(U) Engineering Development Group

Applied Engineering Division

(U) White Paper: Internal Review of Current EDG Testing Practices

23 October 2014

Classified By: 2354260

Derived From: CIA NSCG COL S-06

Declassify On: 25X1, 20641231

Change Log

Doc Rev	Doc Date	Rev By	Change Description	Reference	Authority / Approval Date
Draft	25 Sept 14	XX, XX	First Draft Release	All sections	
Draft-Rev A	26 Sept 14	XX, XX	First Draft Release	All sections	
Draft Rev B	23 Oct 14	XXX	Final Draft for TAC	All sections	

(U) Document Change Revision History

Page #	Description
8	Recommendations section relocated to front of document.
13	Peer Review paragraph altered.

(U) Table of Contents

(U) List of Figures

No table of figures entries found.

(U) List of Tables

(U) TBD/TBR List

TBD/TBR Number	Section	Summary Topic	Resolution Due
	N/A		

1 (U) Scope

(S//NF) This white paper discusses an internal EDG review of EDG's current test practices. This review was established to understand the current testing practices used by EDG developers and contractors when delivering EDG products. This review was also commissioned by the Technical Advisory Cadre (TAC) to provide EDG management with solutions / options for process

improvements that improve how EDG delivers quality tested products to its customers, while ensuring that EDG is able to best use its existing and future resources utilization provided current budget constraints.

1. (U) Background

(S//NF) The Engineering Development Group (EDG) in the CIA's Information Operations Center (IOC) focuses on providing technical capabilities and expertise that give the CIA the cyber technical advantage – to access, exploit, operate, and persist on cyber targets as they adopt the newest technology. In doing so, EDG developers – whether they are internal developers from EDG's Applied Engineering Division (AED) or external (contractor) development teams managed by EDG's Engineering Systems Division (ESD) – follow a standard development cycle in developing and test their products before any tool is used in an operation.

(S//NF) While product development cycles may vary slightly (due to the tailoring applied for a given product development effort) because of operator needs, product requirements, available developer resources, and other things, every product development cycle involves some form of testing. Generically, each project test effort involved multiple stages of testing:

- Developer (unit) testing
- Integration testing
- Acceptance testing
- Operational testing

(S//NF) While each testing period may vary from project to project, the definitions used to describe each testing period varied between developers – leading to differing expectations regarding testing from project to project. These differences had also exacerbated customer concerns about product quality vs. tool need timelines. For example, some customers expressed concerns that the testing process, which involved EDG's Independent Verification and Validation (IV&V) branch, would be slowed due to a lack of test resources for tools even though the test setups maintained by the IV&V branch were not representative of the true operational environment.

(S//NF) Though the example referenced above was not true in a majority of cases, similar perceptions on the value of testing has led to changes in customer behavior. For example, all

customers have acknowledged that testing is important, and many customers indicate in their requirements that tools must go through an "Independent Verification and Validation" process prior to delivery. That said, customers have also complained that many tools have been "caught" in a scheduling loop when it comes to testing, whereby the independent verification and valuation testing for tools of interest is not being completed in a timely manner.

(S//NF) The observed resulting behavior is that testing (while important) becomes a "bottleneck" that delays operations -- some of which is due to a lack of available testing resources and testing resource re-prioritization of tools currently under test. As a result, some operators have two strategies to address this problem. The first strategy was prioritizing their tools as Quick Reaction Capabilities (QRCs) with the following goals:

a) Reduce EDG testing burdens on the tools of interest, while
b) Ensuring that test resources were applied to their tools of interest -- leading to EDG/SED/IV&V to re-prioritize test resources to these projects (and thus delay tools that were not prioritized as QRCs).

(S//NF) To address this problem, COG/NOD established the NOD Prioritization Review Board (NRPB) to level set priorities for tools developed by EDG for COG/NOD. The goal was to ensure that COG/NOD's established priorities were in concert with COG/NOD's operational goals rather than individual operators -- thereby assisting developers with an understanding of how development effort tools be prioritized during the development process at large, and not as a means to dictate changes solely to the testing process. The NRPB believed that once tools reached the testing phase, tools would be placed under test and EDG developers & management would work together to assess tool testing priorities with the central goal of meeting operational needs (i.e., tools with lower NRPB-assessed priorities would not necessarily be pre-empted by tools with higher NRPB-assessed priorities). Unfortunately, while the NRPB prioritization provided guidance to EDG on tool development, EDG's IV&V team also used this prioritization assessment as a means to re-direct testing resources once tools reached IV&V – further exacerbating the re-prioritization of EDG resources for tools under test and leading to further delays of tool deliveries.

(S//NF) Additionally, as operators found themselves "crunched" between tool deployment deadlines and further delays to tool deliveries, a secondary behavior emerged as a way to mitigate further delays and need for training time (prior to tool deployment) in the product delivery process. To gain an understanding of tool performance and gain confidence in product chain's capabilities, many operators began to request "evaluation copies" of tools prior to or as the tool entering the integration and/or acceptance testing phase. As such, operators would

test the "evaluation copies" of a tool in an operationally representative environment and provide feedback to developers -- sometimes leading to new tool release candidates, which would inevitably lead to further delays in the integration and/or acceptance testing phase -- ultimately leading to a delay in the delivery of a product for operations. In some cases, testing “evaluation copies” of tools caused operators to realize the following:

a) The test environment the operators possessed was more representative of the target environment than originally posed in the requirement specification to EDG (and, thus, the testing EDG was engaged in was not indicative of the proposed tool’s operational scenario), and/or
b) The tool performed satisfactorily (or better) in the operational test environment and could meet a prevailing mission need in its current state.

(S//NF) While there is broad agreement that tools are not always developed for one mission scenario, operators could elect to request approval (independent from decision forums to which EDG participated) to use the tool in an operation. To do so, operators requested COG/NOD mission manager (or alike) approval to deploy the “evaluation copy” of a particular tool on a specific target -- despite the tools not completing EDG testing. As a result, there are cases where a tool was in test and officially delivered to COG/NOD weeks after the tool was successfully deployed in operation(s).

(S//NF) While this secondary behavior has generated much concern among many in EDG, it also has generated a positive effect among developers. Since their tools "may" get deployed during the IV&V process and developers found many operators to be diligent in testing their tools integrated with other COG/NOD tools for a specific mission case, some developers resorted to spending more time demonstrating and coordinating functionality and development activities with operators prior to "evaluation copy" release. This additional time spent before "evaluation copy" release to the operation meant that developers felt the need to spend more time examining their tools (at the unit level) prior to sending these tools to IV&V testing. As a result, the quality of these tools improved -- leading to fewer defects in integration and acceptance testing -- while also having a negative effect that reinforced that some tools were "good enough" for operations without finishing the integration and/or acceptance testing process.

(S//NF) In summary, there have been many examples over the last few years that have led to the generation of this white paper and its associated study. The general perception across EDG is that there was a need to gather developer and project manager feedback on testing practices in EDG, understand testing standards, and pursue commonalities and best practices to ensure that all EDG products (whether internally or externally developed) meet a common quality

standard. Members of the EDG community universally agree that EDG product quality is everyone's responsibility, though the definitions and methodologies that must be employed to meet this standard vary group-wide.

2. (U) Recommendation Summary

(S//NF) Following all of the discussion forums, the following recommendation summary has been created to summarize the feedback received from each of the forums and other external reviews.

2.1 Tool Requirement-Related Recommendations

1) **Tool Requirement Mapping to Verification Strategies:** Operators, testers and developers must discuss testing requirements in addition to tool requirements prior to the URD (and SRD) release and acceptance. These discussions must include an assessment of how each requirement will be evaluated during the tool development phase – as each requirement can be verified through a variety of means (e.g., inspection, analysis, demonstrations, testing), where testing represents one method for evaluating a tool's performance, capabilities, etc.

2) **Tool Requirements for Integration Testing:** Operators, testers and developers should also consider (from the beginning) the end-to-end system requirements for a tool (including requirements for integration with other tools). Operational testing scenarios related to a tool's CONOPS should be provided from the beginning, as they will help developers and testers to establish representative development and testing requirements to be imposed on tools.

3) **Tool Requirements for Baseline Test Environment Specifications:** Operators, testers and developers need to discuss the specifications of the environment under which a tool should be examined (e.g., targeted system's hardware, OS, PSP and programs). In general, operators, testers and developers need to decide the scope of this testing, as well as who has the responsibility to perform each examination. For example, for core tools, testers may be required to test a large matrix of hardware, OS, PSP and programs configurations; for other tools, testers may be asked to examine a primary configuration while developers may be responsible for producing automated tests that examine a multitude of others. These automated tests and analysis of their output can then be shared with operators and testers. Since OS-PSP combinations become drivers for IV&V testing, specifying combinations of interest to be tested / demonstrated vs. those that should be analyzed is critical to balancing risk with examination time.

4) **Tool Requirements for Forensics Testing:** Operators, testers and developers need to discuss the specifications / requirements of forensics testing under which a tool should be examined (e.g., targeted system's hardware, OS, PSP and programs). Since forensics testing can take a long time, it is essential to decide what level of forensics testing should be performed on a tool prior to its delivery. General feedback from a multitude to those surveyed indicated that they are not using the results of forensics testing to inform their decision making.

5) **Continue Tool Test Requirements TEMs:** During the writing of this report, operators, testers and developers established TEMs to review testing requirements prior to IV&V's evaluation of tools. This practice should continue, and include a discussion of <u>what</u> requirements should be examined by IV&V and what requirements have been (or will be) verified through other means. These TEMs have already helped clarify tool evaluation requirements and prevent changes to tools after IV&V evaluations have started.

2.2 Testing-Related Recommendations

1) **Establishment of Lexicon Associated with Tool Maturity:** The following definitions should be used when defining the state of a tool:

   i. Tool Evaluation (Eval) Copy – A tool that has not been deemed "releasable" by a developer. This tool has not been submitted to Verification / Acceptance Testing by a developer. Tools at this stage are under the control of the software developer (since the developer has not finished all of the key features of the tool, nor have they cleaned up the code, etc.). Tools at this stage should not be used in any operation.

   ii. Tool Release Candidate (RC) – Developer believes the tool is ready for release, but the tool has not completed Verification / Acceptance Testing, nor has the tool been approved for delivery by COG-EDG ERB

   iii. Released Tool – A tool has finished Verification / Acceptance testing, and was approved for delivery by the joint COG-EDG ERB and accepted for use by the customer.

2) **Developer Understanding of Their Role in Examining a Tool Prior to Independent / External Review:** All developers must understand that they have a responsibility to examine their products prior to an independent and/or external review of their tools prior to producing an RC release of the tool. Developers must embrace the fact that they have a responsibility to do the following:

   i. Perform Unit / System testing: This step includes examining each component / function in their code, verifying that the functionality of each component, examining that the tool will meet all of the requirements outlined in the URD (and/or SRD). Tool release candidates (RCs) should be examined in their entirety by developers or to the best of their ability prior to their release to an independent entity (such as IV&V or the operator).

   ii. Write and run automated testing scripts to verify the tool works as expected

   iii. Develop a Categorization of tool capabilities (as currently provided in TDR documentation)

   iv. Forensic footprint of tool – This includes strings checks and other standard tradecraft best practices

   It is important that developers understand that unit level testing and functional verification testing (i.e., the tool works as intended) is an additional part of a

developer's responsibility, and not solely the responsibility of an independent and/or external reviewer.

3) **Development of Core DART Tests and DART Testing Templates:** A "core" suite of tests should be defined for each product area (i.e., mobile devices, computer implant tool, etc.) based on best practices and lessons learned. If and where possible, this "core" suite of tests should be automated and made available for every developer to use during their unit / functional testing of their tool prior to the release of an RC. Likewise, DART testing templates are needed to provide tool developers a "starting point" for the adoption and development of new DART tests that can be used to support their unit / functional testing evaluations. For example, if a tool is designed to provide a persistence mechanism for other tools in Windows 7, DART test scripts can be composed to look for irregularities when the tool is deployed in a Windows 7 VM while a "simulated user" is surfing the internet. These scripts can be used to examine the tool in the presence of stock / pre-configured OS-PSP combinations automatically – potentially eliminating the need for long OS-PSP characterization matrices composed by IV&V and reducing tool overall test time. While "core" automated tests should not serve as a panacea for all OS-PSP characterizations (since some edge cases should be examined in more detail), they could eliminate some of the manual testing that is performed today and reduce overall testing time.

4) **Evaluation of IV&V Personnel Skill Sets:** A review of IV&V personnel skill sets should be conducted to assess whether current and future IV&V personnel have the necessary skill sets to provide the appropriate level of support for EDG tool support needs, and adjustments should be made to the makeup of the IV&V team to meet EDG support needs. For example, all IV&V personnel should be familiar with core programming languages (e.g. C/C++, Python, etc.) that support the testing of EDG tools – to include building custom automated scripts and the basic troubleshooting of tools.

5) **Recommendation that New EDG Staff (regardless of division) Spend a Period of their Time in IV&V as Tool Testers:** In many software companies and organizations, new employees spend a percentage of their time as members of the organization's test team. This activity promotes an environment where employees are exposed to test and development best practices, items to watch for, etc. while promoting the incorporation lessons learned and best practices into their daily activities and products.

6) **Re-assessment of Tool Forensic Examinations / Testing Philosophy:** Current forensic examinations take far too long as compared to that which is performed by ECG/AFD when requested by developers. Digital forensic examiners from IV&V and ECG/AFD should meet to "exchange notes" on their processes and determine what is the "bare minimum" forensic tests that must be done for any given tool delivery. These results should be published and provided to both operators and developers as a mechanism for discussion and use in defining the depth by which tools should be examined forensically prior to use. As such, a re-examination of "what forensics testing should be performed on a tool" through discussions with operators, developers, testers and EDG/AFD personnel should be prioritized during requirements discussions – in order to both provide developers / operators options for different levels of forensics tool evaluations.

7) **Re-assessment of Considerations when SED/IV&V Personnel attend External Contractor FAT testing events:** Because SED/IV&V resources are limited, special consideration should be required where a COTR/PM is able to request IV&V support for external contractor tool testing efforts. More specifically, EDG SETA and SI should serve as an independent representative (to the COTR/PM of a contract and the external contractor) tasked with assessing the performance of a tool and its viability for delivery.

8) **Establishing "Core" / "Regression" Test Strategies for Long-term Tool Development Efforts:** Every tool development effort that requires continued O&M support should consider the need for developing a long-term testing strategy to be used during verification / acceptance testing. For long-term tool development and O&M efforts supporting multiple product versions, developers should consider developing a "core" set of tests that can be used to evaluate the core functionality of the tool, and a set of additional tests used to verify the changes that were made to the tool. This type of test strategy could be helpful in reducing overall product testing time, while also helping teams improve unit, functional, integration and acceptance tests.

9) **Repurpose Current IV&V Checklist:** The current "IV&V Test Readiness Review Sheet" should be changed to promote a discussion between Developers and EDG Management (specifically Branch Chiefs) on whether tools are ready to be released as RCs. This discussion should incorporate the testing strategy that has and will be used to evaluate the tool during its development. This check (by a branch chief) could help to resolve situations where tools are not properly examined prior to their handoff to IV&V, and likewise, it could also help with IV&V testing prioritization of a branch's tools in IV&V or other tool testing requirements (which may not include IV&V but rather operational testing instead). The revised checklist promotes an environment where both the Developers and Branch Chiefs are held accountable for the quality of the tools leaving the branch, the application of resources applied in testing all tools, and whether tools are ready (and could be conditionally deployed) for operations.

10) **Establishment of a QA Process:** While some tools require a full IV&V test cycle, there are other tools which, following consultations between operators and developers, may be deployed after the production of an RC. These cases include (but are not limited to) situations where EDG does not possess a test environment that is representative of the operational environment. For these tools, an independent Quality Assurance process could be used to examines / verifies two things prior to the tool's use in an operational environment:

    i. The developer assessment that an RC is "ready for release." This assessment, performed by a developer, should incorporate a short, simple checklist of items based on a "common lessons learned" list of items that should be examined before a tool is transitioned to a QA individual for review.
    ii. A list of the "core" regression tests a tool must pass before delivery. This list should be defined by the developer and agreed upon by the operator as part of the requirements vetting process. These tests may be performed as part of the creation of an RC (i.e., before a tool is transitioned to QA for review). This list of tests should include a shortened forensics examination (e.g., strings checks, etc.)

11) **Establishment of Peer Review Practices:** Every development project should have a few people identified as individuals willing to peer review all source code contributions. Currently, peer reviews of source code quality and design are conducted infrequently, and commonly at the end of a delivery cycle when commentary cannot be incorporated into the project. Modern development practices strongly promote regular code review during the development process by all members of a development team. While peer

reviews do not have to be done by people in the same branch, providing developers with a list of individuals willing to support peer reviews would also be helpful in improving the quality of deliverables. Some branches in AED have already embraced this methodology, though a wider adoption of the practice is encouraged.

2.3 Tool Delivery-Related Recommendations

1) **Discussing End-to-End Development Methodology at Start of Project:** Developers, PM/COTRs, operators, testers, etc. should discuss the development methodology / approach applied to a given project at the start of the project. The methodology used should coincide with the project's complexity or maturity and adapt as necessary. Some projects may require multiple RCs to be released to testing because of project's complexity or maturity (i.e., first-of-its-kind efforts). While multiple RCs can be costly, there are situations where delivering multiple RCs to test is not "bad" and these situations should be discussed accordingly. Likewise, developers, operators, and testers should discuss the development methodology / approach applied to a given project up-front to ensure that IV&V testers are examining test cases that are relevant to operational CONOPS and environments.

2) **EDG Management Consent Prior to Tool RC use in Operations:** Unless previous consent is provided by EDG management (e.g., temporary/limited approval can be included as part of repurposed IV&V Checklist recommendation on previous page), Tool RCs should not be deployed by COG or any other operational entity without prior consent from EDG. Consent to deploy an RC should not be granted by a COG/NOD Mission Manager without consent from EDG management, as EDG maintains some risk in the delivery and operational deployment of tools developed using EDG equities. In the event that a tool is deployed outside of the TDR process, the deployed version of the tool should be immediately be placed under CM control and a TDR of the tool should take place immediately.

3. (U/FOUO) EDG Feedback -- "The State of Testing in EDG"

(S//NF) Over a three-week period during the months of April & May 2014, there were three separate group discussions on the "State of Testing in EDG." Each discussion was intended to provide a forum for internal developers, project managers for external development contracts, and EDG's internal testing team to discuss testing / tool lexicon, testing practices & methodologies, customer feedback, recommendations for improvement, and other product delivery-related feedback.

(S//NF) During each forum, the following questions were posed to attendees to initiate discussions on testing:

a) Which parts of EDG have the responsibility to test EDG products?
b) Should different branches of EDG be responsible for certain types of testing, based on areas of expertise?
c) How can EDG add the most value to testing deliverable products with minimal impact to operational timelines?
d) What technical or industry practices can be adopted by EDG to increase testing value?
e) In what areas can testing be automated to increase efficiency?
f) How can EDG maximize the allocation for the limited SED/IV&V resources?
g) Where can SED/IV&V add the most value to the product delivery for each EDG division?

The following sections summarize inputs, feedback and comments from each of the discussions.

3.1 (S//NF) Testing Forum #1: Focused toward Internal Development Efforts

(S//NF) The first EDG testing forum was held on 14 March 2014. This forum focused on testing methodologies and processes used in supporting predominantly internal (EDG/AED) development efforts. While the discussion was scheduled for one hour, the forum lasted well over two hours, where people gathered after the forum to chat about issues and concerns individuals experienced during the tool delivery process.

(S//NF) The following paragraphs summarize the notes taken during the first forum. While there were many points discussed during this forum, these notes provide a summary of the discussion and convey the tenor and ideas presented during the forum.

- Overall, individuals who participated in the forum agreed that there is a need for more and/or “better” testing, which is completed faster to meet operational deadlines;

however, all involved recognize there are trade-offs (i.e., “more, better, faster” may require additional testing resources, time becomes a factor when we get “more & better” testing, and so on).

- There was a brief discussion about requirements, how they are provided to EDG, and the impact to testing. In short, requirements should not be provided piece-meal – if there is an end-to-end system concept / need, this information should be provided to developers from the start so that the system and the testing requirements are assessed from the beginning.

- Throughout the forum, there was broad acknowledgement for the need to standardize the lexicon associated with testing. For example, some developers used the terms “evaluation copy” and “release candidate” interchangeably, while others pointed out important differences in each version. Likewise, there was not broad agreement on the definitions of “unit testing”, functional testing, integration testing, acceptance testing, operational evaluation / testing – and at the beginning of the forum there was not broad agreement on who had the responsibility for each testing phase.
- A large portion of the forum focused on automated testing and the void that DART could assist developers in filling in regard to functional and unit level testing. There was broad agreement that DART and other automated testing platforms have the potential to assist developers with test-driven development posture. For example, if this automated testing is used throughout the development of a product, it should help to reduce the time to delivery (given that IV&V testing could concentrate on edge cases, rather than PSP-OS combinations, which is the current driver for length time it takes for IV&V to complete test cases). Additionally, since operators are also getting a version of the DART system, it could allow operators to use the same test tools / setup used in development when pursuing operational evaluations – and potentially feed test cases back to EDG developers to be used during tool developments. That said, automated testing requires IV&V participation early in test development process (mainly so that they understand the tool under development and testing in process). Likewise, while DART is very useful, there will also be a need to have “bare metal” testing –esp. when specified by the operators.

- Additionally, there was broad acknowledgement that automated testing is not a panacea that could be applied to all tools. As such, if developers write automated test procedures, there will remain a need for someone to look over the tests and/or the tool; otherwise, if (for example) developers write test scripts and IV&V just runs them, the

IV&V person would not serve as an independent reviewer of the tool. That said, for some tools, the automated tests may suffice and the IV&V role would be more that of Quality Assurance (QA) to make sure that the test coverage proposed / implemented was appropriate. In the event that the test coverage was not appropriate, the "QA" person would propose additional procedures / testing that must be performed prior to release.

- There was broad agreement that both customers and developers must be part of the decision for when a tool should be delivered, and the customer should be involved with acceptance testing to set and/or adjust expectations appropriately. As automated testing becomes more prevalent, it is important that a person review the test results generated from automated tests – and, preferably, these results should be shared with both the customer and the developer prior to publishing.

- Concerns were raised over the number of testers available to examine specialized tools or capabilities that are designed for "unique" platforms. For example, mobile and embedded tools may require specialized skills / knowledge. With a reduction of IV&V testing resources, there is a need to ensure that all IV&V personnel have a minimum set of skills (e.g., Python programming – since DART scripts are written in Python, C/C++ programming, etc.).

- All attendees agreed that operator involvement in testing was important; however, there were concerns raised about providing "evaluation" software to operators that EDG later finds out was used in an operation. Hence, the following definitions were recommended to assist developers / testers / operators in understanding the maturity of a given tool:
    - Tool Evaluation (Eval) Copy – Developer must maintain control of the tool (since the developer has not finished all of the key features of the tool, nor have they cleaned up the code, etc.)
    - Tool Release Candidate (RC) – Developer believes the tool is ready for release, but the tool has not completed Verification / Acceptance Testing
    - Released Tool – Finished acceptance testing, approved for delivery by IOC ERB

Given these definitions, developers noted that "Eval Copies" of tools should never be used operationally because the developer did not complete the tool. "Release Candidates" or RC versions of tools could be used operationally, but there are inherent risks the operator is undertaking in using the RC without Verification / Acceptance testing having been completed. Regardless, the forum recommended that every RC should undergo some version of Forensics examination prior to release (see another bullet below), and stressed the importance of customer involvement in acceptance testing.

- With the definitions provided above, there were concerns raised about the COG/NOD decision-making process currently used to evaluate when an RC should be used in a particular operation prior to a tool's official release. Currently, NOD Mission Managers can approve an operator's request to deploy a RC version of a tool – which means that EDG is not involved in (and may not even know about) the decision process to deploy a tool.

- Likewise, there were concerns raised about the length of time tools were taking to get through IV&V – which was cited as the main reason Mission Managers would take the risk to deploy a tool prior to its release. For example, one person asked if it was necessary for IV&V to perform all OS-PSP combinations if the NOD operator was planning to re-do the testing prior to deployment, given that OSs and PSPs are updated all of the time (esp. since OS-PSP combinations were the driver for the length of time a tool is in IV&V testing). Another person noted that IV&V Forensics testing takes many times longer than the amount of time it take ECG/AFD to analyze a tool forensically.

(S//NF) After the forum, the following recommendations were gathered from the minutes. Again, while there were many points discussed during this forum, these recommendations are a summary of the ideas posed throughout the forum.

1. Tool Requirement-related Recommendations
    a. Since tool requirements play a large role in tool development and testing, it is important for operators, testers and developers to discuss requirements prior to their acceptance.

b. Tool requirements for end-to-end system (including requirements for integration with other tools) should be provided from the beginning, as they drive development and testing requirements imposed on tools.
c. Operators need to spend time thinking through the specifications for a given requirement – especially the targeted system's hardware, OS, PSP and programs. Likewise, it is important for testing purposes to specify the exact combinations of interest that will be examined during testing. Since OS-PSP combinations become drivers for IV&V testing, specifying combinations of interest to be tested / demonstrated vs. those that should be analyzed.
d. Requirements should include information about the operational scenario driving the CONOPS for a tool – so as to drive testing scenarios for tools. TEMs / requirements discussions accompanied with demonstrations are very helpful for both operators and developers in understanding both CONOPS and tool requirements.
e. Lastly, operators and developers need to decide up front which requirements truly require demonstrations / testing and those which an inspection / analysis is suitable. Currently, tool requirements from COG do not specify a suitable verification strategy, nor are all requirements "demonstrable" or "testable." As such, many developers do not explicitly decide which requirements have already been verified prior to IV&V testing, thus driving additional testing.

2. Testing-related Recommendations
   a. All developers must understand that they have a role in testing their products. Developers must embrace the fact that unit level testing (i.e., examining the code components and the functionality each component provides) and functional verification testing (i.e., the tool works as intended) are their responsibility. Tool release candidates (RCs) should be examined in their entirety prior to their release to an independent entity (such as IV&V or the operator).
   b. A "core" suite of tests should be defined for each operating environment (i.e., network tests, mobile device tests, computer implant tool tests) based on best practices and lessons learned and, if possible, this "core" suite of tests should be automated and made available for every tool developer to use during unit / functional testing. For example, if a tool is designed to provide a persistence mechanism for other tools in Windows 7, DART test scripts can be composed to look for irregularities when the tool is deployed in a Windows 7 VM while a "simulated user" is surfing the internet. These scripts can be used to examine the tool in the presence of stock / pre-configured OS-PSP combinations automatically – potentially eliminating the need for long OS-PSP characterization matrices composed by IV&V and reducing tool overall test time. While "core" automated tests should not serve as a panacea for all OS-PSP characterizations

(since some edge cases should be examined in more detail), they could eliminate some of the manual testing that is performed today and reduce overall testing time.

c. Tool forensic examinations must be re-addressed by EDG. Current forensic examinations take far too long as compared to that which is performed by ECG/AFD on request by developers. Digital forensic examiners from IV&V and ECG/AFD should meet to "exchange notes" on their processes and determine what is the "bare minimum" forensic tests that must be done for any given tool delivery. These results should be published and provided to both operators and developers as a mechanism for discussion and use in defining the depth by which tools should be examined forensically prior to use.

d. The following definitions should be used when defining the state of a tool:

   i. Tool Evaluation (Eval) Copy – Developer must maintain control of the tool (since the developer has not finished all of the key features of the tool, nor have they cleaned up the code, etc.)

   ii. Tool Release Candidate (RC) – Developer believes the tool is ready for release, but the tool has not completed Verification / Acceptance Testing

   iii. Released Tool – Finished acceptance testing, approved for delivery by IOC ERB

e. There is a need to ensure that all IV&V personnel are familiar with core programming languages that support the testing of EDG tools – to include building custom DART scripts and the basic troubleshooting of tools. A review of IV&V personnel skill sets should be conducted to assess whether IV&V personnel have the necessary skill sets to provide the appropriate level of support for EDG tool support needs, and adjustments should be made to the makeup of the IV&V team to meet EDG support needs.

3. Tool Delivery-related Recommendations

   a. QA Checks – Prior to the creation of an RC, there is a need for the following:

      i. A developer assessment determining that an RC is "ready for release." This assessment, performed by a developer, should incorporate a short, simple checklist of items based on a "common lessons learned" list of items that should be examined before a tool is transitioned to IV&V and/or an operator.

      ii. A list of the "core" regression tests a tool must pass before delivery. This list should be defined by the developer and agreed upon by the operator as part of the requirements vetting process. These tests may be performed as part of the creation of an RC (i.e., before a tool is transitioned to IV&V).

b. Unless previous consent is provided by EDG management, Tool RCs should not be deployed by COG or any other operational entity without prior consent from EDG. Consent to deploy an RC should not be granted by a COG/NOD Mission Manager without consent from EDG management, as EDG maintains some risk in the delivery and operational deployment of tools developed using EDG equities. In the event that a tool is deployed outside of the TDR process, the deployed version of the tool should be immediately be placed under CM control and a TDR of the tool should take place immediately.

3.2 (S//NF) Testing Forum #2: Focused toward External Development Efforts

(S//NF) The second EDG testing forum was held on 31 March 2014. This forum focused on testing methodologies and processes used in supporting predominantly external (EDG/ESD) development efforts. While the discussion was scheduled for one hour, the forum also lasted well over two hours, where people gathered after the forum to chat about issues and concerns individuals experienced during the tool delivery process.

(S//NF) The following paragraphs summarize the notes taken during the second forum. While there were many points discussed during this forum, these notes provide a summary of the discussion and convey the tenor and ideas presented during the forum.

Note: Since there are commonalities with the first forum's discussion, only new points that were not previously addressed in Section 3.1 will be mentioned below.

- Again, in the second forum, there was broad agreement that both customers and developers must be part of the decision for when a tool should be delivered, and the customer should be involved with acceptance testing to set and/or adjust expectations appropriately. Additionally, there was broad agreement that operators and developers must communicate regularly about the tool's requirements, intended CONOPS, and desired testing scenarios. Communication between operators and developers was cited as critical to a tool's success.

- Part of the discussion revolved around the difference between a Factory Acceptance Test (FAT) and Operational Testing. In many cases, Operational Testing (conducted by operators) may be performed without a test plan and centers around the key functions a particular operator group may care about. That said, forum attendees noted that

representative operational tests could be incorporated into a FAT with enough knowledge of the CONOPS and intended use cases.

- Attendees noted that external contractor-developed tools may have more test resources examining a given tool than in-house tools. That said, attendees noted that with additional resources, requirements generation and subsequently test planning is very important – as small changes to requirements at the end of a development effort can result in large financial costs and time delays affecting the delivery of a product.

- A portion of the forum focused on automated testing. Since automated testing for a given contract delivery involves added overhead to support the effort, a number of factors must also be considered before investing in an automated testing effort. For example, if the tool chain is one of many tools developed by the contractor that uses the automated test suite or there are large matrices of test cases involving (for example) several OS-PSP-Program-Configuration combinations, then test automation makes sense for a given tool. That said, the scale and longevity of the effort is important in deciding whether test automation is worth the investment for a given tool.

- The forum also discussed whether SED/IV&V test personnel should be required to attend a FAT associated with external contractor-developed tool. While there were differing opinions on this point, most attendees agreed that the definition of what is required during the FAT is an important consideration when considering IV&V's involvement. Since EDG SETA and SI can also serve in an independent role to the COTR/PM of a contract and the external contractor, it is not necessary to have an SED/IV&V person in attendance to maintain "independence." Additionally, a COTR/PM should be able to request IV&V support for external contractor tool testing efforts, esp. in regard to EDG "core tools" – since many of these tools are integrated with other EDG tools (both in-house developed and external-contractor developed). In all cases, COTR/PMs can consult with IV&V personnel to gain an understanding of test practices, etc. Including IV&V in test discussions for any product at the end of the development cycle promotes limitations and potential issues in the tool develop process (e.g., a tool's requirements can impose testing limitations and testing lessons learned can inform "SMART" tool requirements).

- This discussion also led to the topic of operator-defined testing. Many times, operators have defined whether IV&V resources should be applied to a particular project without EDG input. Participants noted that COG's request should be that an independent person and/or group should be used to examine the end product, rather than define that IV&V resources should be used for a particular project. Additionally, operators should be involved in the testing of a product especially in situations where operators had strong opinions on the tests used to examine a particular project.

- Forum attendees noted that SED/IV&V and Contractor test personnel never take the place of a customer during FAT testing. Multiple individuals questioned whether project officers include mappings of SRD requirements to IMIS requirements (or vice versa). Additionally, several people noted the importance of including operators in (for example) TEMs, project milestone meetings, and test planning sessions to ensure that operator needs were included in the development process.

- Several development cycle models and processes were discussed during the meeting – to include the "V" model, spiral, and agile development cycles. In each case, the importance of project planning was emphasized. For example, one person noted that "… if there are changes to tool development requirements at or near TRR, we feel like we failed…" even in cases where an operator changes the tool's CONOPs late in the development process. While people noted that the development process can be rigid at times, it is important to understand whether a feature is "necessary" and truly affects the end delivery's functionality or if it's a "nice to have." Since many contractors dry run their products many times before delivery, peer and other types of reviews (from the code level to system integration / functional testing pre-TRR) should be used to understand the tool's capability prior to acceptance testing. A "trust but verify" should be used to verify a contractor's delivery – as no one should accept a tool's performance "at face value" prior to acceptance. That said, part of this discussion focused on testing strategies and whether it is necessary at a Government-witnessed Factory Acceptance Test (FAT) to "re-do" all of the contractor's tests. Most people agreed that a tailored approach (developed in concert with the program's RVTM) should be applied – where a set of core regression tests are defined for a tool and additional tests are run based on

changes or a tool's functionality. People also noted that for some tools, the full test suite is "small enough" to run against a tool each time (esp. for "one-off" tools).

- Lastly, some participants expressed a concern that there was a stigma attached to projects that had multiple release candidate (RC) versions that were being tested. In summary, forum participants noted that for some projects multiple RCs could be expected given a project's complexity or maturity. That said, there was an acknowledgement that the testing approach for projects where there were multiple RCs expected should be examined. Additionally, developers, PM/COTRs, etc. should discuss these situations up-front and consider a development approach that coincides with the project's complexity or maturity and adapt as necessary.

(S//NF) Following the forum, the following additional recommendations were gathered from the minutes. Again, while there were many points discussed during this forum, these recommendations are a summary of the unique ideas posed throughout the forum.

1. Tool Requirement-related Recommendations
    a. Since tool requirements play a large role in tool development and testing, it is important for operators, testers and developers to discuss requirements – and their impacts on testing – prior to their acceptance by EDG.
    b. All developers should acknowledge and understand the testing requirements imposed by tool requirements at SRR. More specifically, tool developers and/or PM/COTR for projects should map IMIS requirements to SRD requirements and decide on the appropriate verification strategy. This also includes a discussion of the complexity and maturity of a given project's deliverables from the outset of a project – so as to inform the testing requirements and strategy for a product.

2. Testing-related Recommendations
    a. SED/IV&V resources should not be mandatory participants during acceptance testing of all EDG products; instead, all project deliveries should be independently examined prior to delivery. More specifically, EDG SETA and SI should serve as an independent representative (to the COTR/PM of a contract and the external contractor) tasked with assessing the performance of a tool and its viability for delivery. In special cases, a COTR/PM can request IV&V support for external contractor tool testing efforts – specifically in regard to EDG "core tools" since many of these tools are integrated with other EDG tools (both in-house developed and external-contractor developed).
    b. For all tools, operator-defined test scenarios should be incorporated into acceptance testing. For externally-developed tools, operational test scenarios should be developed as part of the requirements process, but also examined as part of FAT testing. Operators should participate in these operator-defined tests for product validation purposes and to gain familiarity with a tool.

    c. Every tool development effort that will require continued O&M support should consider the need for developing a long-term testing strategy to be used during acceptance testing. For long-term tool development and O&M efforts supporting multiple product versions, developers should consider developing a "core" set of tests that can be used to evaluate the core functionality of the tool, and a set of additional tests used to verify the changes that were made to the tool. This type of test strategy could be helpful in reducing testing time, while also helping teams improve unit, functional, integration and acceptance tests.

3. Tool Delivery-related Recommendations
    a. Developers, PM/COTRs, etc. should discuss the development methodology / approach applied to a given project up-front coincides with the project's complexity or maturity and adapt as necessary. Some projects may require multiple RCs to be released to testing because of project's complexity or maturity (i.e., first-of-its-kind efforts). While multiple RCs can be costly, there are situations where delivering multiple RCs to test is not "bad" and these situations should be discussed accordingly.

3.3 (S//NF) Testing Forum #3: Focused toward IV&V testing methodologies

(S//NF) Following discussions focused on EDG/AED and EDG/ESD development efforts, a third EDG testing forum was held on 3 April 2014. This forum focused on EDG/SED/IV&V testing methodologies and processes used by IV&V to supporting EDG's testing efforts. While the discussion was scheduled for one hour, the forum also lasted well over two hours, where people gathered after the forum to chat about issues and concerns IV&V members experienced during the tool testing and delivery process.

(S//NF) The following paragraphs summarize the notes taken during the third forum. While there were many points discussed during this forum, these notes provide a summary of the discussion and convey the tenor and ideas presented during the forum.

Note: Since there are commonalities with the first two forums' discussion, only new points that were not previously addressed in Section 3.1 and 3.2 will be mentioned below.

- As with the previous forums, there was broad agreement that customers, developers, *and testers* must be part of the decision for when a tool should be delivered, and the customer should be involved with acceptance testing to set and/or adjust expectations appropriately. Additionally, there was broad agreement that operators, developers *and testers* must communicate regularly about the tool's requirements, intended CONOPS, and desired testing scenarios. Communication between operators, developers, *and testers* was cited as critical to a tool's success.

- Participants indicated that there have been many cases (in the past) where tools provided to IV&V have not undergone testing prior their hand off to IV&V. For this reason, IV&V developed a worksheet that must be submitted with tools indicating the conditions by which tools have been examined prior to their acceptance in testing by IV&V. Part of this worksheet was designed to assure IV&V testers that (for example) the developer performed unit and/or functional tests on the tool, the tool requirements were assessed, and any issues were stated to testers prior to IV&V testing. Likewise, the worksheet served to document the software's release candidate that is to be under test.

- One reason for the worksheet also helped IV&V keep track of the version of a tool and its associated documented requirements because of the many cases where tools and/or their requirements (due to requirements creep, operational changes, etc.) have changed post-IV&V acceptance to test the tool. These changes in many cases have caused testers to re-start testing "from scratch" due to the many PSP-OS combinations requested by operators during testing. As a result, IV&V implemented a process by which developers, operators and tester get together to discuss the testing that will occur on the tool – mainly to ensure that all of the requirements / expectations of testing are taken into account before IV&V begins the testing process. Participants noted that tool demonstrations and providing information on the tool's intended operational environment helps testers in their test activities.

- Attendees noted that operators have indicated the importance of independence between IV&V personnel and developers – as IV&V testing serves as an independent examination of EDG (contractor/internal) developer-produced tools. As such, COG commonly requests a number of PSP-OS combinations that must be used during testing. These environments take time to setup – and as such, the more combinations, the longer the testing period will take for a given tool.

- Attendees noted that it is important for IV&V personnel to attend some ESD-contractor developed tool's Factory Acceptance Tests (FAT) because it provides insight into tools that IV&V needs to use during integration testing efforts. That said, not all FATs require the need for IV&V personnel (esp. those where the IV&V individual is just checking off requirements / test steps). Attendees recommended that IV&V personnel should meet with a person requesting their involvement in a FAT prior to attending, and a decision should be made prior to the FAT whether an IV&V member should attend (e.g., in cases where test activities are for tools that will be used / integrated with other EDG tools during IV&V test efforts).

- A portion of the discussion focused on IV&V's forensics examination of tools. Participants noted that while almost every tool sent to IV&V requests a forensics examination, many times the requests do not include an expectation or calibration on what the operator / developer is looking for out of the examination. Forensics

examinations can take a long time – mainly because of the need to setup a clean testing baseline, install the tool and evaluate, and remove the tool and evaluate, which can take up to three days per examination of a tool for a given operating system. Therefore, it is important for testers understand the requirements (e.g., strings check, evidence of the tool in volatile memory, disk, or otherwise, etc.) and concerns of the operator to size the forensics test effort – allowing testers to focus on items of interest to both the developer and operator.

- A portion of the forum also described the discrepancy reporting process during testing. Since developers and testers regularly communicate throughout the testing process, developers want to correct discrepancies as they arise. While understandable, a new release candidate many times causes testers to restart the entire test process. As a result, the developer's checklist helps to document the number of release candidate / revisions a tool has undergone as it goes through the testing process (though sometimes developers do not fill it out when the drop their code in the CM dropbox). Additionally, because of the fact that a number of tools have been revised as they have gone through the testing process, forensics testers wait to test a given RC until an IV&V testing is nearly completed because of the resources and timing that is required to complete a forensics evaluation.

(S//NF) Following the forum, the following additional recommendations were gathered from the minutes. Again, while there were many points discussed during this forum, these recommendations are a summary of the unique ideas posed throughout the forum.

1. Tool Requirement-related Recommendations
    a. Since tool testing requirements play a large role in testing process, it is important for operators, testers and developers to discuss testing requirements prior to the testing of a tool.
    b. All developers should acknowledge and understand the testing requirements – including the depth of forensics testing that will be imposed on the tool.

2. Testing-related Recommendations

a. There is a need to establish an IV&V Test Readiness Review Sheet that is reviewed by an EDG management representative prior to a tool being placed into IV&V testing, where SED/IV&V resources are being applied against the tool. This check (by a branch chief) could help to resolve situations where tools are not properly examined prior to their handoff to IV&V, and likewise, it could also help with IV&V testing prioritization of a branch's tools in IV&V or tool testing requirements.
b. While most ESD-contractor FAT tests do not require participation by IV&V personnel, it is helpful for IV&V resources that will be regularly using an ESD-contractor developer tool for integration/compatibility testing to be involved with the tool's FAT testing product validation purposes and to gain familiarity with a tool.

c. There is a need to re-assess how IV&V forensics evaluations are performed on tools. It is clear that there are some core forensics tests that should be employed on tools, but forensics testing requirements for each tool should be examined as part of the pre-testing TEM. A technical interchange between ECG/AFD and IV&V forensics personnel should occur to determine if (for example) there are lessons learned that can be applied to IV&V testing examinations.
d. There is also a need for developers and testers to talk about core-testing requirements prior to the beginning of an IV&V test effort. In the event that there is a RC-update to a tool under test, a tool should not necessarily undergo a complete "redo" of all IV&V-related examinations. This should be discussed prior to IV&V testing, and include discussions on what testing has already been employed against a tool, what should be re-examined during IV&V testing, etc.

3. Tool Delivery-related Recommendations
   a. Developers, operators, and testers should discuss the development methodology / approach applied to a given project up-front to ensure that IV&V testers are examining test cases that are relevant to operational CONOPS and environments.

4. Abbreviations

(U) The Acronyms/Abbreviations used in this document are shown in Table 5 -1.

Table 5-1: (U) Acronyms/Abbreviations

Acronym/Abbreviation	Term
A	Analysis
API	Application Programming Interface
APN	Access Point Name
C2	Command and Control
CID	Cell IDentification
CONOP	CONcept of OPerations
D	Demonstration
DNS	Domain Name System
ECG	Exploitations Capabilities Group
GPS	Global Positioning System
HTTPS	HyperText Transfer Protocol Secure
I	Inspection
ICD	Interface Control Document
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IOC	Information Operations Center
IP	Internet Protocol
LAC	Location Area Code
LP	Listening Post
MCC	Mobile Country Code
MMS	Multimedia Messaging Service
MNC	Mobile Network Code
MSISDN	Mobile Subscriber ISDN Number
PIN	Personal Identification Number
PN	Poseidon
RF	Radio Frequency
RVM	Requirements Verification Matrix
SIM	Subscriber Identification Module
SMS	Short Messaging Service
SRD	System Requirements Document
T	Test
TA	Timing Advance
URL	Uniform Resource Locator
USG	United States Government
USSD	Unstructured Supplementary Services Data
VOIP	Voice Over IP
XML	Extensible Markup Language

This table in its entirety is classified as SECRET//NOFORN