

Some anti-patterns of agile software development

Sriram Narayan

ThoughtWorks

Velocity as target

- stages within an iteration
- each stage is inventory for the next
- minimize build up of inventory
- velocity is a measure of end to end flow
- purpose of velocity is to minimize waste
- difficult to achieve this without on-site customer

Points estimation under fixed time and scope

- estimating in points makes sense when scope is negotiable
- If budget, schedule, quality and scope are fixed at the outset, then estimation becomes commitment
- commitment is better served by real units of time

Not failing fast enough

- Fail fast is enabled by tight feedback loops
- Feedback must be of good quality
- Best feedback is tested functionality
- Iterative versus incremental
- But what about costs?

Risk adjusted costs

Approach	Cost at zero risk	Total	Estimated Risk	Risk adjusted cost	Total
Single attempt	100	100	40%	167	167
Iterative					
i1	30		30%	43	
i2	50		20%	63	
i3	40	120	10%	44	150

Cost-ineffective pair programming

- For a project that runs for, say six months or more, there should no extra development cost on account of pair programming.
- Pairing pays off in several intangible, hard-to-measure ways.
- How to see if there is extra cost?

Questions? Comments?

sriram.narayan@thoughtworks.com