San Francisco County Transportation Authority

SFCTA Clerk <clerk@sfcta.org>

Re: Public comment on SFCTA CAC item 9

Cyrus Hall <cyrusphall@gmail.com> Sat, Aug 30, 2025 at 2:27 AM
To: clerk@sfcta.org

Correcting a couple typoes:

depredations of sensors -> degredations of sensors

such bugs could be dangerous behavior -> could result in dangerous vehicle behavior

On Sat, Aug 30, 2025, 11:08 Cyrus Hall <cyrusphall@gmail.com> wrote:

> Clerk, CAC members,
>
> I read the SFCTA AV Permitting Framework with interest. While I largely support the approach, and unfortunately lack time while on vacation to analyze the details, I wanted to call out the lack of a framework for continuous validation.
>
> The concept of continuous validation in software systems recognizes that systems that behave a certain way today may not behave the same way tomorrow, either because of exogenous or endogenous changes. Exogenous changes include code and model changes, where "code" defines the logic of the system, and model define logical and semantic relationships reflected up on by the code. Other exogenous changes in the case of AVs can include cell network performance, or weather. Endogenous changes can include depredations of sensors or other control systems.
>
> In modern software environments, exogenous changes are "continuous," ie, happen multiple times a day. The process of continuous deployment includes concepts like iterative rollout to try and discover bugs before they widely impact system performance, but low interval bugs can easy escape such processes. In the case of AVs, such bugs could be dangerous behavior.
>
> In the face of continuous deployment, regulatory frameworks must also define continuous permitting. How do ensure that code and model changes are safe? Size of the change is not a good measure - the change of a single constant in the code or model could have outsized impact. And since it takes time to prove out a system, a continuous permitting system would need to be delicately balanced between safety and speed of change.
>
> I strongly believe any regulatory safety framework must dig deep into this difficult and unsolved process problem in computer science. The current document does not discuss this problem, and needs to go back for some more rounds of consideration, in coordination with staff or outside assistance that have familiarity with continous software deployment pipelines.
>
> Cheers,
> Cyrus Hall