Annex D - IV&V Engagement Checklists

The following Checklists are provided herein which apply throughout various steps during an Engagement. However, these should be considered when formulating an RFP and tender document during Pre-Engagement phase because activities noted on these checklists may need to be included in an RFP or tender document.

Important Note
These are the minimum reporting to be performed during the IV&V engagement.

#	Unique Identifier	Checklist Name	Checklist Description	See page
1.	CK-PSR	Project Schedule Review Checklist	List of project schedule quality-oriented items to review and assess.	D-2
2.	CK-DR	Document Review Checklist	List of documents to review and assess for quality focus and testability.	D-3
3.	CK-RR	Requirements Review Checklist	List of requirements attributes to review and assess.	D-4
4.	CK-HLD	High-Level Design Checklist	List of high-level design items to review and assess.	D-6
5.	CK-LLD	Low-Level Design Checklist	List of detailed design items to review and assess.	D-7
6.	CK-PRP	Process Review Procedures Checklist	List of development and testing process items to review and assess.	D-8
7.	CK-CR	Code Review Checklist	List of code review items to assess.	D-10
8.	CK-UTR	Unit Testing Review Checklist	List of attributes of unit testing to review and assess.	D-12
9.	CK-TRR	Test Readiness Review Checklist	List of attributes to assess whether ICT system is ready for testing to begin.	D-14
10.	CK-SD	System Disposal Checklist	List of attributes to assess whether ICT system is ready for disposal.	D-15

CK-PSR: Project Schedule Review Checklist

- **Instruction:**

(a) Use this schedule review to verify that all appropriate activities have been planned throughout the ICT development lifecycle.

CK-PSR : Project Schedule Review Checklist			
IV&V Team Member:		Date:	
Project:			

#	**Item**	**Description of Issue**
1.	Project Management Plan	
2.	Quality Management Plan	
3.	Requirements Management Plan	
4.	Requirements Documents	
5.	Design Documents	
6.	Master Test Plan includes all appropriate Quality Gates	
7.	Time allotted to testing	
8.	Appropriate Quality Gates included in Schedule	
9.	Schedule does not start next stage before quality gate is completed (i.e. Design does not start until requirements are complete; development does not start until design is complete, etc.)	
10.	Document review timeframes are planned and reasonable based on the document scope and other tasks that occur at the same time	

CK-DR: Document Review Checklist

- **Instruction:**

(a) Perform document reviews is to verify that the technical documents and plans are consistent with project plans, requirements and guidelines established by the Technical Committee.

CK-DR : Document Review Checklist

IV&V Team Member:		Date:	
Project:			

#	Item	Description of Issue
1.	Is document written to appropriate detail?	
2.	Is document consistent with other documents?	
3.	Is material within document feasible as stated?	
4.	Are all required sections included in the document?	
5.	Are all sections in the proper order?	
6.	Is document in compliance with required statement of work?	
7.	Are all statements compatible and consistent?	
8.	Is the level of detail and presentation style consistent throughout the document?	
9.	Are all terms, acronyms and abbreviations defined?	
10.	Is the overall approach sound?	
11.	Is the document well researched and based on proven prototypes?	

CK-RR: Requirements Review Checklist

- **Instruction:**

(a) Use this checklist to provide guidance for verifying the quality of the system requirements.

CK-RR : Requirements Review Checklist			
IV&V Team Member:		Date:	
Project:			

#	Requirements Characteristics	Items Under Examination
	Overall	
1	Standards	Have appropriate requirements documentation standards been followed?
2	Standards	Are all figures, tables, and diagrams labeled and referenced?
3	Standards	Are all terms and units of measure defined?
4	Standards	Are all requirements written at a consistent and appropriate level of detail?
5	Standards	Are individual requirements ranked with descriptions of priority provided?
6	Verifiability	Do the requirements provide an adequate basis for design and system test?
	Completeness and Correctness	
7	Correctness	Are all internal cross-references to other requirements correct?
8	Incomplete	Are all classes of users described? Are the user characteristics described?
9	Incomplete	Does the specification include all known customer or system needs? Are all the tasks the user wants to perform specified?
10	Incomplete	Does each functional requirement specify input and output, as well as function, as appropriate?
11	Incomplete	Have all dependencies on other systems been identified? (Applications, interfaces, databases, communications, networking, etc.)
12	Incomplete	Are user documentation and training requirements addressed?
13	Incomplete	Are the hardware and software environments specified?
14	Incomplete	Have all derived requirements been included? (those implied by the system or software requirements, generally constraints on development or verification)
15	Incomplete	Has full life cycle support been addressed, including maintenance?
16	Incomplete	Are any design or implementation constraints described?

17	Incomplete	Are all reliability, recoverability, business continuity, and performance requirements properly specified?
18	Incomplete	Are all security requirements properly specified?
19	Incomplete	Have all data privacy requirements been included?
20	Incomplete	Are time critical functions identified and criteria specified for them?
21	Incomplete	Have regulatory, legislative, or standards-driven requirements been addressed?
22	Incomplete	Have all quality attributes been properly specified (i.e. efficiency, flexibility, interoperability, maintainability, portability, reusability, usability, availability)?
23	Interface	Have the human interface requirements been correctly addressed?
24	Interface	Are all external hardware, software, and communication interfaces defined? Are they correct?
	Consistency and Clarity	
25	Inconsistent	Does the specification agree with all relevant higher level documents?
26	Inconsistent	Are the requirements free of duplication and conflict with other requirements?
27	Inconsistent	Is each requirement written in consistent, clear, concise language?
28	Ambiguous	Does each requirement have only one interpretation? If a term could have multiple meanings, is it defined?
29	Verifiability	Is each requirement verifiable by testing, demonstration, review, or analysis?
30	Verifiability	Are there measurable acceptance criteria for each functional and non-functional requirement?
31	Traceability	Is each requirement uniquely and correctly identified?
32	Traceability	Is each requirement traceable to its source, including to requirements?
	Other	
33	Other	Is each requirement in scope for the project?
34	Other	Are all requirements actually requirements, not design or implementation solutions?
35	Other	Are the time-critical functions identified, and timing criteria specified for them?
36	Other	Are significant users of resources (memory, network bandwidth, processor capacity, etc.) identified, and the anticipated resource consumption specified?
37	Other	Have internationalization, personalization, and scaling issues been adequately addressed?

CK-HLD: High-Level Design Checklist

- **Instruction:**

(a) Perform design reviews to determine whether all software requirements have been translated into a viable software design. Generally, software projects have two design phases: top-level and detailed design.

(b) The following checklist applies to the high level design.

CK-HLD : High-Level Design Checklist			
IV&V Team Member:		Date:	
Project:			

#	Item	Description of Issue
1.	The functional [or object] partition is consistent with the requirements.	
2.	Security, Reliability, Maintainability, Availability issues have been addressed.	
3.	Each function has a single well defined purpose.	
4.	Software and interface requirements allocated to code.	
5.	Allocation of requirements to source code including COTS.	
6.	All inputs, outputs, functional control and sequencing is defined.	
7.	Internal interfaces and external interfaces are defined.	
8.	COTS applications and interfaces are defined.	
9.	Human factors have been addressed where relevant.	
10.	Contractor configuration management procedures and controls are in place.	

CK-LLD: Low-Level Design Checklist

- **Instruction:**

(a) Perform detailed design reviews to determine whether all software requirements have been translated into a viable software design. Generally, software projects have two design phases: top-level and detailed design.
(g) The following checklist applies to the detailed design.

CK-LLD : Low-Level Design Checklist			
IV&V Team Member:		Date:	
Project:			

#	Item	Description of Issue
1.	Each module has a single, clearly stated function.	
2.	Units are named according to conventions.	
3.	There is a software requirement from which the need for this function arose.	
4.	There is no superfluous processing.	
5.	No necessary processing is missing.	
6.	There are no other types of identifiable errors in logic.	
7.	There are no possible error conditions not provided for.	
8.	Unit interfaces are consistent and well defined.	
9.	Software requirements can be traced to code.	

CK-PRP: Process Review Procedures Checklist

- **Instruction:**

(a) Perform process review to ascertain, based on objective evidence, that approved plans and procedures have been implemented and are being followed.

CK-PRP : Process Review Procedures Checklist			
IV&V Team Member:		Date:	
Project:			

#	Item	Description of Issue
1.	Documented processes in use that provide timely, comprehensive, and accurate processing, reporting, and recording of approved changes to controlled components.	
2.	Documented processes that provide comprehensive implementation of approved changes and dissemination of corrected documentation and software changes.	
3.	Documented processes in use that provide accurate reporting and recording for the status of all proposed changes and change resolution.	
4.	Documented processes in use that provide verification and implementation of identification, change control, and status accounting of descriptive documentation and software materials.	
5.	Internal baseline for documentation.	
6.	Documented processes in use which govern the identification of software documentation and software materials:	
	(a) Identification denotes the component to which it applies.	
	(b) The purpose is described.	
	(c) The applicable baseline is defined.	
	(d) The serial, edition, and change status is identified.	
	(e) The compilation date for each deliverable software component is identified.	

#	Item	Description of Issue
	(f) Visual and machine readable identification for all delivered software media.	
7.	Documented processes in use that govern internal control of documents and software materials in the development support library.	
8.	Documented processes in effect that require bringing each component of the software under configuration control.	
9.	Documented process that governs establishment of Change Control Board (CCB).	
10.	The CCB operates as described in the documented process.	
11.	There are verifiable records indicating that all required CCB members were in attendance.	
12.	Documented processes that define the methods and format for submission of problem reports for problems detected in activities and products.	
13.	Documented processes in use that define the methods for processing problem reports for software and documentation.	
14.	Documented processes in use that control the preparation and dissemination of changes to documentation to reflect approved and implemented changes.	

CK-CR: Code Review Checklist

- **Instruction:**

(a) Perform a code review to determine whether the software design has been correctly implemented in code that adheres to the programming standards and conventions.

CK-CR : Code Review Checklist			
IV&V Team Member:		Date:	
Project:			

#	Item	Description of Issue
1.	Does the module (unit) have a single, clearly stated function?	
2.	From which software requirement(s) did the need for this function arise?	
3.	Does the documentation adequately describe the processing, data, and interfaces of this function?	
4.	Is the developer name, date of development and description of module function or code change included in the comments? Are comments adequate and accurate in describing the processing?	
5.	Are there control flow errors?	
6.	Is there superfluous or dead code?	
7.	Is there missing code?	
8.	Are there other types of errors in logic?	
9.	Are there possible error conditions that are not trapped?	
10.	Are statements "commented out?"	
11.	Does the code conform to the Standards?	
12.	Has the Unit Test Plan for the code under review been completed?	
13.	Does the module achieve its goals as stated in the design documentation?	
14.	Does the module follow program design language in design documentation?	

#	Item	Description of Issue
15.	Are there obvious style problems that affect readability or maintainability?	
16.	Is the file too long or contain too many functions?	
17.	Is there duplicate or similar code that could be combined into a general-purpose function?	
18.	Are there obvious code inefficiencies?	
19.	Are there better ways to accomplish the same results provided by the code?	
20.	Does the function return correct information to the caller in all cases?	
21.	Error cases not handled correctly (including caller program ignoring error status returned by called function)?	
22.	Do error messages provide enough information to understand the problem being reported?	
23.	Has a code review results file been created?	
24.	Has the Development Lead or his/her designee followed up to ensure that any discovered defects are addressed prior to the completion of testing?	

CK-UTR: Unit Testing Review Checklist

- **Instruction:**

The purpose of this checklist is to provide guidance for assessing the quality of unit testing.

CK-UTR : Unit Testing Review Checklist			
IV&V Team Member:		Date:	
Project:			

#	Item	Description of Issue
	Unit Test Plan Review	
1.	Is the purpose/objective of the test stated and it is applicable to the unit in question?	
2.	Is the requirement reference traceable to the unit?	
3.	Is the data recording and analysis method defined?	
4.	Are all required software items and tools identified and available?	
5.	Is the version of each software item and tool identified?	
6.	Is regression analysis defined in case of errors and code update?	
7.	Were any tools employed (test path coverage)?	
8.	Is the test plan consistent with the prescribed process defined by the development team?	
9.	Was the test plan subjected to a peer review?	
10.	Will the test be executed by someone besides the author?	
11.	Will the test be executed by someone besides the author?	
	Unit Test Results Review	
12.	Was the test plan approved prior to the start of testing?	
13.	Are test results retained in the application folder?	
14.	Is there a test report for this unit?	

#	Item	Description of Issue
	Unit Test Plan Review	
15.	Were the required higher level units available?	
16.	Were the results reviewed by an independent evaluator?	
17.	Is there evidence of source code review prior to the start of testing?	

CK-TRR: Test Readiness Review Checklist

- **Introduction:**

The purpose of a Test Readiness Review is to assess readiness to proceed to the Integration or Acceptance Test. This checklist provides guidance for assessing these reviews.

CK-TRR : Test Readiness Review Checklist			
IV&V Team Member:		Date:	
Project:			

#	**Item**	**Description of Issue**
1.	Software Test Plan submitted and approved.	
2.	System Integration or Acceptance Test Plans submitted and approved.	
3.	Configuration of System under test documented.	
4.	Draft Version Description Document submitted before Test Readiness Review.	
5.	Requirements/Test Case Traceability completed.	
6.	Test Environment established.	
7.	Test specific software developed.	
8.	Test Dry Runs completed and results submitted. Results included the number of dry run requirements passed, failed, and not tested.	
9.	Test Schedule prepared.	
10.	Prior milestones completed (e.g., critical design review) in that all of its exit criteria is satisfied and all Action Items responded to.	
11.	Entrance Criteria for the Integration/Acceptance/Alpha/Beta Testing established.	
12.	Exit Criteria for the Integration/Acceptance/Alpha/Beta Testing established.	

CK-SD: System Disposal Checklist

CK-SD: System Disposal Checklist					
System:					
#	Requirement	Compliance			Comments
		Yes	No	N/A	
1.	All information has been moved to another system, archived, discarded, or destroyed.				
2.	Legal requirements for records retention were considered before disposing of the system.				
3.	All information is cleared and purged from the system.				
4.	All information has been removed from storage medium (e.g., hard disk or tape).				
5.	Appropriate steps have been taken to ensure the level of sanitization is appropriate for the type of storage medium (e.g., overwriting, degaussing (for magnetic media only), and destruction).				
6.	All hardcopy media has been destroyed (e.g., shredded, burned, etc.).				
7.	Appropriate steps have been taken to ensure that all contractors implement sanitization policies and procedures for removing information processed or residing on a contractor's site.				
8.	Leased equipment for processing information has been sanitized before returned to the vendor.				
9.	When data has been removed from storage media, every precaution has been taken to remove duplicate versions that may exist on the same or other storage media, back-up files, temporary files, hidden files, or extended memory.				

System Owner Printed Name

Date

Owner Signature