



Evidence Based Scheduling

JoelSpolsky developed a planning and estimation method called EvidenceBasedScheduling:

- <http://www.joelonsoftware.com/items/2007/10/26.html>

He doesn't give any indication that it owes anything to [ExtremeProgramming's PlanningGame](#), but it works in a very similar way, breaking the work down into tasks which take up to two working days, estimating how long each would ideally take, then forecasting how a set of tasks will actually take based on objectively-measured past experience. It even uses the term 'velocity'. It certainly looks a lot like a cheap knockoff to the untrained eye.

However, there are some differences:

- The method produces not just an forecast of how long some set of tasks will take, but a probability distribution - you can say you have 80% confidence they will be done in fifteen days, 95% that they will be done in twenty, 99% that they will be done in twenty-six, etc.
- It tracks velocity at a task level, rather than a whole team level.
- It turns estimates and velocities into a forecast using a MonteCarlo simulation.
- Only the developer responsible for a task is allowed to estimate it, and this responsibility must be known at plan-time; the method thus assumes a kind of [CodeOwnership](#), or at least TaskOwnership, that is alien to XP.

Only the developer responsible for a task is allowed to estimate it

Although alien to XP, this is very practical advice in other methodologies. It is a sucky feeling having to live up to someone else's estimate, or having to provide an estimate for someone else. The two-hour knockoff task for you may get someone else who hasn't had your background and experience stuck for days or weeks.

This kind of thing has made me wonder whether EstimateBidding could work. In a [PairProgramming](#) situation, you'd have every engineer bid on new tasks, and for each task assign the lowest and highest bidder. Hopefully this would provide 1) the best fit of expertise to the problem so it is retired quickly and 2) the best cross-training benefit from the pairing.

[CategoryPlanning](#)

Last edit June 7, 2009, See [github](#) about remodeling.