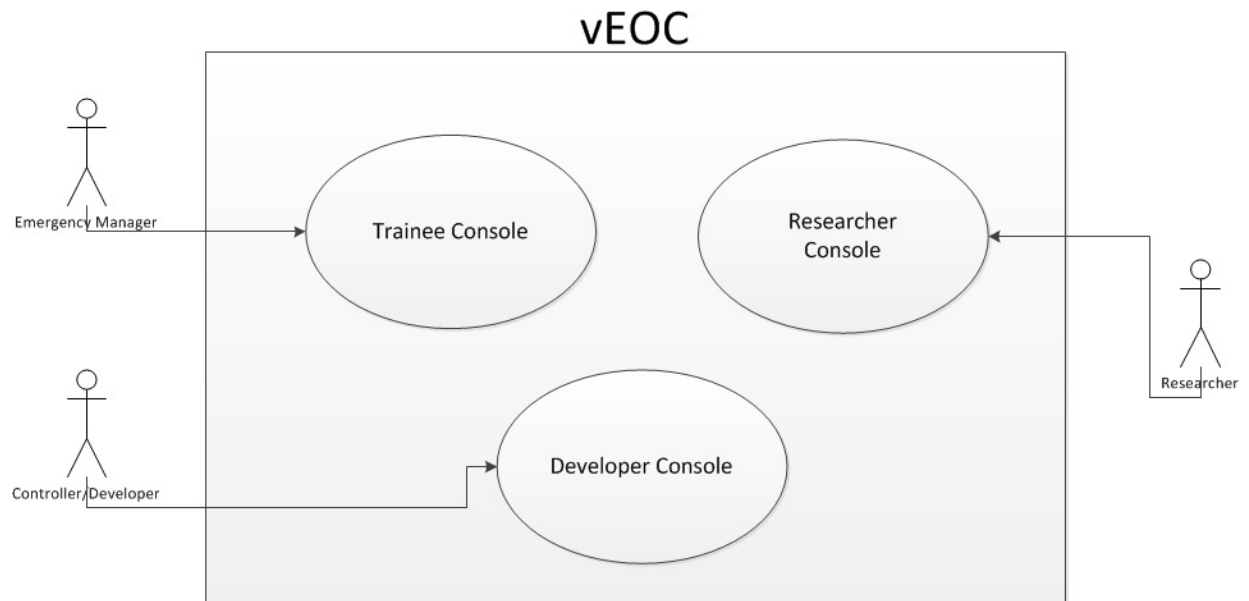


vEOC As Is

The vEOC is designed for 3 different types of users

- The trainee/emergency manager who uses the trainee console to train for emergency situations.
- The exercise developer/controller who develops scripts, and runs the exercises as the trainees train.
- The researcher who studies what the trainees did and how they perform during the exercises.



- **Incident Command** – vEOC users can currently refer to the incident action plans available through their console.

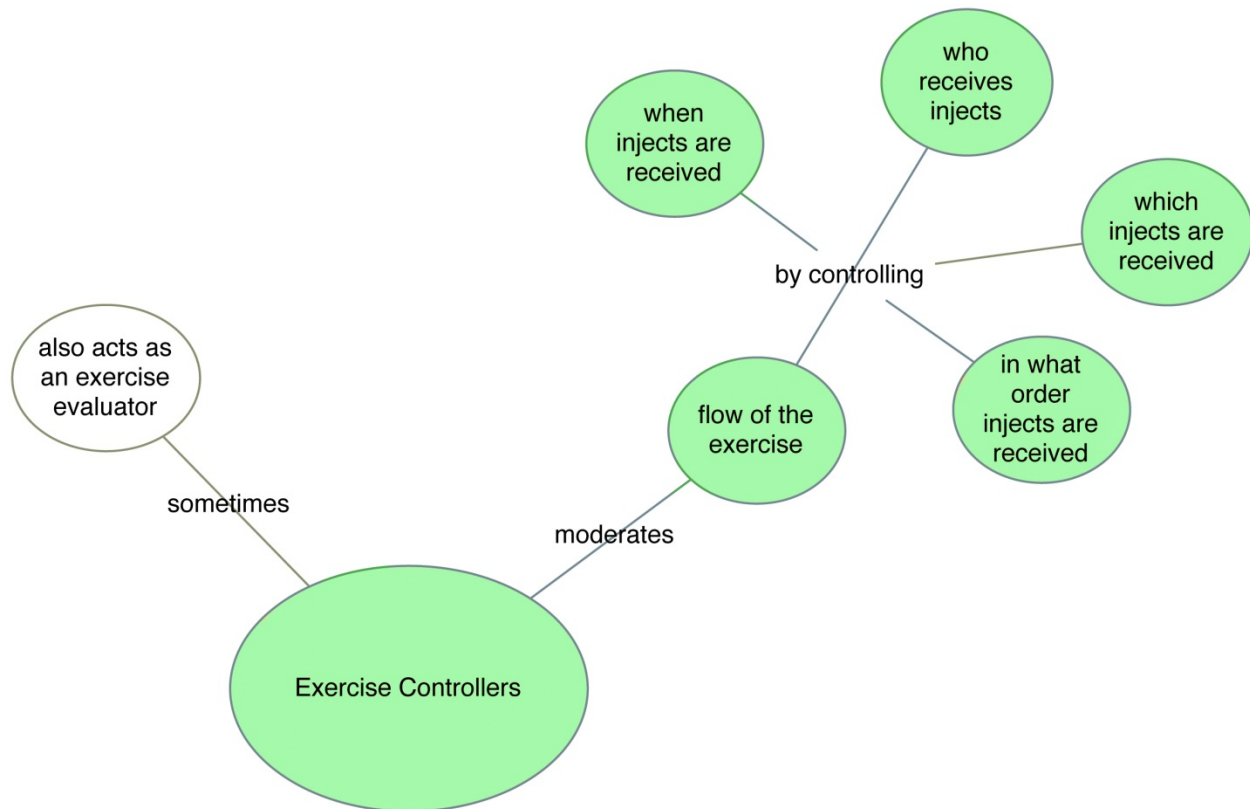
In Development

- **Information** – In addition to the already established ways of sharing information, we are looking for different ways to present different types information to the users, particularly through the dashboards. With the additional information we can make available, we intend to provide a way for the users to assess damages and mitigate risks more effectively.

Proposal for Future Versions

- **SOP** – It has been proposed to have Standard Operating Procedures (SOP) available through the vEOC. This requires us to collect a large number of files we currently do not have. This should definitely be considered for future versions of the vEOC.
- **Disaster Assistant** – For future versions of the vEOC we plan to have a disaster assistant available for the trainees. The disaster assistant would be a resource the trainees could consult with questions, and it would provide direct answers. This requires some AI capabilities and a considerable more amount of time to implement. Having the disaster assistant available could help us leverage the amount of information we provide to the trainees.

Exercise Controller



In the exercise developer console, exercise developers and controllers can find their functions. Exercise controllers are in charge of administering the exercises. Their responsibilities and duties are shown above in the circles. The green signifies the functions are implemented in the vEOC.

Implemented

- **Moderating the Flow of the Exercise** – On this console there is a link to the exercise controller window where you can load the script and begin the exercise. Once the exercise has begun, all players logged in are notified that the exercise will begin and the controller has the option to pause, stop, and fast forward the exercise.

Exercise Developer



Implemented

- **Make Scripts** - Exercise developers utilize the exercise developer console to create scripts. A script is essentially the exercise. Scripts consist of injects, which the vEOC currently has in its database. Developers may also create their own injects to add to the database. Injects are inserted on a timeline to send to one or all players during the exercise.
- **Print Reports** – Evaluators have access to a log of what each player did during the exercise, and when they did it. Currently the log can identify which pages the player

viewed, what they updated/deleted on those pages, who they contacted, and what was said. These reports are available to be pulled up for specific users.

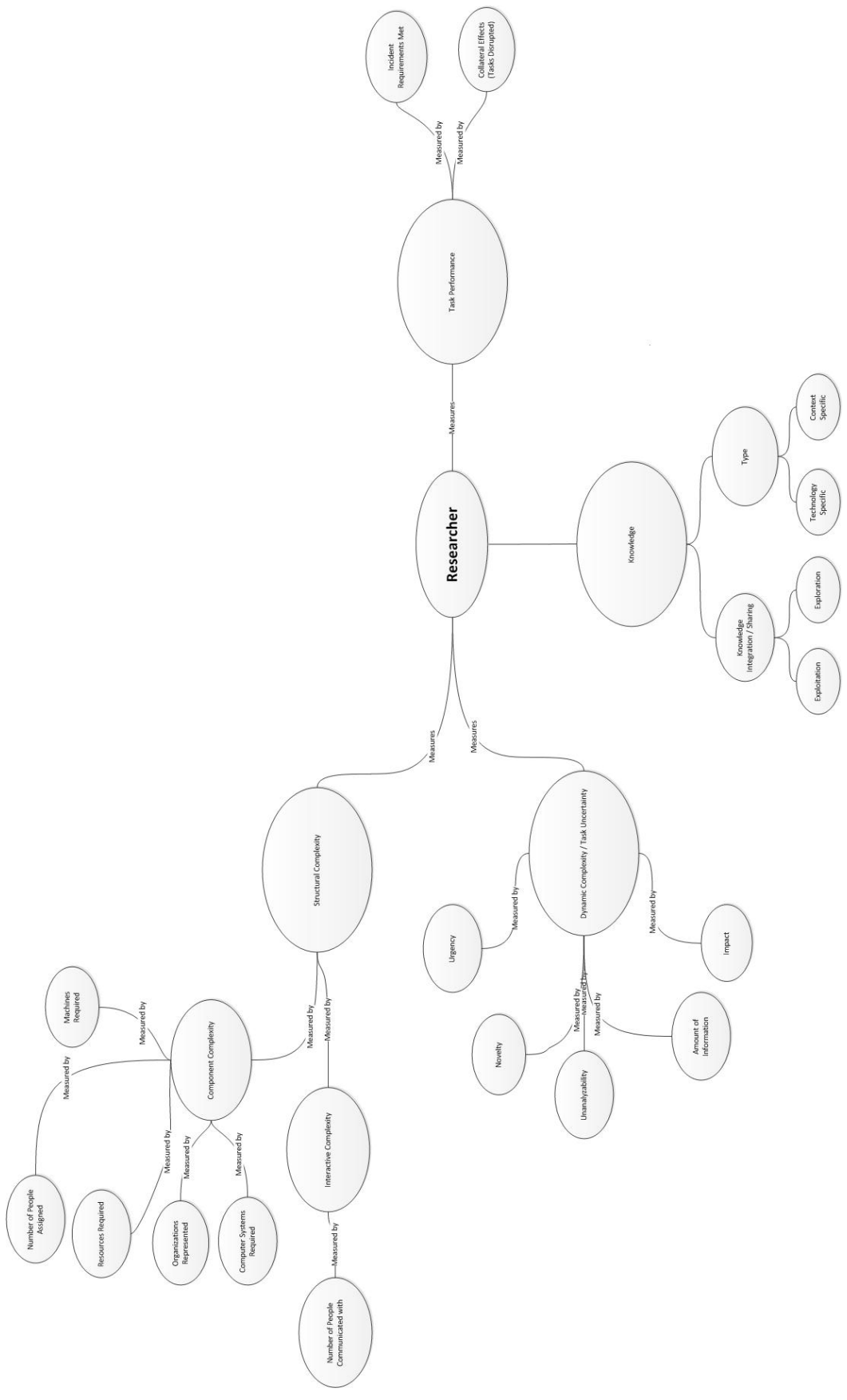
- **Determines Evaluation Metrics** – through this console the developers can assign objectives for the exercise and each inject. Exercise evaluators can use these objectives along with the reports of the exercise to determine how well the trainees performed during with each task.

In Development

- **Creating After Action Reports** – For the current version of the vEOC, we will be adding a template for the after-action reports that the evaluators could use while they are logged into the vEOC.

Proposal for Future Versions

- **Player Evaluations** – Currently an evaluator reads through the script and the logs and determines if the objectives set forth were met. For future versions we would like to have the vEOC itself determine if objectives were met during the exercise.



Researcher

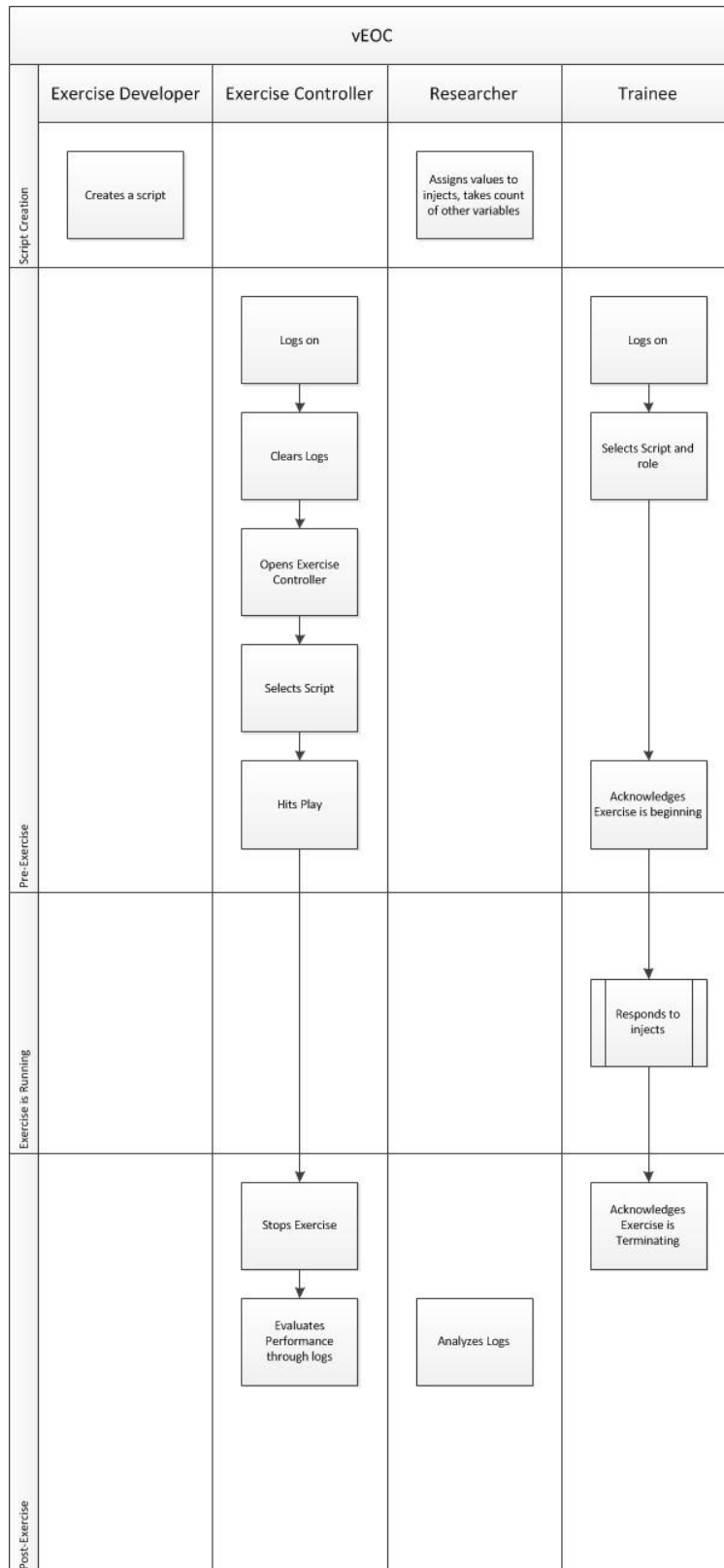
The researcher measures and compares various aspects of the exercise, available through player reports and exercise logs.

- **Task Performance**- is measured first and foremost by the evaluators. From the logs in the vEOC, it can be determined if the objectives were met.
- **Knowledge** – In the vEOC the researcher can observe how knowledge is shared. The logs can be consulted to see what was said by whom, and to whom. Page views are also logged so researchers can see what data the users are looking for.
- **Structural Complexity** – Consists of both the component and the interactive aspect. Both of these can be measured by researchers by taking into consideration how many other users are drawn into the task, and the resources required as it progresses.

In Development

- **Dynamic Complexity** – The measures of dynamic complexity relate to the task. Injects can be assigned values for urgency, impact, un-analyzability, and novelty by the researcher and exercise developer. The amount of information available can be controlled by status reports, maps, injects, and the scenario narratives
- **Export Logs to .csv files** – Currently in development is the capability of the logs and reports to be exported as csv files to be opened in programs like excel for further analyzing.

vEOC Flow



Responding to Injects

