How we got rid of time reports

A story about Waste Elimination
Version 1.0, 2010-09-29

Henrik Kniberg

Intro

Have you ever dealt with time reports? Filled them in? Approved them? Shuffled them around?

Did it feel like well spent time?

Can you imagine a world without time reports?

Disclaimer 1: *Time reports are sometimes useful.*

Disclaimer 2: *But they often aren't.*

The Story

Once upon a time I started a new job as development manager at a software development company.

It was an interesting and exciting job, but one of my weekly annoyances was collecting and approving time reports. Dozens and dozens of time reports, physical papers being shuffled around the office. Consequently, one of the routines of the HR department was nagging me to get the time reports in on time, and consequently I had to nag my developers about this every week.

This was annoying. Not only because it took time and focus from everybody, but also because my “approval” was mostly guess-work. I had no effective way of knowing if a report was correct. Annoying, but not annoying enough for me to raise a fuss about at that moment - I was new and wanted to focus on other more pressing issues.

One of the most pressing issues was that people were working themselves to death. Many of the developers looked like pale ghosts to me, they were there in the office in the morning when I came, and still in the office in the evening when I left. One of the key developers even submitted a formal vacation request for the upcoming Saturday. I thought he had entered the wrong date, but it turned out that he had been working every weekend for the past few weeks and really wanted to go home at least one day next weekend. This Saturday vacation request was a funny but effective way of communicating a serious problem.

Why didn’t the time reporting system issue some kind of Impending Developer Burnout Alert? If it couldn’t even do something as simple and important as that, then what was the point? Imagine an airplane or nuclear power plant with a fancy monitoring system that logs all kinds of data but never actually alerts anyone when something is obviously wrong...

Fortunately I could see the burnout signs myself since I sat with the developers (I was glad I declined the personal office room that was offered to me at the beginning).

Anyway within a few weeks these and other pressing issues were more or less under control, so I decided to do something about these time reports. I had way too many direct reports at the time, but despite all the disadvantages of that there is one advantage: no space for wasteful routines.

I was not only annoyed about the time reports, I was curious. Why do we have them? Where do they come from and where do they go? So I started asking around.

My manager (the CEO) said that finance needed them. Finance said HR needed them. The HR manager said that the company that we had outsourced salary administration to needed them. And so on. It became my background project - whenever I had a few moments of slack I would continue my private investigation The Mystery of the Time Reports.

Finally I find out what happened to the papers. After a series of handovers, they ended up at a third party company, where each report was *manually typed* into an electronic HR management system!

Um...

DUUUUUUH!

Er, I mean “Hey, that doesn’t feel very efficient”.

Being a techie at heart, I immediately started thinking about how we could streamline the process. Each developer fills in their times online, their local team lead looks through each time report online and presses OK to approve them, then I look at a high level summary and press OK to accept that, then the whole thing beams off to the third party company. Wow, slick. Hmmm, maybe we could connect the system to Eclipse and have it automatically log time to the system depending on which development project was open on the computer? Maybe we could integrate directly with whatever system the HR company uses? This probably calls for a SOA....

Wait.

> *"there is nothing so useless as doing efficiently that which should not be done at all"*
> *- Peter Drucker*

I reminded myself that my quest was effectiveness, not efficiency.

- **Efficiency** = The ratio of the output to the input of any system.
  Example: Lines of Code per Hour.
- **Effectiveness** = Quality of being able to bring out an intended results.
  Example: ROI per release.

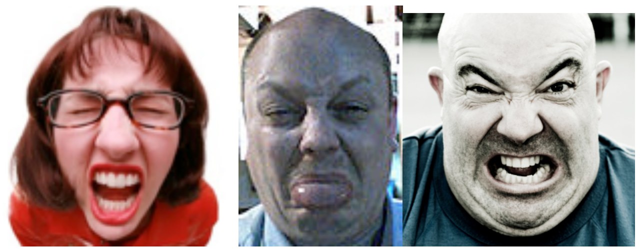
I still didn't know what the time reports were being *used for*. The answer I got was often something silly like "we have to do the time reports because it's our HR routine" (which is like “we have to because we have to”). People told me all kinds of stuff about the time reports, but what I wanted to know was what the numbers were actually *used for*.

So I changed my line of questioning. Instead of asking "Why are these time reports needed?" I started asking questions such as "What would happen if we ‘lost’ the time reports?" or "What would happen if we ‘accidentally’ randomly generated the contents of the time reports?".

Finally the real purpose of the time reports emerged. They were primarily used to figure out how much overtime compensation to pay. There were some other purposes too. But the other purposes were already covered by other reporting systems. So the only important purpose was the overtime stuff. Oh, and to calculate "flex" time, a Swedish industry tradition which basically allows people to come to work a bit late one day if they come to work early another day. The time reporting system was used to make sure it all adds up to at least 40 hours + X per week, where X was overtime.

Ah, the truth is sweet.

Now, armed with The Truth, I gathered a bunch of developers and asked them a rhetorical question - "how do you like filling in time reports?" The look on their faces was all the answer I needed.



I continued with something like this:

"We currently use time reports to manage over-time. But time reports suck, and over-time work sucks even more. We all know that one single hour of energized, motivated, focused development is worth more than a whole weekend of zombie-coding."

(I don't have any concrete references at the top of my head right now, but I've lost count of the number of studies I've read that find little or no correlation between the length of the work week and the amount of value produced - at least when it comes to creative work such as software development. It's all about motivation and focus, not hours).

"So here's the deal. No more time reports. And no more overtime. From now on Overtime means ‘Working when you don't want to’. If you are demotivated or tired, go home. If you want to stay late and code because you are in the zone, then stay. That’s not overtime. If you feel you've been working late the past few days, take a day off. If you've had a slow week, catch up on Saturday if you like. Find your rhythm, find the rhythm of your team."

"There may be exceptions from time to time. Situations where “traditional” compensated overtime work is warranted. We will minimize the need for that and handle them on a case-by-case basis. When in doubt, talk to me."

> *"manage for the normal and treat exceptions as exceptional "*
> *- Edwards Deming*

"Bottom line: I don't care exactly how many hours you work, as long as it is roughly corresponds to a full time job. I do care that you are focused, energized, and motivated when you do work. Can I trust you to handle this responsibly?"

Enthusiastic nodding around the room.

I mustered some courage, went to the CEO, and told him that I will not be delivering time reports any more. The dialog went roughly like this:

- Me: "We will no longer be delivering time reports"
- CEO: "Why?"
- Me: "Because they are a waste of time"
- CEO: "Won't HR be upset?"
- Me: "I'll give them what they need in a more efficient way."
- CEO: "OK, cool."

Whew, that was easier than I thought. I had prepared for all kinds of objections…
I liked that CEO :o)

Then I went to HR.

- Me: "We will no longer be delivering time reports"
- HR: "But you have to. That is the routine"
- Me: "The routine is being changed. I just talked to Mr CEO" (OK, a bit deceptive, I admit)
- HR: "How will we know how much salary to pay people?"
- Me: "You have everybody's employment contracts. Just pay their normal monthly salary."
- HR: "What about overtime?"
- Me: "There won't be much of that any more. When there is, I'll let you know. Assume 40 hours per week in the meantime."
- HR: "But we don't want to change our routines, can't you just keep doing the time reports?"
- Me: "I can have them auto-generated with 40 hours per week, and auto-emailed to you exactly on time every week. Would you like that?"
- HR: "Well, that feels rather silly."
- Me: "So I'll skip sending them. If you like, you can pretend you received a bunch of 40-hour time reports from me, and punch the numbers into your system like you usually do. Less fiddling with papers, and you'll never again have to nag me about late time reports."
- HR: "OK, fair enough, we'll give it a try."

Nobody ever mourned the time reports.

Moral of this story

1. Most organizations are riddled with policy-driven waste, i.e. wasteful practices, rules, and routines. Just keep your eyes open. Look at it positively - these are high-leverage opportunities for improvement.
2. Even the most wasteful routine usually has some reason behind it.
3. If you encounter resistance in removing the wasteful routine, expose the reason behind the waste. Find out which need is supposedly being fulfilled. Offer a less wasteful way of fulfilling that need.
4. Be persistent; deeply entrenched waste takes effort to dislodge.
5. Don't mock or blame the predecessors that put the wasteful routine in place. Maybe it was a good solution to a real problem in the past. Or maybe not. But it doesn't matter, focus on the future.
   “Things are the way they are because they got that way” - Jerry Weinberg
6. Don’t confuse *effectiveness* with *efficiency*. Focus on *effectiveness* first, then *efficiency*.
7. An hour is a unit of effort. Not a unit of value.

FAQ

So with most of the overtime gone, didn't this add up to less stuff getting done?

Well, it did result in less code getting written. Less Bad Code. Less buggy code. Less junk. And therefore, less time hunting down bugs, responding to panicky support calls when the system crashed, maintaining bug lists, releasing hot-fixes, and deciphering messy code.

And therefore, more time to write Good Code.

We all saw a clear & significant improvement in productivity after removing the overtime. Productivity in terms of customer value delivered over time.

Didn't you still need to track time sometimes?

Oh, sometimes there was a valid need to know how much time was spent on some project, for example if there was an external client that needed to be billed. That was easy enough to do without time reports. If one 8-person team spent 3 sprints working on project X, and each sprint was 2 weeks, that's 8 people working roughly full time for 3x2 weeks. So 40 hours x 6 weeks x 8 people = 1920 hours. Oh, maybe they had to spend some of their time maintaining product Y at the same time. In that case I'd ask the team at each retrospective to estimate roughly how their time was allocated between X and Y this sprint, for example 80%/20%. Easy enough to incorporate into the calculation.

Some teams do this on a daily basis - before each team member goes home, they put a mark on a chart on the whiteboard, indicating the rough allocation of their day across different projects. This gives even higher fidelity, while still at low cost.

However I'd be careful there. Multitasking is usually a horrible idea. If we have to create an advanced system to track exactly how people allocate their time across multiple projects then we are definitely focusing on the wrong thing - we should fix the problem (doing too many projects in parallel) rather than the symptom (having trouble keeping track of the time).

Suppose I spent Monday working on four different projects, 2 hours on each, 8 hours in total. I may log 2 hours on each project in the time reporting system, but with all that context switching the *effective* time was probably closer to 30-60 minutes per project. So most of that day was waste.

A tracking system that hides waste *is* waste.

If everyone works on one project at a time, then time tracking is trivial.

> *"there is nothing so useless as doing efficiently that which ..."*
> oh wait, I already mentioned that one. How about this one instead:
>
> *"Time reports are often a waste of it"*
> *- Henrik Kniberg*

Didn't some of the developers miss earning tons of money doing paid overtime?

I raised the salaries to a level where overtime compensation was no longer a driving force. A significant raise in many cases. Nobody complained about this, nobody asked to go back to the old model :o)

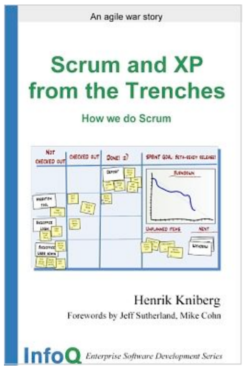
Research shows that cash incentives make performance worse, not better, when it comes to creative and complex work such as software development. Instead, pay people enough to take money off their minds. Here's a short & awesome presentation on this, a must-see:

- Drive: The surprising truth about what motivates us

Paying people to work overtime was the most silly thing of all, that just gave us burned-out developers and technical debt.

What happened afterwards?

I wrote a book about that company: "Scrum and XP from the Trenches". Much to my surprise it became a smash hit within the software development community (I wrote it mostly for my own sake, to get the thoughts out of my head) & I've since lost count of all the people around the world that have told me that the book inspired them to change the way they manage their software development organization. Feels great, thanks for the feedback!


An agile war story
Scrum and XP from the Trenches
How we do Scrum
Henrik Kniberg
Forewords by Jeff Sutherland, Mike Cohn
InfoQ Enterprise Software Development Series

This article is just another side of that story. The important thing is that the stuff we did would never have been possible if we hadn't aggressively and continuously removed waste whenever encountered (time reports were just one example). Hmmm... why did the word "Lean" just pop into my head? :o)

That was 2006. I haven't touched or filled in any kind of hour-based time sheet since then - even though I've spent the past 4 years consulting. If a prospective client brings up the topic I inquire about their real needs and offer better alternatives.