Nha Hoang (aka Liv)
Software Engineering I
Dr. Rex Page
**iEx6 – Programming Environment Survey**
Tuesday, November 8, 2011

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

   A good programming environment should provide three following services: debugger/error detector (both syntax and runtime errors), searchable documentation/useful help desk, and interaction support. For any programming language that the environment is designed for, it should be capable of detecting syntax and runtime errors. Error detector here does not only mean finding errors but also locating errors, determining what errors they are, or even better, recommending how to fix detected errrors. The capability of recommending error fix of a programming environment, in my opinion, will help novice programmers a lot in learning the language, as well as experienced programmers to be more efficient in coding. Along with debugger/error detector service, a good programming environment should provide handy access to good documentation of the language, and trouble shooting of the environment itself. Finally, graphical interaction support is a common need in most applications nowadays.

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

   The first thing I like about Dracula is its simple look and user friendly;
   The secon thing I like about the Dracula programming environment is its error reports. The error to be displayed in the Interactive Window is the first error encountered. In other words, programmers, especially novices, are pointed directly to the error which may be the cause of a chain of other errors in
   The third thing I like about Dracula is its mechanism for reasoning and test suites; this feature make me get familiar to industrial application requirements.

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

   The first thing I dislike about Dracula is its unuseful help desk and not very many documentation about Dracula out there for searching.
   The second thing I dislike about Dracula is that it is uncapble of locating potential portion of code that produces errors, especially when you have a long program.
   The third thing I dislike about Dracula is that I cannot find a way to close tabs in Dracula.

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.

University of Oklahoma

# Memo

To: Dr. Rex Page

From: Nha Hoang

CC: Allen Smith

Date: 11/15/2011

Re: Choice of Programming Environment for Nuclear Reactor Control Software

---

For our organization newly received contract of designing and implementing the control software for a new kind of nuclear reactor to be constructed in a densely populated area, choosing the best programming environment to be used in the project is the essential start for the best final product. Following are the most crucial specifications that a programming environment should have to be the most appropriate choice for our project and reasons why they are essential :

- Firstly, the environment should be extremely "sensitive" with all types of error, for the nature of the project: it is a safety-critical system; a single software or software-engineering error could cause catatrosphic results. Every error and warning should be reported to the programmer until it is fixed. Along with error detection, the environment should have the capabality of analyzing errors and recommending different approaches (if there are more than one) to fix detected errors.
- Secondly, the environment should support testing. Again, the reliability of the final product is our most important goal, it is most likely that the process of this development will follow the test driven development or extreme programming approach. In other words, we will want our final product to pass as many tests of all levels as possible and thus, the environment should support all kinds and all levels of test: "Ad-hoc" tests, automated tests, unit tests, integration tests, acceptance tests, etc.
- Thirdly, the environment should provide useful help desk, as well as handy access to reliable and up to date documentation. In other words, it should provide access to approved site that documents issues other programmers have encountered in the past. (Note that these site may require authentication)
- Fourly, the environment should support graphical interaction. This specification is particular for analyzing purpose. Although it is impossible to anticipate every situation, our clients need to be protected against each worst-case scenario. Hence, graphical demonstrations for analyzing worst-case scenarios and how to recover is a crucial requirement of this project, especially when the nuclear reactor will be constructed in a densely populated area.

It is, however, difficult to find a programming environment that meets all the above four requirements. One recommendation is Dracula but it still need to be improved on documentation, in my opinion. However, the choice of the most appropriate programming environment only gives facility for the best outcome to achieve the highest confidence in the reliability of system designed to protect workers and the public.