Nicolas Lescano

Review of HIT or Miss Chapter 8

Alert fatigue is what this example illustrates with excruciating clarity. Clinician brain-time is a finite resource for a potentially infinite load of tasks. EHR systems can either reduce or increase this load, and the net effect of any HIT deployment depends not on its theoretical intentions, but the downstream consequences of its implementation in practice. Within a fixed capacity system, such as the neurocognitive and executive functions of a human brain, tasks will inevitably compete with each other for limited resources like attention span and working memory. Alerts that interrupt a mental workflow have the remarkable ability to halt it and redirect attention to something that would otherwise have been missed. But this potentially useful disruptive effect can very easily turn into something else. Make the stimulus too ubiquitous and it will become background noise, leading to what we know as alert fatigue. But “fatigue” is only the tip of the iceberg, as dumbclicking through alerts to be able to keep the flow going is only one of the predictable outcomes. Harder to predict behaviors are also likely, including some that not only beat the purpose of the alerts, but create all sorts of new problems that may now have to be patched and troubleshot blindly.

The example demonstrates how costly it can be to adopt a reactive approach upon the possibility of alert fatigue, only making adjustments once the ship is at the brink of sinking. Alert fatigue is not just something that “could” and “hopefully won’t” happen. It is bound to happen with certainty. What matters is to what extent, and: is it worth it? I think a painfully detailed cost-benefit predictive analysis should be part of the minimum requirement not just for CDS alerts, but any addition of cognitive tasks for EHR users. I believe that a piloting phase with a representative sample of the expected users of the HIT is not an unreasonable requirement. Furthermore, if the cost-benefit ratio is not immediately obvious as favorable, it might be wise to go through a set of progressive iterations of feedback and repiloting until it becomes evident that it is worth moving forward with it. Is this a lot of extra work? Absolutely, but not even close to as much as the downstream cost of a rushed implementation like the exemplified one. After a solid pre-implementation evaluation, the post-implementation recalibration will likely be much more effortless and productive.