

## CS3704 Project Workshop - Bagel Devs

Zachary Lahlou  
Maddie Oehler  
Conor Edwards  
Arik Scofield  
Maceo Cardinale Kwik  
Esther Kim  
Annie Tran

0a. Come up with a team name for your group.

**Bagel Devs** (we can make the checkboxes little bagels)

0b. Please list the names and PIDs of the team members who are present today (or knowingly absent)

Zachary Lahlou (lahlouzn@vt.edu)  
Maddie Oehler (mmo815@vt.edu)  
Conor Edwards (conore@vt.edu)  
Arik Scofield (ariks@vt.edu)  
Maceo Cardinale Kwik (Maceo@vt.edu)  
Esther Kim (estherkdy@vt.edu)  
Annie Tran (anniettran@vt.edu)

0c. Provide your preliminary project idea (or set of ideas). This is not a commitment to a project. Using the approved idea for your group's course project, complete the following activities related to requirements analysis.

- **TODO App**
  - **Desktop application to help developers keep track of tasks and prioritize items.**
- FocusBot
  - Help team members to be productive while working remotely (Provide productivity aids, such as pomodoros, daily reminders, incomplete tasks, etc.).
- Virtual Standups
  - Design an online platform to conduct virtual standups with remote teammates

---

1. Provide an example of five hypothetical non-functional requirements for this system. Be sure to include the specific type of requirement discussed in class, with each requirement coming from a unique category.

1. Usability: a user friendly interface for the application (navigation)

2. Performance: quick response time to organize tasks and items , able to store many tasks/items

3. Supportability: Ensuring the desktop application is easily updated and maintained adhering to coding standards

4. Implementation: able to store data on tasks, notify users of upcoming tasks, etc.

5. Reliability: available to access and use any time without error (testing)

2. Provide an example of five hypothetical functional requirements for this system.

1. Stores tasks between launches
2. Allows completing or deleting tasks
3. Displays tasks as a list
4. Stores a description with each task
5. Allows searching by task name

3. Think of a specific task required to complete each of the functional requirements and non-functional requirements mentioned above (10 total). Estimate the amount of effort needed to complete this task using function points (i.e., using the values [here](#)). Briefly explain your answer.

Non-functional:

1. Usability
  - a. Outline and test aspects of GUI to ensure best usability - 5
2. Performance
  - a. Minimize unneeded logic and code, and minimize storage space needed - 3
3. Supportability
  - a. Maintain and update the app with improvements - 2
4. Implementation
  - a. Implement main logic and functionality with storing tasks - 5
5. Reliability
  - a. Watch for and fix bugs and issues after launching - 3

Functional:

1. Stores tasks between launches
  - a. Requires encoding and decoding a task object - 5
2. Allows completing or deleting tasks
  - a. Deleting an object is easy, completing is just moving the task - 2
3. Displays tasks as a list
  - a. Graphical front-end - 8
4. Stores a description with each task
  - a. Just adding a string to the task class - 1
5. Allows searching by task name
  - a. Needs a string fuzzy-matcher - 3

4. Write three user stories from the perspective of at least two different actors. Provide the acceptance criteria for these stories. (Annie Tran)

1. User Story: As a developer, I want to see which tasks I have completed, are in progress, and have not started because it will help me decide what tasks to work on.

Acceptance Criteria:

Scenario: The developer adds tasks to the To-Do list application and labels them by completion. As each task is labeled, it places the completed, in progress, and not started tasks in different columns.

2. User Story: As a team lead, I want to check up on the progress of my team members because it will help me understand where we are in our work.

Acceptance Criteria:

Scenario: Team members are able to share their list to the team lead. The team lead is able to see which tasks are labeled as completed, in progress, and not started.

3. User Story: As a developer, I want to organize tasks by priority so that I can complete my tasks more efficiently.

Acceptance Criteria:

Scenario: The developer adds tasks to the To-Do list application and labels them by priority. As each task is labeled, it organizes the list by priority with the top priority tasks at the top.

5. Provide two examples of risk that could potentially impact this project. Explain how you would mitigate these risks if you were implementing your project as a software system.

1. A potential risk is code getting lost with incorrect usage of version control. We would mitigate these risks by making sure each team member understands how to use git and version control.
2. Another potential risk is a potential conflict within two team members. We will mitigate that risk by being clear about expectations and communication standards so that everyone can be on the same page about what to expect.

6. Describe which process your team would use for requirements elicitation from clients or customers, and explain why.

- Our project would align more with iterative process models because of the flexibility and adaptability that they offer. Specifically, prototyping and incremental models would work best because our project requirements may evolve over time.
- We would use the prototyping model to gather feedback and refine our project requirements. This would work well for this specific project because it would allow for us to visualize early on in our process what the user interface would look like and how it would operate. This way, we as developers can make changes before the final product is developed.

- We would use the incremental approach to break down our project into smaller steps. This would allow us to identify and prioritize features that are most important based on feedback and overall app functionality. This approach also allows us to estimate where we are in the process and determine final deadline dates.