**Component: NAVY**

**Topic #:** N201-081

**Title:** Automatic Coding Standards Validation Tool

**Technology Areas:** Info Systems

**Acquisition Program:** TRIDENT II (D5) in support of Strategic Systems Program (SSP) ACAT I

**ITAR:** The technology within this topic is restricted under the International Traffic in Arms Regulation (ITAR), 22 CFR Parts 120-130, which controls the export and import of defense-related material and services, including export of sensitive technical data, or the Export Administration Regulation (EAR), 15 CFR Parts 730-774, which controls dual use items. Offerors must disclose any proposed use of foreign nationals (FNs), their country(ies) of origin, the type of visa or work permit possessed, and the statement of work (SOW) tasks intended for accomplishment by the FN(s) in accordance with section 3.5 of the Announcement. Offerors are advised foreign nationals proposed to perform on this topic may be restricted due to the technical data under US Export Control Laws.

**OBJECTIVE:** Develop and demonstrate an automatic static code analysis tool that can dynamically adapt to changing Navy organizational software standards and will incorporate Strategic Systems Program (SSP)-directed coding standards such as prologue categories (e.g., authors, notes, description, argument list), structured programming conventions, consistent indentation and comment location, and identification of redundant source code.

**DESCRIPTION:** Formal Navy software developments employ coding standards for C++ that are required to be met to ensure the software is maintainable, readable, and demonstratively of high quality. Currently these type of standards can only be validated manually, which takes time and resources. As the organization moves to an agile software development process that values the use of automation, an automated static analysis tool for C++ is necessary to identify standards violations.

Hardware requirements dictate that this tool be operational and maintainable on both RedHat Linux on Intel x86 and Solaris Sparc systems and that it is able to be executed stand-alone and integrated in an automated development and testing pipeline (DevOps). The tool should have an option to automatically fix the standards violations on an item-by-item basis.

Currently there is a tool to automatically perform static analysis for C code but no tool to do this for C++ code. The tool should perform analysis and correction processing on software consisting of both C and C++ code. An additional benefit would be that the user can tailor the portion of code to be analyzed and possibly fixed.

SSP and Naval Surface Warfare Center, Dahlgren Division (NSWCDD) will be able to provide feedback to allow for expedient operational tool updates.

**PHASE I:** Determine technical feasibility of automated static analysis of those standards on C and C++ code, with a chosen/given set of versatile C and C++ standards. Develop approaches to creating a tool that is easy to adapt to different C and C++ coding standards and determine a mechanism for validating the automated tool. Identify risks to the technical approach and develop Phase II plans that include ways to mitigate those risks.

**PHASE II:** Produce and deliver prototype software and associated test cases. Work with the Navy to fully understand the coding standards implemented and provide a draft installation guide for Linux and Solaris systems and a user’s guide. Work with the Navy to troubleshoot the software and resolve implementation and execution issues. Establish a feedback loop with NSWCDD for implementing changes due to findings during prototype testing.

**PHASE III DUAL-USE APPLICATIONS:** Deliver software that can be dropped into NSWCDD automated testing and development environment and distributed to other Navy technical partners. Work with the Navy to provide updates and fix issues. Establish a maintenance agreement that allows evolution of the tool.
This product will perform static analysis according to customizable coding standards. The fact that the coding standards are customizable makes the product marketable to a wide set of commercial software development applications.

REFERENCES:

KEYWORDS: Automated Testing; Coding Standards; C/C++; Software Validation; Solaris/Linux; Static Analysis; DevOps