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

Team Name: SBT

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

Team Members:
Srikaran Bachu - srikaranbachu@vt.edu
Bhargava Elavarthi - bhargava@vt.edu
Tim Son - tims03@vt.edu

0.c Provide your preliminary project idea (or set of ideas). This is not a commitment to a project.
Automated Code Review Assistant
Intelligent Task Assignment and Prioritization
Automated Test Case Generation

Using the approved idea for your group's course project, complete the following activities related to requirements analysis.
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.
Performance requirement: should be able to process large amounts of code in a short time period (seconds)
Security requirement: Should not leak users code to other users
Scalability requirements: Be able to handle multiple users looking for code reviews simultaneously
Usability requirement: Should be intuitive and easy to navigate through the application.

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

1. User Authentication, and Authorization
2. Code review & workflow management
3. Code Analysis Automation
4. Performance & Scalability
5. Version Control Integration

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.

- Implement the authentication system using a third party service such as OAuth (3 points)
- Horizontal Scaling and load balancing for scalability (6 points)
- Workflow management - developing a system that can manage all code review tasks (5 points)
- Code Analysis Automation - Detection of syntax errors, code style violations etc (8 points)
- Set up a method for identifying system performance times with code review tasks (4points)
- Encryption methodology for securing user credentials and code data (5 points)
- Create auto scaling version to auto adjust server capacity based on user size (5 points)
- Reliability requirement - implement failover systems to ensure high availability for users.

4. Write three user stories from the perspective of at least two different actors. Provide the acceptance criteria for these stories.
Story 1: Developer creates code review
Actor: Developer
Initiate code review for the changes, to make sure quality and requirements are met for the coding standards of the team before I merge it to main.
Acceptance criteria:
Developer should be able to access code from the version control interface
Should be able to add comments and feedback to the code
Should be able to assign code to proper reviewers based on the criteria

Story 2: Reviewer Provides Feedback
Actor: Reviewer
Provide feedback on code changes assigned, to improve the quality of the code base.
Acceptance criteria:
Receive notification when code review task assigned to them
Be able to view and review any code changes and comments on code assigned to them
Be able to submit feedback once they have reviewed all code changes.

Story 3: Project Manager Monitors Progress
Actor: Project Manager

Monitor progress on code reviews across the team to ensure everyone is still on pace to finish on time.
Acceptance Criteria:
Have access to a dashboard showing all current code reviews in progress
Should be able to filter and sort code review tasks based on priority and deadline
Should be receiving notifications on code reviews that are delayed or overdue.

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) Integration and Compatibility Issues
- Before development, analyze the most commonly used tools and environments
- Implement a beta testing phase where the system is tested in a variety of real world environments and fix any integration issues

2) Technology and Methodology Mismatch
- Before fully committing to a particular tech stack or methodology, conduct a study to evaluate its suitability for the projects requirements
- Adopt agile methodologies such as crystal clear or adaptive software development
- Using an agile approach we can more easily adapt our technology choices as the project progresses, based on ongoing learning and feedback

6. Describe which process your team would use for requirements elicitation from clients or customers, and explain why.
- Create a prototype and release beta version to test the likeliness of the product with users
- Interview Stakeholders to get a good understanding of what they are looking for and how we can factor their opinions and needs into our requirements for the project
- Provide customers with options to send feedback and or information
- Begin production with a brainstorming session to introduce new ideas and get an idea of what feature we need to implement from the beginning before we turn to fine-tuning said implementations at the end.