Requirements for self-driving guidance software: *Current approaches are very dangerous*

David Gelperin, CTO
ClearSpecs Enterprises

Abstract

- Requirements for self-driving software are **being developed privately**
- Competing companies are developing their **own defective requirements**
- Some of these defects will **kill**

- Industry-consensus, open-source requirements would be much safer

I hope to raise your awareness of **extremely dangerous requirements practices** for potentially pervasive safety-critical software

Autonomous vehicles may have significant benefits

- May save many lives
- Many increase freedom of movement for physically challenged
- …

Autonomous vehicles are an admirable objective

Innocents will die

Group 0 – Wrong place, wrong time e.g., bridge collapse

Group 1 – System defects

Group 1a – Best efforts, but human errors

Group 1b – Not best efforts e.g., poor requirements

Current approach will **significantly increase Group 1b** due to **deeply flawed requirements strategies** for **extremely complex software**

Proprietary requirements will kill many unnecessarily

Semi-autonomous in the News

- Jan. 2016 - A Tesla autopilot car in China smashed into a street sweeping truck, killing the car's 23 year old driver.
- May 7th 2016 - The crash that killed a Tesla driver in Florida when his car struck a tractor-trailer may mark the world's first fatal accident in which a computer was at the wheel. The crash occurred when the truck turned left across the 2015 Model S Tesla's path and the car's autopilot failed to slow down.
- April 2017, Tesla Model S and X owners filed a class action lawsuit over Tesla's enhanced autopilot software. The lawsuit alleges that the company sold Model S and X vehicles with the autopilot program—costing an additional $5,000—knowing that the program didn't work and wasn't supported with appropriate safety features. The lawsuit affects Model S and X vehicles manufactured between October 2016 and March 2017: 47,000 vehicles in all. Plaintiffs allege in the complaint that they became "beta testers of half-baked software that renders Tesla vehicles dangerous," and that when autopilot is engaged, it is susceptible to "lurching, slamming on the brakes for no reason, and failing to slow or stop when approaching other vehicles."
- March 19th 2018 - Self-driving Uber car (Volvo) kills pedestrian in Arizona
- March 23rd 2018 – In California, the driver of the electric Tesla Model X died in a car crash after he collided with a median barrier and the car caught fire.

WHY TESLA'S AUTOPILOT CAN'T SEE A STOPPED FIRETRUCK

Jan. 22nd 2018 - A Tesla Model S slammed into the back of a stopped firetruck on the 405 freeway in Los Angeles County at 65 mph. No one was hurt.

Tesla manual: "Traffic-Aware Cruise Control cannot detect all objects and may not brake/decelerate for stationary vehicles, especially in situations when you are driving over 50 mph (80 km/h) and a vehicle you are following moves out of your driving path and a stationary vehicle or object is in front of you instead."

Volvo's semi-autonomous system, Pilot Assist, has the same shortcoming. Say the car in front of the Volvo changes lanes or turns off the road, leaving nothing between the Volvo and a stopped car. "Pilot Assist will ignore the stationary vehicle and instead accelerate to the stored speed."

The same is true for any car currently equipped with adaptive cruise control, or automated emergency braking. These systems are designed to ignore static obstacles, otherwise (with current sensor configurations), they couldn't work at all.

Bottom Line: The faster you go, **the more dangerous these controls**.

Proprietary requirements are a very bad idea

Proprietary requirements artifacts will be seriously flawed

- **Glossaries (should have many entries)**
- **Hazard analysis results (should be a very long list)**
- **Quality attributes e.g., safety, security, …**
- **Function sets**
  - **particularly hazard mitigation functions**
  - **e.g., all behavior must be monitored**
- **Design constraints**
  - **e.g., no single points of failure**
- **Implementation constraints**
  - **e.g., must have and monitor coding standards for clear and safe software**

Who decides what autonomous vehicles should do when their motor dies?

- **Will "dead motor" be detected?**
- **What should happen when motor dies and the vehicle is:**
  - **stopped?**
  - **moving, but can "safely stop"?**
  - **moving, but can't "safely stop"?**

Why are requirements likely to be flawed?

"As is well known to [some] software engineers (but not to the general public), **by far the largest class of problems** arises from **errors made in the eliciting, recording, and analysis of requirements**."
[2007 National Research Council report on **Software for Dependable Systems**]

- A competitive race mentality causes errors i.e., speed kills.
- Most US software engineers have never taken a requirements course. It is likely that most embedded software developers, some of whom are EEs and system engineers, have never taken a requirements course.
- Most embedded software developers know little about software safety.

Unintended Acceleration Again

'My gas pedal is stuck' BMW driver tells 911 as he barrels down interstate at nearly 100 mph
Feb 14, 2018

BMW spokesperson called the scenario "implausible"

"All BMW vehicles, .., employ an electronic accelerator pedal which uses **software logic to override the accelerator whenever the brake pedal is pressed while driving**. **This fail-safe software** means that if the vehicle detects that both pedals are depressed, the on-board electronics will reduce engine power so that the driver may stop safely.

No requirement for global monitoring of vehicle behavior

Conclusion

- Those developing AV guidance software are solving (almost) the same complex problem
- Proprietary requirements are concealed and dangerous
- Requirements validation (and verification) is concealed

Cooperative requirements are much safer

I hope you will spread these ideas

Raising Awareness – Directly & Indirectly

- manufacturers
- system/software safety engineers
- test engineers
- product liability insurers
- lawyers
- public interest groups
- press
- regulators

If you're NOT part of the solution, ...