YOUR COMPANY LOGO

Agile Estimation

Adaptations from real-world projects

How do you estimate?

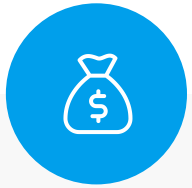
Estimation can be a difficult beast to deal with; more so on an agile project. How do you estimate when you don't have a list of requirements that is complete or signed-off by the customer? Or a nailed-down schedule? What should your currency of estimation be? How do you estimate on distributed teams?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

About me

Leonardo Souza Mattos

IT Principal Consultant - SAP

Agenda



You can replace this text.
This is a sample.

Sample Text

You can replace this text.
This is a sample.

Sample Text

You can replace this text.
This is a sample.

Sample Text

You can replace this text.
This is a sample.

Sample Text

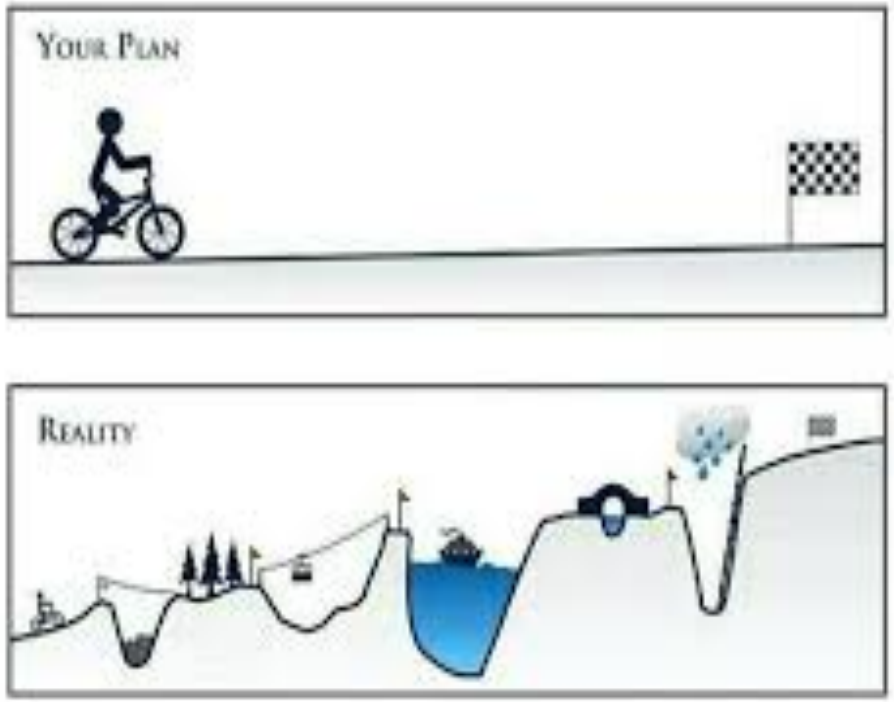
You can replace this text.
This is a sample.

Sample Text

You can replace this text.
This is a sample.

Estimation on Agile

Estimation is hard. For software developers, its 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**



Have you seen this before?

Estimate

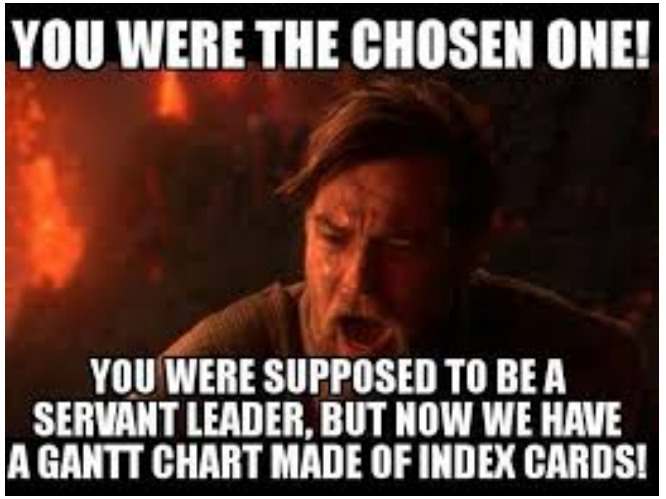
Developers are asked for (or given) estimated for upcoming work. People are optimists, so these estimates tend to be too low, even without pressure to make them this way

Plan

These tasks and estimates are turned to release plans, tracked with burn-down charts and regular status meetings (which most of the demos turn to be)

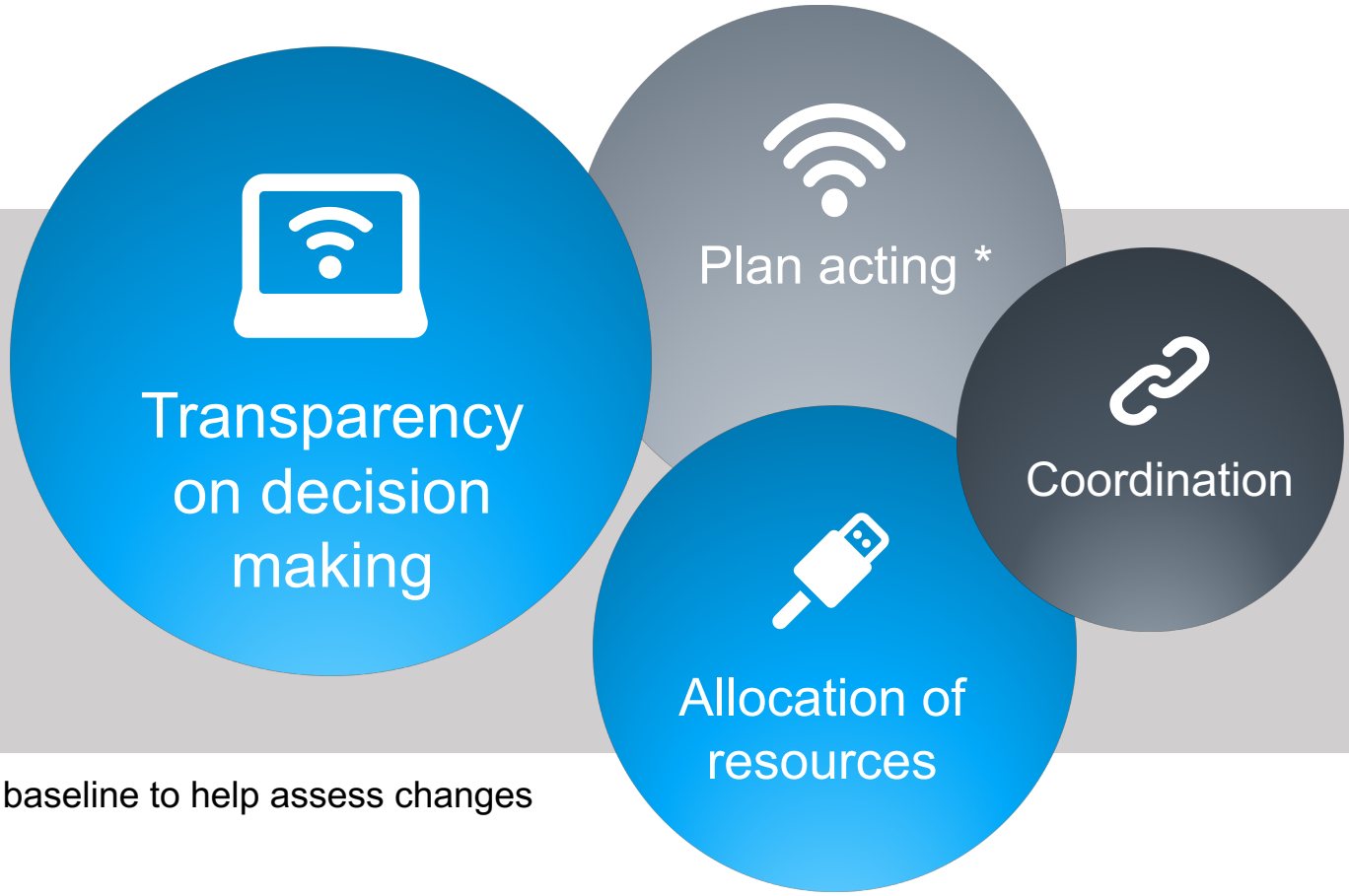
Execute

Time and effort goes into monitoring progress against these plans. Everyone is upset when actuals turn more then estimates. In effort to increase pace with the estimates, developers are told to take shortcuts, which makes things worse.



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

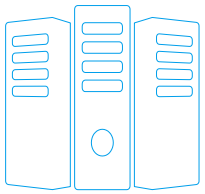
Side effects

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

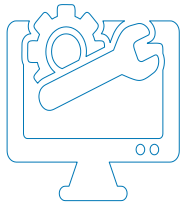
How to do investigate?

Points > Hours

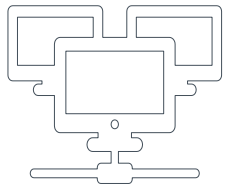
As we agile tend to estimate on relative points, its difficult to give up on linking to a more ”concrete” thing like hours. This is a practice from unmatured teams or teams starting the journey. This is not the preferred way as a mindset shift needs to be enforced.



Dates don’t account for non-project related work that inevitably creeps into our days : emails, meetings, and interviews that a team member may be involved in



Dates have an emotional attachment to them. Relative estimation removes it.



Each team will estimate work on slightly different scale, which means their velocity will naturally be different. This, in turn, makes it impossible to play politics using velocity as a weapon

Once you agree on the relative effort of each story point value, you can assign points quickly without much debate.

Story points reward team members for solving problems based on difficulty, not time spent. This keeps team members focused on shipping value, not spending time

All about points!!!

What does a story point represent?

What is included within a story point estimate?

Are story points an excuse for not being able to estimate correctly?

Who should be involved in story point estimation?

How do we plan/schedule a project using story points?

Can story points be standardized across teams?

How do we know if a team is getting better at estimation?

Why are story points better than estimating in hours or days?

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.

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

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

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.

The bucket theory

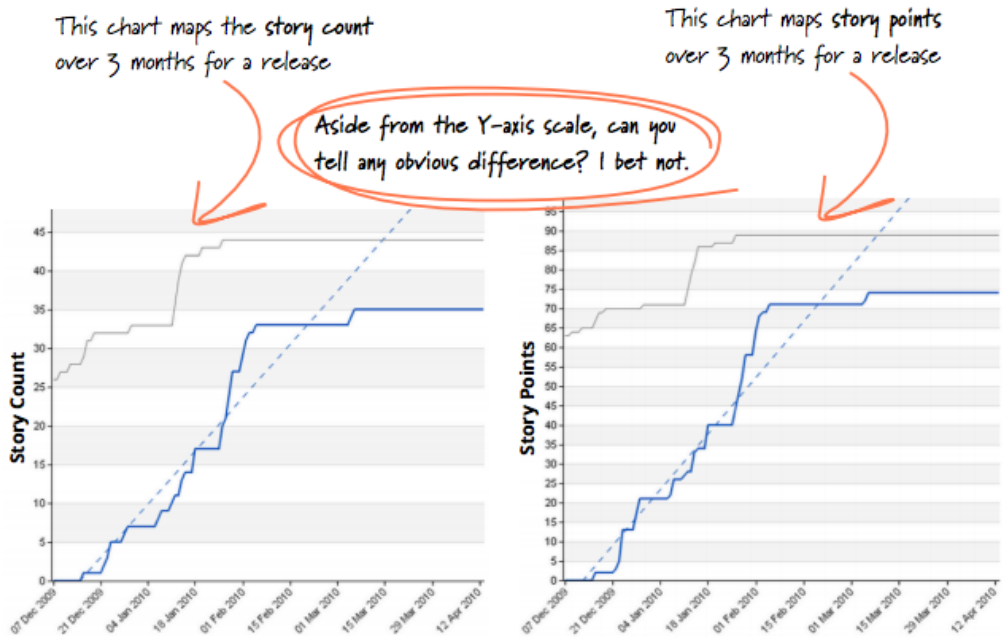
Problem : When we estimate relatively in size, we get different developers would complete the same 2 size story in different speeds.

The theory says that if we calculate the velocity based on speed that each team member completes the same size story, then this deviation on speed will not matter, when we define a scope for a fix period of time

Calculate team velocity

This calculation is considering the bucket theory and the concept of individual speeds over a fix period of time.

Story Counting



Introduced by Thoughtworks consulting company after experience in tens of projects with clients

Over time, when doing the distribution in groups (like buckets) by points, they saw that most og the stories were getting into the same bucket

Comparing the burn up graphs based on story points or number of cards, they started to look all the same, independent of the size of the cards after a span of 2 weeks.

Changed to follow progress using story count on burn up

Still use estimated points as reference for priorization

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

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

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

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

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

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