part 2:

When I was writing the program, I didn't consider the possibility of multiple users running the software simultaneously. Writing the SRS made me think about that possibility, and if I'd written the SRS first, I would have specified that any number of users can run the software simultaneously, and when writing the software I would have found a way to avoid potential race conditions between multiple users.

Part 3:

Giving the SRS definite purpose (Riordan p. 6) can be achieved by describing why features are useful and not just stating that they exist. The feature that allows precompiled headers to be specified for source files is an example of this; many c++ developers don't know that header files can be precompiled, and they may be interested in using them to improve build performance.

The plan/draft/finish cycle (Riordan p. 58) can be followed when writing the SRS. The first part of planning would be looking through the SRS template and thinking about how to divide the content among the different sections. Each major section of the SRS -- 1.0, 2.0, and 3.0 – are created with an iteration of the plan/draft/finish cycle.