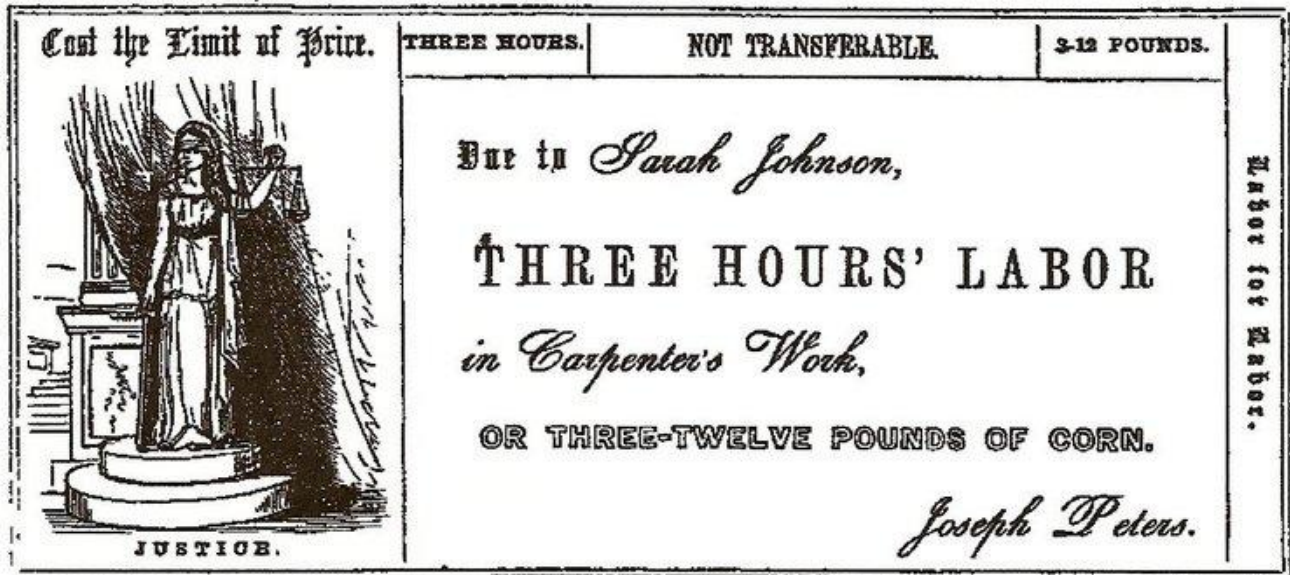


# Cincinnati Test Store

10 September 2012 · agile, clock, questioning, testing



Sample Labor Note

Monday 3rd September 1827, A man steps off the road at the corner of Fifth and Elm, and walks into a store. He's frequented the store a few times since it opened, and he's starting to get to know the owner and his range of merchandise. In fact, like many of people in town he's becoming a regular customer.

He steps up to the counter, both he and the store owner glance at the large clock hanging on the wall and nod in unison. The shop-keeper makes a note of the time, the two then begin a rapid discussion of requirements and how the shop keeper might be able to help. When they've agreed what's needed, the shop keeper prepares the various items, bringing them to the counter, weighed, measured and packaged ready for transport to the customers nearby holding.

The store keeper then presents the bill to the customer, who glances at the clock again, and the prices listed on each of the items arranged around the store's shelves and then pays. The customer smiles as he loads the goods onto his horse, happy that he's gotten a good deal and yet been able to talk over his needs with the store keeper - for the items he knew least about. He also appreciated how his purchases were packed securely. As he was travelling back home that day, the extra cost of packing the goods was worth it given the rough ride they'd likely take on the journey.

The store was the Cincinnati Time Store ([http://en.wikipedia.org/wiki/Cincinnati\\_Time\\_Store](http://en.wikipedia.org/wiki/Cincinnati_Time_Store)), and the shop keeper was Josiah Warren. The store was novel, in that it charged customers for the base 'cost' of the items on sale plus a cost for the labour-time involved in getting the item to and serving the customer. The store-keeper might also charge a higher rate for work he considered harder. The store was able to undercut other local stores, and increase the amount of business he was able to transact.

Imagine if software testing was bought and sold in this manner. Many successful software testers here in London are contractors, and already work for short contracts as and when is agreeable to both companies. But even then, the time is usually agreed upfront ie: 3 months. Imagine if that time was available on demand, per hour?

What drivers would this put onto our work? and that of other team members?

You might want constant involvement from your testers, in which case the costs are fairly predictable. But remember, you are paying by the hour, you can stop paying for the testing at the end of each hour. Would you keep the tester being paid for the whole day? week? sprint? even if they were not finding any useful information? If you found that pairing your testers with programmers during full-time was not helping, you can save some money from the pure-programming parts of your plan. Conversely your tester would be motivated to show they could pair and be productive - if they wanted to diversify their skills.

As the tester, I'm now financially motivated to keep finding new information. To keep those questions, success stories, bug reports coming. I'm only as good as my last report. If the product owner thinks she's heard enough and wants to ship - then she can stop the costs any-time, and ship.

The team might also want to hire a couple of testers, rather than just one. The testers might then be directly competing for 'renewal' at the end of the hour. I might advertise myself as a fast tester (or rapid tester (<http://www.developsense.com/courses.html>)) and sell my hours at a higher rate. I might do this because I've learned that my customer cares more for timeliness than cost per hour. For example the opportunity cost of not shipping the product 'soon' might be far greater than the cost of the team members. I'd then be motivated to deliver information quicker and more usefully than my cheaper-slower counterpart. My higher rate could help me earn the same income in less time and help the team deliver more sooner.

Has your team been bitten by test automation systems that took weeks or longer to 'arrive'? and maybe then didn't quite do what you needed? or were flaky? If you were being paid by the hour, you would want to deliver the test automation or more usefully the results it provides in a more timely manner. You'd be immediately financially motivated to deliver actual-test-results, information or bug reports, incrementally from your test automation. If you delivered automation that didn't help, didn't help you provide more and better information each hour, how would you justify that premium hourly rate? What's more agile than breaking

my test automation development work into a continuous stream of value adding deliverables ? that will constantly be helping us test better and quicker?

Paying for testing by the hour would not necessarily lead to the unfortunate consequences people imagine when competition is used in the workplace. My fellow tester and I could split the work, maximising our ability to do the best testing we can. If my skills were better suited to testing the applications Java & Unix back-end I'd spend my hour there. Mean-while my colleague uses their expertise in GUI testing and usability to locate investigate an array of front end issues.

Unfortunately a tester might also be motivated to drag out testing and drip feed information back to the team. That's a risk. But a second or third tester in the team could help provide a competitive incentive. Especially if those fellow testers were providing better feedback, earlier. Why keep paying Mr SlowNSteady when when Miss BigNewsFirst has found the major issues after a couple of hours work?

I might also be tempted to turn every meeting into a job justification speech. Product Owners would need to monitor whether this was getting out of hand - and becoming more than just sharing information.

I'm not suggesting this as a panacea for all the ills of software development or even testing in particular. What this kind of thinking does is let you examine what the companies that hire testers - want from testers. What are the customers willing to pay for? What are they willing to pay more for? From my experience, in recent contexts, customers want good information about their new software and they want it quickly - so the system can be either fixed and/or released quickly.