

SECRET//NOFORN

—

Table of Contents

(U)Executive Summary.....	4
(U) Current State.....	7
(U) EDG's System Development Lifecycle (SDLC).....	7
(U) Environments.....	8
(U) Future Plans.....	10
(U) Challenges and Observations.....	12
(U) Challenges.....	12
(U) Need for More Functional Integration and Operational Validation in IV&V Test (Challenge 1)....	12
(C) Need for More Forensic/Signature Tests Executed Against All EDG Projects (Challenge 2).....	12
(U) Need for More Communication Between All Stakeholders (Challenge 3).....	13
(U) Need for More Consistent Testing Across Projects and Branches (Challenge 4).....	14
(U) Need for Shorter IV&V Testing Timelines (Challenge 5).....	15
(U) Need for DART Automation Development Strategy (Challenge 6).....	15
(U) Need for an IV&V Database Replacement (Challenge 7).....	16
(U) Observations.....	16
(U) Forensics Testing Outreach (Observation 2).....	17
(U) Distribution of Test Environments (Observation 3).....	17
(U) Recommendations.....	18
(U) Increase Functional Integration and Operational Validation Testing in IV&V (Recommendation 1).18	18
(U) Challenge Addressed: Challenge 1.....	18
(U) Immediate Actions.....	18
(U) Mid-Term Actions.....	19
(U) Implementation.....	20
(U) Implement a Menu of Testing Services to Focus Testing Baselines (Recommendation 2).....21	21
(U) Challenges and Observation Addressed: Challenges 2, 4 and 5, Observation 2.....21	21
(U) Immediate Actions.....	21
(U) Mid-term Actions.....	22
(U) Long Term Actions.....	23
(U) Implementation.....	23

(U) Improve Communications Between All Stakeholders (Recommendation 3).....	24
(U) Challenges Addressed: Challenges 3, 5 and 7.....	24
(U) Immediate Actions.....	24
(U) Mid-Term Actions.....	25
(U) Long Term Actions.....	25
(U) Implementation.....	26
(U) Focus Dedicated DART Automation Support (Recommendation 4).....	26
(U) Challenges Addressed: Challenges 5 and 6.....	26
(U) Immediate Actions.....	27
(U) Mid-Term Actions.....	27
(U) Long Term Actions.....	28
(U) Implementation.....	28
(U) Connect Test Environments Through a Central Shared Services Environment (Recommendation 5).....	29
(U) Challenge and Observation Addressed: Challenge 4 and Observation 3.....	29
(U) Immediate Actions.....	29
(U) Mid-Term Actions.....	30
(U) Long Term Actions.....	30
(U) Implementation.....	30
(U) Roadmap.....	31
(U) EDG Vision.....	33
(C) Appendix 1: EDG SDLC Process.....	35
(C) Appendix 2: Example Walkthrough of EDG Vision Process Flow.....	37
(C) Appendix 3: Stakeholders Providing Feedback.....	3
(U) Figure 1: Vision for EDG SDLC Environments and Processes.....	6
(S) Figure 2: EDG SDLC Environments.....	8
(U) Figure 3: Recommendation 1 Roadmap.....	20
(U) Figure 4: Example Test Services Menu.....	21
(U) Figure 5: Recommendation 2 Roadmap.....	23
(U) Figure 6: Recommendation 3 Roadmap.....	26
(U) Figure 7: Recommendation 4 Roadmap.....	28
(U) Figure 8: Recommendation 5 Roadmap.....	31
(U) Figure 9: Overall Recommendation Implementation Roadmap.....	32
(U) Figure 10: EDG Development and Test Environment Vision.....	34

Y

(C) Table 1: Challenges and Observations Table.....	4
(S) Table 2: Recommendations Table.....	5
(S) Table 3: EDG and COG Environments.....	9

(U)Executive Summary

(U) This review was conducted to provide an independent look at the overall enterprise test and evaluation processes that support the Engineering Development Group (EDG) and help determine a way forward for improving those processes. The goal of this effort was to have an independent reviewer: interview key stakeholders in EDG's development and test processes; review documentation relevant to those processes; identify areas where EDG could improve their processes; and provide recommendations for how to improve them. During this review 21 stakeholders were interviewed from multiple branches within EDG, including SED, AED, and ESD as well as with the customer organization COG and they are identified in Appendix 3. This report consists of three main parts:

- (U) A review of the high level System Development Lifecycle (SDLC) at EDG, the different environments involved in those processes, and future plans for improving EDG processes.
- (U) Challenges and observations uncovered during interviews with stakeholders and reviews of documentation.
- (U) Recommendations for addressing these challenges and observations as well as a roadmap and vision for moving EDG test processes and environments into the future.

(U) Overall, EDG's SDLC and test processes have been successful, as evidenced by the large number of systems/tools and updates delivered each year (400+). Over time, the number of systems/tools and updates in development have increased, the amount of dedicated test resources has decreased, and the sophistication of the target's system security has improved. These evolving changes will require EDG to adapt their SDLC and test processes to keep pace and continue to deliver quality products while maximizing efficiencies and time to market.

(U) Through interviews with stakeholders and documentation reviews, a number of challenges and observations were found. These challenges focus on problems that are currently affecting EDG's development and test efforts, where observations focus on areas that could be problematic in the future. The table below provides the high level Challenges and Observations, with detailed discussion found in the (U) Challenges and Observations Section.

(C) Table 1: Challenges and Observations Table

Challenges	
1	(U) Need for More Functional Integration and Operational Validation in IV&V Test
2	(C) Need for More Forensic/Signature Tests Executed Against All EDG Projects
3	(U) Need for More Communication Between All Stakeholders
4	(U) Need for More Consistent Testing Across Projects and Branches
5	(U) Need for Shorter IV&V Testing Timelines
6	(U) Need for DART Automation Development Strategy
7	(U) Need for an IV&V Database Replacement
Observations	
1	(U) Forensics Testing Outreach
2	(U) Distribution of Test Environments

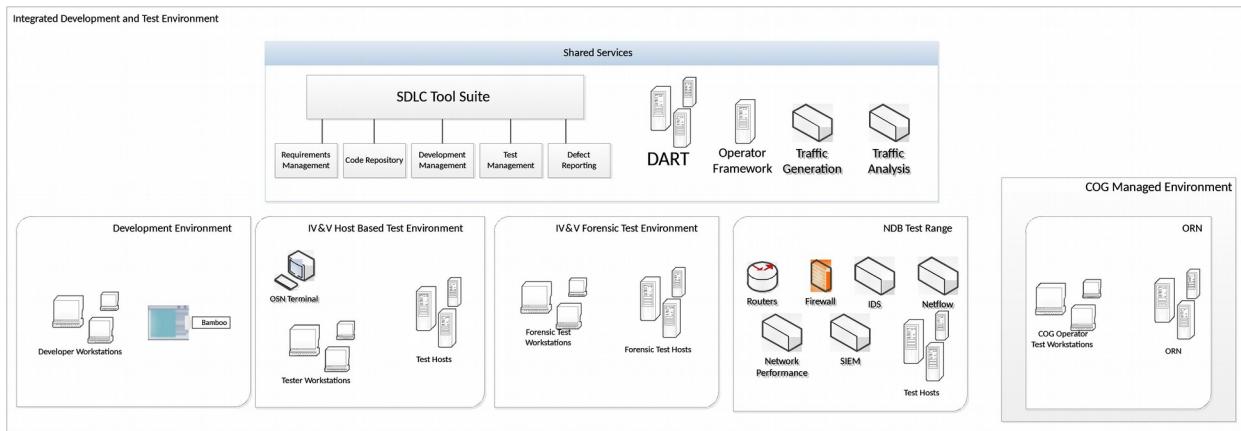
(U) To mitigate these challenges and observations, a set of recommendations has been developed, with specific immediate, mid-term and long term actions. The Table 2 below provides the high level overview of these recommendations, as well as the Challenges and Observations they address.

SECRET//NOFORN

(S) **Table 2: Recommendations Table**

Recommendation	Immediate	Mid-Term (6-18 months)	Status	Long Term (18+ months)	Status	Challenges/Observations Addressed
1. (U) Increase Functional Integration and Operational Validation Testing in IV&V	<ul style="list-style-type: none"> (S) Install an OSN Terminal in the IV&V Lab (S) Provide Additional Detail in the CONOPS (U) Send IV&V Testers for Day in the Life Sessions with Operators 	<ul style="list-style-type: none"> (S) Provide IV&V with the Operational Frameworks and Tools Most Commonly Used for Deployments (C) Provide IV&V with Training on COG Tradecraft (U) Embed IV&V Testers within Different Branches on a Rotating Basis 	<input checked="" type="radio"/>			Challenge: 1
2. (U) Implement a Menu of Testing Services to Focus Testing Baselines	<ul style="list-style-type: none"> (U) Define the Test Services Each Organization will Provide (U) Define the Minimum Testing Required for All Projects (U) Educate All Stakeholders on Test Services Menu Concept 	<ul style="list-style-type: none"> (U) Implement the Test Services Menu (U) Track Tier 1 and Tier 2 Metrics 	<input checked="" type="radio"/>			Challenges: 2, 4, 5 Observation: 1
3. (U) Improve Communications Between All Stakeholders	<ul style="list-style-type: none"> (U) Add an Additional Tag-up/Design Review (for projects longer than 1 month) (C) Identify and Procure new IV&V Database 	<ul style="list-style-type: none"> (U) Implement Agile SW Development Methodology for Large Projects and Tools That Require Continuous Development 	<input checked="" type="radio"/>			Challenge: 3, 5, 7
4. (U) Focus Dedicated DART Automation Support	<ul style="list-style-type: none"> (U) Establish a Central Repository for Sharing DART Scripts (U) Identify Most Critical COG Tools that Require Repetitive Testing (U) Prioritize DART Automation on Critical COG Tools 	<ul style="list-style-type: none"> (U) Prioritize Regression Testing for Top Tools That Undergo Continuous Development (U) Add DART Script Development to the Project Development Process 	<input checked="" type="radio"/>			Challenge: 5, 6
5. (U) Connect Test Environments Through a Central Shared Services Environment	<ul style="list-style-type: none"> (U) Identify the Organization that will Own and Administer Shared Services Environment (U) Define the Services, Tools and Capabilities of the Shared Services Environment 	<ul style="list-style-type: none"> (U) Define the Detailed Shared Services Environment Architecture (U) Start Shared Services Environment Build-out 	<input checked="" type="radio"/>			Challenge: 4 Observation: 2

(U) The goal of these recommendations is to move EDG closer to a vision where SDLC processes and environments maximize efficiencies, optimize time to market, and improve product quality across EDG branches by: improving the effectiveness of communications throughout the development process; and increasing the level and rigor of testing accomplished by setting a baseline level of testing for all projects. Increasing the efficiency of testing is enabled by sharing of tools, tests, analysis and data through a common set of Shared Services available to all test environments. An example of what this environment might look like is provided in Figure 1 below.



(U) Figure 1: Vision for EDG SDLC Environments and Processes

(S) Increased communication between developers, operators and testers is achieved through the use of an SDLC Tool Suite, such as Crystal Castles or Rational Jazz, and the use of development methodologies such as Agile, which focus on more developer, operator and tester interactions. Increasing the level and rigor of testing will be accomplished by implementing a baseline set of tests all projects must accomplish and sharing key test resources and tools across all environments. This would include having a common DART environment, where any branch can easily share and use DART scripts developed by other branches, and network traffic analysis equipment that can be utilized to provide network signature analysis while other branches are conducting their tests, which would eliminate the need for a specific traffic analysis event. A more detailed discussion of this vision can be found in the (U) Recommendations Section below.

(U) To track progress against the recommended actions in Table 2 above, there should be a monthly pulse check between the key stakeholders involved in these activities to ensure that progress is being made. The goal of the pulse check would be to track the completion status of each individual action, identify any roadblocks or issues that are delaying or might delay the implementation of an action, and discussing the activities scheduled to be completed during the next month.

(U) Current State

(U) EDG's System Development Lifecycle (SDLC)

The Engineering Development Group (EDG) produces a large number of different systems/tools (400+) each year. These span a wide range of software (SW) and hardware (HW) based systems, however the vast majority of systems/tools are SW based. From discussions with SED, AED, ESD and COG, an overall high level SDLC exists. A detailed walkthrough of the SDLC for normal development efforts is provided in (C) Appendix 1: EDG SDLC Process. QRC developments are much more dynamic, and harder to capture in a simple diagram as they will streamline the number of the steps through this process, depending on the tool and operational need.

(U) Overall EDG's SDLC follows a traditional waterfall development model where:

- (U) Requirements are identified, documented and approved
- (U) Developers develop the system/tool to meet those requirements in a single release
- (U) The system/tool is delivered for verification and validation
- (U) Operations formally accepts the tool based on the test results and then conducts their own validation

(U) As evidenced by the number of systems delivered each year, this process has been successful in delivering key capabilities; however improvements in efficiency and time to market can be made. In recent years the rate of development has increased, the number of dedicated test resources has decreased, and the sophistication of the target system's security has improved, which will require improvements in these SDLC and test process to continue to deliver quality products. From reviewing these processes there are a few areas that point to some of the challenges that will be discussed later in this document. Most of these areas of concern stem from the differences in the process between AED led development efforts and ESD led development efforts.

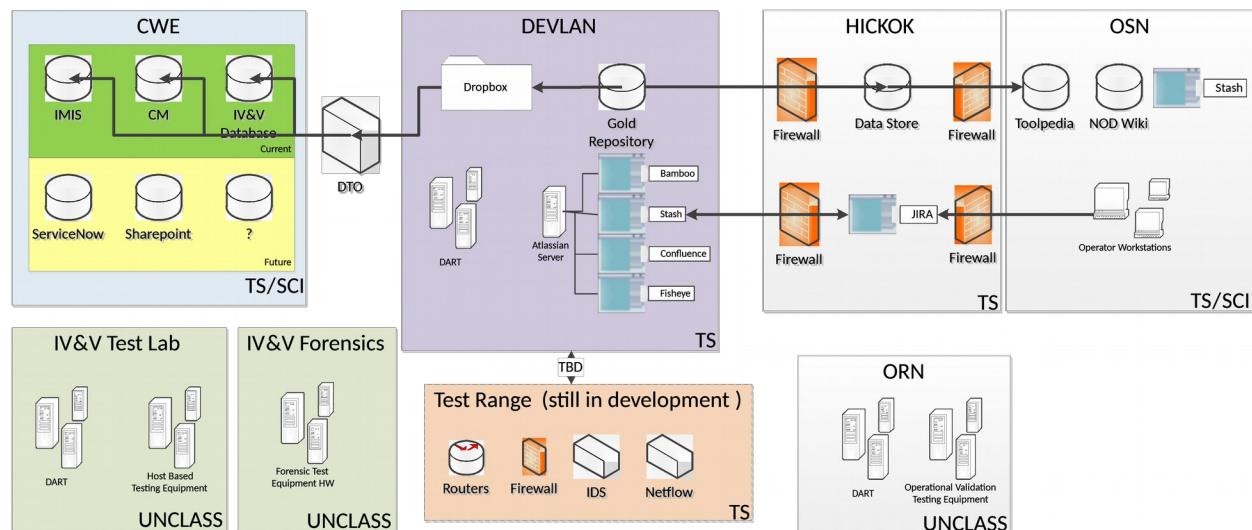
(U) The first area of concern, primarily for AED development efforts, is the limited number of interactions with all stakeholders (operators, developers, testers) present between the initial requirements generation and the start of formal testing. When requirements are initially generated for a project, there may be many unknowns about the target system, which leads to more general requirements. Overtime, more information about the target system is developed, which improves the specificity of these requirements. This is unique to EDG, since more traditional SW development efforts are able to define their requirements in detail at the beginning. In the current process, there is no formal meeting or mechanism in place to easily enable operators to convey this updated information to both the developers and testers. Consequently, these updates are often not relayed to all stakeholders until the test team's Tag-Up meeting, which is after development is complete. This often leads to changes that require AED to update the project and changes in the scope of testing that IV&V must accomplish. The rework that AED must accomplish either reduces the amount of time for IV&V to conduct their testing, or expands IV&V's test timeline, and by extension the time to deliver the system to COG. This area of concern can be addressed through one or more additional design review meetings prior to the tag-up,

such as a design adequacy assessment (DAA) or critical design review (CDR), where operators can provide updates to requirements to developers and testers during the development timeframe.

(U) The second area of concern is the difference in development testing between ESD and AED. Many ESD projects go through a formal Factory Acceptance Test (FAT) prior to delivery, while AED projects only go through unit testing, which can vary on a developer by developer basis depending upon the developer's experiences. This is due to the more formal nature of contracting out development activities versus conducting development in-house. The FAT testing, which is witnessed by both ESD and COG operators, provides a formal verification event prior to delivery for IV&V. AED does not have a formal level of test required before they deliver their systems to IV&V, which can lead to inconsistencies in the maturity of products given to IV&V for testing. In some cases this can lead to IV&V returning projects to AED to fix critical defects, which will require a full retest of the project. EDG's Technical Advisory Committee (TAC), conducted a State of Testing analysis, which addresses AED's testing and proposes some recommendations to improve it. One improvement was procuring Bamboo, which is an automated build and unit testing tool that will make it easier and less time consuming for developers to conduct their unit tests. Other recommendations included conducting peer reviews of code and having more test focused development processes. As this was a recent effort, the formal report is in the process of being finalized, and the implementation of these recommendations is still in work. Formalizing these recommendations across AED's development efforts would increase the consistency of testing across AED.

(U) Environments

(U) There are a number of environments that get used throughout EDG's SDLC. These consist of development environments, operational environments and multiple environments for conducting testing. (S) Figure 2 below provides a high level overview of those environments and the connections between those environments.



(S) Figure 2: EDG SDLC Environments

(U) The following table provides a brief high level description of each environment and where it fits within the overall SDLC process.

(S) Table 3: EDG and COG Environments

Environment	Classification	Purpose
(C) CWE	TS/SCI	<ul style="list-style-type: none"> • (U) Houses the SDLC tracking tools and mechanisms <ul style="list-style-type: none"> ○ (C) IMIS ○ (U) CM database ○ (U) IV&V database • (C) IMIS is in the process of being replaced by ServiceNOW • (U) The CM database is planning on being replaced with SharePoint • (U) IV&V Database will be replaced • (U) Documentation can be shared between utilizing DTO.
(S) DEVLAN	TS	<ul style="list-style-type: none"> • (U) AED and ESD are the primary users • (U) Houses the tools and systems that enable AED development and test • (U) Houses a drop box that outside users can utilize to grab the tools for testing • (U) Final SW baselines and documentation are housed in the Gold Repository. • (U) DART used for automated testing • (U) Houses Atlassian Tool Suite <ul style="list-style-type: none"> ○ Bamboo - Automated build, unit testing and deployment ○ Stash - Source Control ○ Fisheye - Source Review ○ Confluence - Wiki • (S) A connection between NDB's Test Range and DEVLAN is planned • (S) SW and documentation are transferred between DEVLAN and the OSN, through HICKOK
(S) HICKOK	TS	<ul style="list-style-type: none"> • (S) Used for transferring tools and their associated documentation between DEVLAN and OSN • (U) Consists of two firewalls, a datastore that houses the files to be transferred, and the JIRA instance for discrepancy reporting
(S) OSN	TS/SCI	<ul style="list-style-type: none"> • (U) Environment COG operators use for their operations. • (U) Two key systems that can be useful to other users. <ul style="list-style-type: none"> ○ (U) Toolpedia - houses the tools and documentation COG uses ○ (S) NOD Wiki - houses user supplied information on how certain tools are used as well as tips and tricks for using those tools. • (U) Also houses an instance of Stash for Source Control
(U) IV&V Test Lab	UNCLASS	<ul style="list-style-type: none"> • (U) Where IV&V conducts their testing • (U) UNCLASS to allow download of the latest PSP products and updates • (U) Consists of the HW and SW required to conduct host based tests. • (U) DART used for automated testing
(U) IV&V Forensics Lab	UNCLASS	<ul style="list-style-type: none"> • (U) Separate from IV&V Test Lab to ensure the environment remains forensically clean • (U) Set of HW and SW to conduct their forensic examinations
(U) ORN	UNCLASS	<ul style="list-style-type: none"> • (U) Where operators conduct their operational validation and training • (U) Small environment with a limited amount of HW. • (U) UNCLASS to allow download of the latest PSP products and updates
(U) NDB Test Range	TS	<ul style="list-style-type: none"> • (U) Still currently under development • (U) Will house the equipment necessary to create a more realistic network

Environment	Classification	Purpose
		<p>environment</p> <ul style="list-style-type: none"> o (U) Network security devices such as firewalls, intrusion detection systems (IDS), intrusion prevention systems (IPS), and netflow analyzers o (U) Traffic generators and equipment to allow the simulation of different network performance parameters such as jitter and latency. • (U) The estimated IOC is end of CY 2014

(U) Future Plans

(U) During discussions with stakeholders, new tiger teams, working groups, and updates to tools and environments were identified. The focus of these groups and changes is to improve EDG's processes and the quality of EDG's products. Many of these initiatives will be addressing the Challenges and Observations discussed below, and where appropriate will also appear in the Recommendations Section.

(U) NDB Test Range

(S//NF) ESD/NDB is currently developing an NDB Test Range that will allow NDB to conduct the performance and operational testing currently performed by PlumWaffle (PW). The Test Range is expected to have an initial operational capability by the end of December 2014. The range will consist of network performance equipment that will vary operational characteristics such as MTU, packet loss, and latency, as well as network security related devices such as firewalls, IDS/IPS and netflow connectors. Although the initial goal of the Test Range is to support primarily NDB systems, the longer term goal is to create an environment that provides communications based signature analysis of host-based systems and network attack devices.

(U) Technical Advisory Committee (TAC) State of Test Analysis

(U) The TAC recently conducted their own assessment of the state of testing within EDG, and identified their own challenges and recommendations for addressing them. This analysis was initially out briefed in July 2014 and a more detailed report will be delivered in the near future. The TAC study reviewed all EDG testing, however based on the TAC briefing, their observations and recommendations focus more on the testing accomplished by AED. The observations and recommendations from that assessment correlate well with this report, and where appropriate are referenced in the appropriate recommendations or challenges.

(C) IMIS Replacement

(C) The IMIS system currently used for system development and requirement tracking is old and does not adequately meet the needs of EDG. Additionally the current CM repository uses a Lotus Notes database, and will need to be replaced. A replacement system for IMIS, ServiceNow, has been procured and will be installed on the CWE by the end of the calendar year. This system is a significant upgrade over the current system, and will likely improve the tracking and coordination of artifacts currently used by IMIS. The proposed replacement for the CM repository will be SharePoint, which again provides additional coordination and tracking mechanisms.

(U) Forensics Working Group

(U) A forensics working group is being started to coordinate the forensic testing activities that IV&V, ESD and AFD are conducting and define the types of forensic services each organization provides. The goal of this working group is to more clearly delineate the activities of each organization and communicate the full range of forensic services available to projects within EDG.

(U) Test Tiger Team

(U) A test tiger team is being formed to help reduce the IV&V backlog. It consists of three AED developers, three ESD SETA staff as well as IV&V team members. In addition to helping reduce the backlog, this tiger team will also be looking at near term tactical actions that can be taken to improve overall test processes. This will include ways to streamline processes, reduce bottlenecks and implement automation through DART.

(U) Challenges and Observations

(U) Challenges

(U) Throughout discussions with members of SED, AED, ESD and COG, a number of challenges were noted with both the processes that support testing and how testing is accomplished today. The following challenges were identified either through discussions with the stakeholders listed in Appendix 3 or through review of the documents that support or describe EDG's SDLC processes. Many of these challenges were identified by multiple stakeholders and they are the challenges that have the highest impact on EDG development and testing efforts today. They are provided in order of priority, based on the feedback received.

(U) Need for More Functional Integration and Operational Validation in IV&V Test (Challenge 1)

(U) The most common challenge cited was that the current structure of IV&V testing focuses too greatly on requirements verification and does not cover the functional integration and operational validation of the tools that the operators need. The COG operators would like to see their tools tested in a manner similar to how they plan to use them, which includes using the other tools necessary to deploy, execute, exfiltrate data or delete the tool under test. In most cases, the tools that undergo testing by IV&V are used in conjunction with other COG tools, and only verifying functionality without verifying interoperability with those tools does not always adequately test functional requirements. Functional integration and operational validation, in conjunction with verification, is what the operations need. Implementing this type of testing will likely increase the complexity of IV&V testing, which may lead to longer test timelines, however implementation Recommendation 2 (Test Services Menu) will mitigate this.

(U) A key impediment to implementing this type of testing though has been providing IV&V testers the tradecraft resources necessary to accomplish this type of testing. In some cases the IV&V team does not have access to the tools the operator use. The tradecraft used to operate these tools can be very complex, changes frequently and most IV&V testers don't have the tradecraft training and knowledge necessary to use them as the operators would. Limited COG training resources and the structure of COG training have made it challenging to support COG training opportunities for testers. Additionally, the CONOPS that provides the high level use case for a specific tool, which is provided as part of the User Requirements Document (URD) developed at the beginning of the SDLC Process, does not always contain the depth necessary to conduct functional integration and operational validation testing. The additional detail needed would include listing the other tools and processes that must interoperate with the tool under test, either to deploy the tool on target, operate the tool, exfiltrate data or remove from the system. This data will be necessary for IV&V testers to create an operational like scenario to test each tool.

(C) Need for More Forensic/Signature Tests Executed Against All EDG Projects (Challenge 2)

(C) Recent events, such as operational compromises and analysis by the Advanced Forensics Division (AFD), have highlighted a need for a more rigorous testing of the signatures produced by the tools developed within EDG. In the current process, COG determines the level of signature and forensic testing that will be conducted for each project. Since the forensic test requirements vary on a project by project basis, the rigor and depth of forensic testing is inconsistent. Currently COG does not require forensic testing from EDG for all projects, and those projects that request forensic exams usually focus on basic host based analysis. Additionally, COG will utilize resources outside of EDG to conduct forensics examinations for some of their tools, the results of which are not always received and tracked by EDG.

(C) In the current process, the only branch conducting network based signature analysis is NDB, and they are only conducting that analysis for their projects. Unless the project under test contains a network device, or includes network communications functionality, COG does not normally request a network based signature analysis. With the speed at which commercial security companies are advancing their products, many stakeholders expressed concern that a more sophisticated network analysis capability will be commonly available within the next 5 to 6 years, which would put COG tools at risk. Getting ahead of those tools, and understanding if the combination of commonly used COG tools creates network signatures, will be important for ensuring EDG is ready for those advances. The development of NDB's Test Range provides a new opportunity for more projects to take advantage of network based signature analysis.

(U) There are three primary organizations conducting forensic testing for EDG projects: IV&V, ESD and AFD. All three provide their own skills and forensic services that can be applied to a wide range of projects. A Forensics Working Group has been started for these organizations to coordinate forensic test responsibilities and activities. There is a tremendous amount of host and network based forensic capability that these organizations provide, which more projects should take advantage of. Defining a baseline set of host based and network forensics that all projects must accomplish can mitigate the risk of future compromises and improve the overall rigor of EDG's test process. Recommendation 2 (Test Services Menu) below provides a framework for implementing a baseline set of forensic tests, without delaying delivery to operations.

(U) Need for More Communication Between All Stakeholders (Challenge 3)

(C) Per the process outlined in the EDG's SDLC Section, there are two main times when AED, SED and COG formally meet to discuss a development effort. The first is during the requirements development phase, which culminates with the ERB where a project is approved. The second is at the Tag-up that immediately precedes the start of IV&V testing. Between those two events there is normally minimal communication that involves all stakeholders (operators, developers, testers). Those initial requirements are normally more general, with many aspects of the target system unknown. Over time, more knowledge of the target system is gained, and additional detail can be added to the requirements, however there isn't a formal mechanism for conveying that information to both the developers and testers. That understanding in many cases isn't transferred to the development and IV&V teams until the Tag-up meetings, which is the point in the process when development is supposed to be complete and IV&V testing is supposed to start. This new understanding can result in development rework, which

occurs during the timeframe set aside for IV&V testing. For development efforts that are short (e.g. - less than a month in length) this may not be a major challenge, but for longer duration development efforts this can have a significant impact on both development and IV&V. Implementing additional mechanisms to communicate these changes earlier in the development process, will minimize the amount of rework required later in the development cycle, and allow more time for the IV&V testers to adapt to these changes.

(C) After projects are delivered, there isn't a feedback mechanism that notifies all stakeholders when an issue is found operationally. Operations will notify the developers of discrepancies they find, however IV&V isn't notified as part of that process. When those projects go back through IV&V, they may not be aware of past issues, and may not know to test them. An instance of JIRA, a discrepancy tracking tool, has been deployed to better status and track discrepancies found. Use of JIRA is just beginning, but including IV&V as part of the JIRA implementation can address this. As an example: operators would create the initial discrepancy in JIRA; after developers have reviewed it they could set the status of that discrepancy to Accepted; once the fix is complete and the developers have checked it the status could be set to Tested; and finally after IV&V has regression tested the software the status could be set to Verified, and the fix would be released to operations. Including IV&V as part of the defect management process will ensure they know about past issues and can test them to ensure they were fixed.

(U) Need for More Consistent Testing Across Projects and Branches (Challenge 4)

(U) As discussed in the SDLC Section above, the level of development test prior to IV&V is inconsistent. The level of unit, system and regression testing accomplished on AED developed projects varies on a developer by developer basis, and there is no formal standard for how much developer conducted testing must be accomplished before the handoff to IV&V, which can lead to defects found in IV&V and rework. For many ESD projects, a contractor FAT occurs in addition to unit, system and regression tests, which provides a formal requirement verification event and more confidence in the maturity of the system. This does not apply to exploits developed by ESD, which do not include a FAT because by their nature, those exploits either work or fail, with no grey area that would require extensive testing.

(U) After development testing is complete, not all projects go through IV&V testing. Most AED developed projects go through IV&V, with the exception of projects where IV&V can't recreate the target environment. Fewer ESD development efforts go through IV&V testing though, which is mostly due to the challenges discussed in Challenge 1 (Need for more Functional Integration and Operational Validation). The addition of more functional integration and operational validation in IV&V will likely lead to more ESD efforts being scheduled for IV&V testing.

(C) Finally, as discussed in Challenge 2 (Need for more Forensic and Signature Testing) above, there is also a great degree of variability in forensic testing across all projects. Creating a baseline set of tests that all projects must accomplish will help provide more consistency and rigor across the test program. Recommendation 2 (Test Services Menu) below provides a mechanism for creating a more rigorous test baseline that can still be executed in a way that optimizes delivering projects within COG's operational timeline. At a high level this would involve each organization defining the types of tests and test services they provide, and those services would fall under one of two tiers. Tier 1 would be the tests that must be

accomplished prior to operational deployment, such as functional verification, operational validation, binary analysis, dirty word searches, and a small set of OS and PSP tests. Tier 2 would be tests that aren't a high priority for COG, but are still useful for future analysis or planning. Tier 2 tests would include a network based analysis of the operational scenario, a more extensive matrix of OS and PSP combinations, and more detailed host based forensics such as live state analysis, volatile memory capture and analysis and fuzzy hashing. Specific tests would be defined as applying to all projects as either a Tier 1 or Tier 2 requirement, with the remaining services left to the operator to request. Dividing testing into these two tiers can allow operators to focus on only the testing that needs to be accomplished for their upcoming operation, while ensuring that more extensive testing is accomplished to support other operations.

(U) Need for Shorter IV&V Testing Timelines (Challenge 5)

(U) Another challenge mentioned during stakeholder discussions was the time required for IV&V to conduct their testing. IV&V test timelines frequently do not meet the operational need dates set by COG. In most cases the length of IV&V testing is driven by two key factors. The first are projects that require additional development to fix defects. Any time a project is sent back to development, the full suite of IV&V tests must be re-accomplished when it is returned, which can significantly lengthen the timeframe required to conduct testing. Projects sent back at the Test Tag-up due to changes in requirements can lengthen the test timeframe as well.

(U) The second key factor is the number of OS, PSP and language pack combinations identified in the test matrix. An extensive combination of OS, PSP, and language pack combinations can significantly increase the amount of time required to conduct testing, as to date this testing has mostly been accomplished manually on bare metal HW. From discussions with stakeholders, testing these large matrices is less focused on supporting the current operation and more focused on supporting future possible operations against a larger target set. The recent addition of DART to the IV&V environment will hopefully help mitigate this issue, by automating extensive test matrices. Additionally, many stakeholders questioned whether extensive PSP testing should be accomplished in IV&V, because COG normally conducts their own PSP evaluations prior to deployment. This is due to daily PSP update cycles, and the need to test the tool against the most recent PSP packages. Minimizing the number of these combinations will help reduce IV&V test windows. As discussed in Challenge 4 (Need for More Consistent Testing) and Recommendation 2 (Test Services Menu) below creating a two tiered system of test services, can address the need for more consistent testing prior to IV&V, while allowing extensive OS and PSP matrices to be tested in Tier 2, which will reduce the Tier 1 timeframe and speed the time to COG delivery.

(U) Need for DART Automation Development Strategy (Challenge 6)

(U) The DART tool suite provides compelling functionality that can bring new capabilities to EDG testing and improve current test processes, however many stakeholders interviewed were concerned about trying to utilize DART in ways that can create inefficiencies. As an example, automation is normally most effective for tests that need to be run repeatedly, or in the case of DART, tests that need to be run against a large number of OS, PSP and language pack combinations. In traditional SW development efforts, the industry best practice for implementing good automated testing, where tests can be executed overnight without human interaction, is to plan for automated test script development

timeframes that are approximately 8 times as long as developing a manual test procedure. Therefore, if a test won't be run more than 8 times, or have more than 8 combinations of OS and PSPs, then automated testing becomes less efficient than just running the manual test. This extended timeframe results from many factors including: the need to create a manual test initially to base the script off of; coding the script and including additional features such as error handling and fault tolerance to support true automation; running the script and comparing the results to the manual test; and then troubleshooting and correcting any issues. Within EDG, this can be extremely effective in testing OS, PSP and language pack combinations; "Patch Tuesday" tests against commonly used COG tools; as well as regression tests of tools that frequently undergo updates. Utilizing DART to test an exploit that is only designed to run in one operation and against a limited target environment is likely to be more inefficient than running a manual test.

(U) The DART development process requires new skills that are currently in limited supply in some branches. Although AED, ESD and COG have a good base of personnel comfortable scripting with Python, IV&V hasn't traditionally required that skillset and are still getting familiar with the tool and development process. IV&V is trying to increase this skill set by bringing new testers on-board with Python experience, however the process for finding personnel with those skills, a test background and the proper clearances takes time. Since AED, ESD, IV&V and COG are all utilizing DART, one way to accelerate the learning process for all users would be through sharing DART scripts between organizations, which would allow each organization the opportunity to modify scripts for a specific tool instead of having to create those scripts from scratch. Currently, there is DART repository in AED's Stash source control tool, although COG does not have access to this instance and most users still share scripts through e-mail and drop boxes. Additionally, there isn't an easy way to identify what scripts exists other than by asking specific developers what they have already created. Setting up a DART Script Repository, or using AED's Stash and connected with COG's version of Stash, and allowing all DART stakeholders access would enable the sharing of existing scripts. Folders for each tool could be created and DART developers could load the scripts they have created for those specific tools. Other DART developers would then be able to see what scripts already exist without having to hunt down specific developers working on a tool.

(U) Need for an IV&V Database Replacement (Challenge 7)

(C) The current IV&V database utilizes a Lotus Notes database, and will have to be retired soon. The IV&V database is a useful tool that all of the branches use to track the status of testing through IV&V. Loss of this capability will make it more difficult for AED, ESD and COG to track the status of projects through IV&V, as well as make it more difficult for IV&V staff to keep configuration management of test results, track their resources and generate IV&V metrics. Finding a suitable replacement will be necessary before the Lotus Notes databases are decommissioned. The IV&V team is currently exploring options for this replacement.

(U) Observations

(U) Throughout discussions with members of SED, AED, ESG and COG, some observations about processes within EDG were noted. Unlike the challenges above, these observations may not be currently

impacting overall development efforts, but may impact development efforts in the future. The following observations were those commonly expressed across the stakeholders.

(U) Forensics Testing Outreach (Observation 2)

As discussed in Challenge 1 (Need for more Functional Integration and Operational Validation), IV&V, ESD and AFD all provide forensic testing services to EDG projects. Through discussions with stakeholders it became clear that not all stakeholders and branches were familiar with the range of forensic testing services available to EDG. Some stakeholders weren't aware of all of the capabilities of IV&V's forensics team, and likewise other stakeholders didn't have a good understanding of the ESD forensics team's capabilities. All three organizations have their own skills and forensic services and a more coordinated communication effort to make all of EDG and COG aware of those abilities can help increase the level of forensic testing accomplished across projects. A Forensics Working Group has been started to address this, and define each group's forensic testing capabilities. This in conjunction with creating the test services menu described in Recommendation 2 (Test Services Menu) will increase the visibility of the forensic testing capabilities of each organization and the overall rigor of the forensic testing program.

(U) Distribution of Test Environments (Observation 3)

(S) There are multiple test environments currently running throughout EDG, including testing within DEVLAN, the IV&V Test Lab, the IV&V Forensics Test Lab, and the Operational Research Network (ORN). Additionally, by the end of the calendar year there will also be the NDB Test Range, which will have a connection to DEVLAN. Each environment has their own set of equipment and capability, including their own instances of DART, suited to the type of testing they focus on. There are a number of capabilities that would be useful across environments, including an ability to easily share and browse DART scripts, tools for capturing netflow, and generating a generic network traffic flow. Currently there is no easy way for sharing tools across environments, and sharing tests and scripts is done manually either through e-mail or drop box. Creating a more connected set of environments, where the different test users could share tools, scripts, and tests easily, would increase the efficiency of overall test efforts and could more easily increase the scope of testing that can be accomplished with minimal additional resources. As an example, when IV&V is conducting one of their tests, they can use a shared services network traffic analysis capability to capture netflow for later network signature analysis, eliminating the need for a dedicated traffic analysis test by the Test Range. Likewise, a developer creating a new capability for an existing tool, could use an automated regression test created by IV&V to check their code against and ensure it doesn't break existing functionality. There are likely security concerns that may complicate what services can be shared, however exploring what is possible and enabling as much shared functionality will improve the efficiency of testing in the long run. This may become necessary if the rate of development at EDG continues to increase.

(U) Recommendations

(U) The following recommendations are a combination of suggestions from stakeholders across EDG, as well as recommendations derived from the independent reviewer and industry best practices. Each of these recommendations traces back to a specific challenge and/or observation, and includes immediate, mid-term and long term actions. Immediate actions would be accomplishable within 6 months, mid-term actions would fall within 6-18 months and long term actions would be 18+ months out. Finally, each recommendation includes an implementation section that discusses the timelines and resources that would be associated with executing each action. Where possible these recommendations were discussed with appropriate stakeholders to ensure their feasibility. The goal of these recommendations is to mitigate the challenges and observations discussed above.

(U) To track progress against these recommendations, there should be a monthly pulse check between the relevant stakeholders involved in these actions to ensure that progress is being made. The goal of the pulse check would be to track the completion status of each individual action, identify any roadblocks or issues that are delaying or might delay the implementation of an action, and discussing the activities scheduled to be completed during the next month.

(U) Increase Functional Integration and Operational Validation Testing in IV&V (Recommendation 1)

(U) Challenge Addressed: Challenge 1

(U) The feedback that was most consistent across EDG was the need for IV&V testing that is more focused on functional integration and operational validation. This can be accomplished in conjunction with the current IV&V functional verification by utilizing COG tradecraft (tools and processes) to deploy the tools and exploits under test. As mentioned in Challenge 1 (Need for More Functional Integration and Operational Validation) however, the IV&V testers don't currently have all of the tradecraft resources needed (e.g. - operator tools and operational knowledge) to accomplish functional integration and operational validation. One important note is that increasing functional integration and operation validation testing is likely to lengthen IV&V test timeframes, however implementing Recommendation 2 (Test Services Menu) may help mitigate this. The following are immediate, mid-term and long term actions that will support the implementation of this recommendation.

(U) Immediate Actions

1. **(S) Install an OSN Terminal in the IV&V Lab:** Installation of an OSN Terminal within the IV&V Lab will provide the IV&V testers with access to the NOD Wiki. The NOD Wiki provides key operator insights into how COG tools work, common challenges encountered and recommended ways to use these tools. This will give IV&V testers an extra resource to consult when conducting tests that use COG tools if they run into challenges with the tool, are unsure how the tool works, or if the tool is working correctly. This should be the quickest way to provide IV&V testers with additional operational knowledge.

2. **(S) Provide Additional Detail in the CONOPS:** As mentioned in Challenge 3 (Need for more Communication), the CONOPS for a specific tool is included as part of the tool's URD. The CONOPS currently provided don't normally provide the depth necessary for IV&V to conduct the types of functional integration and operational validation that are being requested. Updating the CONOPS to provide additional detail, including the other tools and processes that must interoperate with the tool under test, either to deploy the tool on target, operate the tool, exfiltrate data or remove from the system, will help IV&V provide more realistic operational scenarios.
3. **(U) Send IV&V Testers for Day in the Life Sessions with Operators:** Periodically send IV&V testers to shadow an operator during one of their operations. This will give IV&V testers a more real world example of how operators use and deploy the systems that they test. These sessions have been provided in the past, and SED is currently working with COG to start implementing them again.

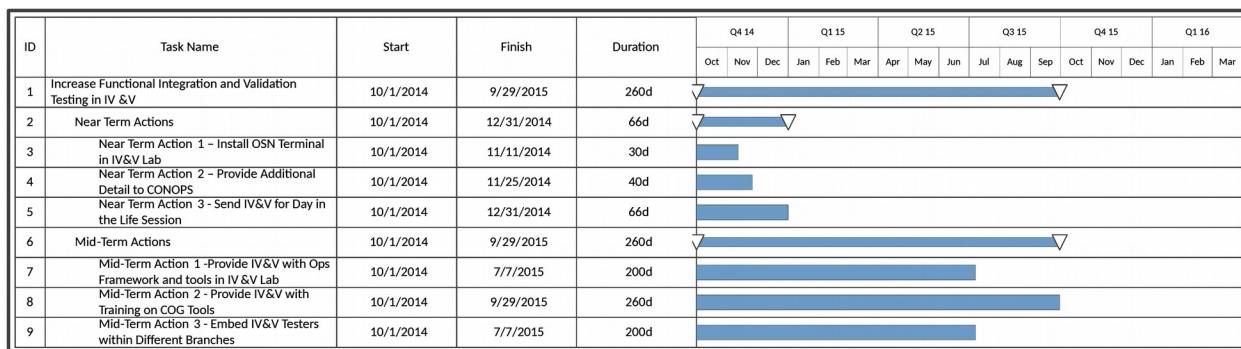
(U) Mid-Term Actions

1. **(S) Provide IV&V with the Operational Frameworks and Tools Most Commonly Used for Deployments:** The IV&V team doesn't currently have access to all of the operator tools and frameworks required to conduct functional integration testing. Not all operational tools and frameworks would be necessary, just the most commonly used tools. From the interviews conducted, it seems like many deployments use some combination of the following tools that IV&V does not currently possess: Windex, Mission Control/Quaffle, and Winshell. Focusing on implementing the most commonly used tools would likely cover 80-90% of the scenarios that IV&V would need to test.
2. **(C) Provide IV&V with Training on COG Tradecraft:** As mentioned in Immediate Action 2, there is a small subset of tools commonly used in deployments. Providing IV&V with focused operational training on the tradecraft and basics of tool use will ensure that the IV&V team is using the tools correctly, and help them develop more operationally relevant tests. Ideally IV&V staff would sit in on the Networking Exploitation Tradecraft Course for the most relevant COG tools, however with the long timelines associated with this training, the limited COG training resources and the structure of the current course, this may not be easy to accomplish. COG training has discussed offering more a la carte type training offerings, and if resources permit those types of offerings in the future, it would be beneficial for IV&V to participate.
3. **(U) Embed IV&V Testers within Different Branches on a Rotating Basis:** To increase the knowledge of different branches test needs, it may be beneficial to have testers embedded within the different EDG branches. Working day-in and day-out with one of the different branches will give the IV&V testers better insight into what their test needs are and how the operators are using these tools. The testers would then work with the developers, operators and SETA staff in that branch to help review and shape development testing. If IV&V testers have participated in the development test process, that may reduce the amount of dedicated IV&V testing that would be required after tool delivery. After 12 months, new testers could rotate in while the embedded testers would move back to IV&V with a better operational understanding of ESD tools and processes.

(U) Implementation

(S) The Immediate Actions in this recommendation are likely the easiest to execute and they could be implemented in parallel. For Immediate Action 1 (Install OSN terminal in IV&V), the process for getting new OSN terminals installed is well understood and already exist within other environments in EDG. From discussions with COG personnel getting an additional OSN Terminal in IV&V's lab would not be difficult. Addressing Immediate Action 2 (Provide Addition CONOPS Detail) would occur at the URD development phase initially, with CONOPS updates as more details become available throughout the rest of the SDLC process. This could be accomplished quickly after having discussions with the operators about providing these extra details. Immediate Action 3 (Day in the Life Sessions) is already in the process of being implemented. Day in the Life sessions with operators were accomplished in the past and SED is in the process of resurrecting them.

(S) Implementing the mid-term actions is likely to be more complicated, as shifting resources to accomplish these tasks will be necessary, and would likely take 6-12 months to implement. Mid-Term Action 1 (Provide Ops Framework and Tools to IV&V) will require some support from COG, as different COG tools have different architectural needs and an analysis of what can be accomplished within the IV&V environment may be necessary. Even if it is not possible to make every tool available to IV&V, getting as close as possible to providing IV&V with all of the core COG tools will go a long way toward increasing the ability to conduct functional integration and operational validation. As mentioned previously, limited COG training resources and the structure of the Network Exploitation Tradecraft Course make implementation of Mid-Term Action 2 (Provide COG Tradecraft Training) difficult. This action is still highly recommended and worth pursuing if resource constraints improve, and would be the best avenue providing IV&V testers with better insight into COG tradecraft. Mid Term Action 3 (Embed IV&V in Branches) would require reallocating IV&V staff to specific branches and assigning them responsibility for tests within those branches, which would change current IV&V processes. Additionally, a plan for rotating staff would need to be created and the number of IV&V testers who would be deployed would need to be determined. These decisions are best left to the IV&V leadership to develop in conjunction with the other branches, as they will have a better understanding for the impacts and tradeoffs to the full suite of IV&V test efforts. The Gantt Chart below provides a proposed timeline for implementing these actions.



(U) Figure 3: Recommendation 1 Roadmap

(U) Implement a Menu of Testing Services to Focus Testing Baselines (Recommendation 2)

(U) Challenges and Observation Addressed: Challenges 2, 4 and 5, Observation 2

(U) Many of the concerns voiced during interviews with stakeholders revolved around inconsistency of testing and testing timelines that don't meet operational need dates. One way to address these concerns would be to implement a menu of testing services that each organization provides, with two tiers associated with each service. Tier 1 would be tests that must be conducted before the tool can be deployed for operations, while Tier 2 would be tests have less operational risk, but are still important from either a forensic perspective or as risk reduction for future operational implementations. Tier 2 tests can be accomplished after the tool is delivered to COG, relieving the pressure to try and fit all testing within a narrow timeline pre-deployment. (U) Figure 4 provides an example of what a test services menu could look like. Ideally, certain test services would always be identified as Tier 1 or Tier 2, which would create a consistent test baseline. Certain test services, such as OS, PSP and Language Pack combinations, could also be separated into Tier 1 and Tier 2 lists to help alleviate schedule pressures from conducting testing against a large test matrix. This concept is very similar to how QRC testing is accomplished currently. For QRCs specific tests will be designated as QRC or non-QRC tests, where QRC tests are conducted prior to deployment and non-QRC tests are conducted after. Separating the tests into two tiers will shrink the timeline required for testing prior to operational deployment, while increasing the overall rigor of the enterprise testing program.

(U) Immediate Actions

1. **(U) Define the Test Services Each Organization will Provide:** Each organization that conducts testing, including AED, IV&V, ESD and AFD, would develop a list of test services that they provide. Examples of those services can be found in (U) Figure 4. The Forensics Working Group is a forum where IV&V Forensics, ESD Forensics and AFD are conducting a similar exercise to delineate the capabilities that each organization can provide and coordinate forensic activities. Folding this activity into the that working group to define the full forensic test services suite, while having AED, ESD, NDB and IV&V provide a list of their test services would complete this activity.
2. **(U) Define the Minimum Testing Required for All Projects:** Once the menu of test services has been established, convene either an existing working group such as the TAC or a new working group to define the baseline set of test services that all projects must accomplish. The working

Test Services Request – Project Name		
AED Testing	Tier 1	Tier 2
Unit Testing		
Integration Testing		
Regression Testing		
Acceptance Testing		
ESD Testing	Tier 1	Tier 2
Unit Testing		
Regression Testing		
System Testing		
Factory Acceptance Testing (FAT)		
Ad-Hoc FAT Testing		
IV&V Testing	Tier 1	Tier 2
Functional Verification		
Functional Integration		
OS and PSP combinations		
CONOPS Validation		
Host Performance Testing		
IV&V Forensics Testing	Tier 1	Tier 2
Binary Strings Analysis		
Live State Analysis		
Volatile Memory Capture and Analysis		
VM forensics		
Fuzzy Hashing		
Restoration of media formats to OEM configuration		
Dirty word searches		
Mobile device forensics		
NDB Test Range Testing	Tier 1	Tier 2
Adverse Network Performance (e.g. – high packet loss, high latency, etc.)		
Traffic Analysis		
Network Security Analysis		
Network Performance Testing		
Operational Rehearsal/Readiness		

*Full List of OS, PSP and language pack matrix would be addendum with specific OS, PSP and language pack combinations marked as Tier 1 and Tier 2.

(U) Figure 4: Example Test Services Menu

group should include members from each branch of EDG, plus COG to ensure every organization has a stake in the process. During the interviews conducted for this report, many stakeholders already had opinions and examples of test services that they felt would be required for both Tier 1 and Tier 2. Examples include binary analysis and dirty word searches as Tier 1 tests, and traffic analysis and network security analysis for Tier 2 tests.

3. **(U) Educate All Stakeholders on Test Services Menu Concept:** After an initial baseline is established, the test services menu concept and baseline can be communicated to both COG and the EDG development and test community. Educating the community on the concepts and benefits of this approach will be important to the successful implementation of this process. The key message to convey is that Tier 1 tests, outside of those identified as part of the baseline, should focus only on testing required to support ***the specific operation*** that the tool was designed for. Testing that would support possible future operations involving this tool should be identified as Tier 2 tests, however again a focus on testing that provides the most future value should be stressed. The time to market benefits of this process are predicated on more narrowly focusing Tier 1 testing.

(U) Mid-term Actions

1. **(U) Implement the Test Services Menu:** Once the stakeholder community understands the concepts and benefits of the new Test Services Menu process, implementation should be started. The Test Services Menu would be completed during the initial requirements development process, and then updated over time if necessary to adapt to updated requirements and more target information. The Test Services Menu would be reviewed at the ERB along with the URD, providing the formal mechanism for accepting the defined test services. For the initial implementation, conducting the new process with a small set of initial development efforts as a pilot would provide the opportunity to compare the benefits and impacts of the new process against the old and tweak the menu and process before a full rollout. After the pilot efforts are complete and any initial changes are implemented, the Test Services Menu can be rolled out to all new development efforts.
2. **(U) Track Tier 1 and Tier 2 Metrics:** One key to measuring the effectiveness of the Test Services Menu will be to track metrics on Tier 1 and Tier 2 testing. These metrics would include Tier 1 and Tier 2 estimated timelines versus actual timelines. Tracking these estimated and actual timelines and comparing against the historical IV&V timelines will quantify the time benefits of using this process, and help determine whether small changes in the baselines are necessary. In general, Tier 1 timelines should be shorter than current IV&V timelines, however that may not be the case initially, as IV&V testers start creating more functional integration and operational validation tests. Another metric to track would be defects found in operations and tools compromises. Again, comparing these metrics against historical information will provide the ability to quantify the impact of these new processes. With the additional rigor of testing in this process, the assumption is that both defects found in operations and the number of tool compromises should start to fall. If these numbers don't decline or there is an initial decline followed by an increase, then it may point to areas where tests need to be added to the Tier 1 or Tier 2 baselines.

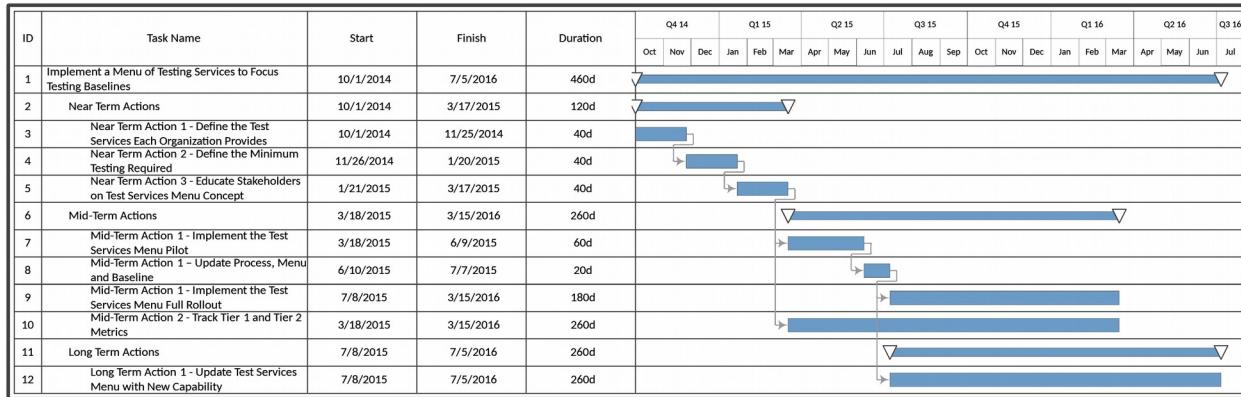
(U) Long Term Actions

1. **(U) Update and Refine Test Services Menu:** Over time, test priorities may change, especially as new security capabilities become more widely available. Additionally, the metrics tracked in Mid-Term Action 2 may point to areas to increase rigor. Periodically revisiting the Test Services Menu and baseline set of tests will be important to ensure that this process reflects those priority changes. Additionally organizations conducting testing may add new test services that they can perform, which will need to be reflected on the menu. Examples of these services would include host based performance testing in IV&V or a security and incident event management (SIEM) analysis capability in the Test Range.

(U) Implementation

(U) The implementation of the Test Services Menu concept will require stakeholder coordination up front, followed by a rollout of the new process. Immediate Actions 1 (Identify Test Services) and 2 (Identify Baseline) will likely require a core group of stakeholders from AED, ESD, SED and COG as part of the associated working groups. The outcomes of both actions, the list of test services and the baseline set of Tier 1 and Tier 2 tests, will likely require approval by the appropriate branch chiefs within AED, ESD, SED and COG as well. Many stakeholders already have defined the services they offer and what they feel the baseline should be, however in order to give both the working groups time to coordinate and upper management time to review, each action was estimated at 2 months. Immediate Action 3 (Educate Stakeholders) would likely rely on the working group members to provide the training and education on the Test Services Menu concept to their respective organizations. Again, 2 months was estimated to give time for those working group members to educate all of the appropriate members of their organization.

(U) Mid-Term Actions 1 (Implement Test Services Menu) and 2 (Track Tier 1 and Tier 2 Metrics) would be accomplished in parallel. As mentioned in Mid-Term Action 1, initial Test Services implementation would be best with a small set of developments as a pilot program (3 months), followed by menu, baseline and process improvements (1 month) and then the full rollout (8 months). Long Term Action 1 (Update and Refine Menu) would be a reoccurring activity that would be accomplished indefinitely to tweak the Test Service Menu and update the test baselines. The Gantt Chart below provides a proposed timeline for implementing these actions.



(U) Figure 5: Recommendation 2 Roadmap

(U) Improve Communications Between All Stakeholders (Recommendation 3)

(U) Challenges Addressed: Challenges 3, 5 and 7

(U) One challenge identified from reviewing the system development lifecycle was the minimal communications between AED, COG and IV&V between the initial requirements development and the test tag-up meeting. Improving communication between those three stakeholders would help improve the overall quality of the tools delivered as well as allow both the developers and the IV&V team time to react to changes in the tool target or capability.

(U) Immediate Actions

1. **(U) For Projects Longer than 1 Month in Duration, Add an Additional Tag-up/Design Review:**
For development efforts scheduled to last longer than 1 month, add an additional tag-up/design review between the requirements ERB and the test Tag-up meeting. Having design reviews at the 50% and/or 75% design/development completion point is common for many SW developments. For development efforts that are estimated at 6 months or more, adding more than one additional review may be warranted. The design review should include at a minimum an operator knowledgeable in how the tool/exploit will be used, the developer of the tool, and the IV&V tester who will be testing the tool, and it should focus on three key items:
 - a. Changes to requirements and the target system since the initial URD (COG)
 - b. Demonstration of the current state of the tool, with an opportunity for the operator to provide feedback (AED)
 - c. A walkthrough of the test scenario for this tool, which should include the tools used to deploy, execute, exfiltrate or erase the tool under test, as well as the basic steps involved in those processes (IV&V)

This extra review will allow COG provide new information they may have that can impact development, see the current state of the tool and give feedback, and review how IV&V plans to test the tool and provide ways to make those tests more operationally impactful. Having this review at the 50-75% completion point will give AED and IV&V time to react to these changes and minimize the impact on schedule.

2. **(C) Identify and Procure new IV&V Database:** The current IV&V database is the one of the key communications tools for showing the status of projects within IV&V. The IV&V team is already

investigating replacements for this database before Lotus Notes Databases are decommissioned. This would be a good opportunity to expand that investigation into implementing a more robust test management tool, such as Rational Quality Manager or HP's Quality Center. It may be beneficial to have a more full featured test management capability to support Recommendation 2 (Test Services Menu), as there will be multiple groups conducting testing and a central location to track all test activities would make it easier to generate metrics and track the effectiveness of activities in the long run. Although these tools are generally more expensive than a simple SharePoint or Access database, procuring a more robust tool now would be a way to start Long Term Action 1, and ease integration with the other SDLC tools later.

(U) Mid-Term Actions

1. **(U) Implement Agile SW Development Methodology for Large Projects and Tools That Require Continuous Development:** There are a number of COGs tools that are frequently updated and upgraded. Although most evidence is anecdotal, past SW development efforts at AED that utilized Agile like methods had better success meeting operator expectations. Agile development methodologies stress more frequent communications between, users, developers and testers, and are a good fit for the COG tools that are used the most and are frequently being updated. Agile would also benefit larger scale new development efforts that involve long development timeframes or coordination among multiple developers. The benefits of implementing Agile on smaller scale, one off developments may not exist, however increased communication per Immediate Action #1 will still benefit these projects.

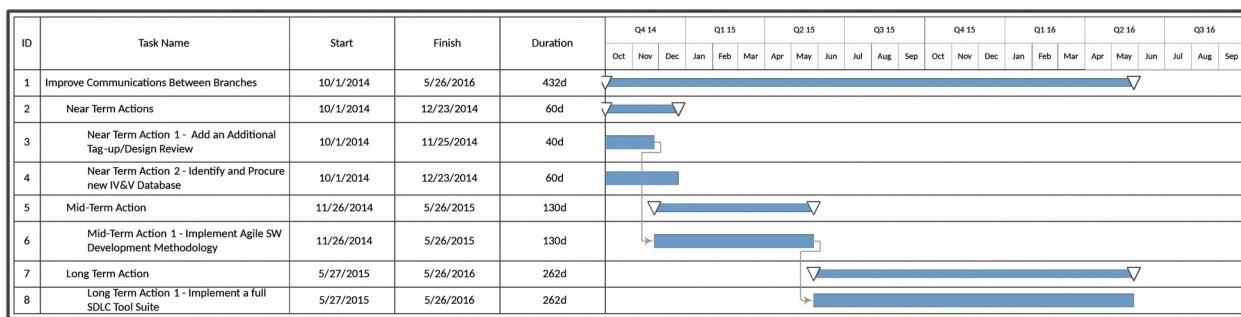
(U) Long Term Actions

1. **(U) Implement a full SDLC Tool Suite:** A full SDLC tool suite, such as Rational Jazz or possibly Crystal Castles, can provide an easier way for developers, operators and testers to communicate and share information without having to schedule meetings. These tool suites also make it much easier to track metrics and manage projects throughout the SDLC, including allowing individual users to set up their own metrics dashboard that focus only on their metrics of interest. EDG already has many pieces of a typical SDLC tool suite with the Atlassian tools, which include Stash, JIRA and Bamboo. These pieces can be integrated with the other SDLC tools such as requirements and test management tools, to create the full lifecycle tool suite. The newest versions of these tool suites commonly use web interfaces and include social media like functionality to make communication between the different stakeholder groups easier and more automated. As an example, an operator can update a requirement or CONOPS in the system once they have more information about a target. Once that change is made both the developers and testers working on that project receive a message, alert or e-mail a message notifying them that a change has been made and they need to review it. This happens automatically within the tool, without requiring the operator to take any additional actions. Additionally, these tools can be set up to keep the changes as draft until a certain personnel have reviewed and approved the changes.

(U) Implementation

(U) Implementing Immediate Action 1 (Add Design Review) is something that could be accomplished immediately, by adding an additional meeting at the 50-75% development completion point for projects longer than one month. After adding the meeting, it shouldn't require extensive effort for IV&V, COG and AED to put together their respective discussion topics. Immediate Action 2 (IV&V Database Replacement) is already in work, as IV&V is in the process of identifying a replacement for the IV&V database. If this effort were expanded to include more robust test management tools, then it may increase the timeline for acquiring the tool as funding for the tool and getting approval for hosting on CWE would be necessary. Additionally, acquiring a dedicated test management tool could be deferred to Long Term Action 1 (Implement Full SDLC Tool Suite).

(U) Mid-Term Action 1 (Implementing Agile Process) proposes a process that requires more communication and coordination between all key stakeholders. Implementing this on a larger scale will require some coordination between stakeholders to ensure they have the resources positioned to support the extra meetings and demonstrations associated with Agile. In order to implement Long Term Action 1 (Full SDLC Tool Suite) the SDLC processes should be stable and all stakeholders should be comfortable with them before integrating new tools. Additionally, there are a number of pieces of the SDLC tool suite that already exist within the Atlassian tool suite resident in the DEVLAN. Identifying how to integrate those tools, with the existing requirements management tool ServiceNow and any additional test management tool suites, or whether to transition to an already integrated tool suite will require detailed analysis and planning, pushing this effort out by 18 months or more. The Gantt Chart below provides a proposed timeline for implementing these actions.



(U) Figure 6: Recommendation 3 Roadmap

(U) Focus Dedicated DART Automation Support (Recommendation 4)

(U) Challenges Addressed: Challenges 5 and 6

(C) One recommendation that was expressed by a number of stakeholders, was having automated tests that would test the most used/relied upon COG tools against OS patches and PSP updates on a regularly occurring basis. DART is likely the best way to implement this type of automated testing. Focusing initial DART development on these types of use cases, will provide the most initial value for EDG and COG overall. Since almost all EDG branches will have their own instances of DART running, these efforts can be shared between branches, providing additional value. Finally, focusing on reducing the overall skill set required to work with DART, by creating a Test Harness or set of libraries for commonly executed

tasks within tests, will allow new staff or staff with minimal Python experience to more quickly come up to speed and develop DART scripts.

(U) Immediate Actions

1. **(U) Establish a Central Repository for Sharing DART Scripts:** Currently, DART scripts are shared either through drop boxes or e-mail. Today, there is no way for DART developers to know what scripts may exist for a specific tool, other than calling the different personnel developing or testing against that tool. Setting up a Central Repository of DART scripts, that is accessible by COG, AED, ESD and IV&V, will allow for easier sharing and reuse of scripts across the DART user community. In addition to having a central location accessible to all stakeholders, the repository should be structured so it is easy to find scripts associated with specific tools or functionality. As an example, each specific tool would have a folder where scripts developed specifically for that tool can be placed. There could also be a folder for scripts that focus on OS, PSP and language pack combination testing, or other similar tests that would be useful across a wide range of tools. After the central repository is in place, all DART users would be instructed to upload whatever scripts they have created to the appropriate folders in the repository. Utilizing AED's implementation of Stash, which is a source control repository tool that already has an area for DART scripts, may be the easiest way to accomplish this. The operators also have an instance of Stash in the OSN, however synching between the two instances would not be possible without changing the location of the operator's version. It may be possible to relocate the operator's Stash instance to HICKOK, which would allow synching with AED's version of Stash, similar to how the JIRA instance works today.
2. **(U) Identify Most Critical COG Tools that Require Repetitive Testing:** Work with COG to identify the top tools that would require "Patch Tuesday" and continuous PSP type testing. This initial effort should focus on only the most critical tools to keep development timelines in the near term.
3. **(U) Prioritize DART Automation on Critical COG Tools:** Identify DART development staff that can focus on creating "Patch Tuesday" and PSP tests for these critical COG tools, and have those staff develop the DART Scripts. The focus of these tests should be to create scripts that can be run on a schedule, likely overnight, without any human interaction. This will require these scripts to have error handling and fault tolerance built-in, so that tests will continue to run, even if encountering small issues or failures that don't have an effect on the functionality being tested.

(U) Mid-Term Actions

1. **(U) Prioritize Regression Testing for Top Tools That Undergo Continuous Development:** After creating a suite of tests for conducting "Patch Tuesday" testing and PSP testing for critical COG tools, DART development should focus on creating full regression tests for those tools that are continuously updated or enhanced. Again, the focus of these scripts should be creating tests that can be run automatically without human interaction.
2. **(U) Add DART Script Development to the Project Development Process:** During the requirements definition phase, when the Test Services Menu is being completed, any automated tests that would logically fit within those test services should be identified. As part of the development process, the developer would create a draft of those scripts that would be

provided to IV&V at the Test Tag-up Meeting. IV&V would then be able to refine and update that initial draft, instead of having to develop new scripts from scratch. Both the developer's script and IV&V's script would then be provided to COG as part of the tool delivery, giving COG operators a starting point to create their own automated test scripts if needed.

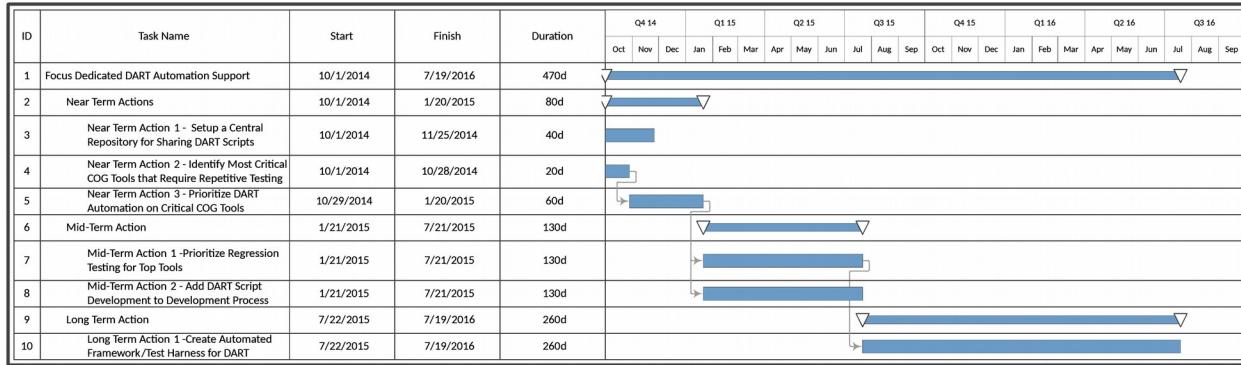
(U) Long Term Actions

1. (U) **Create Automated Framework/Test Harness for DART:** As DART developers get more familiar with creating DART scripts, there will likely be a set of functionality within these scripts that are exercised by the majority of DART tests. Developers should focus on creating a set of libraries, or a test harness, that simplifies those common activities shared across scripts. User guides or training will also likely be required to teach all DART users how to operate the test harness. This will allow personnel with limited Python experience to more quickly get up to speed and scripting, by calling those libraries instead of having to script those steps themselves.

(U) Implementation

(U) Implementing Immediate Action 1 (Shared DART Repository) will likely provide the greatest positive impact on initial DART development. As mentioned previously, AED and IV&V already have access to the current repository in Stash. Identifying a mechanism for synching this repository with COG's version of Stash may be the easiest and quickest mechanism for accomplishing this. Immediate Action 2 (Identify Critical COG Tools) can likely be accomplished quickly, as many COG stakeholders have already identified their most critical tools. Once those tools are identified, Immediate Action 3 (Prioritize DART Automation) can be accomplished. The overall timeline for this activity will likely depend on the skills of the developers selected. The personnel used to develop these prioritized scripts should be comfortable and familiar with DART development. It is likely that development of these scripts will get easier as each script completes, as some functions will be reusable between scripts. Assuming that knowledgeable developers are used, and development focuses on only the most critical tools, this effort should be achievable in three months.

(U) Mid-Term Actions 1 (Prioritize Regression Testing) and 2 (Add DART Script to new development efforts) are activities that can be conducted in parallel. Ideally Mid-Term Action 1 would use the same staff that accomplished Immediate Action 3 (Prioritize DART Automation). Since regression tests are normally more extensive than the patch or PSP tests, they will likely take longer to develop. Mid-Term Action 2 would be added as part of the development and planning effort. At the URD phase, any automated tests would be identified, and then would have to be planned as part of the overall development effort. Initial drafts of the automated tests could be delivered by AED, who then had those tests over to IV&V for review and completion. This package of DART tests would then be delivered along with the tool to COG. Long Term Action 1 (Create a Test Harness), could utilize the same staff from Immediate Action 3 and Mid-Term Action 1 or a new set of personnel who have been identified for their DART expertise. Creating a framework/test harness will take time as the harness will need to be tested and training will need to be developed to help DART users understand how to operate it, which would likely push the completion of this effort out past 18 months. The Gantt Chart below provides a proposed timeline for implementing these actions.



(U) Figure 7: Recommendation 4 Roadmap

(U) Connect Test Environments Through a Central Shared Services Environment (Recommendation 5)

(U) Challenge and Observation Addressed: Challenge 4 and Observation 3

(U) Currently, there are multiple environments where tests are conducted, but very little connectivity between those environments. Most sharing of data, tools and tests seems to occur through primarily manual processes such as e-mail, drop box or thumb drives. Having a central environment that can be used to test related services, tools and data would make testing across all branches more efficient. As an example, housing DART within a common environment would allow developers, testers and operators to share and execute DART scripts easily, without having to manually transfer and modify the script to reflect any changes in their implementation of DART. Another example would be having netflow analysis equipment available in the common environment for testers to send the netflow generated by their tests.

(U) Immediate Actions

1. **(U) Identify the Organization that will Own and Administer Shared Services Environment:** The first action should be to identify the organization that would own and manage the Shared Services Environment. The initial plans for the NDB Test Range included some of the concepts around shared services, however current plans are to focus on the specific needs for NDB testing for the near term build-out. Since the NDB Test Range is still looking for a permanent location and in the process of planning their future architecture and services, the NDB Test Range may be the most logical option for taking over this responsibility. Additionally, many of the capabilities that are proposed for the NDB Test Range, such as netflow analysis, traffic generation and network performance emulation, fit well within the Shared Services concept. Other organizations that would be alternatives would be IV&V and AED, however neither is currently in the process of defining their environment architecture or looking for new space for their environments.
2. **(U) Define the Services, Tools and Capabilities of the Shared Services Environment:** Convene a working group to define the services, tools and capabilities that can be shared. In addition to stakeholders such as COG, AED, ESD, and IV&V, an organization intimately familiar with the C&A concerns of connecting these environments and services should be included as part of the

working group. Initial candidates for shared services would likely include DART, the operator framework, netflow analysis, traffic generation, network performance emulation and any automated forensic analysis.

(U) Mid-Term Actions

1. **(U) Define the Detailed Shared Services Environment Architecture:** Due to the security risks and concerns associated with each environment, an effort to develop detailed network architectures to support connections between the different environments that exist today will likely be necessary. This effort should focus on ensuring that the detailed design can support the services, tools and capabilities defined in Immediate Action 2 above.
2. **(U) Define Shared Services Migration Plan:** As shown in the Environments section above, there are many different environments that are being used to conduct testing. Based on the detailed network architecture, a migration plan should be created to identify the order of environments to be integrated and the schedule for integration. An initial environment that can be used to test the shared services connectivity should be identified.

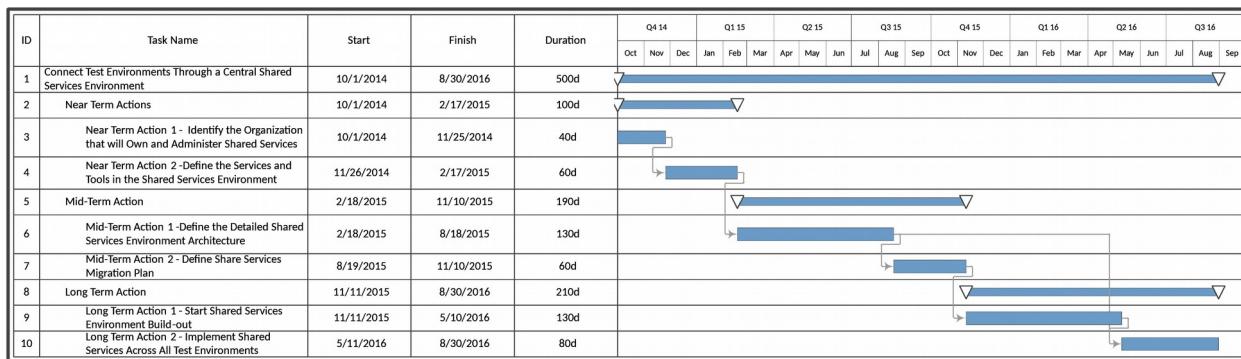
(U) Long Term Actions

1. **(U) Start Shared Services Environment Build-out:** Start building out the Shared Services environment in accordance with the final architecture. With one environment connected to shared services, test each service as it is installed and checked out. Initial tests should include some performance tests to start identifying a performance baseline.
2. **(U) Implement Shared Services Across All Test Environments:** After the Shared Services are running, and have been checked out by the initial test environment, continue the integration of environments in accordance with the migration plan. As each new environment is connected, checkout tests should be run and at least a minimum level of performance testing to compare against the performance baseline.

(U) Implementation

(U) The implementation of this recommendation is likely to be the most challenging of all of the recommendations in this report. Due to the differing classification levels of the environments and policy surrounding them, it may not be possible to integrate all of the shared services recommended above. Even if only half of those shared services were implemented, it would likely still provide meaningful efficiencies for future testing efforts. The first step would be to accomplish Immediate Action 1 (Identify Shared Services Organization) to identify who will own and manage the shared services environment. As stated above, the NDB Test Range is still in development, the future plans for both their environment and equipment needs is in work, and many of the services they will provide would fit within the shared services model making them a logical choice. Once an organization is selected, they can lead a working group to determine Immediate Action 2 (Define Shared Service and Tools). This working group should include stakeholders from AED, ESD, IV&V, and COG as well as personnel knowledgeable about the network restrictions and policy related to connecting the different test environments. This effort should allow ample time for coordination and approvals, and which would probably require three months.

(U) Once the services are identified Mid-Term Action 1 (Detailed Network Design) can be accomplished. This would likely be an extensive effort (6 months), as the design would likely require approvals from network security and other IA stakeholders. Following the network design would be Mid-Term Action 2 (Migration Plan), where coordination with stakeholders from each test environment can be initiated to identify the best timeframe to integrate their environments with shared services. Once the migration plan is complete, Long Term Action 1 (Shared Services Build-out) can start, which again will likely require an extensive timeframe (6 months) as each service will need to be tested, after it is built to ensure proper function and baseline performance. Finally, after Build-out is complete, Long Term Action 2 (Integration with All Environments) can be started. This will likely take one month per environment, as the initial connections will need to be made, services will need to be tested and performance baselined, all while the normal workload of testing continues. The Gantt Chart below provides a proposed timeline for implementing these actions.

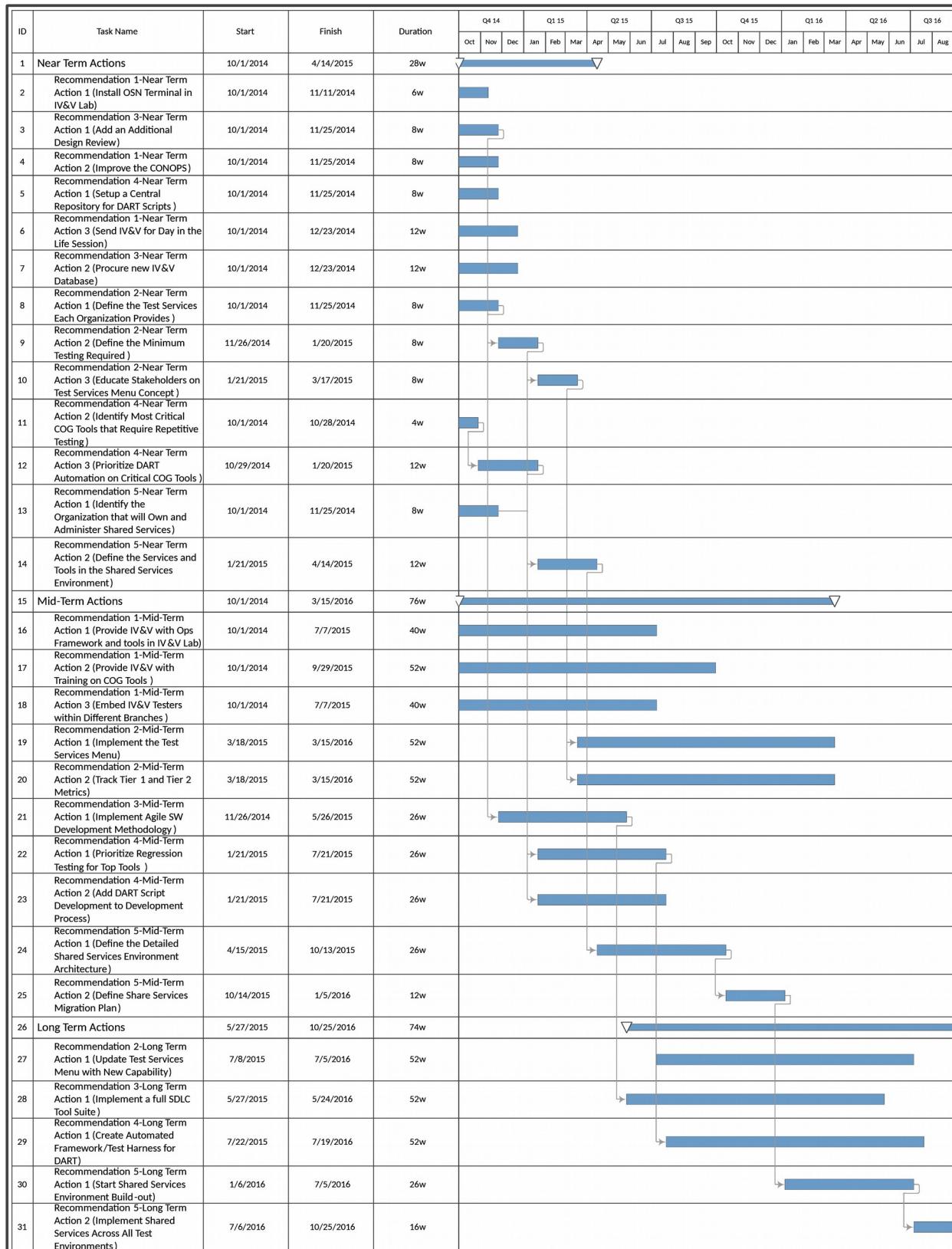


(U) Figure 8: Recommendation 5 Roadmap

(U) Roadmap

(U) The recommendations above each have a series of immediate, mid-term and long term actions associated with them, consisting of 28 total actions. Some of these actions will be easier to implement than others, and many of the actions require coordination between stakeholders to define processes and services. Additionally, some of these actions can be accomplished in parallel, while others may be reliant on another action completing, either to define a process or because the stakeholders or resources used to accomplish an action are already being used by another. The following Gantt Chart proposes a way to sequence these actions, which could become a roadmap for the rollout of these recommendations. As is shown, the easier actions to implement are identified at the beginning, as well as the actions that can be accomplished in parallel.

SECRET//NOFORN

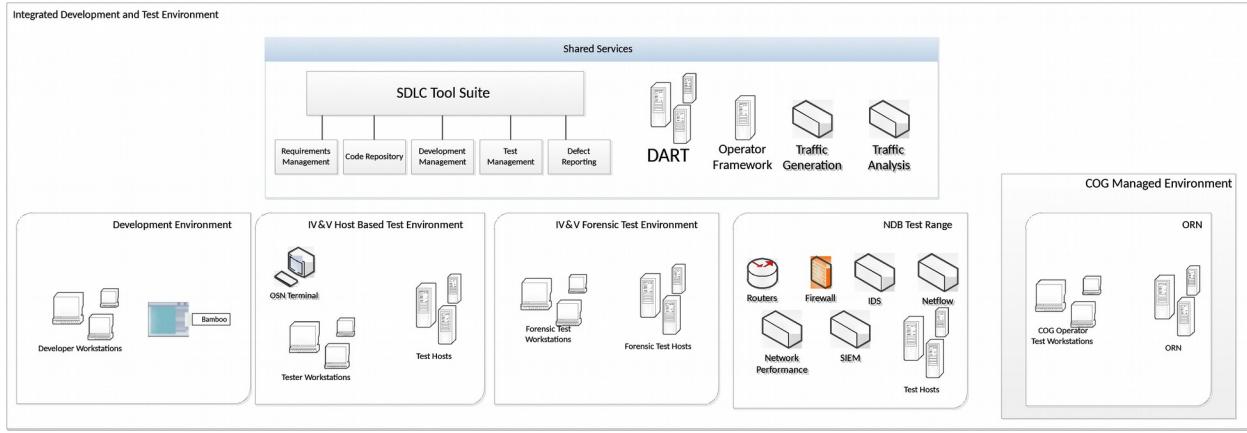


(U) Figure 9: Overall Recommendation Implementation Roadmap

(U) EDG Vision

(U) The goal of the recommendations provided above is ultimately to progress EDG's processes and environments towards a future EDG Vision, where the development and test processes provide maximum efficiency while minimizing time to market. This EDG Vision is based on what the processes and environments could look like if someone were to develop them from the ground up with no existing processes and environments for reference, using systems engineering and development best practices. With the large number of systems/tools produced and updated each year, the goal of this vision would be to maintain high quality product deliveries while maximize the efficiency of the resources available. There are two key principles that define this. The first principle is: developing high quality products requires communications between developers, operators and testers that are effective and timely throughout the development process. Changes in requirements need to be communicated to everyone as quickly as possible, so development and test plans can adapt without delaying schedules. Likewise operator feedback into development and test efforts is crucial to ensuring the end product meets operator expectations. This would be accomplished through the use of an SDLC Tool Suite, as detailed in Recommendation 3 above, which can automate change notifications and approval processes, without requiring extensive additional meetings. An SDLC Tool Suite, used in conjunction with more Agile-like development methods, which include some additional targeted meetings between stakeholders at key points in the development process, will ensure that developers, operators and testers stay in sync throughout the development process and minimize disruptions to delivery schedules.

(U) The second key principle is: developing high quality products requires increasing the rigor of testing for each project developed. This increased rigor would ensure that all projects are tested against a baseline set of functional, performance, host-based and network forensics, with additional tests tailored to specific tools as necessary. Implementing the Test Services Menu from Recommendation 2 will set the baseline level of rigor necessary across projects, while keeping test schedules within operational need dates. In order to accomplish this testing without increasing the number of resources, it will be necessary to set up mechanisms to easily share and reuse tests, data and tools. A shared services environment, as proposed in Recommendation 5, would allow developers, IV&V, the Test Range, and operators to quickly and easily share tools and tests, which would reduce the overall workload across all branches. (U) Figure 10 below provides a high level overview of what the EDG Vision could look like.

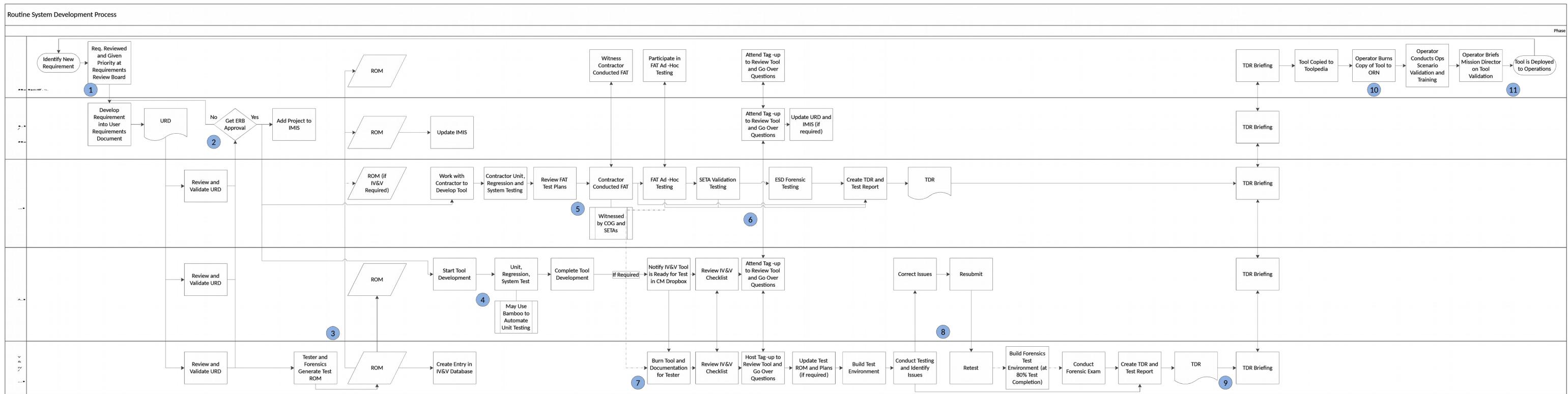


(U) Figure 10: EDG Development and Test Environment Vision

(U) As mentioned above the key feature of this type of environment is the ability to quickly share tools, tests and data between different environments and test stakeholders. As an example, developers would be able to test their code against IV&V developed regression tests in DART, before delivery to IV&V. IV&V would run their tests through Network Traffic Analysis equipment, while generating a typical network traffic profile to provide a network signature analysis test, without adding an additional network traffic test event. Additionally, all users would be able to utilize a common operator framework, which would include all of the tools and frameworks used by operators to deploy and operate the systems under tests. Also within this environment would be the SDLC Tool Suite to make it easier to track, manage and provide metrics on all of the development and test activities. (U) Appendix 2 provides a walkthrough of the updated SDLC that incorporates the use of these tools as well as the implementation of recommendations such as the Test Services Menu.

(U) It may not be possible to implement a full shared services environment as envisioned above, however implementing the recommendations contained within this report will likely provide EDG with measurable improvements to tool quality, testing rigor, process efficiency and time to market.

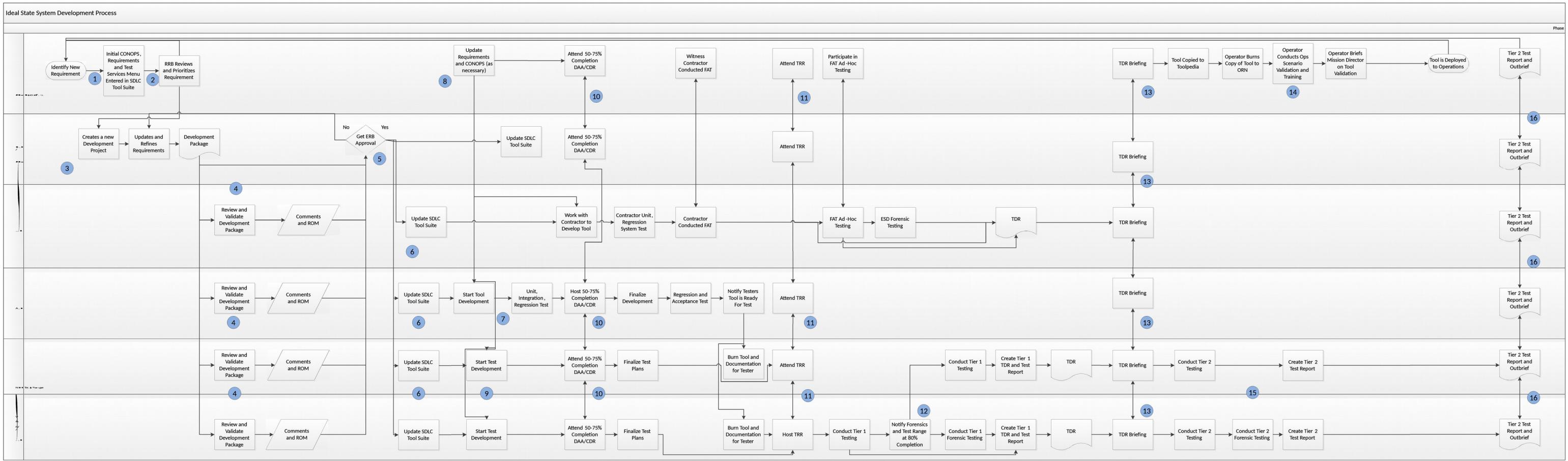
(C) Appendix 1: EDG SDLC Process



- (U) COG Operators define a requirement to add a new capability to an existing tool or a request for a new tool. The requirement is added to the list of other new requirements, reviewed at Requirements Review Board and if accepted given a priority. COG/OED takes the initial requirement(s) and develops a User Requirements Document (URD).
- (U) AED, ESD, and SED/IV&V review and validate the URD and give initial feedback. The URD then goes to the Engineering Review Board (ERB) for formal approval, where AED/ESD formally agree that they can develop the system/tool according to the requirements and timeline in the URD. After approval the URD is entered into IMIS and is tracked as a development effort.
- (C) SED/IV&V is notified that the URD is approved and they develop a rough order of magnitude (ROM) for a test schedule. SED/IV&V team tasks a tester and a forensic analyst to review the URD in detail and develop a ROM for IV&V testing and a forensic exam (if required). AED, ESD and COG will review the ROM, and provide feedback. After the ROM is accepted, IMIS will be updated to reflect the time required for IV&V.
- (U) AED developments will take the URD and start development. This normally follows a waterfall development model, where the developer produces a system/tool in a single release. Some projects have used a more Agile development process, where there are more frequent discussions and demos with the COG operators throughout the development process, and anecdotally this has produced better results. Unit testing is left to the developer, and the level of unit testing accomplished varies on a developer to developer basis. AED has procured a software system called Bamboo that will do automated deployment and unit testing on a regular basis to help improve the level of unit testing accomplished, but use of Bamboo is not required at this time. During development, some projects may produce an evaluation copy of software for the operators to use to get a feel for the tool. This evaluation copy normally has had little to no testing and is not generally considered ready for operations, however there have been cases where evaluation copies have been used operationally. Once development is complete, the developer will notify IV&V that the system/tool is ready for testing.
- (U) ESD staff work with an appropriate external contractor to develop the system/tool to meet the URD requirements. This process includes the normal unit, regression and systems tests culminating in a Factory Acceptance Test (FAT), and recently has been updated to include ad-hoc testing at the end of the FAT. ESD staff (government and SETA as appropriate) will review FAT test plans and procedures, and along with COG operators, travel to the contractor facility to witness the FAT. There are some occasions where IV&V staff have witnessed FATS as well; however that is not a regular occurrence. At the conclusion of the contractor led FAT, ESD staff and COG operators will conduct ad-hoc testing, which normally allows the operators to test the limits of the tool and validate a limited set of use cases. Because contractor facilities vary, the level of operational validation that can happen varies on a case by case basis. At this point, if the tool will go to IV&V for testing, they will be notified that it is ready for test.
- (U) ESD may conduct an additional Validation Test. This testing is accomplished by the ESD SETA contractors and focuses on operational validation of the tool. Additionally, ESD's forensics team may conduct their own forensic exam against the tool.
- (C) IV&V is notified that a tool is ready for testing. They go to the drop box on the DEVLAN and burn a copy of the tool and any accompanying documentation, and the IV&V checklist is reviewed to determine the level of testing that was accomplished against the tool. The tester reviews the documentation, and hosts a Tag-up meeting with AED/ESD and COG to review the tool, documentation and any questions with the developer and operators. At this point, it is likely that the operator has additional information on the target, and the capabilities of the tool and the scope of testing may change. If tool changes are required the tool goes back to AED and documentation is updated.
- (U) Once the tool is ready for IV&V, the tester will build the test environment and conduct testing in accordance with the test plan. If issues are found they will communicate those issues with the developer, and send the tool back to be fixed if necessary. Testing will halt and will require a retest of all completed tests. At the 80% completion mark, the IV&V forensics testers will start to build the forensic environment for the tool. IV&V forensics conducts their examinations in a separate forensics environment on a clean system to ensure they are getting accurate forensic data. Forensics will conduct their testing, and follow the same process if they find an issue.
- (U) The testers and forensics team will create the TDR briefing and a test report. SED, AED/ESD and COG will meet to review the TDR results and COG will formally accept the tool.
- (U) COG will have a copy of the tool and all documentation loaded onto Toolpedia. They will then burn a copy of the tool and deploy it to the Operational Research Network (ORN) to conduct operational scenario validation and training. The operational scenario validation is not a procedural driven test and is more like an operational rehearsal to ensure that the tool fits within the operators work flow. Operators may run another test against the target OS and PSP combination, to ensure the PSP updates since IV&V testing don't flag the tool.

11. (U) The COG operator will brief the Mission Director on the all of the testing and validation that has been conducted against the tool and request that the tool be deployed to operations.

(C) Appendix 2: Example Walkthrough of EDG Vision Process Flow



- (U) COG operator identifies the need for a new SW based exploit tool to support an upcoming operation. The COG operator logs into the Requirements Management section of the SDLC Tool Suite and:
 - (U) Defines a CONOPS for the tool.
 - (U) Includes what other COG Tools will be used in conjunction with the new exploit.
 - (U) How the operator intends to use the tool after it has been deployed.
 - (U) Defines high level functional requirements
 - (U) Defines target system at a high level
 - (U) Defines initial Tier 1 and Tier 2 testing
- (U) COG branch chiefs/deputy branch chiefs are notified of the new requirement by the SDLC Tool Suite and it is added to the Requirements Review Board (RRB). At the RRB, the priority of this new development effort, the final need date, and the Tier 1 and Tier 2 test requirements are defined.
- (U) COG/OED is notified of the requirements approval by the SDLC Tool Suite, and they create a new development effort within the tool. OED can then work with the COG operator to refine the initial requirements into a set of requirements that can be used for development.
 - (U) As part of this package creation, attributes can be set for requirements and artifacts can be linked to different parts of a package. As an example, a CONOPS could be linked to the tools that are used in conjunction with the tool under development. Later a tester could search on that same set of tools identify any other CONOPS that use a similar tool set. If one of those CONOPS is similar to the current CONOPS, then they may be able to modify those test plans and test scripts instead of creating those tests from scratch.
- (C) SED-IV&V, NDB Test Range and AED are notified that a new development package, which includes URD (with CONOPS) and Test Services Menu, is available for them to review and comment on. SED-IV&V, NDB Test Range and AED provide comments to the requirements and CONOPS through the comment system in the SDLC Tool Suite. Also within the tool suite, IV&V, NDB Test Range and AED can provide draft schedules for both development and testing efforts.
- (U) The final development package is reviewed at the Engineering Review Board, along with the draft development and test schedules for review. This review can happen using either the tool itself or artifacts automatically generated by the tool. Final approval of development package artifacts is completed.
- (C) IV&V, NDB Test Range and AED start creating tasks within the tool and assign those tasks to specific testers and developers as necessary. Each task includes a date or schedule for when that task will be complete. Managers in IV&V, NDB Test Range and AED can then develop their own dashboards within the tool to track task status and metrics.

7. (U) While AED is creating the tool developers check their code into and out of the code repository, which is integrated with the SDLC Tool Suit and automated build and unit testing tools like the currently procured Bamboo tool. Each day the developer will work on specific tasks within their effort and at the end of the day, check in their code. The SDLC Tool Suite will then do an automated build and test against the code overnight. The next morning, the developers review the results of the test and determine if there are any new defects that need to be corrected. Defects would be created and tracked within the tool, either using the defect tracking mechanisms of the tool suite itself, or by integrating the existing JIRA system into the tool suite. Fixing these defects then becomes a new set of tasks, which provide additional granular detail to SW development metrics.
8. (U) Over time, the COG operator will gain new insight into the target system, and be able to provide additional detail to the development package. The operator will log into the requirements management subsystem and either change or add new requirements to the development effort. These changes and/or additions could be related to the CONOPs, functional requirements, target system or Test Services Menu. The testers and developers that are assigned to this development effort will be automatically notified by the tool suite that changes have been made and need to be reviewed. As in step 4, comments and discussion on these changes or additions can happen within the tool itself. The tools will also keep a history of changes for every requirement and artifact. The system can be set so that these changes are not final until one or multiple people approve the change. Handling changes within the tool, will allow the operators to make updates as quickly as they can identify them, maximizing the amount of time the developers and testers have to address those changes.
9. (U) While development is in work, testers start developing tests plans, procedures and other test artifacts, against the Tier 1 and Tier 2 tests identified in the Test Services Menu. These plans and procedures can be based off of templates created within the tool suite. Ideally, test data sets and automated test scripts will also be integrated within the tool suite, to allow easier sharing between tests. If the new tool's CONOPs closely matches that of another tool, previous test scripts may be able to be leveraged to lessen the amount of new development required.
10. (U) At the 50-75% Completion point, AED, COG and IV&V conduct a design review to discuss status of exploit development; review changes in requirements, target system and CONOPS; demonstrate initial concepts; and review initial test plans and strategies. For Agile Development projects, this type of meeting would occur for every Sprint within a release. After this meeting, developers will complete and finalize code and testers will complete and finalize their test plans.
11. (U) Hold a Test Tag-Up/Test Readiness Review (TRR). This meeting will review any final changes to requirements and/or CONOPS and the results of development testing. The outcome of this meeting is to formally hand over the system to the testers to complete their evaluations.
12. (C) After the TRR, IV&V will execute their Tier 1 functional verification, integration, and operational validation tests to ensure the tool operates in accordance with both the requirements and the other tools and processes used by the operators. Like in the current process, at the 80% completion point, Forensics (IV&V/ESD/AFD) as well as the NDB Test Range would start building their environments and executing their Tier 1 tests.
 - a. (U) While IV&V is conducting their functional tests, they would use shared services, such as Network Traffic Analysis tools to collect data that can be used by the NDB Test Range for later analysis. For updates to existing tools they may use some shared DART regression test scripts.
 - b. (U) Discrepancies found during testing would be entered into JIRA or through another part of the SDLC tool suite. These defects would be linked to specific tests, requirements, CONOPS or tool interactions, which over time would allow for more detailed analysis of development efforts. As an example, by linking discrepancies to tool interactions, you could create queries that show the amount of defects associated with specific tool interactions, and identify the most complex development areas. If a specific tool is repeatedly causing defects, updating that tool with a better API or integration interface may reduce complexity of operations and future developments.
 - c. (U) All test results would be input and tracked through the test management portion of SDLC tool suite. Depending on the SDLC tool suite there are multiple ways to enter and store test results, including a number that will take csv or xml results generated by specific tools. Branches across EDG and COG would have read access to this part of the tool suite to track testing progress and see test results.
13. (U) After completion of Tier 1 testing, results from all IV&V, Forensics and the NDB Test Range tests would be consolidated into a Tier 1 Test Report and presented at the TDR, as is done in the current process.
14. (C) After TDR out brief, COG operators would conduct whatever their operational rehearsals and training. By adding more functional integration and operational validation in IV&V, COG's operational rehearsals can focus more as a dry run for the actual operation, rather than the operational validation that is done today. As in the current process, the operator would still brief the Mission Director on all testing accomplished before getting approval for deployment.
15. (U) While COG operators are conducting their rehearsals and training, IV&V, Forensics and the NDB Test Range would continue executing the Tier 2 tests identified in the Test Services Menu.
16. (U) At the conclusion of Tier 2 testing, a Tier 2 Test Report would be created and out briefed to all stakeholders. Although Tier 2 testing would likely start immediately following the Tier 1 TDR, it is likely that most Tier 2 testing wouldn't be complete until after the initial operational deployment for that tool.

(C) Appendix 3: Stakeholders Providing Feedback

Interviewee	Organization	Date
Andrew K.	EDG/AED	18 Aug and 25 Sep
Brian J.	COG/OED, Vencore	15 Aug
Carole G.	EDG/SED, Vencore	7 Aug
Albert M.	COG/NOD Branch Chief	7 Aug and 26 Sep
Jeff P.	EDG/SED-IVV, Booz Allen	30 July
Sonja B.	EDG/SED-IVV, Booz Allen	30 July and 16 Sep
Marcus D.	EDG/SED-IVV, Booz Allen	31 July
Tyrone T.	EDG/SED-IVV GPOC	6 Aug and 16 Sep
Kara W.	EDG/ESD/SDB	21 Aug
Keith F.	EDG/SED Division Chief	7 Aug
Mike P.	EDG/ESD/NDB	14 Aug
Mike S.	EDG/ESD/SDB	27 Aug
Misty L.	EDG/ESD/SDB, Booz Allen	15 Aug and 16 Sep
Russell B.	EDG/ESD	25 Aug
Stephen P.	EDG/ESD Deputy Division Chief	21 Aug
Suzanne B.	EDG/ESD/SDB, Booz Allen	15 Aug
Mario V.	EDG/AED/RDB Branch Chief	26 Sep
Mike W.	EDG/SED, Vencore	7 Aug
Amy C.	EDG Deputy Group Chief	14 Aug
Kevin K.	EDG Technical Director	14 Aug
Robert W.	EDG/SED,TREMOR COTR	16 Sep