Assignment 4.02

Systems Theory

Dodgers Group

Claire Hughey, Alex Centorbi, Alex Leduc, Kevin Zeng

	Wholes	Interconnections/relationships	Non-linear relationships	Stock and flow relationships	Dynamic behavior	feedbacklops	Acknowledging that systems are important	System as the cause of its behavior	System structure generates behavior	delays
CHAOSS/Augur	x	x	x	x	x	x	x	x	x	x

Perspectives:

- As a technical system
    - Able to source revenue from the service.
    - Community health would improve from analysis of Augur data.
    - Computed data can be used to help make accurate medical decisions.
    - Further implementation as an open source software can help create stable future updates.
    - Stability seems like a crucial point for this system, if it ever goes down lives could be lost.
    - API can be easily distributed for data access.
    - Can be used to easily display collected metrics and data.

- As a sociotechnical system for evaluating open source software project health and sustainability.
    - Augur allows users to evaluate free, open source software (FOSS) projects in a software community.
    - It shows the contributions made by multiple individuals collaborating on a project.
    - The health of the community as a whole can be measured.

- As a sociotechnical system for
    - Wholes Rather than parts
        - Augur allows you to see the whole scope of an open source project
        - It gives you statistics to see another whole side of the project that was before inaccessible.
    - Interconnections / Relationships
        - This is shown with Augur's ability to connect open source project data and make it display relevant and useful information.
    - Non-linear relationships
        - This is represented in Augur's metrics. Not everything is one to one. For example, you could have one user add lengthy commits hundreds of times to a project, but it doesn't exactly mean he's doing the best contributing, or even contributing at all.
    - Stock and flow relationships
        - Stock and flow is represented in Augur's ability to allow users to see the latest commits/changes.
        - This allows people to see how active a project is.
    - Dynamic Behavior
        - This is represented because since this works with projects in development, Augur's metrics are constantly changing.
    - Feedback loops
        - Augur provides metrics for open source projects that give developers an idea of the health and sustainability of those projects. This knowledge helps developers to make better informed decisions about which projects to become involved with. This increases the likelihood of success for open source projects as more support is given to sustainable projects rather than unsustainable projects.
    - Acknowledging that systems are important
        - Because Augur is capable of seeing which open source projects are healthy and sustainable, it allows its users to identify which systems are important to their communities.

        - Attaching good metrics to communities implies their worth to people viewing their project, meaning that projects with good metrics are important.
    - System as the cause of its behavior
        - Augur is capable of creating metrics based on past data.
        - This allows further analysis of future metrics based on past data.
    - System structure generates behavior
        - As an open source system, the data is easily and widely distributed. This means that, as Augur as a system improves, other developers are using the source code to create subsystems that use Augur code.
        - This open source software subsystems created using Augur git can be used in Augur, which can analyze the metrics and determine the quality of the newly created subsystems
    - Delays
        - When new open source systems are made from Augur, the rating system puts it at the bottom, and it must be evaluated by other users/developers before it can be recognized by the main Augur developers
        - This creates a delay from using Augur