SE 216 – SOFTWARE PROJECT MANAGEMENT
SOFTWARE MEASUREMENTS DOCUMENT

PROJECT NAME: **Servicify**

GROUP NUMBER and MEMBERS: **GROUP 1 (Barış Can Ceylan, Alperen Demirezen, Ege Sezak, Yasin Kızıltaş, Buğra Yurtsever)**

Questions to identify measurements:

What did the team produce? (sprints)

How much effort did this project require?

How much effort went to testing?

How many times has the code been tested?

How many commits have been done?

What is the defect density in the code?

Identified measurements:

Number of hours spent working.

Number of times code committed.

Number of sprints finished.

Number of test cases written.

Number of times that tests have failed.

Measurement storage and collection:

Number of hours spent will be entered into a pre-specified project spreadsheet at the end of each week at 6pm in real number data in hours.

Number of times the code has been committed to the repository will be tracked and entered into a pre-specified project spreadsheet at the end of every day at 6pm in real number data.

Number of times the tests have failed will be tracked and entered into a pre-specified project spreadsheet at the end of every day at 6pm in real number data.

SE 216 – SOFTWARE PROJECT MANAGEMENT
SOFTWARE MEASUREMENTS DOCUMENT

Number of sprints finished will be tracked in a pre-specified spreadsheet and entered once a sprint is finished.

Measurement Type	Description	Example Measurements
Hours spent working	This measurement measures how much time has the team spent working on the project.	At the end of each week, a monitoring program tracks the number of hours spent working by a developer.
Number of commits	This measurement measures how many times the code has been committed by a team member.	The amount of times a member committed code daily.
Testing	This measurement measures how often the code is tested.	A development team runs unit tests regularly before every commit.
Errors	This measurement measures errors that occur during the testing phase of the software and represent undesirable behavior.	In a form validation test of a web application, an unexpected error message is displayed when the user receives invalid input.
Defect Density	The number of faulty components or faulty lines of code in a product.	During a sprint, 1000 lines of code are written and 10 of them are buggy in subsequent testing. In this case, the error density is 1%.