

“Make the Choice”



The Terror of Being a Software Architect  
an essay

by Charles Wolfgang Keane Tuomi



I worked as a software engineer and then a software architect, for more than twenty years. An English major in college, I taught myself programming from reading books and by annoying engineers at the company I worked for (originally as an editor) with constant questions about how the software worked underneath the hood. I did for this three reasons. It was interesting intellectually, like solving puzzles. I felt pressure to make more money as a new husband to compete with friends' situations, and software developers made a lot more money than editors did. And the editorial team I ended up managing needed tools the engineering teams did not have time or resources to build, so I built them myself.

As years went on I got quite good at it. I ended up architecting key aspects of a system that handles hundreds of millions of requests a day from users all over the world. This scared the hell out of me in my early years in the profession, due to the lack of confidence bred in me by my upbringing and the fact that, unlike most of the architects and engineers with whom I worked, I did not have an educational background in engineering. So I was constantly wondering, what have I missed? What don't I know that other people in the room do?

My short story "you then asia" is, on one level, a description of the terror I felt upon monitoring the usage of the first somewhat major system I had a significant hand in building: personalization functionality that allowed users to create and manage user accounts, and add items to something very much like a shopping cart but a little bit fancier, with a few more options than a typical shopping cart.

I sat there in the front of a computer screen the day we released and enabled it, and I watched the rows in the database that stored the account information, occasionally refreshing the screen to see the changes over time. "Ooh. There's our first user!" Refresh. Now there are several more. Refresh again. Now the whole screen is filled up with rows of users.

"Oh, shit. What did I miss? What problems are people going to report?"

"The infants will not shut up. Their crying? Incessant. Their shrieking? Inescapable."

Etc.

"For some reason" I was the only doing this watching of a brand new application come to life (like a baby) among the team members who built the thing, the only one who could "hear" the users banging away at the database like knocking on a cabin, like pounding on a chest during CPR. My co-workers were laissez faire about it. This was in part because some were more experienced than I was. Some of it was because others didn't give a shit, at least not nearly to the extent that I did.

I was scared, though, I admit it. Which may seem silly. I mean, worst case, maybe I made a terrible mistake and the application would have to be rolled back and if it was stupid enough I guess I might have gotten fired. But failure to me of any kind was terrifying. I had to exceed expectations, always, to validate my presence in the world.

What if I was to blame? What if I had mistakenly written a snippet of code that ended up executing an infinite loop that trapped users forever waiting as their browser just hung there, spinning and spinning and spinning...

I could just picture the humiliating meeting to discuss the defect.



“Mad Genius Behind Immortality Epidemic Identified”

All this worrying was for naught in this case. And in pretty much every other one in my software career. I jumped around a lot from team to team, as I bore easily and liked to work on different parts of the system, which over time gave me a holistic perspective that came in very useful. A VP told me once that I was the only one in the company who, when asked about where a problem was coming from, no matter what part of the system was involved, I could not only describe it but bring up the specific source code. But I would sometimes revisit old code that I wrote in other “lives” to see how it held up over time. I was gratified when I did this to find that, aside from changes to add or revise functionality, my code rarely needed to be revised due to any kind of serious defect. Because I am super careful, and I catastrophize, and when writing code I have even the most unlikely edge cases in the back of my head, trying to account for everything, including how readable and comprehensible the code is for future engineers who need to work with it.

Unfortunately this was due to a deep sense of unworthiness. I did not have the educational background. I was “less than” in general. “Less than” everyone. But if I worked and worked and worked, people would let me stay. They would even praise me, like Rudolph the feak with his useful red nose.

It’s funny to watch these things bleed into a feature film, or into a web site that I am making right now, for which I am writing this essay. Still trying to justify my existence, screaming myself hoarse in a storm, trying desperately, in utter futility, to locate someone, anyone, who actually cares as much as I do about things.

If you watch Wen in *Knock at the Cabin* (which is infected with “you then asia” via the fact that *The Cabin at the End of the World* is contaminated by it) you can see her hesitant, anxious curiosity as the first user account appears in the database. Wen is neW to this level of software development.



Then oh, shit, a lot more. Here they come. Actual users!





Here Wen retreats into her mental space in panic.

“Wen will my mystery mistake, whatever it is, the hidden defect in my code, be uncovered?!”

“What did I miss? What should I be doing? There are so many people using this thing!”

One makes so many choices as an engineer and architect. You design database schemas to hold the data. Pick a type of database management system to use. You assign responsibilities to subsystems, which involves balancing sound engineering principles with practical concerns, including, sometimes, political ones (like, “the manager of team X is being a recalcitrant asshole and protecting her turf, so our team needs to build this functionality ourself even though it would be a better architecture if the system that her team owns did its job”).

While writing code, you create abstractions to represent the real world the software is designed to represent. Which is a bit like creating characters in a story.

Speaking of which.

#

Here is how I architected Heaven.

In a previous document in this series, I described the fact that “you then asia” is in part inspired by the death of my uncle Dick. I was at his bedside when he passed away, just the two of us. He had lost his wife only a few months earlier, and people who were there when she passed told me that he said to her then that he would be with her soon.

I understood what he meant. He wanted to leave this world and spend eternity with her.

I also describe elsewhere how I used Zeno’s Paradox to impose immortality on human beings in my story. To summarize it again, Zeno’s Paradox questions how motion can be possible if to get from any Point A to Point B, one first has to get halfway to that Point B, and to get to that point, one first has to get halfway to that point, and so on, a process of slicing distance in half which seems like it could go on forever and make actual movement impossible.

[Here is an article that describes a “solution” to Zeno’s Paradox.](#) You can decide if it does. I have mixed opinions on this.<sup>1</sup> But the essence of the classic solution is mathematical, and uses the mathematical truth that “the sum of an infinite number of intervals that are (each) half as long as the previous interval adds up to a finite number.”

---

<sup>1</sup> For instance, here is an article that kind of disagrees:

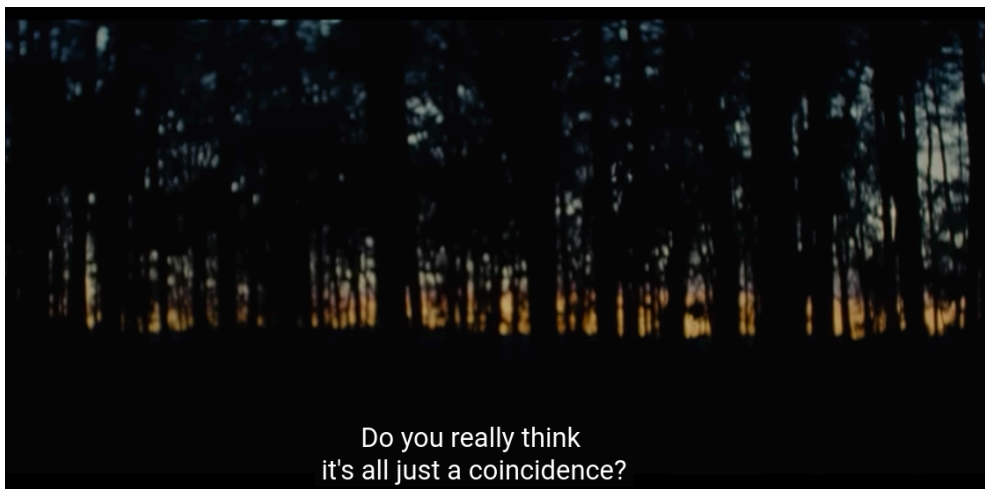
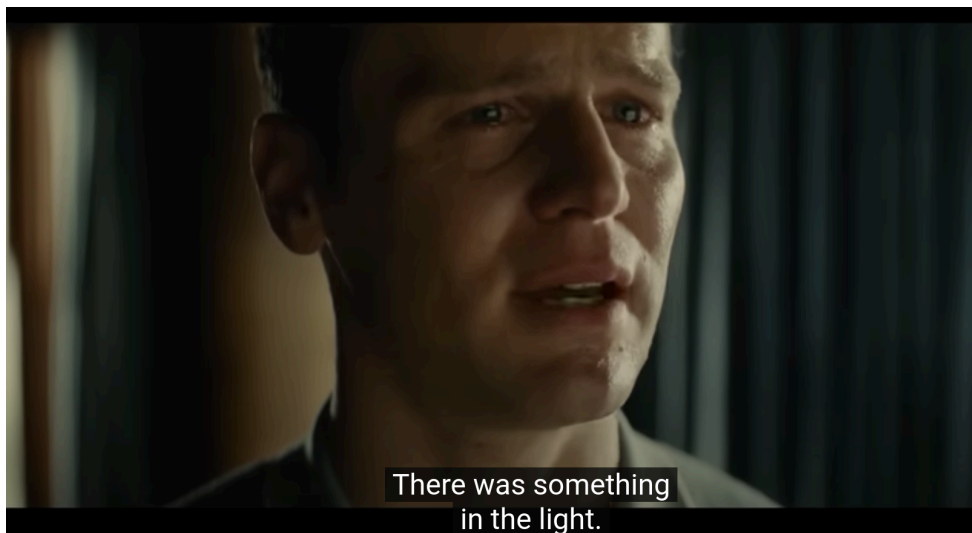
<https://www.forbes.com/sites/startswithabang/2020/05/05/this-is-how-physics-not-math-finally-resolves-zenos-famous-paradox/>. Which I also question as a complete solution. The Interweb is so confuscating sometimes.

This mathematical and therefore abstract description assumes that motion involves an infinite number of intervals. Which may or may not be true, I don't know. But it does reference time, which as we know from Einstein, is relative to a point of reference. Immortality in "you then asia" can be thought of as something like Einsteinian "time dilation," which specifies that time passes differently for different observers.



It's all relative, baby.

If motion does in fact work like the mathematical solution to Zeno's Paradox, it seems to me that someone watching someone else move from one point to another might very well be bombarded, by light, with an infinite number of images, pouring through their eyes, endlessly, in a finite amount of time.



Just someone opening a door. Or walking toward you. Or a smile forming on a lover's face while you dance with one another.

An *infinite* number of intervals.

An *infinite* number of moments, taking place in a *finite* amount of time.

Is there a better description of a miracle?





Actually, yes. Actress Oona Chaplin's *impossibly perfect* face.  
But this Zeno stuff may be the next best thing.  
Seriously, though. Blow this picture<sup>2</sup> up and stare at it. It's unbelievable.  
Text me and ask me if I have a proof for the existence of God,  
and I will simply send you a link to this photo. Case closed. Mona Lisa in real life.  
Have it digitally analyzed.  
If you can identify *one single molecule* that  
is not *exactly where it should be*, just one molecule, one defiant atom,  
and I will sign over rights to everything I have ever written to you.  
Until then, Oona...



Love,  
"Charlie"

---

<sup>2</sup> Oona Chaplin at International Week at Cine de Valladolid, 2016, Rubin Olmedo  
<https://creativecommons.org/licenses/by/2.0/>

Anyway, my uncle, lying there in a coma, his mind still functioning, had an *eternity* with his wife inside of his mind. *Every single time* they danced together or kissed one another was a photo album that could be flipped through *forever*. Just one occasion contained *an infinite number of intervals*. And not only her, but also his children, two boys and a girl, my cousins, who I lived with and called my siblings. He can see them arrive in their new home for the first time, hug them for the first time, find them and then lose them and find them again, forever.



And I took a snapshot of Dick there, before he passed, and I wrote a short story about him.

And now you may have read that story. And perhaps you have read a novel that was partly influenced by that story. And maybe you have watched a movie that was based on that novel. And because of the way story DNA works, my uncle is there, in that moment, all over the place, in all of those works. And each version of my uncle in the many places you can find him has those infinite number of moments with his family inside of his mind, an eternity to spend with her. And other writers and artists have read *The Cabin at the End of the World*, or watched *Knock at the Cabin*, or in a much smaller number of cases, perhaps even read “you then asia,” and been impacted by those works, and created new works that contain them, and therefore passed him, them, on unknowingly as parts of other stories. This process will go on, and on and on. Passed on, yes, literally *passed on*, my uncle and my aunt, who will be dancing forever, or kissing, until there are no more humans or human-like beings who know any of these such stories to think about them.

And all of this magic was possible simply because I *shared my feelings about my experience* at my uncle’s death bed in a story.

Of course this is all just metaphorical.

Storytelling.

Fiction. Kids stuff.

Not real magic.

They aren’t *really* still *actually* dancing with one another, the man with the head injury, and the girl he loved more than anything in the world.

Right?



You tell me.



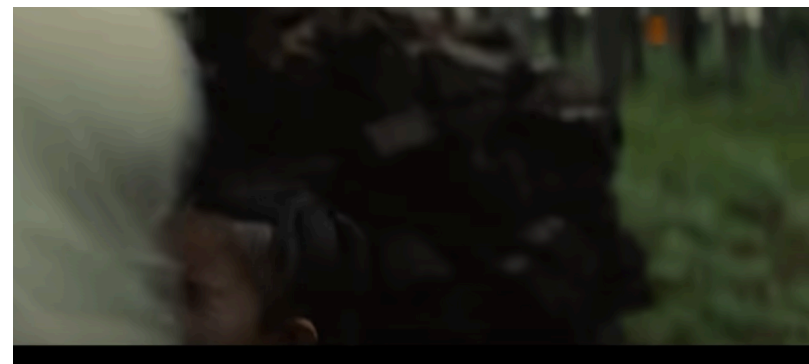
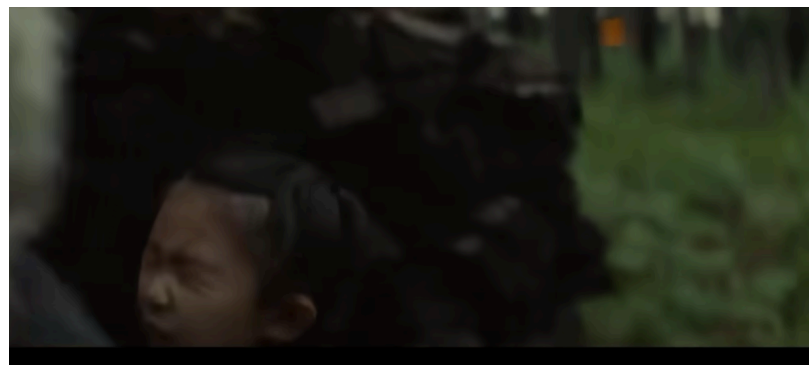


Losing Mary Anne (you then asia: Maria) as she passes away.  
Losing his children as he does, soon thereafter.



Getting them all back...through a story.

“Animated Shadow Dancing Through Trees”  
“you then asia”





“The infants will not shut up.” Ever, for all practical purposes.  
Isn't that grand?

[https://youtu.be/RO5FqAi6oZQ?si=o2GURoEAU\\_TBz4BC](https://youtu.be/RO5FqAi6oZQ?si=o2GURoEAU_TBz4BC)