



Movie Physics and the Software Industry

We go to movies for entertainment, and of course, most films are fiction, not real; they are make-believe. However, when a scene blatantly disregards the laws of physics it is enough to make an engineer retch. Yet Hollywood's bad physics can provide insight into our own industry's faux pas. For example, consider the following:

Flashing Bullets

An action movie's climax is the gunfight complete with ricocheting bullets that emit bright flashes of light upon impact with anything on the set. While the flashing bullets are exciting, they are highly unlikely. The majority of bullets are made of copper-clad lead that does not spark when struck, even by steel objects. Professionals employ the same copper-lead-alloy in flammable work areas to prevent sparks.

In software engineering, flashy product features with marginal value are the flashing bullets of the industry. We get so excited about new ostentatious features that we lose sight of our original product intent. Focus on the target, not flamboyant red herrings.

Endless Ammunition

Where do all those bullets come from? A Mac 10 expends a thirty-round magazine in a mere 1.8 seconds of sustained fire, an Uzi in 3.0 seconds. Nevertheless, a movie star can sustain fire during a five-minute scene without a single reload. If you are thinking bigger magazines, think again. To sustain a three-minute burst of fire, a Mac 10 gunman would have to carry 100 pounds of lead and 3,000 cartridges cases – not very agile or likely.

In software engineering, requirements are endless. Baselines are hollow, freezes futile. Customer needs pile up and weigh a project down like 100 pounds of lead. Hit your target before you aim at the next.

Visible Laser Beams

From security systems to light-sabers, filmmakers treat us to conveniently visible laser beams. One problem, we can only see laser light when it hits a repercussive object, revealing a dot, not a beam. Any laser pointer user can confirm this fact. True, smoke, dust, or mist can reflect the laser light and create an apparent beam, but that is short lived unless you are a heavy smoker in which case you are short lived.

Software requirements, like laser beams, are only visible when they hit home with the developer. They are not guiding beams but elusive rays of light that require reverberation and discernment. In this case, it is okay to blow smoke or sprinkle dust to better perceive and focus in on customer requirements.

Pushy Buckshot

Now move from bullets to the famous sawed-off shotgun, blasting thugs violently backwards into the nearest plate-glass window (see cut-free glass). That seems real, right? No, using conservation of momentum, you find the velocity of the thug is proportional to the ratio of the buckshot's mass to the thug's mass, which is tiny. The net effect for an average person and standard buckshot would be 0.4 miles per hour. The average human walks at 4 miles per hour so the only direction the thug is going is down, due to another force called gravity.

Software managers are looking for silver buckshot that will change the momentum of a project. Why do we continually believe that the tiny thrust of a new technology, process, or consultant will overcome the impetus of the project? Avoid sawed-off initiatives.

Flaming Cars

Why are movie cars always bursting into flames the instant they collide with anything? If you watch closely, some explode before they hit anything as if the gas tank gets panicky and detonates at the mere thought of collision. A car explosion requires a tank rupture that spews a fine mist of gasoline vapor-air mixture of 0.8 percent to 6 percent and a source of ignition, typically found at the other end of the car – possible but improbable.

Software project leads deal with flaming gurus whom they feel they cannot afford to lose. They tiptoe around in fear that their fumes will ignite and explode the project. Coddling prima donnas is more harmful than helpful to a project. Their ignition probability is lower than perceived. They may burn but it is improbable that your project will detonate. They are replaceable. Stick with stable, reliable, and efficient project fuel.

Cut-free Glass

A shattered window contains thousands of

incredibly sharp dagger-like edges. Little force is required for one of these daggers to lacerate flesh. However, thespians frequently crash through plate glass without a scratch. There are individuals who have accidentally fallen through windows without sustaining serious injuries. There are people who have survived shark attacks. However, in both cases the odds are meager.

The software industry has programmers that believe they can code their way through a project a week before delivery. Even if you make the deadline without a scratch, you leave a product full of lethal shards for your customers to navigate. Fortunately, most customers have developed thick calluses from other software products. Do not procrastinate – respect your client.

Space Explosions

I am sorry Trekkies and Jedi, but there are two problems with explosions in outer space: With no air to transmit sound, outer-space explosions are virtually silent, save the wisp of expanding gas that passes by your X-Wing. In addition, with no gravity or air to slow them down, fragments from the explosion would travel outward until they hit something with the same kinetic energy they had during the initial blast. The first exploding Tie-Fighter would indiscriminately wipe out most of the fighters from both sides of the battle. The lucky few left would have second thoughts about setting off another shrapnel shower.

Software design errors are like space explosions. They are seldom heard and hard to spot. You do not see them coming but when they hit, they hit with more energy than they did when initiated. Once found, you spend the rest of the project dodging their debris. Spend your time and attention in the design phase. Knock out design flaws while they are small; do not wait until they are death stars.

Luckily, when we leave the theater, we realize what we saw was just a movie, fantasy, or make-believe. Unfortunately, that is not the case with software projects. These are real problems with real consequences. So sit down; grab a soda, some popcorn, and Red vines; and chew on that for a while. Enjoy your project.

— Gary Petersen
Shim Enterprise, Inc.