More Requirements Engineering Adventures with Building Contractors

Daniel M. Berry
School of Computer Science
University of Waterloo
200 University Ave. West
Waterloo, Ontario N2L 3G1
Canada
dberry@uwaterloo.ca

August, 2002

© Copyright 2002 by Daniel M. Berry

Abstract

This Viewpoint describes some lessons about requirements engineering I learned while being a customer in a house remodeling. The main lesson is the importance of the customer insisting on following a full requirements engineering process, including goal identification, requirements elicitation, analysis, and specification, and validation of the specification.

Introduction

In 1998, I published in these pages a Viewpoint in which I described lessons about combatting requirements creep learned while being a customer in a house remodeling and in a house construction [2]. The main point is that we requirement engineers have much to learn about our own professional processes, especially with regards to controlling requirements creep, by being customers of a process similar to our own, such as building or remodeling a house. A secondary point is that we can learn to be better customers of house construction or remodeling from what we know from requirements engineering, especially in combatting building requirements creep, which enriches only the contractors.

In 2001, in the same Viewpoint series, Roel Wieringa wrote about his adventures renovating two houses and building a third from scratch. The lessons he learned concerned the inordinate complexity of building houses and software, the relentlessness of requirements changes, and the importance of human communication and writing skill in requirements engineering [13]. The next section discusses some of these lessons in detail.

The present paper describes some additional lessons learned about requirements engineering and life cycles learned while being a customer in a recent remodeling effort.

Wieringa's Lessons

I assume that you have read Wieringa's Viewpoint, as space limitations do not permit building up his arguments enough to make this discussion self contained.

Wieringa discovered that in some cases of house remodeling and building, requirements engineering continued until after the product, the house, was delivered (although in this case, it was really the people that delivered themselves to the house). The point of my first article was that perhaps we can learn to avoid this continuance by doing careful full-process requirements engineering, with walkthroughs, inspections, and other validation techniques. In my case, I applied enough of these validation processes that I did not have any requirements creep after the plans were finalized. After all, the domain of house building is well understood, much like that of compiler

building. On the other hand, if even people like us in requirements engineering, with all of our knowledge about requirements management, cannot avoid requirements creep in a well-understood domain such as house building, what hope is there of the average customer, either of houses or software, avoiding requirements creep in a typical, not well-understood domain? Maybe we need to accept that requirements engineering is inherently difficult.

Wieringa observes that a typical house buyer and user deals with requirements failures in his or her house by fiddling around with the house after delivery for several years, until it fits, and perhaps even looks, like an old shoe. He adds that a typical software user is less fortunate since he or she has to live with what is delivered in the software, adapting to the software rather than the other way around. Unsaid, but certainly implied, is that the results of fiddling with the house after delivery are never quite as good as if the requirements problem had been detected in time for the delivered building to be changed to deal with it. I add the observation that some house users, namely those with building skills, and some software users, namely those with programming skills, know enough about fiddling to achieve better adaptation of the product to their needs than the average. I am, and Wieringa seems to be, in this category in both software and house building. Most, however, have to live with the delivered products more or less as they are, adapting to them rather than the other way around.

I agree that there are fundamental differences between software and house construction stemming from the fact that software and its specifications are in basically the same medium, while houses and their specifications are in radically different media. Nevertheless, the requirements process for the two are similar enough for lessons to be learned in both directions.

Finally, Wieringa's exhortation for requirement engineers to learn how to write carefully cannot be understated. At the University of Waterloo, the Software Engineering program has the students take a writing course. I have offered to teach it, when the mainline requirements engineering course is stable enough for a less expert person to take over. I add to Wieringa's recommendation of Ben Kovitz's book on requirements writing [11] recommendations for Lyn Dupré's book titled, appropriately enough for software engineering, *Bugs in Writing: A Guide to Debugging Your Prose* [7]. This book does a better job than *any* other writing style book, including the very famous ones, at covering issues that crop up in specification writing. If I may be so crass as to plug some of my own work, I and some colleagues have been doing some research on writing user's manuals as requirements specifications [4] and on avoiding and detecting ambiguities, often stemming from poor writing, in natural language requirements specifications [3, 8, 10, 9, 6, 5].

Specification writing and house plan drawing each involves elements of both technology and artistry, both in what is specified and in the specification itself. Just as architects must learn how to draw plans, requirement engineers must learn how to write.

The rest of this paper deals with the promised additional lessons learned about requirements engineering and life cycles learned while being a customer in a recent remodeling effort.

Request for Proposal

After moving to Canada, I bought a house and soon thereafter, I discovered some problems that required the expertise of professional builders and electricians to fix. I prepared a Request for Proposal (RFP), a high-level requirements description in which I thought I had stated only requirements.

> My shower stall is leaking and the plaster on the outside of the shower is crumbling. I guess I will need to have the showerstall [sic] rebuilt.
>
> I would like to have the amperage of one outlet increased so that using the toaster at it does not dim the lights in the whole house.
>
> The power to the baseboard heaters needs a bit of work. I think one of the heaters is malfunctioning.

> One phone jack is dead; I would like the wire damage located and fixed.
>
> Can you do these things?
>
> If so, when can we meet?

I soon realized that what I described in fact bordered on being a description of a solution and was not really a pure requirements specification. However, since what I needed to do was clearly constrained by the existing house structures, I thought that what I described could be considered requirements.

I faxed or e-mailed the RFP to about 2 dozen remodeling contractors in the Kitchener-Waterloo area. Some were recommended by friends. In any case, each had either an e-mail address or a fax number listed in its Yellow Pages entry. With an e-mail address or a fax number, I could communicate with each in textual form rather than voice, and thus avoid the difficulty of reading lips over the telephone.* Many contractors never replied. In the end, I got replies from only three, hereinafter known as C1, C2, and C3, with the digit reflecting the order of replying to me. I use these designations to refer to both the company and the representative person who dealt with me, the owner in each case.

Contractor C1

The first to reply, C1, sent me a fax and proposed a meeting at my house for him to see the work sites. He took my RFP as explicit requirements and discussed various implementations.

With regard to the electrical work, C1 proposed replacing the existing wiring from the fuse box to the dinette outlet by heavier wiring so that he could put in a higher amperage fuse. However, he noted that the living room and dinette were on the same fuse. Allowing more current on the line would endanger the television set, the video cassette recorder, and other entertainment appliances in the living room. Thus, C1 said that we should split the line and fuse into two, one for the living room and one for the dinette, each with its own amperage. However, there were no more fuse slots in the fuse box. I suggested attaching the living room line to the dining room line, to leave the former dinette–living room line for only the dinette. Since the dining room had no appliances, the load on the new dining room–living room line should be fine.

With regard to the shower, C1 proposed a three-piece unit with seams overlapping only downward, to prevent dripping leakage, rather than redoing the tiling. No matter how well the tiling is done, tiles and grout eventually leak.

Two weeks later, he sent an estimate that covered all tasks in a complete middle-level requirements specification. He said that if I chose him, he would produce detailed plans for the shower work.

Contractor C2

About a week after the meeting with C1, but before C1 sent his estimate, C2 sent the following fax to me:

> Mr. Berry;
>
> ...
>
> We are happy to quote you a fixed price to repair and/or rebuild your shower as is found to be necessary. This is work we regularly do. Please visit our website at ... to see some of our work.
>
> Also, we can service you [sic] 110 v.a.c. toaster outlet, service power to your electric baseboard

---

* For those of you who do not know me, I am hearing impaired and depend on reading lips to understand spoken language.

> heaters, and service your malfunctioning telephone jack. Our standard labour rate for such service is $39.00 per hour per service person, plus materials, plus applicable the taxes [sic].
>
> If you choose to retain us for this service work, we will inspect your shower when we are on site, make recommendations, and follow up with a written proposal containing a stipulated price for the decided upon shower work.
>
> We shall look forward to receiving your written reply, and shall respond to you promptly via facsimile.
>
> Thank you.

When I read this fax, I remarked to myself, "My G-d! He's proposing an XP [(extreme programming)] approach for the electrical task followed by a waterfall approach for the shower task!" That is, he was not planning on ironing out requirements ahead of time. He was planning to come, sit with me, get the four stories, and immediately begin implementing each of the stories. I, the customer would be on site the whole time to make sure that the implementations proceeded to my requirements. He might even have more than one of each craftsperson for the equivalent of pair programming. In fact, C2 had said "per electrician", leaving open the possibility of more than one electrician! Even the tests were effectively written before implementation, because I could take a shower, I could turn on the toaster and watch the lights in the kitchen, I could turn on the heaters, and I could make a phone call. This is surely classic XP or perhaps XR (eXtreme Remodeling) [1]. I immediately decided against C2, because I realized that I would be paying $39.00 per person hour plus taxes, not only for the implementation of the electrical solution, but also for an on-the-fly requirements analysis and specification phases during which time all the electricians except the leader would be sitting, doing nothing but watch their leader and me haggle out the specifications. Since I would be anxious to get started on the implementation as quickly as possible so as not to spend more than necessary on the requirements specification, I would be under great pressure to accept the first workable specification produced and would not be interested in finding the best approach.

As I said, I turned C2 down immediately because I had at least one contractor, C1, that was willing to write requirements specifications for the electrical task at his own expense. I knew that the waterfall approach [12] would be more to my advantage than the XP approach simply because with the waterfall approach, I would pay only for implementation based on requirements that had been more thoroughly thought through and reviewed by me, all at the contractor's expense.

Contractor C3

The third to reply was C3. When C3 came to my house, he surprised me by doing real requirements engineering. That is, he said that he thought that my so-called requirements for the electrical work were too complicated both as requirements and to implement. He looked around in the kitchen that adjoins the dinette that needed stronger outlets. Figure 1a shows what he saw. He said that there is a better way to achieve what I *really* want, which is simply to have stronger outlets for the toaster and microwave oven. I agreed that he captured my basic goal. He showed me the wall that the kitchen and dinette shared and showed me a pair of outlets in the kitchen that were unused. They were unused because I did not know that they existed; I have never noticed them. He explained that by the building code, kitchen outlets are already strong enough to support toasters and microwave ovens, two of the hungriest appliances in the house. He asked, "So why not plug the toaster and microwave in those outlets in the kitchen?", as shown in Figure 1b. I explained that I did not want the cords running on the floor along the wall around the archway between the two rooms. He explained how he could put in another pair of plugs on the other side of the wall, opposite the existing kitchen pair of outlets, so that there would be a pair of outlets inside the dinette, coming off the existing strong line in the kitchen. Figure 1c shows what he intended.

I was really impressed with C3. He saw the real requirements. He found two cheaper ways to implement them, one so cheap that he would not make any money at all on the electrical work. That zero cost way, running the cords around into the kitchen as shown in Figure 1b, was ugly; so, I opted for the other method, still quite a bit cheaper than what I had envisioned with my RFP.

C3 then inspected the shower and proposed essentially the same as C1, right down to the brand of shower unit. As expected, C3's estimate was considerably lower than C1's since the electrical work was relatively trivial. The part of the estimate dealing with the shower was basically the same as that of C1's.

My Decision

With C1's and C3's estimates in hand, with the knowledge that they were essentially equivalent with respect to the shower work, I asked each when he could begin. It turned out that C1 could start 2 months earlier. So I went with C1, but modified the electrical requirements in the way that C3 suggested. C1 lowered his electrical estimate. I sent C3 a gift check for $50.00 for his time and the idea that led to the new electrical requirements and the lowering of C1's estimate.

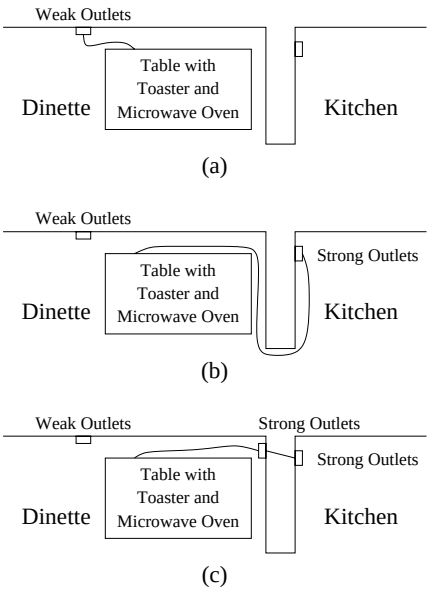

Figure 1: Floor Plans

Observations and Lessons Learned

It was interesting to be on the customer's side of a requirements engineering effort with the knowledge of a requirements engineer. As the customer, I firmly resisted working in any way in which implementation would start before the requirements and costs were agreed upon. I did not want to pay for on-the-spot requirements analysis or prototyping. The building profession has a tradition of the contractor bearing the costs of requirements engineering. Therefore, it was prudent for me, as a customer, to stick with the contractors that followed the waterfall model completely and did the requirements engineering up front and at their own expense.

It is important to understand what the real requirements are and to specify requirements and not solutions. Too often solutions are offered as requirements. On the other hand, it is the professional's job to recognize that the customer has specified a solution and to ask questions that ferret out the real requirements if he or she cannot see them on his or her own. The professional has the domain knowledge necessary to make this recognition. In my case, contractor C3 knew enough about the building codes to know that there were strong enough outlets in the kitchen; he knew, and could verify by observation, that the dinette would be near the kitchen. Thus, he was able to see an alternative implementation to the real requirements and then abstract to the real requirements.

It is essential for the customer of either a house building or a software development to insist on following a full requirements engineering process, including

1. goal identification,
2. requirements elicitation,
3. requirements analysis,
4. requirements specification, and
5. repeated validation of all of the above.

Conclusions

We requirements engineers have a lot to learn about our own processes by being customers of a similar process. In my opinion, for us to learn what it is really like to be a customer, it is necessary for us to be customers in a process, such as house building, not directly related to our own professional expertises, computer-based systems. I have been a customer for software construction [14, 4] and it was not as difficult, anxiety producing, and educating as being a customer for a house building. Perhaps, I know too much about software development to be as mystified as most customers are! Perhaps?

One idea that has occurred to me is for us to write a manual to be given to potential software construction customers and users explaining how to play their roles in a software requirements engineering process. It should tell them about all the processes involved and their roles in them. It should admonish them not to accept any attempt by us to pull the wool over their eyes. It should admonish them not to allow us to proceed to any next step until they have validated the work done up to then. Finally, it should admonish them to bug us to death with questions about *any* thing that they do not understand about what we are doing. In other words, it should teach them how to be as good customers to us as I was to my contractors. By "good" here, I mean for the customers' own benefits and not for the benefits of the requirements engineers' income or laziness.

Perhaps, a slight variation of this manual could be written for customers of house buildings.

Acknowledgements

I thank Nancy Day, Jing Dong, Roel Wieringa, and Didar Zowghi for useful feedback. I was supported in part by NSERC grant NSERC-RGPIN227055-00.

References

[1] Beck, K., *Extreme Programming Explained: Embrace Change*, Addison-Wesley, Reading, MA (1999).

[2] Berry, D.M., "Software and House Requirements Engineering: Lessons Learned in Combating Requirements Creep," *Requirements Engineering Journal* **3**(3&4), pp. 242–244 (1998).

[3] Berry, D.M. and Kamsties, E., "The Dangerous 'All' in Specifications," pp. 191–194 in *Proceedings of 10th International Workshop on Software Specification & Design, IWSSD-10*, IEEE CS Press (2000).

[4] Berry, D.M., Daudjee, K., Dong, J., Nelson, M.A., and Nelson, T., "User's Manual as a Requirements Specification," Technical Report CS 2001-17, Computer Science, University of Waterloo, Waterloo, ON, Canada (May 2001).

[5] Berry, D.M., Kamsties, E., Kay, D.G., and Krieger, M.M., "From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity," Technical Report, University of Waterloo, Waterloo, ON, Canada (2002).

[6] Denger, C., "High Quality Requirements Specifications for Embedded Systems through Authoring Rules and Language Patterns," M.Sc. Thesis, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern, Germany (2002).

[7] Dupré, L., *Bugs in Writing: A Guide to Debugging Your Prose*, Addison-Wesley, Reading, MA (1998).

[8] Kamsties, E. and Paech, B., "Taming Ambiguity in Natural Language Requirements," in *Proceedings of the Thirteenth International Conference on Software and Systems Engineering and Applications*, Paris, France (2000).

[9] Kamsties, E., Berry, D.M., and Paech, B., "Detecting Ambiguities in Requirements Documents Using Inspections," pp. 68–80 in *Proceedings of the First Workshop on Inspection in Software Engineering (WISE'01)*, ed. M. Lawford and D. L. Parnas, Software Quality Research Lab at McMaster University in Canada, Paris, France (23 July 2001).

[10] Kamsties, E., “Surfacing Ambiguity in Natural Language Requirements,” Ph.D. Dissertation, Fachbereich Informatik, Universität Kaiserslautern, Germany, also Volume 5 of Ph.D. Theses in Experimental Software Engineering, Fraunhofer IRB Verlag, Stuttgart, Germany (2001).
[11] Kovitz, B.L., *Practical Software Requirements: A Manual of Content and Style*, Manning, Greenwich, CT (1998).
[12] Royce, W.W., “Managing the Development of Large Software Systems: Concepts and Techniques,” in *Proceedings of Wes-Con* (August 1970).
[13] Wieringa, R., “Software Requirements Engineering: the Need for Systems Engineering and Literacy,” *Requirements Engineering Journal* **6**(2), pp. 132–134 (2001).
[14] Wolfman, T. and Berry, D.M., “flo — A Language for Typesetting Flowcharts,” pp. 93–108 in *Electronic Publishing ’90*, ed. R. Furuta, Cambridge University Press, Cambridge, UK (1990).