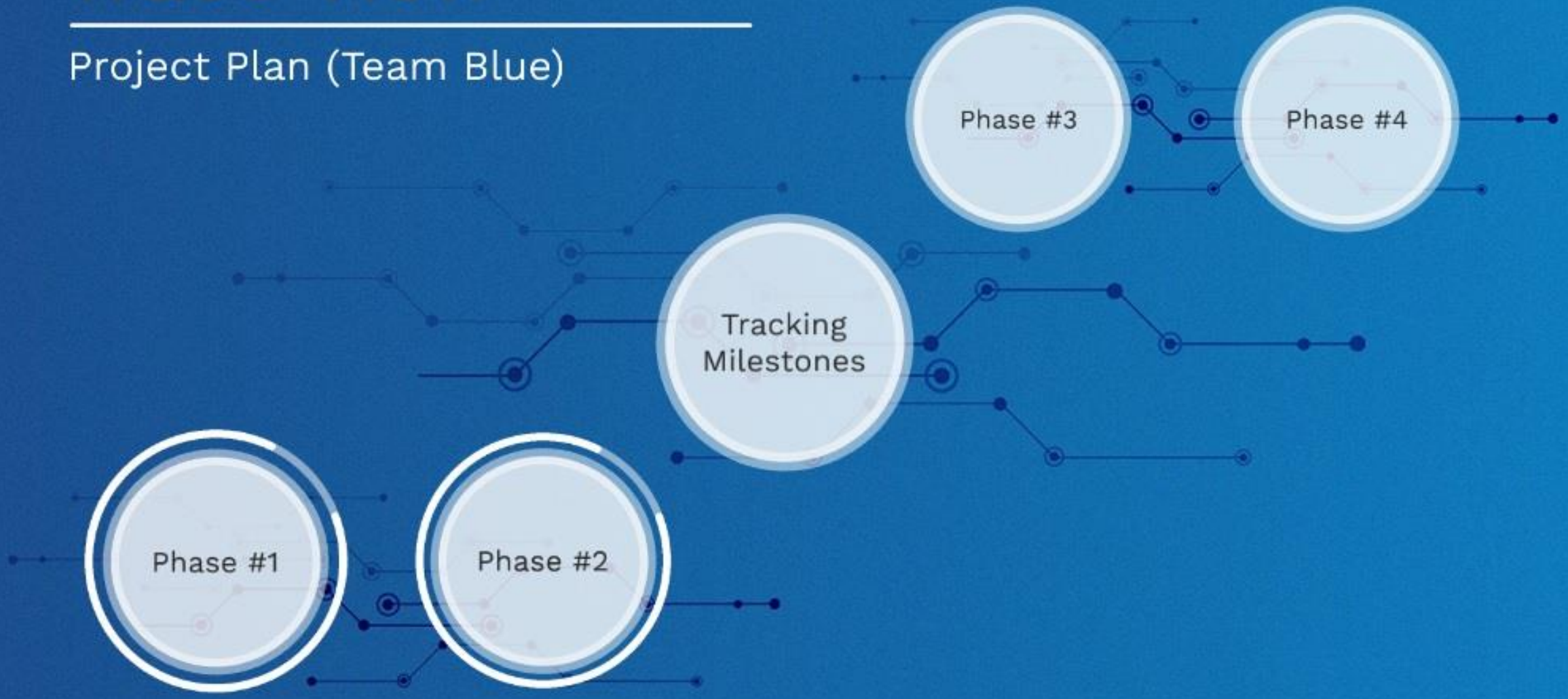
Trace View

Project Plan (Team Blue)



Trace File
(Static)

- Input Trace files to VS Code.
- Test operations (Create adjacent data structure for given trace file).
- Understanding and development of dynamic data structure for the trace file.

Task Breakdown

Task Breakdown

Task #1: Input Trace files to VS Code

- Input different static trace files to VS Code (Entire Team).

Task #2: Setup a custom data structure from trace file

- UML/EF diagrams according to trace files structure (Shah).
- Creating classes from the diagrams in VS Code (Alexis, Sarim).

Task #3: Base data structure for dynamic trace files

- Finalizing the data structure to be used in the entire project (Sarim, Alexis).
- Testing the created data structure and cross platform independence with Linux (Shah).

Milestones:

- An extension that reads any trace file and stores it into its built-in data structure for further manipulation.

Graph

- Research and select appropriate library to create graphs (Chart.js).
- Use data from the static file to create a graph.
- Add graphical features to extend data representation.
- Test

Task Breakdown

Task Breakdown

Task #1: Select appropriate library to create graphs

- Research and select the most suitable library (Entire team)
- Create graphs using sample data using the selected library (Entire Team)

Task #2: Create a graph using a static trace file

- Retrieve data from trace file to plot a graph (Alexis, Sarim).
- Integrate the created graph extension to VS Code (Shah).

Task #3: Add graphical features

- Select the data representation techniques (features) to be used in the graphs (Sarim).
- Implement the selected features (Alexis, Shah).

Task #4: Test

- Match the plotted graphs with the data from the trace file (Sarim).

Milestones:

- A graph is created from a static trace file with all its graphical features

Expected Milestones and next steps

Dynamic Trace Files and Graph

Trace Files

Graphs

Integration

Task Breakdown

Task #1: Input Dynamic Trace Files

- Using previous approach, implement a method to input dynamic trace files (Sarim, Shah).

Task #2: Integrate trace files with created data structure

- Create and implement methods to integrate dynamic trace file to previous data structure (Alexis).

Graphs

Task #1: Create Dynamic graph from dynamic trace file

- Use dynamic trace file input to create dynamic graph (Sarim, Alexis).
- Integrate created extension to VS Code (Shah).

Task #2: Adding features to dynamic graph (Entire team)

- Update previously created features and implement them to dynamic
- Add remaining features for dynamic graphs (dynamic scrolling, data points, legend).

Integration

Integrate created extension to VS Code

- Add created Trace View extension to VSCode (Entire Team).
- Testing created extension with multiple trace files (Entire Team).

Final Testing

- Cross platform independence testing.
- Testing of all functional requirements.