Clients, Customers, and the Celebration of Change

Richard P. Gabriel

Software, when successful, enjoys a variety of sources of influence—the notion of influence more useful (being more indistinct) for describing the different people, organizations, clients, customers, users, maintainers, developers, writers, sellers, principles, philosophies, religions, cultures, lifestyles, and everything else whose actions, hopes, plans, and desires shape the forces coming from the outside that, along with the nature of software with its internal forces, constraints, requirements, the quirks of programming languages, and the inflexibilities of components and frameworks, conspire to shape the final product, form it as if through a geologic or evolutionary process, make it not only a tool but an actor in the world, creating forces that shape not only other software, not only other artifacts, but the lives and futures of people and civilization, and that thing is this: a software system—over its creation.

That such a thing as software can be closer or less close to perfect is a fact of human design and construction, of human weaknesses and strengths, of the nature of people to learn and adapt over time; and making a thing perfect is an act of exploration, discovery, invention, creation, consideration, observation, and revision, each step fractally contained within the others so that regardless of how close one looks at the matter or how far away, the intertwined spirals present similarly, and a participant is often puzzled about how to describe what he or she is up to at any given moment, so confused seeming is the process. No one would claim that any piece of software large enough to have more than the simplest purpose cannot be improved, and two questions arise: what place does the concept of perfection take in the universe where software is made, sold, commissioned, and bought, and (this is raised by agile methodologists) what rôles do customers and clients play in the dance of creating perfect software—or is the idea that such folk play a rôle a fantasy or the product of thinking directed by greed? Let’s look: Shorter OED says, in its apparently most relevant definition, a customer is:

> *A person who makes a purchase or gives business, esp. habitually to any particular seller or establishment.*

A customer, by these lights, then, is an actor in a mere commercial exchange and holds no special place in the perfection of the software aside from that which enables the purchase. And but so when the programmer—as I shall refer to the worker enmeshed in the fractal work of making a software system—is engaged with the customer, the goal—the only goal and the paramount one—is to secure the purchase and then (perhaps) to secure the habituation of the customer to purchase recurrently. One imagines that satisfaction may play a part, but dependence often suffices, and the question becomes whether satisfaction lies on the path to perfection—I am interested and we should all be interested in perfection.

And what of the client? Again the OED:

> *A person using the services of any professional; a customer.*

Aside from foregrounding service over mere purchase and bringing in a professional, a client is just a customer—a customer of someone providing a service, of a professional someone providing the service; and this brings us to the idea that the professional may or may not do a good job (during the service), or perhaps that the service can be done more or less well. When the customer is a mere purchaser, then what might be considered acceptable or even exemplary products or services may or may not include the enjoyment of the products or service; sometimes it might include reports from others—real users and not simply the buyer—of satisfaction or enjoyment or correct performance; sometimes or most certainly almost always it might include a satisfactory or enjoyable purchase price. But does this have anything to do with perfection? And—and this is a key question—and does perfection have anything to do with satisfaction or enjoyment or the suitability of the purchase price—does perfection contain the seeds of habituation that characterizes a customer from a mere buyer or procurer?—or is perfection the enemy of habituation; if a system is perfect, does it mean that a customer will not return, and the habit of the purchaser is to be thankful and walk away until something else perfect is needed or desired?

Ah, but perfection comes to the rescue: perfection + complexity + people = the impossibility of perfection—can’t be done, so the merchant need not fear selling, by accident as commerce would appear to require, perfection. In this case, does the pursuit of perfection lead to satisfaction, enjoyment, and repeat business? Or is subjectivity essential—does it matter more that the software be what the user (client? customer?) wants or desires than it be perfect? How different from perfection is subjectivity? Is the Mona Lisa—arguably perfect—less desirable than a less perfect Mona Lisa? Or let’s look at it another way: da Vinci sitting by a patron—call the patron the customer—responding to the patron who helps make paint selections, who helps pose the woman, who places her within the canvas, who determines the smile. How much more would the patron enjoy that painting to the one da Vinci painted on his own using his own (—let's call them—) muses?

To be sure, the customer wants slash needs the software to do something—this is sometimes not negotiable (though what if what is desired is illegal or immoral or against cultural standards—like backdoors for voting fraud, a tunnel for private bedroom webcams to be spied from afar, or a mechanism to distribute pornography?)—and if the customer needs to use the software—not pay for it only for someone else to use and perhaps suffer through because the customer is purely in a commercial relationship to the supplier—the comfort and enjoyment of the software by the customer is important but...

Consider the bicycle frame. The frame maker does not take all his/her cues from the customer who in this case is the bicyclist who will ride the bike for thousands of miles but (the frame maker) uses her/his expertise about frame making / about bicycling / about the ultimate comfort of a cyclist and sets parameters/offers choices but never relies or depends on the cyclist to make all decisions. The perfect bicycle frame for the customer is a perfect bicycle frame adjusted for the cyclist. It is the perfect bicycle frame.

Let's pretend: that perfection is the goal. Wouldn't a customer accept perfection? Perfection does not (often enough) arise the first time. Evolution shows that; our fractal view of creation demonstrates it in the act of making things. Perfection comes from improvements which comes from two things: deliberate improvements and changes—even random changes. Consider simulated annealing (or any act of creation). But first: consider the small change, the what-if, the what-does-this-do? The maker doesn't know what will happen but just does it; maybe one thing leads to another and a step is made toward perfection. Changes—this is what creates the opportunity for creational perfection. Never sit still. Go for it. Simulated annealing—sure it's got its metaphorical connection to making steel and all that, but its key step is the random choice to do it (a change) anyway, even if there is no reason to believe progress will be made.

> *A simulated annealing algorithm always accepts moves that decrease the value of the objective function. Moves that increase the value of the objective function are accepted with probability*
>
> $$p=e^{\Delta/T}$$
>
> *where Δ is the change in the value of the objective function and T is a control parameter called the temperature.*
>
> *By analogy with the physical process, the probability of accepting a move that increases the objective function is initially high. The temperature is gradually decreased as the search progresses. That is, the system is* cooled *slowly. In the end, the probability of accepting a move that increases the objective function becomes vanishingly small. In general, the temperature is lowered in accordance with an* annealing schedule.

Change—even random change—is where improvements, optimizations, and perfection come from. It isn't the customer *per se* but change *per se*. What does our friend-document the Agile Manifesto say about this?—:

> *Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.*

Welcoming changing requirements is part of welcoming change. Requirements are part of the what-the-thing-should-do and also of the how-the-thing-should-be, but change of requirements is only one category of change. Perfection demands more. And welcoming is only half way there: During different stages—during early stages—changes must be encouraged, not just welcomed. The customer's changes are perforce requirements changes, but the customer cannot be depended on to propose all the changes that are needed to perfect the software. This would be like thinking that a cyclist should be depended on to propose all the changes in perfecting the bicycle frame. Welcome changes: sure; but encourage them too.

What are the key influences on making successful aka perfect software? The most key is change—finding sources of change (a customer can do that; a client can do that) and using them makes a thing and software more perfect—and more perfect is more satisfying, more enjoyable, causes habituation of purchase, is the ~~dream promise~~ hope of free-market capitalism, gratifies the muses and artists among us. But don't rely solely on the customer or the client—what do they know about creation anyway (maybe a lot, but)?

Encourage changes during the early stages, even changes that don't obviously make things better; and as design and development move forward, decrease the temperature and slowly cool the software until it is perfect.