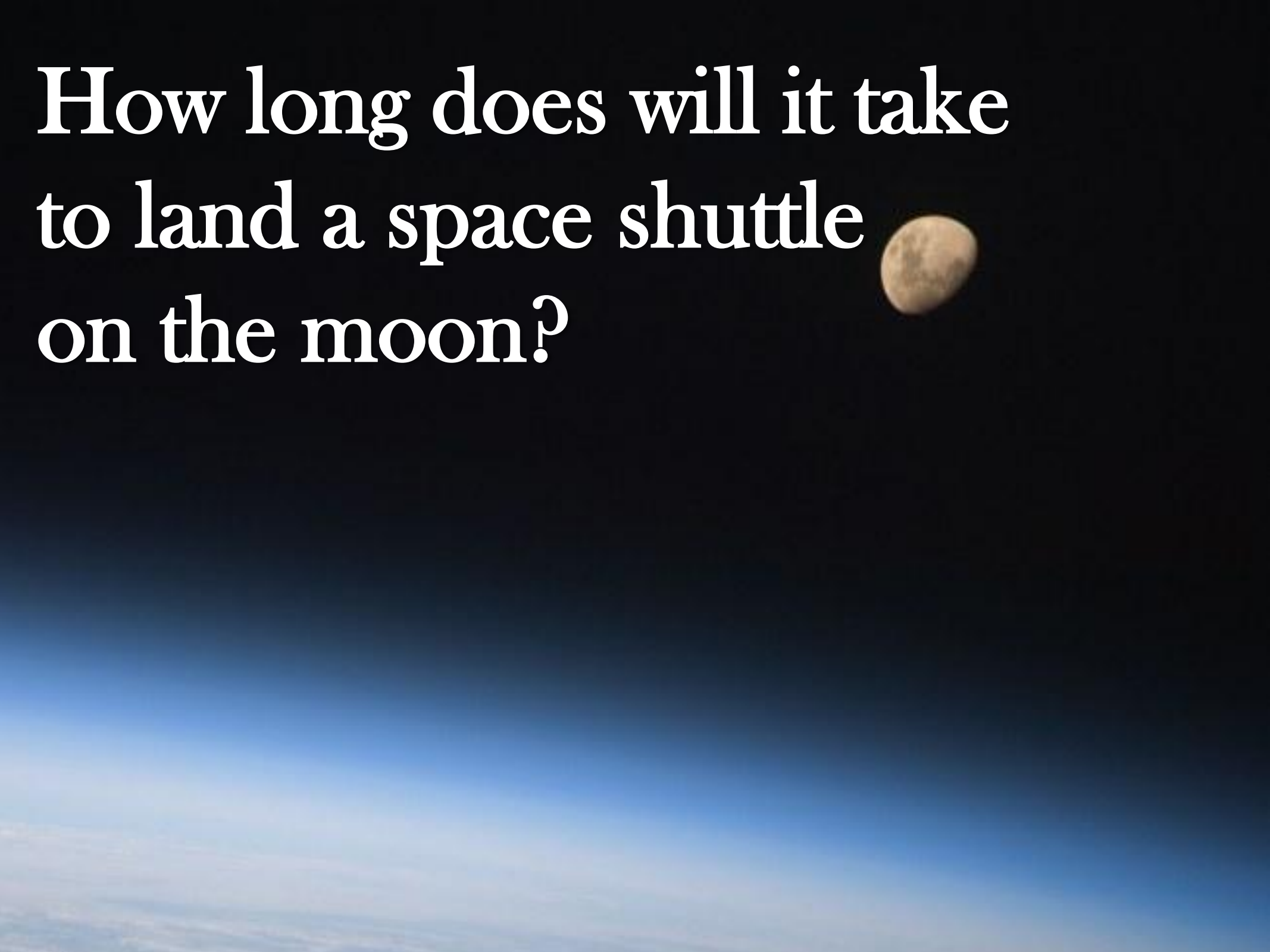
Agile Estimation and Planning

Ben Dewey

Twitter: @BenDewey

Email: ben @bendewey.com


How long does will it take
to land a space shuttle
on the moon?

What is Estimation?

Wikipedia: Estimation is the calculated *approximation* of a result which is **usable** even if input data may be **incomplete** or **uncertain**.

In software engineering, estimation is the process of speculating the amount of **effort** required to complete a task or set of tasks.

Why Do We Estimate Work?

- Resource Planning
- Cost vs. Benefit Analysis
- Coordination with other projects
- What else?

Why Do We *Really* Estimate Work?

To avoid ***stressful*** *situations!!*

Common Problems with Estimation

Calendar time is *not* a measure of effort.

Manager: “How long will this take?”

Engineer: “Do you mean if I worked full time on JUST THIS, or with my other work?”

Common Problems with Estimation

One's *mountain* is another's *molehill*.

Manager: "Joe said it would take three weeks."

Engineer: "What!? He's wrong, the MountainCrusher library already does that, and I have a copy right here."

Common Problems with Estimation

Estimates have a *short shelf-life*.

Manager: “Why is this taking so long!? You said this would only be two days!”

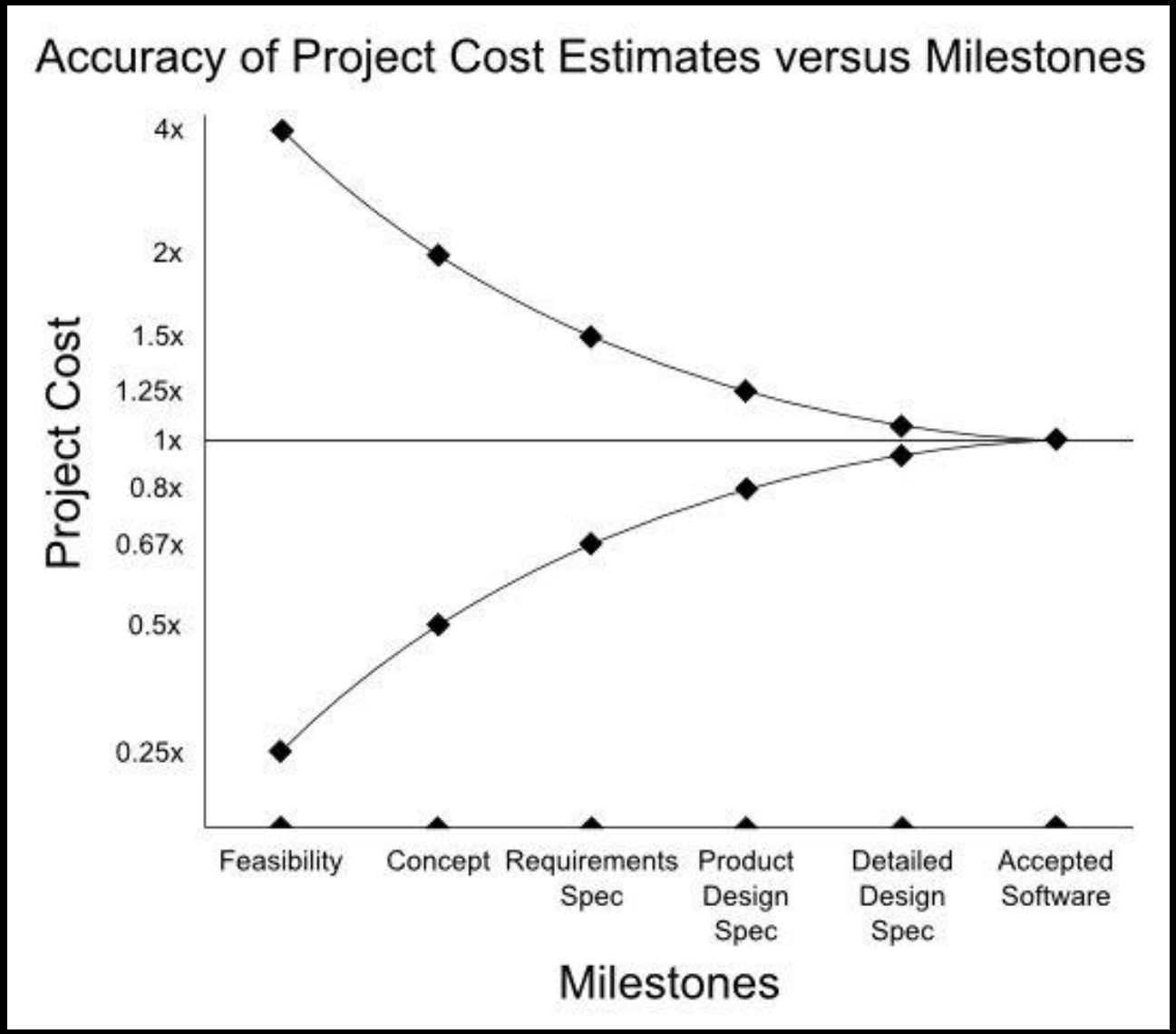
Engineer: “But I said that 4 months ago! The current system architecture makes this feature much harder to build.”

Common Problems with Estimation

It is easier to estimate the *near* future.

Manager: “How long will it take to do Z?”
Engineer: “I haven’t even done B yet!”
Manager: “How long for B?”
Engineer: “Hmm… Maybe a week or so.”

Cone of Uncertainty



Agile estimation avoids these common problems (and more) by taking a more *practical approach* to estimating work.

The Agile Estimation Process

1. Define your work (User Story Modeling)
2. Estimate collaboratively (Planning Poker)
   - Use a unit of work, not time (Story Points)
   - Increase accuracy with relative sizing
3. Measure real progress (Velocity)
4. Adapt to change (Re-estimation)

User Story Modeling

Narrative:

(Who) wants (what) so that (why)

- A story is a conversation starter, and gets more detailed over time

User Story Modeling

Good stories satisfy INVEST:

- Independent
- Negotiable
- Valuable
- Estimable
- Small
- Testable

GOOD:

Billing **wants** to see a summary page of all unpaid accounts, **so that** they can collect payments.

BAD:

Users want rounded corners on the search button.

Our company wants a new website to increase sales.

User Story Modeling: Acceptance Criteria

Story: Users want to import music from a folder so that they can include their own music in the library.

Given a user is on the "Library Folder" screen

- *When* the user clicks Add *Then* the Add Folder screen shows a radio option for "Search for Music”
- *When* the user saves a folder with "Search for Music" checked *Then* the folder is added to the list of folders.

Story Points

… are an arbitrary unit of size/complexity that we use to estimate user stories in lieu of calendar time.

An alternative unit: Ideal Days

Planning Poker - Preparation

- After all the stories are written, *prioritize*! We will estimate the high priority stories first.
- Pick a scale
  - Examples: Fibonacci sequence or Powers of 2
- Pick a baseline story to set the unit size
  - Make the smallest story 1 unit OR
  - Choose a midsize story

Planning Poker - Gameplay

1. Have the story owner give a *brief* overview (1 min)
2. Everyone chooses *one* estimate (KEEP IT SECRET)
3. On the count of 3, we reveal our estimates!
4. Do we agree?

   YES – We're done! Move on to the next story.

   NO – The high and low outliers defend their positions in a *short* open discussion

   1. Record any assumptions we need to make
   2. Repeat steps 2 - 4 until we (mostly) agree

Mock Poker

Company:

Robotics Inc. sells sensors and circuit boards for electronic enthusiasts. They have an ecommerce website where they sell the products.

Story:

"The User wants to be able to see all the categories so that they can select a category to view its products."

Acceptance Criteria:

- Given a user is on the home page
  - When the page loads then the user will see a list of categories.
  - When the user clicks a category then they will go to a product details page.

Story Backlog

- This is just a bucket for all your stories
- Keep it prioritized and organized!
- Pull estimated stories off this list to create an iteration plan

Iteration Plan

- Usually 2-4 weeks
- Do high risk / high priority work first
- How much can you do in one iteration?
  - Make sure everyone can be kept busy
  - Compare with previous work, if possible
  - If not, *guess*, it's OK to be wrong!

Consistancy & Accuracy

Team Velocity

- Velocity is the number of points you **completed** in previous iterations
- Over time this number will stabilize (usually after 3 iterations)

Total Points / Velocity * Iteration length =

Calendar Time!

Re-estimation

- Re-estimate when you have new information that affects your previous estimates
- Review your upcoming story estimates before each iteration, do they still make sense?
- Re-estimate if you change a story
- **NEVER** change an estimate after a story has entered development!

Lab - Planning Poker

- Break up into groups
- Define roles you will represent:
  - Business users (2) and team members (design, dev, QA, etc)
- We work at Expedia.com and have 20 million customers. You are doing an add-on for the Customer Service team. Do a planning poker for this user story:

"Customer Service wants to search for customers by their first and last name, so that they can quickly retrieve customer information when on a call."