

iEx6
Jon Rollman
11.15.11

1.

Firstly, the ability of the environment to verify syntax, auto-indent and provide streamlined access to documentation is very helpful and greatly increases efficiency with respect to development time. Secondly, the development of a non-trivial software product creates the opportunity for inconsistency; therefore, a robust testing suite is preferred. Lastly, with the vast array of devices currently available and their continued diversification, it is preferable to develop in an environment which supports the compilation into various machine specific binaries, if compiled, and, if interpreted, supports interpretation into various formats compatible with associated systems currently in popular use.

2.

I enjoy programming using the Dracula environment for the following reasons. Firstly, the Dracula programming environment provides an easy to use interface for basic testing and development. Secondly, Dracula provides useful libraries for testing and development. Thirdly, Dracula and its associated components are reasonably well documented.

3.

I believe that the Dracula programming environment could be improved by addressing the following issues. Firstly, the Dracula programming environment makes use of a somewhat complex syntax for test input specification. Secondly, Dracula has limited intrinsic test input data structure generation capabilities. Thirdly, Dracula provides limited information regarding test failures.



Undefined Software, Inc.

853 Main Street, Nowhere, OK 72156

Interoffice Communication

Date: November 14, 2011
To: Charles Nevel, COO
From: Jon Rollman, Computer Scientist
Subject: Reactor Control Software Development Environment

Executive Summary

Successful development and deployment of the nuclear reactor control system relies directly upon the efficiency, correctness, completeness, and computability of every component in the developed software system. The proper choice of development environment will play a key role in ensuring this outcome. Therefore, it is clear that the development environment must support a functional language, provide robust component testing and promote effective interaction among developers.

Rational

Firstly, procedural programming languages introduce inconsistency into software systems. For high risk systems, a functional language protects against this negative side-effect. Secondly, the development of any non-trivial software product will produce inconsistencies that must be eliminated prior to deployment. For this reason, a robust testing suite is required. Lastly, due to the scope of a project of this magnitude, the effective collaboration among developers is paramount. An effort to promote effective communication and a streamlined software development process is critical.

Recommendation

For the above mentioned reasons it is my recommendation that two distinct development environments be used for this project, Eclipse and Dracula. Eclipse is a versatile integrated development environment providing substantial build flexibility, developer interaction and development in a variety of languages including scheme, and Dracula is a functional development environment with substantial testing and code validation capabilities. In addition to the above mentioned rational, existing organizational familiarity with these environments will ensure a seamless transition into the development phase of this project.