Joshua Holmes
112702995
Software Engineering I
11/15/11

iEx6 - Joshua Holmes

1. a) Easy-to-use debugger
   b) Simple commands to compile and run
   c) A way to save states during debugging

2. a) The syntax highlighting was good, especially with regards to parenthesization
   b) The theorem checker was nice, it was really easy to verify if your work was actually working the way it was supposed to.
   c) It's multi-language, which is great to have in an IDE (even though I only used ACL2)

3. a) The debugger was impossible for me to figure out
   b) Errors were sometimes hard to decode if it wasn't a simple syntax error
   c) There is no auto-complete or anything similar

Joshua Holmes
112702995
Software Engineering I
11/15/11

4. To whom it may concern:

A decision has been made that the programming environment we will use for our nuclear reactor control software will be Microsoft Visual Studio 2010. This decision has been made factoring in a number of issues, including the ease of development in the Visual Studio environment, the intuitiveness of its debugger and its ease of use in team environments.

Visual Studio provides support for multiple languages, including C++ and C#. These two languages seem to be the industry standard these days (aside from Java), and if we choose to go with either of these languages, we should be fully prepared for the tasks we will have to deal with. Its native language support includes things such as Intellisense (Microsoft's auto-complete system that will give suggestions and complete text for you, as well as do things like show function call parameters and types) and excellent syntax highlighting, which should greatly expedite the development processes as well as ease the strain on our developers.

Microsoft Visual Studio's debugging environment is one of the best around as far as ease of use and intuitiveness are concerned. It's incredibly simple to set breakpoints on specific lines of code and stop execution exactly when you need to. The debugger provides a great environment for stepping through execution line-by-line, as well as see all local variables and their values, visible names, call stacks and manipulating memory in real-time. The use of all of these great debugging features should make it a much more manageable task to maintain the codebase for such an expansive piece of software, and especially considering the importance of the task at hand and the sensitivity of nuclear control software, bugs are simply unacceptable in a project of this magnitude.

The Visual Studio environment also includes built-in support for version control systems such as Subversion. This will make it a breeze to integrate changes made by the various developers across all development machines without having to purchase and learn how to use third-party software; this is simply built into the environment. Again, with a project so expansive and critical, smaller luxuries such as these will surely come in handy and prove more than useful as development gains its footing in the new environment.

I hope everyone feels the same as I do on this matter and as you can see, Visual Studio should prove to be a wonderful development environment that will ease the burden on our developers.