Ruth Shih
CS 4263
Software Engineering I
iEx6
11/15/2011

1. Describe three services that a good programming environment should provide.

A good programming environment should provide a highly accessible way to compile and run a program. It should also provide a debugger and a clean, color-coded source code editor.

2. Describe three things you like about the Dracula programming environment.

I like the interactions console in the Dracula programming environment. I also like that it clearly shows which parentheses pair up with each other in the code editor and that it auto-tabs the lines to the correct place.

3. Describe three things you dislike about the Dracula programming environment

I dislike that the way to place breakpoints in the debugger is not obvious. I also dislike how hard it is to see where the arrows are pointing in “check syntax” mode for large files and that there are no keyboard shortcuts to comment and uncomment lines.

4. Suppose you are a software engineer in an organization that has received a contract to design and implement the control software for a new kind of nuclear reactor to be constructed in a densely populated area. You have been assigned the task of choosing the programming environment to be used in the project. Draft a one-page memo that specifies the programming environment and provides a convincing rationale for that choice.

TO: Project Manager

FROM: Ruth Shih, Software Engineer

DATE: November 15, 2011

SUBJECT: Nuclear Reactor Control Software Programming Environment Decision

I have been asked to choose a programming environment for the new nuclear reactor control software. Since this reactor will be located in a populated area, I took error-checking into account in my decision. I also believe it would be best to write the software in a high level language such as C++ so that narrowed my choice down to programming environments that provided C++ compilers.

I believe Visual Studio 2010 is an appropriate choice it has many features suitable for this task. Visual Studio provides

- a straightforward and detailed debugger,
- keyboard shortcuts for convenience,
- automatic compile-time checking,
- an outline view of the project architecture,
- and the ability to search through large projects relatively quickly.

More importantly, it provides a way to create unit tests and view code coverage. Safety is the biggest concern for this project, and I believe these mechanisms will help greatly in testing the code to check its correctness before it is deployed to the reactor.