

Estimation : Stop guesstimating and start sizing to reach predictability

Leo Mattos, Agile Delivery Operations CoE, SAP



@mattos_leonardo



Leonardo.mattos@gmail.com



MESSAGE FROM LEO MATTOS



Technical Coach

IT Services
Principal Consultant



Developer

20+ years
In the industry



Soccer coach

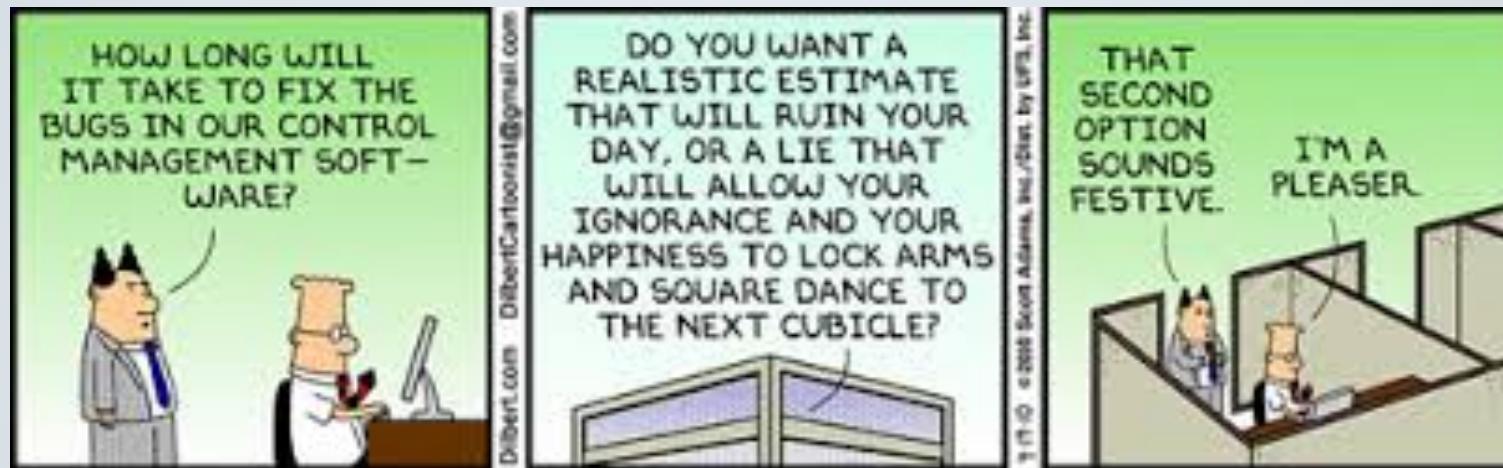
Licensed for kids
U10 - U16



Adventurer

Meyers Briggs
abbreviated test

Story estimation



Estimation can be a difficult beast to deal with; more so on an agile project. How do you estimate when you don't have a list of requirements that is complete or signed-off by the customer? Or a nailed-down schedule? What should your currency of estimation be? How do you estimate on distributed teams? Is it worth estimating at all?

Why do we estimate?



Why do we estimate?



Estimation is valuable
when it helps you
**make a significant
decision**

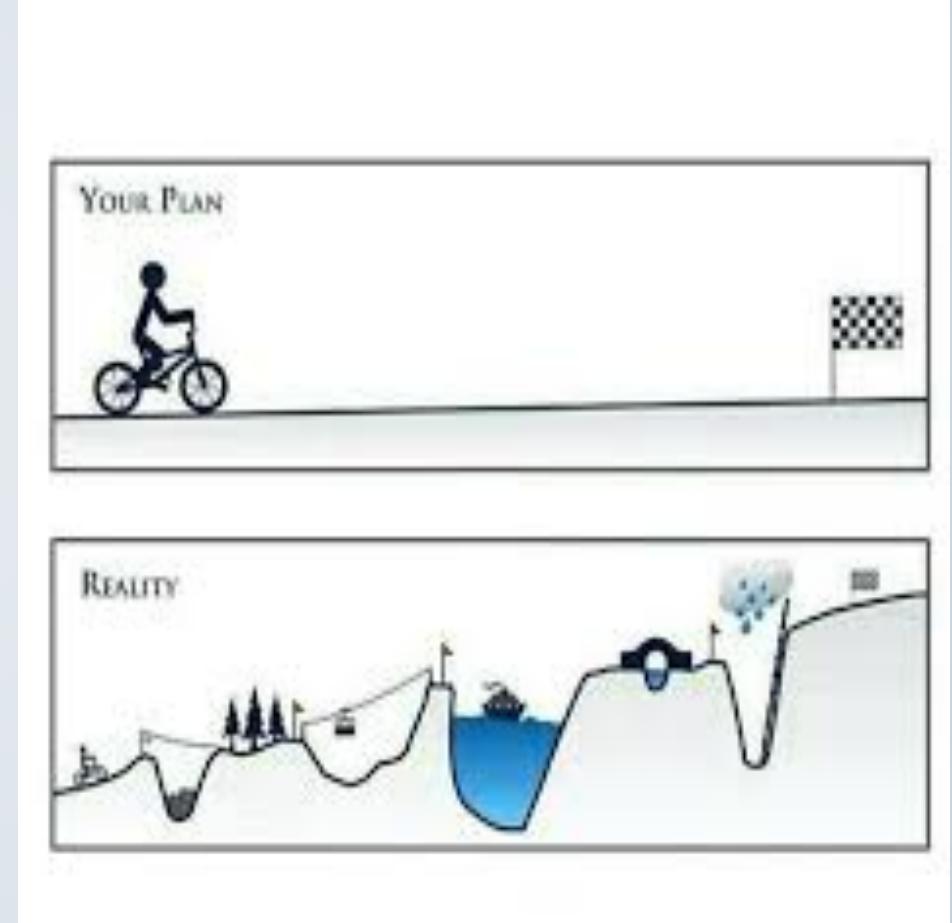
* As baseline to help assess changes

What are the
decisions you
think estimation
can help with on
your space?



Estimation on Agile

Estimation is hard. For software developers, it's among the **most difficult – if not the most difficult – aspects of the job**. It must take into account a **slew of factors that help product owners make decisions** that affect the entire team – and the business. With all that at stake, it's no wonder everyone from developers to upper management is prone to getting their undies in a bunch about it. But that's a mistake. **Agile estimation is just that: an estimate. Not a blood-oath**



Side effects



If you get “too good” on estimating,
you might stop doing it.

How do you estimate?



Why not hours?



Emotional attachment



Political



Not accurate anyway



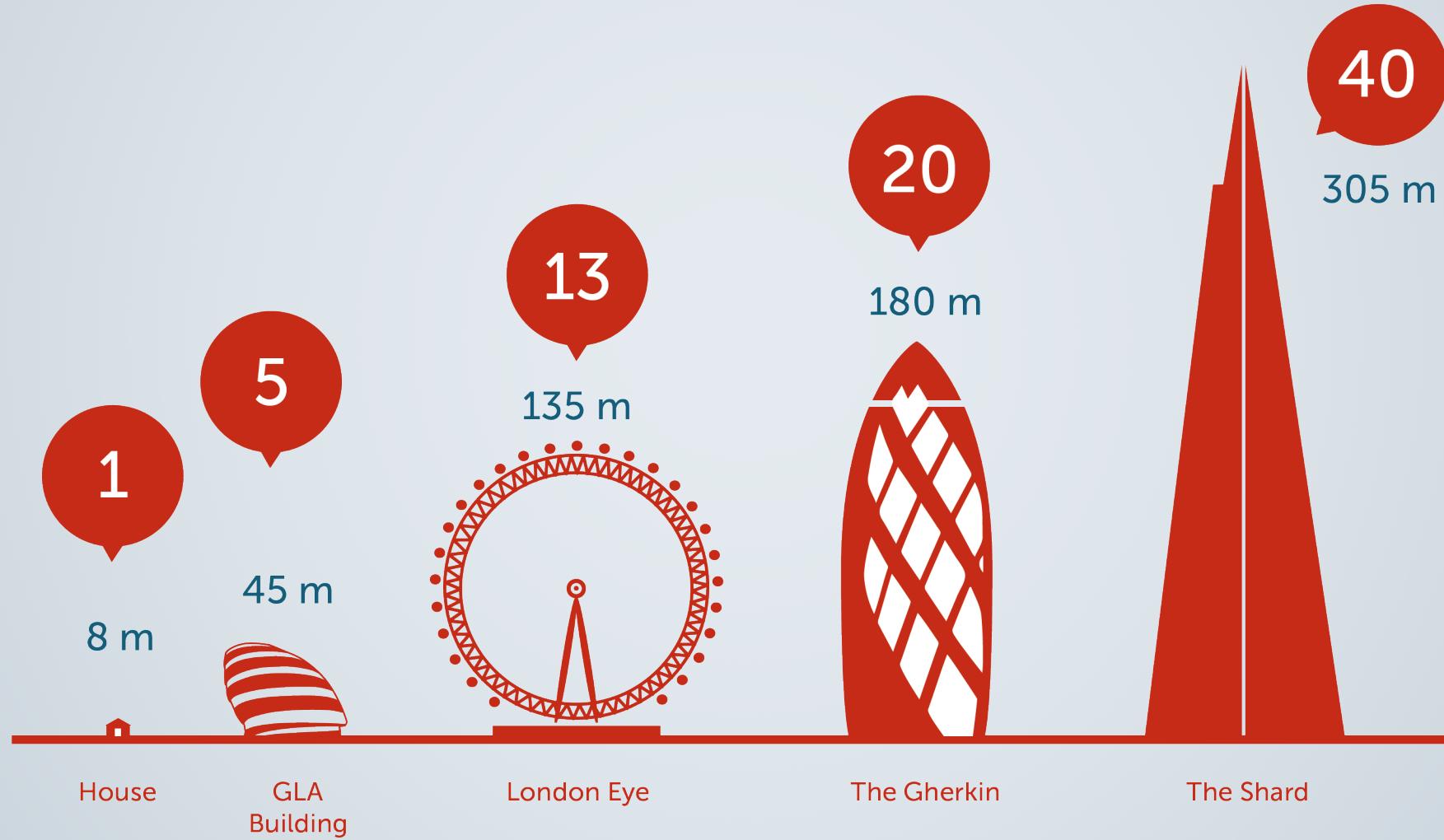
Increase complexity



Reward the wrong thing



Relative estimation



How to size effectively

This is a iterative process you must do during your PBRs (my preference) or as part of estimation meeting, for all stories marked as “ready” on your backlog.



Define the buckets you will have. My suggestion to have at most 4 buckets.

Always estimate stories against each other. We thus need a frame of reference, to relatively size stories

Relatively size each story against benchmark story by discussing only the implementation details that affect its size

Place each story into a bucket

At the end, for each story bucket, do a quick review of the stories in them. Validate they are all reasonably close in “size” to each other.

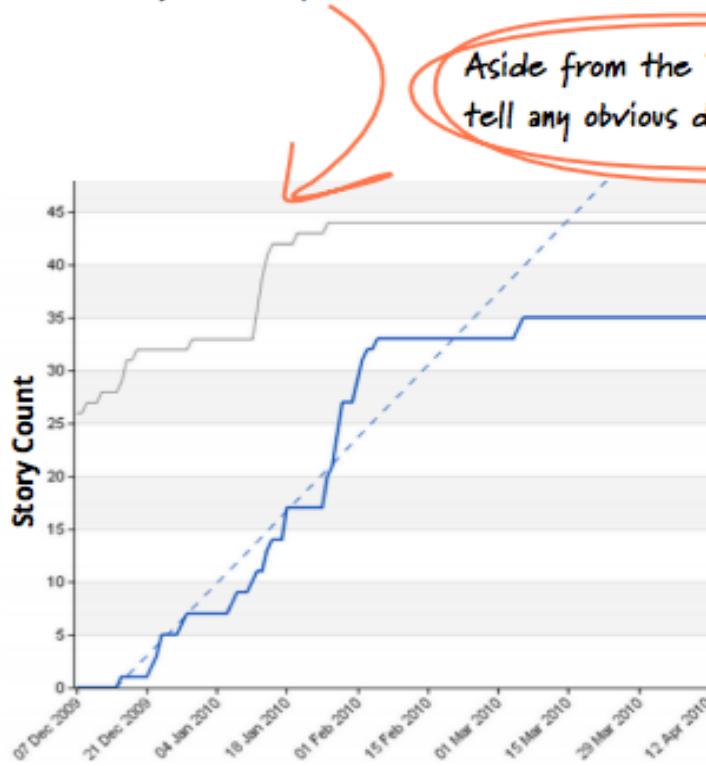
Side effects



If you get “too good” on estimating,
you might stop doing it.

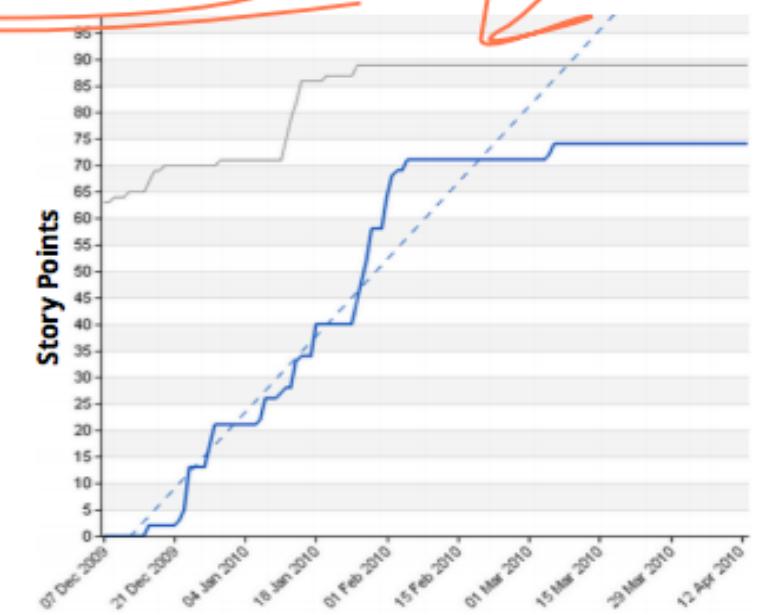
Story Counting

This chart maps the story count over 3 months for a release



Aside from the Y-axis scale, can you tell any obvious difference? I bet not.

This chart maps story points over 3 months for a release



Changed to follow progress using story count on burn up



Still use estimated points as reference for prioritization



Still keep the estimations sessions. There are still high value on team conversation catalyzed by gauging the size of the work

In summary



Revisit the purpose of estimation



In agile model, the estimation is not to track progress against a plan. Its to help define scope of a release



Explore different ways to estimate and pick one that suits your team/product



As important as the estimation is the conversation and understanding it generates



Understand that each team's approach to estimation evolves as the product progresses or team matures



Understand velocity is as important as understand the size of the story

Thank you!!

“ So whenever you’re thinking of asking for an estimate, you should always clarify what decision that estimate is informing. If you can’t find one, or the decision isn’t very significant, then that’s a signal that an estimate is wasteful. “

Martin Fowler



@mattos_leonardo



Leonardo.mattos@gmail.com