Assignment 4.2

Group 2

Michael Boeding
Drew Patel
Shuaipeng Yun
Lingshu Hu

In your groups, discuss these perspectives on Augur as a system

- **As a technical system**

  Components: APIs, database, algorithms for metrics, data visualization

  Interconnections: Augur uses APIs to collect data and store data in database, and then uses data from database to calculate the metrics, visualize them and present them to the users.

  Purpose: to collect data of open source software projects, generate metrics, and display them.

- **As a sociotechnical system for evaluating open source software project health and sustainability.**

  Components: open source software projects, community, evaluation

  Interconnections: augur evaluate the open source software projects in a community.

  Purpose: to evaluate open source software project health and sustainability.

- **As a sociotechnical system for**
  - **Wholes Rather than parts**

    Augur allows you to see the whole picture for an open source project. Whether that is being able to see how an open source project is being maintained or how active the community is. It allows you to see another side of the project that would not be possible otherwise. It also allows you to see a lot more relevant and accurate data about a project in one place.

- **Interconnections / Relationships**

  Interconnections/Relationships are represented in Augurs ability to connect open source project data and normalize it to show useful information. This allows the user to see relationships and patterns that might have been hidden before. Which results in a better selection of open source software.

- **Non-linear relationships**

  Non-linear relationships are represented in augur's metrics. In those metrics, the health and sustainability shown might have non-linear relationships. For example, if a project made 1000 commits, it might be better than one that made only 10 commits. However, one project that made 5000 commits might not be necessarily better than the 1000 ones.

- **Stock and flow relationships**

  Stock and flow Relationships are represented in Augur by allowing users to see the latest commits/changes to a project. This allows the user to see how active a project is and then make an educated decision on if they want to implement this open source project into their system based on the activity/maintenance of the project.

- **Dynamic Behavior**

  Since projects are developing, the augur's metrics are dynamic reflections of those projects. In this sense, the health and sustainability of a project is changing over time, and thus augur's metics changes accordingly.

- **Feedback loops**

  Augur's gives you stats of a repo with a goal of showing the health of the open source project. Developers will interpret that data and use higher quality libraries. These libraries will get more support and donations leading to better LTS which leads to better health that Augur will show.

- **Acknowledging that systems are important**

  Augur allows us to see and recognize different systems that may be in place in an open source project. This is very important because it allows us to see what kind of structure a certain open source projects. This helps us make a good decision on which project to use.

- **System as the cause of its behavior**

  Augur allows us to also analyze projects as a system based on the past behavior. Which allows us to see where the open source project may be headed in the future and gives us insights about things that may be added/removed.

- **System structure generates behavior**

  The open source ecosystem as an entire system is quite scattered. You have forks, you have uncommitted developers, and you have varying skill level of developers. Because of that structure you have repos that are both high quality and have LTS but also repos that are low quality and aren't being sustained. This exists in all things that are run by volunteers. Time commitments and life happens leaving the unpaid, side projects to suffer. This is where Augur becomes useful. It helps developers decide which repos to go with based on metrics.

- **Delays**

  There is a delay in a project's Augur rating. When an open source project is initially made, Augur won't rank it highly. However, every open source project has to start from somewhere. This leads to a delay until the software is seen as healthy in the eyes of Augur. It's basically the chicken and the egg problem.

	Wholes Rather than parts	Interconnections / Relationsnhips	Non-linear relationships	Stock and flow relationships	Dynamic Behavior	Feedback loops	Acknowledging that systems are important	System as the cause of its behavior	System structure generates behavior	Delays
Augur	x	x	x	x	x	x	x	x	x	x