A Case Study: Injecting Safety-Critical Thinking into Graduate Software Engineering Projects

Jane Cleland-Huang
University of Notre Dame
Workshop on Emerging Software Engineering Education, Indian Software Engineering Conference

Partially funded by US National Science Foundation Grant CCF:1319680
Summary

• Taught a 5 month, project-based, graduate level Software Engineering course in which teams of students developed non-trivial products using UAVs and/or various types of sensors.

• Our projects represented “light-weight” safety applications that might normally be built without thinking about safety implications.

• Exposed students to some basic practices for identifying and mitigating safety-related risks.

• Evolved into a more stable course in Software Engineering for UAVs.

Project Domains and Ideas

**Medical Fleet**
Fleets of drones are coordinated to deliver medical supplies in a natural catastrophe and to use aerial reconnaissance for tracking.

**E-Health**
A person recovering from a medical issue such as a heart attack, is safely monitored during his/her rehabilitation.

**Environment**
Crowd-sourced pollution detection in case of a chemical spill. (Also for environmental pollution).

**Mission-Control**
Monitor position & health of a rescue team. Provide instructions and information via visor.

Are these projects really safety-critical?
Example Project: Medical Fleet

Central controller, dispatch algorithms, drone tracker.

Drone supply base  Request for help
Project: Rehabilitation

Doctor or therapist creates a safe exercise plan for the patient.

Exercise sessions are monitored and data is tracked via the mobile to the server. You can also include outside exercise and GPS tracking.
Project: Rescue Mission

Health of team members continually monitored using health-sensors.

Centralized control tracks mission and plans route.

Group dynamics, route plan, and instructions displayed on glasses.
Web-server captures information about pollutants (aka dangerous chemicals) and displays them on a map.

Crowd-source the collection process.

Ordinary users can download an App that warns them if they approach unsafe areas.
Drones used to perform aerial survey. Images processed into map.

Central dispatcher tags interesting sightings. Data is streamed from copter to central server. Tracking is performed in real-time.
Our very own “Q”

We hired an Undergraduate two months in advance of the start of the course to test all the interfaces to physical devices and created and collected instructions for how to send commands and/or extract data from the devices.

We distributed physical devices and hardware to each team in Week 3.

At that time, met with each team to explain how to control and/or to retrieve data from your device.

He has also set-up instructions online at http://sarec.nd.edu/studio
Process/Artifacts *(Steps 1-3)*

**Step 1:** Start creating a glossary of terms. Add to it incrementally *(Doors Next)*

**Step 2:** Define clear system goals *(Doors Next)*

**Step 3:** Create 3-5 high-level use cases. *(Doors Next)*

**Outcome:** You know what your product should do.
**Process/Artifacts (Steps 4-)**

**Step 4**: Identify technical risks and plan/start work on architectural spikes.
- Interface with physical devices
- Mobile iOS/Android
- Web-Service
- Project environment
- Tools
- Team collaboration mechanisms
- Other project specific??

**Step 5**: Start defining requirements or user stories (agile/upfront?) (Doors Next)

**Step 6**: Start defining performance/quality related requirements.

**Step 7**: Safety Analysis. Construct a FMECA (Doors Next)

<table>
<thead>
<tr>
<th>Item</th>
<th>Potential Failure Mode</th>
<th>Potential Cause of Failure</th>
<th>Current Prevention Controls</th>
<th>Current Detection Controls</th>
<th>Recommended Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brake System</td>
<td>Mechanical linkage</td>
<td>Designed per material</td>
<td>Designed per material</td>
<td>Change material to stainless steel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>brake</td>
<td>standard</td>
<td>standard</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Master cylinder failure</td>
<td>Designed per material</td>
<td>Designed per material</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>standard</td>
<td>standard</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Loss of hydraulic fluid</td>
<td>Designed per material</td>
<td>Designed per material</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>standard</td>
<td>standard</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Outcome**: You know more about your product.
**Process/Artifacts (Steps 4-)**

**Step 8:** Security Analysis

Security Cards

*(Doors Next)*

**Step 9:** Architectural Design. Consider multiple solutions. Use UML.

**Step 10:** UX Design

Mockup your main screens. Determine task flows.

**Outcome:** You know that your product is viable.
Process/Artifacts \textit{(Steps 4-)}

**Step 11**: Code, Test (Start this early – for architectural spikes)

**Steps 1-12**: Create trace links as-you-go.

**Step 12**: Build an initial safety-assurance case.

\textbf{Outcome}: You deliver a fully functioning architectural spike.
Managing the Artifacts

This is an interactive Traceability Information Model.

Click on any node to modify/view data.

Commit your changes via Github.
What did Students Think?

Did you like the Safety-Critical nature of the Project?

- Yes: 18
- No: 4
- No Preference: 4

Did you like working with physical sensors/UAVs?

- Yes: 18
- No: 2
- No Preference: 6

Yes □ No □ No Preference
What Aspects were helpful for understanding safety related issue?

- Readings on Safety Critical Systems: Mean Ranking 2.3
- Safety Case Tool and tutorials: Mean Ranking 1.8
- Interactive Team meetings: Mean Ranking 1.6
- External Safety-Speaker: Mean Ranking 1.5
- FMECA's and Tutorials: Mean Ranking 1.5
- External Panel (Final Presentations): Mean Ranking 1.3
What did students say?

The focus on CPS:

1. Added a “real-world” aspect to the course,
2. Forced us to “consider the user of the product”,
3. “gave insight into some of the more bureaucratic processes that software development has to go through”
4. offered “a first-hand experience in developing safety-critical systems” and…
5. Required us to “think about what could go wrong and do risk assessments.”
What’s next?

• Focus on one type of CPS (e.g. UAVs)

• Provide “more” getting started help:
  • Programming UAVs
  • Analyzing hazards and writing mitigating requirements.

• Building Safety Assurance Cases
Take 2!  Software Engineering for UAVs
Our Infrastructure

http://Dronology.info
Not all smooth sailing

Bent prop drone

Rescued Drone

Trajectory challenged, upside down drone

Broken leg drone

Drowned and missing drone
Challenging assignments

Scoped architectural design problems
Requirements, Code, Testing, Safety Analysis
A Case Study: Injecting Safety-Critical Thinking into Graduate Software Engineering Projects

Jane Cleland-Huang
University of Notre Dame
Workshop on Emerging Software Engineering Education, Indian Software Engineering Conference

Partially funded by US National Science Foundation Grant CCF:1319680