform FMEA. Clearly identify the remaining risks and worst case scenario 4. Develop risk management/mitigation plans

CATEGORY	COMPETENCIES (10)	SOFTWARE INDICATORS (guidance on example content that will demonstrate the competency)
	1.4 Apply engineering knowledge to design solutions.	1. Conduct Business or Industrial Process Analysis 2. Identify functional requirements and non-functional requirements (relevant quality attributes) 3. Apply Software Engineering Principles to System Synthesis and Design of a software-intensive system to obtain optimal results (Develop or select a suitable system and software architecture; Select an appropriate. development technology (framework, language, tools) Decompose the system; design interfaces between software components; etc. Trace key design choices to functional and non-functional requirements, etc.) 4. Demonstrate the development of a unique solution which could not be accomplished with a standard design solution 5. Adopt application-proven and field-proven design approaches whenever feasible
	1.5 Be able to understand solution techniques and independently verify the results.	1. Demonstrate an understanding of the application, selection and adaptation of available software engineering tools to support specific tasks in the software development life-cycle 2. Show /describe steps taken to independently verify the results of available software engineering tools, APIs, frameworks, libraries etc.
	1.6 Demonstrate your knowledge and awareness of Canadian regulations, codes and standards pertaining to safety.	1. Demonstrate compliance to local design and programming standards, including privacy regulations, language, product quality and functional safety standards, cybersecurity standards , system and software quality requirements and evaluation, requirements for specific regulatory purposes and ethical considerations 2. Demonstrate application of crossover domains and protection of data 3. Demonstrate incorporation of elements in software design that protects/ensures public interest, user protection, security and safety 4. Demonstrate post-failure analysis of causes of failure and suggest solutions to prevent further disasters 5. Identify risks through operations with respect users goals, how you ensure safety of data and user integrity and how implement xxx to ensure user integrity 6. List the risk or fault exclusion terms or conditions
	1.7 Demonstrate understanding of systems as well as of components of systems.	1. Demonstrate a decomposition and understanding of the modularity of a software solution 2. Show an understanding of the software's operating environment (e.g. operating systems, network components, physical devices, software libraries and frameworks, external systems), and constraints in the behaviour of the overall system 3. Demonstrate understanding of systems architecture from a safety perspective as well as of components of systems on their safety data or statistics from manufacturers

CATEGORY	COMPETENCIES (10)	SOFTWARE INDICATORS (guidance on example content that will demonstrate the competency)
	1.8 Exposure to all stages of the process/project life cycle from concept and feasibility analysis through implementation.	Demonstrate the applicant's contribution to the following stages of the project lifecycle: 1. **Identification/Elicitation of Requirements:** confirmation of operational distribution or deployment mission profile or scenario, critical system performance parameters, utilization requirements, operational life-cycle and operating environment, user expectations and system operational requirements 2. **Preparation:** effective use of software development process or life cycle and adaptation of development process to the requirements, including use of third-party simulation schemes such as HIL techniques 3. **Appraisal:** analysis of the project or features from technical, timeline/financial, economic, social, user requirements, institutional and environmental perspectives 4. **Preparation of specifications, design team integration and subcontracting:** preparation of design specifications, integration of design team, engagement and integration of external system suppliers or subcontractors 5. **Implementation, validation and monitoring:** implementation of project activities with on-going checks on progress and feedback; project management (agile or waterfall/traditional)t, training and support of users including user documentation 6. **System Life Cycle Support:** system maintenance, review of system efficiency, iterative optimization, system retirement
	1.9 Demonstrate your understanding of the role of peer review and quality management that is essential to engineering practice in Canada.	1. Ensure user and stakeholder expectations are met 2. Ensure the software design meets technical expectations 3. Prepare quality control plans for the entire design and implementation process, including frequency and test parameters, for specific processes or products 4. Evaluate test results, determine adequacy, and develop recommended action 5. Demonstrate independent peer review and validation of design 6. Demonstrate that completed project, systems or sub-systems meet project objectives in terms of functionality, operational performance and cost 7. Understand design limitations and exclusions in terms of safety 8. Demonstrate root cause analysis in production/live systems
	1.10 Transfer design intentions to drawings and sketches; Understand transmittal of design information to design documents.	1. Demonstrate the ability to prepare or review software-specific drawings and sketches (ER diagrams, UML (Unifed Modeling Language) diagrams, IDEF0, statecharts, pseudo-code etc.) of others and communicate findings and issues, including suggested alternatives 2. Demonstrate communication of ideas and concepts to project team members 3. Demonstrate understanding of value of project completion reports and lessons learned reports to application in future projects by self or others 4. Produce software engineering design documentation with respect to requirements, architecture, technical (code, algorithms, interfaces and APIs) and end user documentation, including training and standard operating procedures, and document design and specifications for acceptance by the client/user and approval by regulatory authorities