

Manumation, the worst best practice.

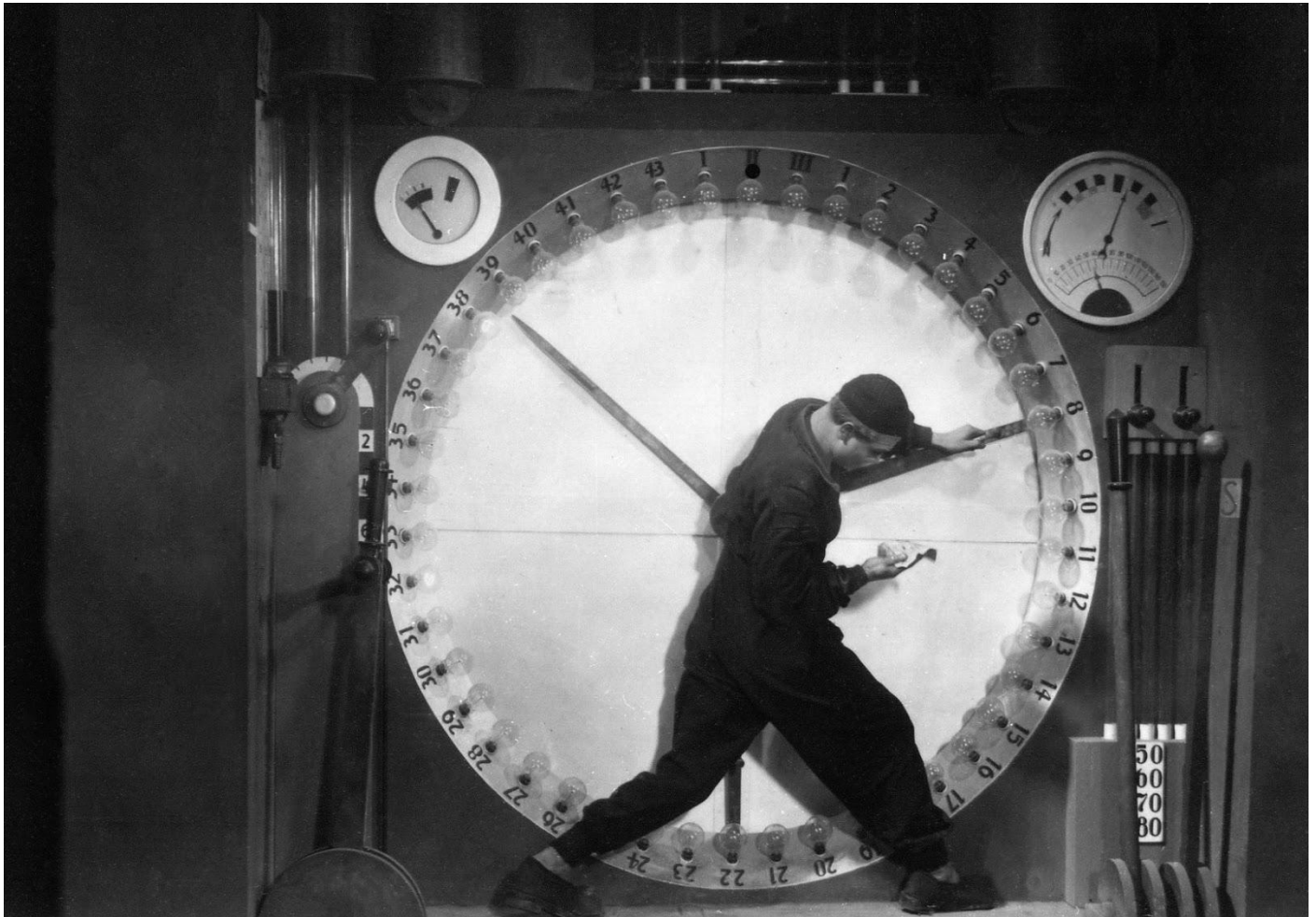
13 February 2018 · automation, critical thinking, skeptic

There is a pattern I see with many clients, often enough that I sought out a word to describe it: Manumation, A sort of well-meaning automation that usually requires frequent, extensive and expensive intervention to keep it 'working'.

You have probably seen it, the build server that needs a prod and a restart 'when things get a bit busy'. Or a deployment tool that, 'gets confused' and a 'test suite' that just needs another run or three.

The cause can be any number of the usual suspects - a corporate standard tool warped 5 ways to make it fit what your team needs. A one-off script 'that manager' decided was an investment and needed to be re-used... A well-intended attempt to 'automate all the things' that achieved the opposite.

They result in a manually intensive - automated process, where your team is like a character in the movie Metropolis, fighting with levers all day, just to keep the lights on upstairs. Manual-automation, manumation.



Metropolis (1927)

The answer is to use... Sorry, I don't have a magic tool, with a funky name, light-hearted README and a 3 word install command.

When presented with this situation I advise people to take a step back. Think about what they want for their team, and themselves. We want to deliver faster and easier right?

Often removing a tool, can save time and work. Could that complicated deployment system be replaced with a 5 line bash script?

Could that test suite, focus on the sort of techniques computers are good at? (reading and writing to APIs at speed, randomisation, graphing, comparison/diffing of complicated documents etc.) Many teams fall into the trap of 'look no hands' as if they are trying to spin plates rather than build quality software, fast.

Sounds simple, but when people are wedded to a tool or an idea of how things 'should be' because it was in that book - by that guy "its a best practice!" then it can be difficult to get things simplified.



Step back before things become comical. Safety Last! (1923)

But simplification is often the easiest and quickest place to start. It rarely makes sense to mix in more, to something that is already a muddle. If your team is already manually keeping the 'automation' going, then letting someone do the process manually - for a few hours will probably help you figure out what can be reliably automated, and what are the sticky complicated bits - people are quicker at doing.

Software testing is often overtaken by the above sort of [broken] tools feeding frenzy. The look no hands evangelist, may have had great ideas, but did that suite of tests really make your life easier? Did it free up time for finding the important bugs? Or are you now finding the real bugs in the test automation, while the software your product owner is paying for is hobbling along slowly and expensively to production?