Topics in Software Safety

[Reading assignment: Just these slides, nothing in the book]
“Even though a scientific explanation may appear to be a model of rational order, we should not infer from that order that the genesis of the explanation was itself orderly. Science is only orderly after the fact; in process, and especially at the advancing edge of some field, it is chaotic and fiercely controversial.”  

- William Ruckelshaus  

1st head of the EPA, subsequently acting director of the FBI and Deputy Attorney General of the US.
Software and safety-critical systems

• We are now using software in systems that we call safety-critical. These are systems that, if they fail, will have very serious consequences:
  – nuclear reactor monitoring
  – flight control systems
  – software controllers on X-ray machines
Software and safety-critical systems (Cont’d)

- So far, we have been fairly careful about introducing software into safety-critical systems:
  - extensive testing, code reviews, formal proofs of correctness
  - use of good engineering principles, KISS, limit frills
- So far, there have been relatively few failures of safety-critical software systems.
But …

• There is great temptation, on both technological and economic grounds, to go rushing in and move a lot more safety-critical system features into software systems.

• This is NOT the first time in history that we have been tempted by technology in this way.

• “Those who cannot remember the past are condemned to repeat it.”
  - Santayana (1863-1952)
A brief history of steam engines

- Heron of Alexandria, in 60AD experimented with steam power.
- 16th and 17th century “exploded” with interest in steam power.
- Thomas Savery (1650-1715) produced the first workable steam engine.
History ...

• Newcomen in 1700 designed a steam-driven cylinder and piston engine that achieved widespread use.

• In 1786, James Watt (1736 -1819) greatly improved the Newcomen engine.
  – Watt worked at University of Glasgow.
  – He had interactions with professors, good knowledge of heat.
History …

• Meanwhile, in the north of England (mainly), the Industrial Revolution was creating an amazing demand for cheap and efficient power sources.

• Watt and Matthew Boulton (a manufacturer) came up with a practical, winning design that transformed heavy industry.

• The Boulton and Watt machines
History ...

• Fast forward to 1800: Watt’s patent expires.
  – Now anyone is free to make high-pressure steam engines (HPSEs)!

• Two designs appear (one US, one UK)
  – No separate condenser; instead, steam is used to push pistons directly.
History ...

• First widespread use of HPSEs is steamboats.
• It’s highly successful!
  – Cheap, efficient.
  – Makes transportation more affordable to the masses.
  – Steamboat companies make money too; helps the growing economy.
History ...

• BOOM!
• Oh yeah, HPSEs tend to explode too.
• Steamboat passengers and crew blown up, scalded to death, drowned, impaled by hot iron, ...
• HPSEs also used in manufacturing industry. Guess what happens?
So what’s the problem?

• Well, HPS is dangerous stuff, but also:
  – low standards of workmanship
  – use of cheap, inferior materials
  – poorly trained workers
  – poorly trained operators
  – bad quality control
Why?

• There was an awful lot of money to be made.
• No real economic advantage to being responsible.
• Companies could just turn out more HPSEs and pay off whoever they had to when an HPSE exploded.
• So what's to be done in a situation like this?
History ...

• In the US, there were calls for standardization of training and professionalism, suggestion for a government academy of steam engineers.
• Back in the UK, Watt and Boulton tried to raise the alarm; they succeeded in slowing the adoption of HPS technology.
Boiler technology

• The technical Achilles’ heel was the boiler, which was apt to explode.
  – Boiler technology lagged behind the rest of steam engine technology.
  – Not cost-efficient to consider boiler improvements.
  – Little understanding of underlying scientific principles.
  – While boilers had been around for eons, they were only now being used in such stressful situations.
Progress ...

• What was needed was R&D into issues such as high stress, corrosion, decay, materials, construction.

• Public pressure forced some changes. Hence, the addition of two new safety features:
  – A safety valve to reduce steam pressure when it reached “dangerous” levels.
  – Fusible lead plugs that would melt when the temperature in boiler got too high.
Result?

- **BOOM!**
- The # of boiler explosions continued to increase.
- Why?
  - Engineers still didn’t really understand the underlying problems of high pressure steam and boilers. That took quite a bit longer.
Why (Cont’d)

• Design engineers didn’t understand how their systems would be used:
  – installation environment
  – operator training, ignorance
  – owner ignorance, greed
  – over-riding of safety features
Who was usually blamed?

- operators ("pilot error") usually
- owners sometimes
- ... but never the design engineers.
Enter the government!

• The steam engine was considered an icon of a forward thinking, prosperous society.
• “Too much is at stake.”
• “The private sector will regulate itself.”
• “The market will self-correct. Bad corporate citizens will be punished by the consumer.”
• Sound familiar?
So we get more HPSEs

• BOOM!

• In 1817, UK parliament decides to investigate; forms a Select Committee to investigate dangers of HPS.

• The Committee recommended, among other things, frequent boiler inspections.
No one pays attention to the results

• Soon after, the city council of Philadelphia tries to raise an alarm.
• The matter is referred to the state legislature, where it dies.
Time marches on ...

• BOOM!

• Between 1816 and 1848 in the US:
  – 233 steamboat explosions
  – 2562 human fatalities
  – 2097 human injuries
  – $3,000,000 property loss
Research ...

• Back in Philadelphia, the Franklin Institute begins a six year investigation on boiler explosions. The US government also kicks in some money.
  – This is the first US government grant for technology research
Research results ...

• The result is a series of reports that:
  – Expose common errors and popular myths about steam engines and boilers.
  – Set out guidelines for design and construction.
  – Recommend that US congress enact regulatory legislation, especially with regard to engineer training and practice.
Also …

• Public pressure in US and UK force laws requiring compensation to victim’s families.
• BOOM!
• Explosions continue!
• Public pressure increases again.
• Newspaper editorials and popular literature reflect growing frustration.
Legislation

• Finally, in 1852, US congress passes a law to require certain changes in steamboat boilers.
• This was the first successful US law regulating product of private enterprise.
• Steamboat boiler explosions start to decline!
• ... but unsafe HPSEs are still being used in locomotives and heavy industry.
Tougher standards

- Later, UK parliament passes very tough standards, which are enforced.
- In 1905, the number of deaths due to HPSE explosions are:
  - 14 United Kingdom
  - 383 United States
- Eventually, US follows suit and introduces tough standards as well.
“Exploding software?”

- We are now in the computer age
- What are the parallels between HPSEs and safety-critical software systems?
Analogies

• Boiler technology lagged behind improvement in steam engines themselves.
• So, too, software engineering lags behind hardware (electrical) engineering.
What to do?

• Use time-tested, good engineering principles:
  – KISS, essential services, testing & verification, double & triple checking, safety engineering principles

• Learn to love computers a little less. Our mistrust is fading and this is a bad thing.
  – Therac-25 radiation therapy machine

• Being careful need not stop progress, but we should consider the issues in detail.
SE foundations

• There was little scientific understanding of the causes of boiler explosions.
• Similarly, ours is a young discipline and we’re still working on the foundations.
  – What’s a good design?
  – high-level abstractions of software components
  – safety-critical systems
  – role of formalisms and formal methods
  – verification and validation
  – system evolution
Problems

• We aren’t sharing as much information as we should (partly due to corporate paranoia), and there isn't that much careful, analytical data anyway.
• Info-tech is a fast-paced, fad-happy, innovation-driven, big money game.
• There has been little time or money for careful reflection, evaluation, and condensation.
Working on engineering foundations

• No one denies that innovation and invention are vital, but we also need to work on the engineering foundations too:
  – criteria for evaluation
  – means of comparison
  – theoretical limits and capabilities
  – means of production
  – underlying rules, principles, and structure

• We need mathematical models and careful experimentation (real-world validation)!
Questioning new methods

• “Formal methods are math. Math is good. Therefore, formal methods will improve software quality.”

• It is not clear that this is true!
  – What kinds of FM?
  – Training of practitioners?
  – Political issues? Costs? Scale?
  – Tool maturity and appropriateness?
  – Are resulting systems better? safer? smaller? bigger? more understandable? more opaque?
Understanding

• The safety features designed for the boilers did not work as well as predicted because they were not based on scientific understanding of the causes of accidents.

• Something that sounds good isn’t necessarily a good idea. You need to develop a deep understanding.
A good idea in one field is not necessarily good in another field

• For example, consider N-modular hardware redundancy:
  – Use $N$ identical hardware components in the same role. If they always agree, fine. If not, take a vote.
  – This is a highly-trusted engineering design principle for safety-critical hardware systems.
A software analogue ...

• The software analogue is called N-version programming (NVP):
  – Have N teams each write a version of the required program independently given the same requirements.
  – Run all N programs; when results differ, take a vote.
NVP under scrutiny

• What are the potential problems with NVP?
  – Software failures are not like hardware failures. All software failures are design failures, not material failures.
  – Often, programmers make the same kinds of mistakes, misinterpretations, and have similar biases.
  – Requirements are often misleading, wrong, vague, etc.
  – What if only one of the N teams actually has the correct interpretation!
Recovery blocks

Try algorithm 1

Algorithm 1 → Test for success → Acceptance test

Acceptance test fails → Retry

Retest → Algorithm 2

Retry → Algorithm 3

Continue execution if acceptance test succeeds
Signal exception if all algorithms fail

Recovery blocks
Recovery blocks

• Force a different algorithm to be used for each version so they reduce the probability of common errors

• However, the design of the acceptance test is difficult as it must be independent of the computation used

• There are problems with this approach for real-time systems because of the sequential operation of the redundant versions
Watch out for “wishful labeling”

- software diversity, expert systems, AI, software engineering
- Also watch out for “proof by definition”:
  - fault tolerant = uses redundancy
  - safe system = uses monitors & shutdown routines
“Wishful labeling”

• People tend to confuse an ideal with its implementation
  – *E.g.*, All you need is monitoring and a shutdown routine to have a safe system.

• Need a much greater understanding of the *human* element:
  – cognition, politics, social factors, training, ...
Workmanship standards

• The early steam engines had low standards of workmanship, and engineers lacked proper training and skills.

• There were more jobs for highly-trained and experienced technologists than there were suitable people to fill them.

• What do you think happened?
Safety engineering

• There exists a wealth of knowledge and experience outside the realm of software development/engineering.

• Safety engineering defines safety in terms of *hazards*:
  – Attack problem of system safety by reducing or controlling hazards.
Basic approaches to safety engineering

• **Avoidance**: Stop hazards from occurring, or minimize their occurrence.
  – *E.g.*, If fire is a concern, use non-flammable materials and minimize chance of sparks.

• Disadvantages:
  – cost
  – performance
Basic approaches to safety engineering (Cont’d)

• **Recovery**: Control hazards if/when they do occur.
  – *E.g.*, sprinklers, fire doors, smoke detectors

• Advantages:
  – cost, can be added after-the-fact

• Disadvantages:
  – often less safe
  – cost
  – performance
Safety engineering (Cont’d)

• In practice, a combination of the two is used.
• Each system is different and requires careful analysis of:
  – risk
  – design
  – cost
  – performance
High-pressure steam engines and computer software

“As Edison argued with respect to electricity, increased government regulation of our technology may not be to anyone’s benefit; but it is inevitable unless we, as the technology’s developers and users, take the steps necessary to ensure safety in the devices that are constructed and technical competence in those that construct them.”
You now know …

• … Historical analogies between steam engine reliability and software reliability
• … N-version programming
• … safety critical software
• … safety engineering