0. A. Our Team Name is Team TODO.
   B. The names and PIDs of the team members are:
      i. Ryan Szilvasi (rszilvasi@vt.edu)
      ii. Tyler Streator (tys305@vt.edu)
      iii. Junxiang Feng (junxiang@vt.edu)
      iv. Kechen Yu (kechen21@vt.edu)
      v. Xiaolong Xuan (xxiaolong@vt.edu)
   C. Our current preliminary project idea is a TODO application for a Computer Desktop.
1. Five hypothetical non-functional requirements could be:
   a. The text on the application has high contrast with the background
   b. The application's skin theme can be changed to a day theme or a night theme.
   c. The application should be implemented in a cross-platform language and work on all major operating systems
   d. The application can be eye-catching without affecting the user's actions.
   e. The application is free to drag and resize.
2. Five hypothetical functional requirements could be:
   a. The application emails/notifies the user of the work they need to complete by the end of the day.
   b. The application must allow users to input some TODO events.
   c. The application can adjust the priority of the event at any time.
   d. The application should be able to categorize or tag tasks to organize them.
   e. The application should support sharing and assigning tasks to other users.
3. The specific tasks required to complete the non-functional requirements would be:
   a. The specific task required to complete the high-contrast functional requirement would be a part of the HTML/CSS side of the application. We believe that this task will not be hard to implement because you set to variables in your program, font size and background color, which we would rate this task a 5 on the amount of effort scale.
   b. Theme skin requires being able to interact with the background color palette on the front-end framework and be able to set the whole application theme with just one click. With the task's nature, I would rate it a 5 on the effort scale.
   c. The application must be programmed in Java and at minimum tested on Windows, Linux, and OSX. The language choice will be made early in the process and is unlikely to greatly increase the implementation difficulty. Java is already a largely portable language, so the cross-platform testing probably won't uncover too many problems, and the difficulty of this task rates around 15.
   d. The application needs to test whether it will cause user confusion and affect the user's attention under normal use. I would rate it a 20 on the effort scale.
   e. The application needs to implement efficient data loading and rendering to optimize its performance. We believe that this task is important for people since it is about efficiency. I would rate it a 20 on the effort scale.

   The specific tasks required to complete the function requirements would be:
   a. The specific task required to complete the function requirement would be to have the user input their email or phone number so that they can be notified. We don't believe

that this task is particularly hard as it is asking for user input. We expect that this would be rated as a 13 on the amount of effort scale, in case the user does not input an email.

b. The specific task required to complete the TODO events function requirement would be able to accept user input like event name, event description, event assignee, etc. With the complexity of the user input and dealing with the user input, I would rate this task a 40 on the effort scale.
c. The system must allow users to edit event details and should implement a priority queue for displaying tasks. I would rate it a 20 on the effort scale.
d. Each task must include a list of keywords or tags that users can edit, and the display system must allow filtering or sorting based on this. I would rate it a 40 on the effort scale.
e. The application needs to test its messaging function. I would rate this task a 40 on the effort scale.

4. User Story 1: One example of a user story can be between software developer, Bill, and his manager, Tom. Bill has several important deadlines coming up as per Tom's request. As an unorganized individual, Bill needs a way to help manage his tasks, thus he could utilize something such as this Desktop application. Bill can be notified at the start of the day with the tasks he must complete before the business closes. The application's high-contrast features will ensure that Bill does not miss any important features.

   User Story 2: Another example of a user story is that Tom sometimes forgets to add people to the required meetings for the project. Utilizing the application sharing and assigning features, Tom can invite individuals to attend these important meetings to ensure their tasks get finished on time without much interruption. The individuals can look at the priority of these meetings to evaluate their importance.

   User Story 3: The final user story involves Jack, a poor student who was rushing to meet his due dates for a CS project, CS homework, and a math exam. He was very confused and did not know which one to start with and how to use his time reasonably. This is when he could utilize something such as this Desktop application to help him prioritize his coding assignment tasks.
5. Two risks that could potentially impact this project would be the requirement changes and defective components. Requirement changes are a huge risk when developing software applications. Clients may not be the best at explaining what they need, which may result in changes in the system. A solution to mitigate this risk would be to constantly check in with clients and users to ensure that our system meets their needs. Secondly, defective components would be another risk in this system as this type of application is new to us all as a group. To mitigate this risk, we need to provide adequate testing on our system to ensure that all components work as intended.
6. If this application were to be created, we suggest utilizing the prototyping process, as the developers can receive ample user feedback to ensure the application works successfully and meets the user's expectations. This would ensure that all parties involved with building, maintaining, and using the applications are well-informed and pleased with the results.