

SE 216 – SOFTWARE PROJECT MANAGEMENT
SOFTWARE MEASUREMENTS DOCUMENT

PROJECT NAME: RecipeBank

GROUP NUMBER and MEMBERS: Group 7 - Atakan Acaroğlu, Hasan Efe Ünal, İdil Sanem Gürsoy, Fulya Aydın, Mert Can Erdoğan, Barış Özdemir

Questions to identify measurements:

- 1- How many people will be needed for this project?
- 2- How much time will be needed for this project?
- 3- Which reason could cause a delay?
- 4- What are the challenges for writing the test cases?
- 5- How will the validation be done?
- 6- Is there enough documentation for the future works?
- 7- What are the bottlenecks and areas for improvement to drive process improvement activities?
- 8- Is the system able to scale, load times of pages or key functionality?
- 9- How likely is it that malicious users might breach the system or gain access to sensitive information?
- 10- How easy is the system to debug, troubleshoot, maintain and integrate?

Identified measurements:

- 1- Schedule and effort:** Track the date a task was accomplished vs. the date it was scheduled to be finished.
- 2- Time-accounting data:** Helps compare overtime vs. underutilization.
- 3- Number of subsystems:** This metric indicates the overall complexity of the project by counting the distinct functional components or modules within the system.
- 4- Lines of code:** Tracks the total amount of code written to develop the application.

SE 216 – SOFTWARE PROJECT MANAGEMENT

SOFTWARE MEASUREMENTS DOCUMENT

5- Amount of media: The amount of media refers to the total number of photographs, videos, or audio files in the recipe. This feature allows viewers to better appreciate the visual richness of the material.

6- Defect count : Used to indicate, not prove, how well the system is implemented and how effective the testing process is at finding defects.

7- Number of changes:

Budget is measured by money

Security is measured by number of vulnerabilities, time to resolution, deployment of updates, number and severity of security incidents.

Maintainability is measured by static code analysis and code complexity

8- Time Calendar: Team velocity measures the number of story points, quantifying the number and size of product features completed by the team in the sprints.

9- Open-close rates: Tracking production issues reported in a specific time period.

10- Application crash rate(ACR): Dividing how many times an application fails by how many times it used $ACR = F/U$

Measurement storage and collection:

1. Keep track of the number of personnel assigned to various project responsibilities (testers, designers, developers, etc.) as the development process progresses. This can be kept in a spreadsheet or project management application.

2. Monitor the dates of significant project milestones' scheduled and actual completion. Using a project management application, record deviations and the causes of delays.

3. Examine the causes of the delays in the project. These can be unanticipated dependencies, technological difficulties, or limitations on resources. Keep track of each cause's frequency to aid in future planning.

4. Describe particular difficulties you had when creating test cases for the different features of the RecipeBank application. Keep a record of this data in a document or test management system.

SE 216 – SOFTWARE PROJECT MANAGEMENT

SOFTWARE MEASUREMENTS DOCUMENT

5. Describe how the application is validated to make sure it complies with the standards. This could involve security, performance, and user acceptance testing. Keep validation plans and outcomes organized in a special repository.
6. Examine and rate the documentation that is currently in place for upcoming development and maintenance projects. Determine which parts of the documentation want improvement.
7. Continually pinpoint locations where development, testing, or deployment might be enhanced as well as process bottlenecks. Use a project management tool to keep track of these problems and suggested fixes.
8. Track and measure how well the program performs with different load levels (the number of users utilizing it at once). Utilize a monitoring tool to store performance indicators such as response times and resource usage.
9. Monitor and assess security events (data leaks, attempted attacks). Document the event's specifics, its effects, and the steps taken to resolve it.
10. Conduct recurring assessments of the maintainability of the codebase according to criteria such as the complexity of the code and the quality of the documentation. Maintain the evaluation findings and corrective action items.

Measurement Type	Description	Example Measurements
Response Time	The time it takes for the system to respond to users requests.	Response time (ms) should be less than 5 seconds.
Scalability	Applications ability to handle workload variation.	Databases storage capacity should be expandable
Product Size	Characteristics of the software product.	Number of components, line of code, documentation size

SE 216 – SOFTWARE PROJECT MANAGEMENT
SOFTWARE MEASUREMENTS DOCUMENT

Effort Distribution	The time spent at every step of the project.	Time spent by each member at each stage.
Quantitative	Tracks user completion rates for AI-guided tutorials or cooking steps.	Completion rate for a specific AI-powered recipe tutorial.
Quantitative	Measures application responsiveness and loading times.	Average page load time for displaying recipes.