To: Jim Beam, Provost
From: Carla Bradley, Dean (CCIS)
Subject: choice of programming language and IDE for new project

A memo starts with these three lines, and that's all there is to addressing the recipient.

After careful considerations of our new customer's requirements, my team and I have come to the conclusion that the Racket language is the best possible choice. We will develop the code in the DrRacket IDE and the Emacs text editor, using Greg Henderschott's well-known `racket-mode` package.

Start with an "executive summary."

Racket is an emerging programming language that permits a wide range of programming styles, from functional to object-oriented, from dynamically typed to statically typed. Racket's core is, like JavaScript and Python, a dynamically-typed, memory-safe scripting language. This language accommodates functional, dysfunctional, and object-oriented programming, which permits the rapid, interactive, and incremental development of a prototype. Beyond these ubiquitous abstractions, Racket also supports dynamically linkable modules and syntactic abstraction capabilities, all of which greatly enhance our ability to create single points of control within the software system. Critically, though, the language comes with an optional but sound type system, meaning that we can easily "harden" code, once we know that the prototype is to be converted into a full-fledged product. By adding meaningful types, we will significantly lower the cost for future maintainers who will need to understand and modify this code base.

Cover each aspect of the choice with a single paragraph, opening with a thesis statement.

Although Racket's on-line package repository is smaller than that of scripting languages, we have confirmed with preliminary experiments that it covers all requirements: JSON reading, TCP/IP connectivity, a parser for mark-down text, GUI wizards, and more. Indeed, I propose to contribute some of our modules to the open-source repo once the project is done; I am sure this act of generosity will attract attention.

This paragraph is about the libraries. Note how it sneaks in that we have done our homework ("experiments") and do not just make marketing claims.

The DrRacket IDE, a 20-year old product, supports numerous languages: Racket, Typed Racket, a variant of Datalog, and many more. It also comes with standard tools such as a debugger, a unit-testing reporting tool, and a plug-in for measuring expression-level test-coverage.

Next we move on to the IDE and tooling issue.

Finally, we have also come to the conclusion that the choice of Racket will attract outstanding developers, making up for the currently relatively small developer community. As in many cases, the developers who study and appreciate an emerging technology make up a self-selecting, highly productive group of people who love their work. For some evidence in support of this conjecture, look at our competitor, which increased its productivity by 50% to 100% after switching to Racket.

You may be surprised but this last paragraph is Jane St.'s rationale for using OCaml. Other trading houses are following its example, too.

This last point is BS; nobody knows how to measure the productivity of developers properly.