

The Agile Physician

Any software development project has at least two kinds of issues: “build the right thing” and “build it right.” In recent years, agile practices such as test-driven development have made a real dent on the build-it-right front. This has provided much-needed relief

for developers who can now work more confidently and comfortably than ever before.

On the build-the-right-thing front, agile methods’ ability to deliver stable value early and often has also helped. However, it hasn’t resulted in the same level of relief for customers as for developers. Having been in both roles on a number of projects, I can vouch that the customer role is by far the more stressful. As a customer on agile projects, I often feel like the patient in this tongue-in-cheek dialogue with an “agile physician.”

Physician: Good morning, Mr. Brown.

Patient: Good morning, doctor.

Physician: What can I do for you?

Patient: Well, this lump recently grew at the back of my skull. Since then, I often feel dizzy and have violent headaches.

Physician: Sounds like you might have a big problem. What treatment would you like us to perform on you?

Patient: Huh? I was hoping you could tell me that.

Physician: I see This is your first time consulting an agile physician, isn’t it?

Patient: Yes.

Physician: I can tell. Well, let me start by telling you a bit about the principles of agile medicine. In agile medicine, we don’t believe that physicians are in a position to choose the right treatment for your condition. You’re the one with the illness and experiencing the symptoms, and you’re the one who’ll suffer the consequences of treatment. So, *you* are in the best position to determine the

appropriate treatment. My role as agile physician is limited to expertly doing exactly the treatment that you feel is right for you in a timely and cost-efficient manner. Isn’t that great?

Patient: I guess so ... but what if I choose the wrong treatment and die?

Physician: Don’t worry. Even if you die, we’ll still get paid.

Patient: But I’ll be dead! That’s a big concern to me.

Physician: Well, you can’t have it both ways. You can’t have the complete freedom of choosing your own treatment without assuming responsibility for those decisions. It wouldn’t be fair to us physicians.

Patient: I guess so ... but isn’t there *anything* you can do to help me figure out what’s good for me?

Physician: Oh yes, absolutely! We realize that making this kind of decision is very hard, so we take an iterative approach. You choose the treatment that you think is most appropriate for your condition, and then we do a little bit of it. We do the SmallestBitOfTreatmentThatCouldPossibly-CureYou. Then *you* get to evaluate if it worked or not. You also get a chance to change your mind—either do more of the same thing or try a different treatment. Makes sense, no?

Patient: I guess so

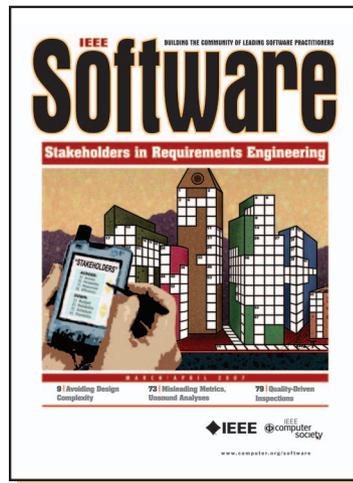
Physician: Absolutely! And, on top of that, we can estimate fairly accurately how much each bit of treatment will cost you.

Patient: That’s good. Money is important to me.

Physician: It is to us too, I can assure you.

Patient: Okay, so can you tell me some treatment options and their cost?

Physician: Sure. We can do a CAT scan for \$2,000, a regular x-ray for \$500, a course of chemo for \$2,000, and a course of radiotherapy for \$2,500. Surgical removal of the lump is about \$1,500. Acupuncture, biofeedback, and medita-



We welcome your letters. Send them to software@computer.org. Include your full name, title, affiliation, and email address. Letters are edited for clarity and space.

tion lessons are in the \$50 range per session. Bloodlettings are free.

Patient: Hmm. I don't think I want to go for bloodletting.

Physician: Your call.

Patient: But I'm not sure I want to go for the more expensive treatments at first. So, I'll go for the regular x-ray at \$500.

Physician: Sounds reasonable. If you go into the next room, my assistants will start treatment right away.

Patient: Thanks, but I feel kind of ...

Physician: Empowered? Yes, I know.

Like this patient, customers on agile projects are often asked to make critical, project-defining decisions, and very little in the methodology can help them make those calls. The situation on agile projects is no worse than on more waterfall-ish ones, and, in fact, agile does make the customer's job easier. My sense, though, is that we haven't made as much progress there as we need to. We need to develop agile practices that will let customers

carry out their work with the same level of confidence and comfort as developers can now. A good place to look for such practices is user-centered design, and a growing body of experience shows how to use lightweight versions of those practices in an agile context.

Alain Désilets

*Research officer, NRC Inst. for
Information Technology
alain.desilets@nrc-cnrc.gc.ca*

Fun vs. tedium

The title of the March/April 2007 Loyal Opposition column says "Is Software Engineering Fun? Part 2," but the first heading uses the word "programming." Worlds of difference there.

I think a lot of the fun is doing something in which you're interested, whether or not you're getting paid for it (double fun if you are). Tedium occurs when you get stuck doing something in which you're not interested, but which for other reasons (job security, medical insurance,

mortgage payments, and food all come to mind) you're compelled to perform. You can't always take the high moral road that Grady Booch does and "speak truth to power" when you can't risk your livelihood.

For example, my Sun Fire T1000 Evaluation (www.grebyn.com/t1000) was a lot of fun that I didn't get paid for. The fact that Sun Microsystems eventually recognized it as a winner in their Open Performance Contest by awarding me the machine was a fortuitous bonus (a non-goal, as some would say).

"Do what you like, the money will follow," they say. "Show me the money," they also say.

Some folks in my family can't understand why I enjoy software engineering/programming/coding/hacking. But then, they can't understand why I like asparagus either. ☺

Karl A. Nyberg

President, Grebyn Corp.

karl@grebyn.com



ICGSE Series

International Conference on Global Software Engineering

Call for Participation

**August 27-30 2007
Munich, Germany
www.icgse.org**



Developing software in globally distributed teams offers many benefits but also poses new challenges.

ICGSE is an international IEEE conference that brings researchers and industry practitioners together to share and explore the state-of-the-art and state-of-the-practice of global software engineering.

Register to attend and benefit from the renowned expertise of this international global software engineering community.

Topics include: benefits and risks, culture and communication issues, lessons learned from case studies, managing multi-site projects, collaboration infrastructure to support distributed development teams, teaching global software engineering, supplier management, requirements engineering, architecture, testing, etc. in a global software engineering context

Tutorials & Workshops: August 27
Conference: August 28-30

Early registration deadline: June 9