10x Software Development

Numerous studies have found 10:1 differences in productivity and quality among individuals and even among teams. This blog contains Steve McConnell's thoughts about how to move toward the "10" side of that 10:1 ratio.

Building a Fort: Lessons in Software Estimation

Also Known as: How I Spent My Summer Vacation

My big project this summer was building a fort for my kids. I'd wanted to build a clubhouse or treehouse or fort or something for the past few years, but we didn't have a good place to put it. Then while clearing some blackberries in the spring I discovered that our property extended about 20 feet further into the adjacent overgrown area than I had thought, and that was the perfect place for a fort.

Whenever I do a physical construction project like this I try to pay attention to which attributes of the project are similar to software projects and which are different. The comparisons are made more challenging by the fact that my construction projects are recreational, whereas I'm trying to draw comparisons to commercial software projects. For the first half of the project, no good similarities jumped at out me. But as the project started to take much longer than I expected, I began to see more and more similarities between my estimates on the fort and problems people run into with software estimates.

Original Work Plan

Here was the work plan I had carried around in my head for a few weeks before I started the project:

**Day 1:** Dig holes for footings, pour concrete for footings, haul building materials from my driveway up the hill to the fort.

**Day 2:** Cut posts and beams to length. This was planned as a half day because I didn't want to put too much stress on the concrete footings until Day 3.

**Day 3:** Finish beams and joists and install decking; do some of the deck railings as time permits

**Day 4:** Complete the fort's framing, minus the roof

**Day 5:** Frame and install the roof

**Day 6:** Install door and windows; finish deck railing; install trim boards

**Part time the next couple of weeks:** Finish loose ends

I won't go through all the errors in my estimates for the whole project, but let's take a look at what I really did on Day 1.

**DAY 1**

**Task 1.1 Clear brush from site (~1 hour).** I'd known that I had a little brush still to clear, but I thought it would take me about 10-15 minutes. Once I started looking at where I needed to put the footings, I found that I really couldn't put them where I'd planned because I would be inside the setback for the property. So I needed to move the fort back about 5 feet, and that meant clearing a bunch of brush including scrubby trees that I hadn't planned to clear.

**1.2 Survey the site and determine placement of footings (~3 hours).** I'd originally planned to build the deck with 2 beams and 2 posts per beam. After looking at some span tables, I concluded that I could *probably* get away with 2 beams with 2 posts each, but what I was building was right on the border between 2 and 3 beams and between 2 and 3 posts per beam. I decided to err on the side of caution, and that meant I needed 9 footings instead of 4. Meanwhile, I had never really adjusted my time expectations to digging 9 holes instead of 4. Siting the 9 holes also turned out to be an issue because of a big stump in the middle of my area.

The site overall had more of a slope than I had realized. I wanted to stake out the corners and use string to locate the position of each hole and make sure the holes were square. Due to the slope, the stakes I was using weren't tall enough on the downhill side of the site, and I spent time pounding in stakes that ended up not being tall enough, then pulling them out, hammering together makeshift taller stakes, and then pounding those in.
I ended up spending a lot of time moving stakes and string around trying to figure out how to get 9 holes that were (a) not blocked by roots from the stump, (b) not blocked by roots from the tree in back of the fort, (c) far enough back from the property line to meet the setback requirement, (d) square relative to each other (which was hard to determine at this stage because of the slope I was building on).

1.3 Dig post holes (~2 hours). I had to dig 9 holes, 12" in diameter, 24" deep. This actually went quicker than I expected. I used a clamshell digger and for the holes where I didn't run into any roots it was something like 5 minutes per hole. The difficult holes were the holes where I ran into roots partway down and then had to hack them out. Some of the holes had quite a few roots.

1.4 Haul 20 80# bags of concrete up the hill (~1.5 hours). I had originally thought I could haul the concrete up the hill using a wheelbarrow, but the hill was just too steep. So I had to hand carry each 80# bag one at a time. It was also about 80 degrees and 95% relative humidity at this point, which meant I needed to rest and drink water every couple of bags. The change from 4 holes to 9 holes also increased the number of bags I had to haul from about 10 to about 20.

1.5 Pour 2 Footings (1.5 hours). At this point I was pretty worn out, but I also really wanted the feeling of completion from pouring at least one of the footings. So I ended up pouring 2 of the footings and calling it a day since there was no way I was going to complete all 9 of them at that point in the day.

End of Day 1. The picture below shows how far I got at the end of Day 1.

What Went Wrong with My Estimate for Day 1

- I hadn't examined my planned site well enough to know what I didn't know -- i.e., my originally planned site wouldn't work and I didn't understand how much slope there was.
- I never revised the expectations I had created while planning a 4-footing Day 1 to more appropriate expectations for a 9-footing Day 1. That one mistake affected my site layout, concrete hauling, hole digging, and concrete pouring.
- Brush clearing just took longer than I expected, and I hadn't included it in my estimate at all.
- Surveying the site also just took longer than I expected, and would have even without the change from 4 holes to 9.

DAY 2

What I could complete on Day 2 was limited by the fact that hadn't poured all the footings on Day 1, so about all I could do on Day 2 was pour the remaining 7 footings and haul the rest of the building materials up the hill. The rest of the footing pouring went fine and took about 4 hours. Then I needed to haul the materials up from the driveway. The pile of stuff didn't look all that intimidating:
Superficial appearances aside, however, there are 10 16' 2x8 pressure treated joists in that pile, and those suckers are heavy. There are also 3 12' 4x8 pressure treated beams in that pile, and those suckers are *really* heavy! And then there were 70 2x4s and 50 lengths of 5/4” decking, and 100 2x2 balusters for the railing, and about 15 sheets of plywood, and 2 bundles of roofing shingles, and a lot of other stuff, and all this stuff just starts to add up after awhile. It took me at least 50 trips up the hill, and that ended up taking me about 3 hours.

At the end of Day 2 I was about where I thought I'd be at the end of Day 1 after 2 pretty long days. For the record, here's what I had done at the end of Day 2:

What Went Wrong with the rest of My Estimate for Day 1 (i.e., the Work I Did on Day 2)

- Hauling the building materials up the hill took longer than I planned, mostly because I'd never bothered to break down the "hauling" task and realize that it was going to take 50 trips, not 10.

DAYS 3-6

Days 3-6 went about like Days 1 & 2 had gone, which is to say there were lots of little tasks that turned out to be medium-sized tasks, there were little tasks that I just hadn't anticipated, and most things took longer than I had planned. By the end of Day 7 (my buffer day), I was done with the tasks I had planned for Day 3 and had a tiny start on Day 4, which is to say that I'd completed the decking, hadn't started on the railings or framing, and had one wall of the fort framed, but that was all.

DAYS 7 AND FOLLOWING

Since I'd used up my planned full-time days on Day 7, the rest of the fort had to be completed after work, so I had to work on it only a few hours at a time, and I couldn't work on it every day. So my calendar time overrun started stretching out faster than my effort overrun did.

OVERALL RESULTS

My original plan had called for about a week full time and then another couple of weeks of finishing up loose ends like painting, installing trim, and so on. I finished the fort about 6 weeks after I started it, so I was about 100% over my planned schedule, and I ended up at 2-3x my originally planned effort. Here are some pictures of how it turned out:
ESTIMATION LESSONS LEARNED

I mentioned some of the specific estimation mistakes on Days 1 & 2. As I got into the project I noticed several other issues that the estimates I'd made for my project had in common with errors in software estimates that we see with our clients:

1. Numerous unplanned problems collectively added up. Here are a few of the problems I ran into:

   - When I opened my chainsaw case, which I hadn't used in a couple of years, I found that the oil plug hadn't been screwed in tightly, and the chainsaw was covered in oil, as was the case itself. I had to clean up the chainsaw and then dispose of the oil.
   - When I was digging the holes for the footings, I chopped my layout strings a couple times with the post hole digger. I had to spend time repairing the strings and making sure that everything was still square.
   - I dropped a little piece of my laser level down the side of one of the footing holes, between the concrete form and the dirt, after I'd poured the concrete. The piece was perched about 8" into the hole, just where I couldn't reach it. If I touched it wrong, it would drop the full 24" to the bottom of the hole where there was no way I could retrieve it. So it took me awhile to figure out how to get the piece out without risking losing it altogether.
   - My jigsaw has a little compartment/drawer to put spare blades in, and it kept coming loose and spilling blades on the ground. I looked at it several times before the sunlight finally hit the opening just right to see that there was a blade stuck under the slot where the drawer is supposed to slide in that was preventing the drawer from going in quite right. Getting that blade out took about half an hour.
   - I had trouble stabilizing the deck-railing posts by the "drawbridge" (gate). In hindsight, I should have used double joists on that side of the deck, but I didn't. I spent a lot of time staring at the these two posts, wiggling them, adding blocking to the joists below, and so on.
   - There was no adequate footing for a ladder on the backside of the fort, and there was a big tree that made putting a ladder in awkward. Tasks that took 10 minutes on the front of the fort (like nailing up a facia board under the roof) took an hour on the back.

I think these problems are EXACTLY like the kinds of problems that show up unexpectedly on software projects -- two new tools you buy don't interface with each other like they're supposed to, and you have to figure out why. Or you install a new tool and suddenly your code doesn't compile anymore. Or you have a module that keeps producing errors because the design isn't quite right; you think you can't justify completely redesigning and rewriting it, but you end up nickling and diming your way to a higher cost than if you had bitten the bullet and redesigned and rewritten it.

2. Underestimation of unfamiliar tasks. My estimates weren't too far off for a lot of the work that I'd done before. But some things, like mapping out the site for the footing holes, I assumed would be 15-30 minute task ended taking several hours.

3. Not decomposing big tasks into smaller subtasks. I'd planned out my project in whole days. At a birds eye view nothing seems obviously wrong with planning "frame the fort in one day." But when you break it down and say, What's involved in each of the 4 walls, and then you realize that one wall includes a door, another wall includes an angled top plate, a third wall includes an angled top plate and a window, and so on, and then you think about what's involved in each one (measuring, cutting, checking for square, recutting anything that wasn't quite right, tilting the wall up, checking again for plumb and square, attaching and then removing the temporary supports, etc.), you start thinking, can I really do a whole wall in 2 hours? If the answer's even close to "no," then you start to realize that the whole estimate for that big task is probably wrong.

4. Substituting a target for an estimate. I had 7 days to do the project, and my estimate turned out to be 7 days. That's a little suspicious, and I should have known better than to make that particular mistake!

5. Sweeping numerous little tasks under the estimation rug. There were lots of tasks that I knew needed to be done, but I didn't want to admit that they were going to affect my schedule, and so I tried not to think about them or I minimized their impact (I realized this later). These tasks ranged from clearing brush from the site, to painting the trim, to installing the door knob. I think this is strictly similar to software estimates in which people just don't want to acknowledge that data conversion takes time, installing new tools takes time, converging each release takes time, etc., etc.

6. Never creating a real estimate. The fact of the matter is that I carried around a rough plan in my head for weeks, but I never actually committed a schedule to paper, and I never even considered creating a detailed estimate for the project. Of course the likelihood of making the other estimation mistakes I mentioned is higher when you don't officially create an estimate!

7. All's Well That Ends Well. My kids love their fort, and I had a great time building it. "All's well that ends well" is one reason that companies don't improve their software practices more often than they do. If people like the software that the team produced, and the software is successful, then that reduces the incentive to try to do better next time.

Some Differences

There were a few differences between my fort experience and a typical commercial software project:

   - There was no way I was going to compromise quality for the sake of schedule. I couldn't build something that would be hazardous to my kids or their friends. So the schedule was going to be whatever it was going to be -- it was clearly a secondary priority. We don't see many companies in which quality trumps schedule to the degree it did on this project.
   - My schedule overrun was free. My extra time was essentially free on this project -- maybe even a bonus since I was enjoying the project. So my overrun didn't imply a cost penalty. On a software project, you'd be paying for extra staff time, and so you'd have a cost overrun along with the schedule overrun.
The estimation error didn't really matter, because I was going to do the project regardless of what the estimate turned out to be. If I had created a real estimate and had learned that the project was going to take 100 hours instead of 50 hours, I would still have done the project. It's much harder to justify not estimating and then flying blind for a business project, especially in the era of Sarbanes Oxley.

What Do You Think?

Are there other lessons I should have learned? Are these the right lessons? Are these parallels between fort building and software estimation valid at all? Let me know what you think.

Published Sep 23 2007, 10:31 AM by Steve McConnell Filed under: estimation, Project Management

Comments

Patrik said:
Wow. Really good work. I'm impressed.
September 23, 2007 4:18 PM

Alan Stevens said:
Steve,
I think the conclusions you draw are valid. Unfortunately, most of us will move on to the next fort... er, project and experience a repeat of the same poor project planning.
Very cool fort, BTW.
++Alan
September 23, 2007 4:36 PM

Dan said:
I can't believe you cut down Blackberries!
September 23, 2007 4:56 PM

Paul said:
This was a very detailed article and process you described; in your younger years did you have a parent or role model with the same attention to detail, or perhaps opposite, that influenced you to be this way? On the point of sacrificing quality for schedule, this to me cuts directly to the difference between someone with a software engineering and detailed process mindset, to an individual that has a more brute force approach to plowing through problems, such as: more sloppy cleanup of the oil problem and less proper disposal, disregard for the jamming of any blades and thus towards prepping the tool for proper use next time, misuse of drill bits to the point where they are all dull and overall less effective (not mentioned in your article)
I'm not really speaking about the safety point and overall robustness of the project, but rather the perhaps excessively high degree of quality aimed at here. Most people doing such a project, particularly without a computing mindset behind them, will have standards tweaked to what they consider to be a reasonable level.
September 23, 2007 5:38 PM

Steve McConnell said:
Paul, Keep in mind that I don't apply the same quality standards across the board. I was really careful on the footings, posts, and joists because I didn't want the fort to fall down and hurt someone. I was way less careful on things like painting, which is a case of, Who cares? It's just a kids' fort. So I don't think it was an excessively high degree of quality.

I don't think this orientation is the result of a parent or outside influence. I think it's just that I've done enough of these projects that I know that many of the apparent short cuts really end up taking longer. Other shortcuts (like doing a sloppy job on the painting) really can be short cuts as long as you can live with the results.
September 23, 2007 7:28 PM

Reg Braithwaite said:
You say this is pretty typical of a software development project. I'd say it's typical of a software development project *where the requirements remain fixed*.

I shudder to think of what would happen had you set out to build a fort but finished with a structure is a play house, a guest house, and has storage for garden implements.

All designed on the fly as you build it :-}
p.s. Many thanks for your terrific writing.

September 24, 2007 5:44 AM

Pawel Brodzinski said:
I find one more difference - in the Fort Project you were both the client and the vendor and the whole project was set up in your head. It doesn't happen in the real life.

Yet still, most of the conclusions are valid. There's one which I don't quite agree. I don't think the main reason for not improving our practices are outcomes which are acceptably good. You can find a lot of projects, which can't be qualified to "all's well that ends well" category and still organizations do nothing to improve their performance. If I had to guess why is so I'd point a number of reasons including lack of knowledge (but there's laziness behind), lack of will to change, no focus on quality (quite often one), wrong people in charge or lack of flexibility.

When you're aware what good enough outcomes mean in your example you probably know you need to work on your practices and you don't suit to any of above categories.

September 24, 2007 6:28 AM

Peter Schwer said:
I love the construction analogy. I've been watching a home-improvement TV show that does post-mortem analysis of failed projects. The criticism I see in that show can apply almost directly to the sort of problems in software construction. Their approach towards dealing with old/legacy construction is also very instructive.

September 24, 2007 7:09 AM

Pedro Rabinovitch said:
Great post! But out of curiosity... How *did* you get the laser level piece from the footing? ;)

Cheers!

September 24, 2007 8:28 AM

Steve McConnell said:

> Great post! But out of curiosity... How *did* you get the
> laser level piece from the footing? ;)

Funny question, because I don't remember! I *think* I used a pry bar with a crook in the end to go under the piece and scoop it up. I have a nagging feeling that I really used some kind of garden tool, but I can't think of any tool I have that would have worked. Funny that I remember the problem but not the solution!

September 24, 2007 8:48 AM

Andy W said:

Great article Steve. I'll be planning my deck in the spring, so I'll have to remember to estimate properly, so that my wife doesn't get any wrong ideas about timelines ;)

Also, I would say you were not the client, your kids and their friends were, and I guess we'll have to take your word for it that they loved it ;). But, since it has a drawbridge, I couldn't imagine who wouldn't love it.

September 24, 2007 10:44 AM

Steve McConnell said:

Andy, good point about my kids being the client, not me. And also good point about them loving it because it has a drawbridge. You could refer to the drawbridge as one of the "bells and whistles" of the fort. There's also a trap door in the decking, which they also love.

Those have to be their two favorite features of the fort, and together the time it took to make them was <5% of the total effort of the project. Just like (some) software clients, they don't appreciate the fact that the decking won't collapse under its own weight; they take that for granted or aren't even aware of the possibility that it could collapse, whereas a substantial portion of my time went into ensuring that wouldn't happen. I think software projects are like that too. We spend a lot of time on infrastructure that clients aren't necessarily even aware of, and they don't necessarily perceive value in that infrastructure even when it's necessary to support other things they do perceive value in.

Another parallel might be the fact that I should probably allocate a few more % effort to adding more bells and whistles to the fort to further increase my kids' enjoyment of it. Similarly, sometimes we can take an overly utilitarian view of the software that we write, and we would benefit from focusing a little more on the aesthetic aspects and the bells and whistles.

September 24, 2007 11:07 AM

Glen said:

Very good article and good job in drawing parallels between software and building (hardware? ;-) ) construction.

I think you possibly missed a point where you could have taken a shortcut because you were overbuilding another area. Since you bumped up the
number of posts and beams, you probably would have been safe placing the posts in more convenient locations rather than spacing them "perfectly". The frame still would have ably distributed the load.

It also sounds like you made a few mistakes because you were tired. I would venture a guess that the laser level 'exercise' was at the end of a long day. I have seen many a developer's work undone by being tired. Oh wait, I think you call that out in one of your books, too. :-) And a tool tip for you -- sharpen the ends of the clamshell digger. It makes cutting through roots a lot easier. I normally use my dremel to put an edge on the tool, but a file would work just as easily.

Kudos for not sacrificing "core" quality -- framing -- and finding places where you could optimize -- painting. Did you consider enlisting the kids to help with the painting or other small tasks like cleaning?

September 24, 2007 11:23 AM

Eric Gunnerson said:

I've done a bunch of this sort of project, and the parallels to software are obvious.

What I find amusing is that people in software often say, "we should be more like 'real engineers', because when you build a house/bridge/dam/whatever, you know exactly how long it will take". And if you talk to those people, they run into exactly the same problems as software projects do.

As a future tip, it's far easier to do post/beam sorts of foundations if you preset all the beams in the locations where you want them (near to the ground, not up in the air if they're elevated) using temporary supports on the ends. That will tell you *exactly* where to put your footings, and when you go to pour, you can use the nice 'embed in concrete' saddle connectors hooked to short posts hooked directly to the beam, which you can ensure is dead level and exactly where you want it.

When the concrete sets, pull of the temporary posts, put in the real ones, set the beam on top, and it's precisely where you want it.

And no chance of the post being in the wrong place.

September 24, 2007 11:35 AM

Steve McConnell said:

Glen, I'll remember the clamshell digger tip. The laser level thing wasn't when I was tired, although I've certainly made those kinds of "tired mistakes" before. I did actually use the approach you suggested on the middle posts because of my overbuilding in other areas. The middle posts are placed on concrete pier blocks, which in turn are resting on poured footings. Whereas all the end posts are tied directly to the poured footings.

I was able to enlist my kids on some of the tasks. My daughter painted a lot of the trim boards before I installed them, and I learned that she is really, really fast at that and still does a good job.

September 24, 2007 12:07 PM

Steve McConnell said:

Eric, I agree that people often have an oversimplified view of other engineering disciplines when they make comparisons to software. I talked about that back in June in my post about "Rumors of Software Engineering's Death are Greatly Exaggerated" (blogs.construx.com/.../software-engineering-ignorance-part-ii.aspx).

Thanks for the tip on the posts!

September 24, 2007 12:09 PM

JK said:

re: Utilitarian vs. Bells and Whistles...

The Six Sigma discipline has terminology for this. (As do all of the others, I'm sure.) The expected features, those things that are taken for granted, are known as "satisfiers." They are expected; if you don't have those, you don't have a product. (Imagine a source code control system without a diff utility!) In your fort project, these would be things like a solid foundation, a roof that doesn't collapse, etc. You don't get to brag about them because they *are* the project. The trapdoor and the drawbridge, on the other hand, are "delighters," and are those things that make your product more interesting than your competitors. Sometimes a little time dedicated to those delighters makes all the difference in client acceptance. Never to the detriment of the satisfiers, though.

BTW, Code Complete is *still* my favorite software book. Thanks for it.

September 24, 2007 12:09 PM

Tim said:

Were the estimates actual time, or just time spent on the project? You mention not factoring in time to drink and rest, but what about phone calls and all the other interruptions? I've given project estimates, only to be called on them because I did not factor in being pulled off to deal with client issues, questions from coworkers, etc.

The perfect example of this comes from hiring a contractor last year to do deck cover around the back of my house. He gave me an estimate of two weeks. He started in March, and was finished at the end of May. Weather, problems getting material delivered, and conflicts with other projects all contributed to the delay. The finished product is excellent, but his estimates were way off.
Steve McConnell said:

Tim, a point I think I didn't make clearly enough is that I never really created an estimate for the fort that I would call a real "estimate" if I were talking about a software project. I just had a number I had been carrying around in my head that wasn't based on a whole lot of *anything.* So I didn't include details like time for phone calls, but that estimation mistake is masked by other, much larger estimation mistakes!

September 24, 2007 1:22 PM

Chris Woodill said:

Love the article...one very geek note - when I read the sentence "while clearing some blackberries" I thought you were talking about wiping memory from your cell phone!

September 24, 2007 1:49 PM

Aaron Korver said:

Ok, now you need to go rent Mr. Blandings Builds His Dream House

www.imdb.com/.../tt0040613

September 24, 2007 3:16 PM

Steve McConnell said:

Aaron, that movie looks funny. I'll add it to my queue. In the same vein, when I was a kid I knew a couple people who were building their own houses. And I don't mean they were general contracting it. I mean they were out at the job site many evenings and weekends literally building their own house. After seeing how much time it takes to build a 48 square foot *fort* with no electricity, no plumbing, no HVAC, no interior ceilings, no interior wall finishings of any kind, etc., etc., I just can't imagine how much time it would take to build a real house by yourself.

September 24, 2007 3:23 PM

starhawk laughingsun said:

Omg i laughed all the way thru this, I read Jeff Atwoods blog and hence found this. I'm currently a masonry contractor but educated as a programmer the mistakes you made in estimation are of course how shall i say it Noob mistakes. I've learned never underestimate how long it takes to do something and to account for even the smallest tasks. And btw I would hardly consider 16' 2x8 pressure treated joists heavy I've used 16' 2x10's for scaffold boards and deal with it daily. Altho I'm impressed the fort looks nice. Thanks for the amusement!

September 24, 2007 4:10 PM

Adam Cole said:

Delightful article. Thank you.

I recently was in the position where I had to “think in reverse”. I was given a small task and a large budget and I had to decompose the task to justify the budget. Normally we tend to estimate optimistically. In this case I was motivated to estimate pessimistically. I had to adopt a reverse mindset. I somewhat amazed myself at the number of little details I was able to quickly come up with. I remember thinking how I would normally never decompose to this level. (Hmmmm, maybe I’m on to something. Take your target, multiple by 4, plan, …then look for another job. ;-) 

September 24, 2007 8:06 PM

Jesper Holmberg said:

"I just can't imagine how much time it would take to build a real house by yourself"

The construction of my sister's house was PM'd by a family member. It took about 2x the estimated time.

It would have taken longer, but they were date driven and optimized in both ends - laid foundation before getting building permit; declared "feature complete" before bells'n'whistles such as roof tiles, bath tubs and most of the kitchen were in place.

The house is nice, tho. Now that it's at SP2.

September 24, 2007 8:39 PM

Nikola Gedelovski said:

Mr. McConnell--

A very interesting reading indeed. I wished our managers, when guilty of under-estimation, hauled cement bags and big wooden boards uphill.

September 25, 2007 1:19 AM
Rob Cooper said:

Excellent Post Steve.

I am a budding young developer and I really try to keep these sort of issues in the back of my mind, the way I see it is the sooner I get used to estimating my work as accurately as possible, the better! We have all been victim of bad estimates. My current project is about 2 months over schedule because a two-line spec was quoted at four months, and there is still more work to do! (wasn't my estimate!)

Very cool fort by the way, I am jealous!

Thanks for Posting!

September 25, 2007 2:18 AM

David Markle said:

Good post, Steve. I love your work.

I know you talked about *estimation* in this post, so this may be a bit OT, but it's important that readers draw the right lesson here. While there may be real parallels between fort building and software construction in the way you make your estimates, there's a big, big difference about what you actually can DO when the schedule starts to slip.

I think it's worth mentioning Brooks' Law here. With the fort, you could use the promise of free beer and a phonecall to your buds reminding them of the times you helped them move to add more post-diggers and concrete-luggers to your project. With more people, you'd be back onto your estimates.

But with software, the stakes for estimation are much higher. If your estimates are wrong, God help you if you try to add more bodies to the project -- there's probably no way for your project to truly recover from that schedule slip -- you'd end up with a system with less features, or one that took more time to build, or was of lesser quality, or all of the above.

September 25, 2007 10:30 AM

AlsoKidHouseBuilder Rob said:

Your craftsmanship and attention to detail shows. But, I have a different philosophy. I built a 6' x 6' interior clubhouse for my kids with shed roof and siding. It took me however long it took me. I just figured it out as I went along. It sits alongside the side of the garage in the yard at the back. It matches the house in siding, trim, and roofing, so it looks pretty good. The kids use it, but mostly for climbing up on the roof and hanging out, then jumping off. That is why it has 2x8 roofing beams spaced 16" on center. It ain't coming down.

My point? It is a kid's clubhouse. Did you spend a lot of time with your kids letting them figure out how to build it or help? Did you teach them about how to use a level? Do they have great memories of nailing on shingles, or deciding how big the windows are going to be, or figuring how to lay out the posts so it will be (kinda) square to the house? Did they like it because they had a say in how it came out?

My point is that you can be a great PM scheduler, or planner, but if you don't spend much of your effort figuring out how to include the sw engineers they won't really like working nearly as much, and they won't have affection for the finished product. They'll just look at what you built and think, "Well, I guess he really had a good time building it." And, they'll find it really handy in their teenage years when smoking doobage :)

I have done this sort of thing myself in the past a lot with my family or professionally. I'll come up with a great system, or product, or project, and the people I want to use it just kind of look at it, and say, "oh, thats nice." Then they mostly ignore my beautiful work and go on about their lives.

Feeling ornery today,

Rob

September 25, 2007 12:19 PM

Stan said:

Wonderful story. How soon did you know you were in trouble? What did you tell your "customer" kids?

September 25, 2007 12:28 PM

Travis said:

You could have cut some of your workload if you just spent $50 and went to Labor Ready, or found some friends to help you carry all this stuff. OR if you maybe had a truck or something you could just drive it up the hill if its accessible?

Point being...like in software development...construction requires good teamwork and the right equipment :)

September 25, 2007 1:07 PM

Chris Loosley said:

Steve,

I have always enjoyed your writing, and this post is up to your usual standard. Your analogy is a good one:

My kids are grown now, but I remember how my own small construction projects would often take much longer than initially expected, because of poor (or non-existent :) planning. I also lived through a home remodeling project that began in August 1986 with a promised completion date of
Thanksgiving. My architect acted as the general contractor, and this was his first big project. It was finally completed by Memorial Day, 1987.

On reflection, these events paralleled my work experience at IBM's Santa Teresa Lab (now the Silicon Valley Lab). When I joined the development team in 1978, the second product release of the new "Eagle" relational database project was scheduled for 1981. In actuality, what became DB2 V1 was finally released in 1983.

So now that I'm once again embarking on another major remodeling project, I'm much less optimistic, and I'm not even setting a firm target date for completing the work. But thanks for reminding me that I probably need to do even more detailed planning than I have so far!

September 25, 2007 3:04 PM

Vladimir Levin said:

I think this is a great article. However, I suspect that different people will have varying opinions on the conclusions to draw from it. My personal conclusion was that there wasn't necessarily anything terribly wrong with Steve's original plan. While decomposing the problem into smaller parts may have led to a better estimate, there's no guarantee of that. It's possible that it would have led to an enormous over-estimate also, as projects tend to balloon ever outward as we inspect them more closely - sort of like the coastline of Norway :) Sometimes it's better to actually start doing some work and see how that compares with a simple outline like the one Steve created.

The simple fact is that the best way to estimate something is to already have experience doing something very similar. Next time Steve makes a playhouse for his kids, he'll probably be closer to the target with his estimate, even without doing a more detailed plan!

I also think that agile/iterative planning techniques do help with this type of problem: As you start comparing reality against estimates you can see whether or not you're on track, and you can make decisions fairly early on: Abandon the project or reduce scope or extend deadline.

September 25, 2007 4:38 PM

Neil Barnwell said:

I love this article. I agree entirely - I'm in the process of renovating our kitchen, and I had no idea how much I would need assistance with, or how long it would take. I made a big mistake thinking I could complete the project in 8 weeks - I started over 4 months ago and it's still not finished. I'm glad to say I don't develop software this way (at least I mostly know what I'm doing) but it's a similar feeling to that I've had in the past delivering software. Outsourcing, anyone?

September 25, 2007 4:41 PM

Hk said:

Darn.

Another confirmation. The universe is just some imagination of me, and my real me is some psycho in an asylum, hallucinating.

This is *exactly* what I experience atm.

I dive a little bit into game development, and well, I am as critical as ever.

And suddenly, such things as "usability" and "GUI design" leap at me like aliens from the darkness. (I dont think of the media creation process. No. I wont. Its the mother alien. Its evil!). Creating some good looking, useable GUI without too much friction is something really beasty.

In fact, its quite amazing how big a little problem can grow, if you think about it. Its like some function thats log*(n) for all n < 100 (like, 0-2), and ack(n,n) (err.... big?) for all other n.

Slapping together some little boring space shooter can be done in several hours, yes. However, creating some civl-clone throws all kinds of evil things around. But heck, if its a huge problem, I can smash his knees easier...

September 25, 2007 4:43 PM

Susan said:

Draw bridge and trap door! Why aren't you my dad?!

September 25, 2007 6:43 PM

Kim said:

Thanks for the great article - and applying the construction metaphor to software development into practice.

It is interesting that the timing of this article is coinciding with a decision we made last week as a group of developers to read "Code Complete 2" and have weekly discussions until we are finished. Just yesterday I found this article and asked them to read it. Only later realizing that you had posted this email only 2 days prior!

September 26, 2007 10:45 AM

Mike Gruber said:

I had to laugh.

I built a 400 square foot deck last summer. I figured it would take about a week. It took 200 hours at least over the course of the whole summer. One problem is that we are engineers, and we do things *right*. Everybody says it's the nicest deck they've ever seen. It ought to be, it took me long enough.
Nice fort. :-)  
September 26, 2007 10:54 AM

Mike Ramm said:
This is a brilliant lesson of how many mistakes we can make when we are put in a situation where we are not too familiar with the nature of the problem even if we are very experienced as project managers.

Thank you, Steve!  
September 27, 2007 2:03 AM

Simon Parmenter said:
Fascinating. Could you provide an estimate on what effort and resources will be required to estimate another similar but much larger scale project?  
September 27, 2007 9:37 AM

Steve McConnell said:
Simon, If you're estimating a larger *similar* project (i.e., a construction project), I have to claim ignorance of how to do that. For larger projects I suspect that you'd need to know a lot more about the nature of the specific work being done, which I don't know much about.  
September 27, 2007 12:06 PM

Simon Parmenter said:
Steve. My career since 1997 has been in 'software engineering'. Previously, I had a 3 year stint as a planner and estimator for electrical installations and services for buildings. In this role I had an old 'rate book' that gave the hours for a particular task. Say 1 hour for so many metres of trunking of a given size to be erected. It was quite extensive. However, it was out of date in some areas due to the fact that equipment etc had evolved. Having more than a decade of experience of doing construction / maintenance myself I could use that to tweak the estimate. For larger projects, more people, more tasks, the effort required to estimate the larger project at the same level of detail as given above became prohibitive, especially for that resource named time. I wish I had access to today's planning software with that rate book's data.

The other problem I had on larger projects was a lack of detailed, reasoned experience and knowledge. This I solved by asking experienced people and accessing other appropriate resources. I gained useful experience and knowledge.

My experience in the IT 'profession' has been somewhat disappointing.

What is common to both, indeed all, jobs is the that estimation of the number of tasks to be performed for a given project is always significantly below that of what will be performed and significantly below what is required for a successful result. An estimation is tightly associated with a plan. Both are part of the definition of the 'problem' to be solved. In some sense then, problems are not adequately defined.

Is there a rate book for producing software? If so, is there a rate for writing a linked list?

I think I need to buy and read your book. I do have books on project management, quality and software metrics etc.

I still feel that most estimates in software are guesses not estimates.  
September 28, 2007 6:29 AM

Sean W said:
Wow, after reading through this it brings to mind my High School Programming final project.

It was a group project and me and two friends decided to make a RPG including a full GUI for it (it was kind of like a basic DND with a GUI). We had originally planned to make it in about 3 weeks, we had 5 total. Thinking that everything would link together, and that all our different programming styles would work smoothly together. Man were we wrong.

The project went down to the last minute, with the GUI being completely tossed all together. We realized afterwards that it wasn't our programming styles that limited us, rather our knowledge (we had only been programming for about 4 months at this point) and estimates were the problem, and they were getting to us. After it was done we had a huge sigh of relief and had a good time playing it.

Turns out the teacher did too and we got a 100. I also won the Programming Award that year too :P

Ever since doing that project (this was all done during last semester of last year's school year so it hasn't been THAT long!) I've been heavily working out a schedule for projects now, taking atleat a day to think them out fully, making sure that even the small parts get a little bit of extra time to be done.

Reading this also made me think of when my father built me a Fort in out backyard when I was just little. We loved that ting!  
September 28, 2007 8:55 AM

About Steve McConnell
Steve has worked in the desktop software industry since 1984 and has expertise in rapid development methodologies, project estimation, software construction practices, and third-party contract management. In 1998, readers of Software Development magazine named Steve one of the three most influential people in the software industry along with Bill Gates and Linus Torvalds. Steve was Editor in Chief of IEEE Software magazine from 1998-2002.

Steve is on the Panel of Experts that advises the Software Engineering Body of Knowledge (SWEBOK) project and was Chair of the IEEE Computer Society’s Professional Practices Committee. Steve earned a Bachelor’s degree from Whitman College and a Master’s degree in software engineering from Seattle University. Read more about Steve at www.stevemcconnell.com.