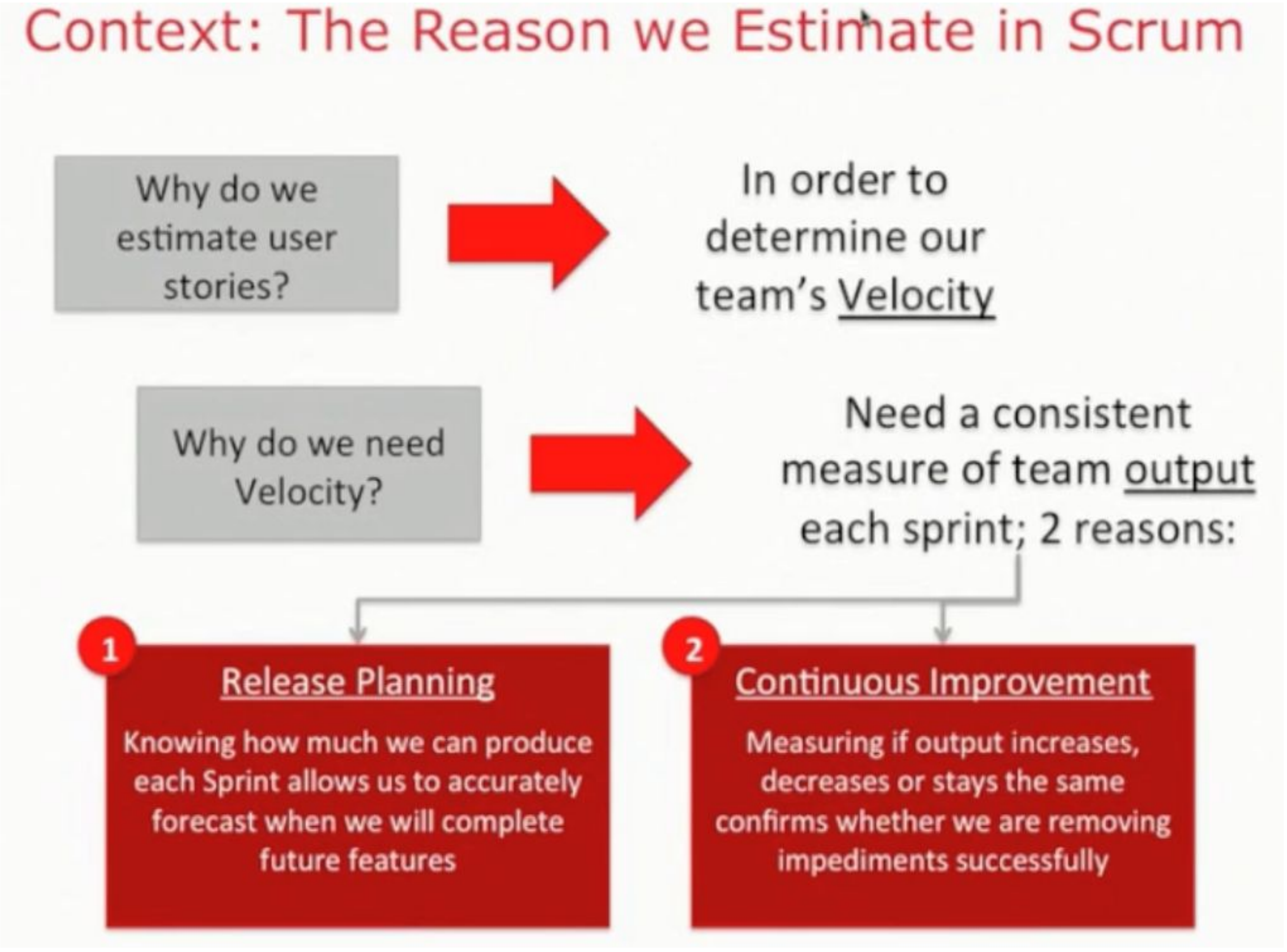
Agile Estimation Techniques

Problems with Traditional Estimates

- We spend too much time doing/redoing it , but we rarely get it right –
- Very difficult to effectively estimate until there is detailed analysis of feature and there is a functional and technical specs documents (expensive)
- Estimating for unknown issues is as good as seeing in future
- People are optimists, so estimates tend to be too low than actual
- Quality usually gets scarified to match estimates
- Fear of failure
- Lack of confidence/experience
- Many unknowns, changes, dependencies

Story points help in improving estimates without doing detailed upfront analysis



Context: The Reason we Estimate in Scrum

Why do we estimate user stories?

In order to determine our team's Velocity

Why do we need Velocity?

Need a consistent measure of team output each sprint; 2 reasons:

1 **Release Planning**

Knowing how much we can produce each Sprint allows us to accurately forecast when we will complete future features

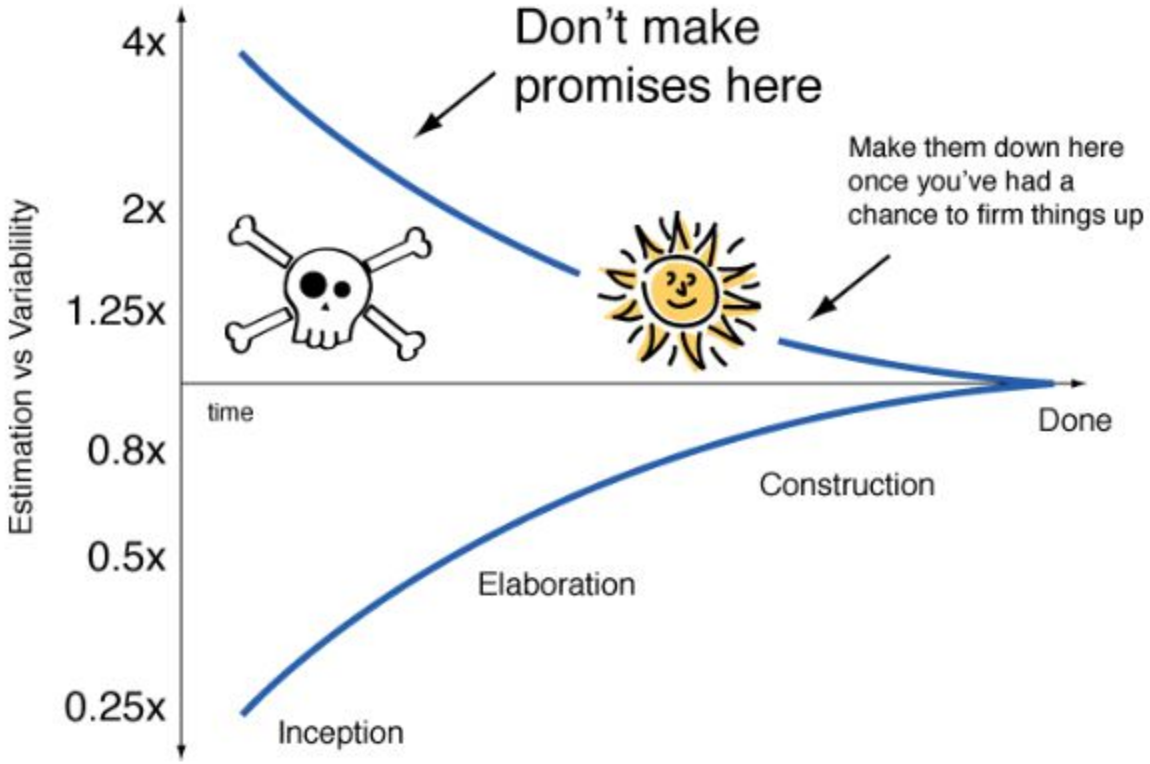
2 **Continuous Improvement**

Measuring if output increases, decreases or stays the same confirms whether we are removing impediments successfully

Cone of Uncertainty

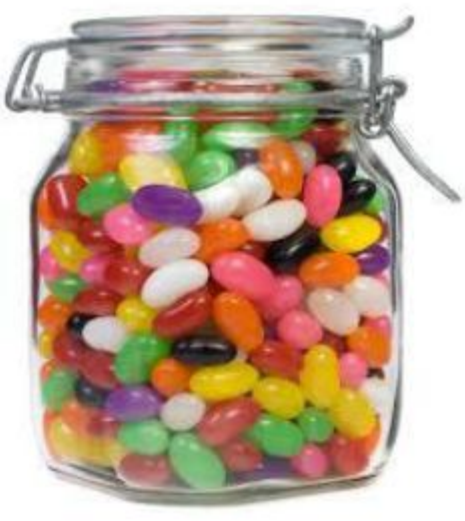
- In waterfall we spent lots of time estimating early in a project, giving us 4x error rate

- In agile estimation we :
  a. Use sizing methods appropriate to where you are in the cone
  b. Avoid spending vast amounts of time estimating work, especially at the start of a project when so much is unknown



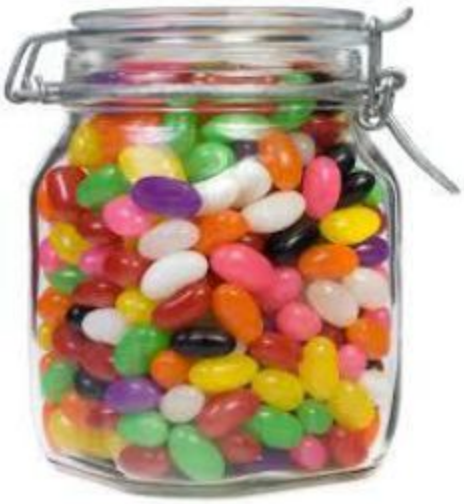
Guess

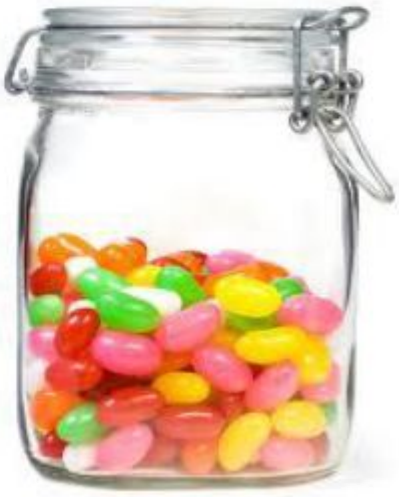
How many jelly beans are there in the jar ?



Another guess

How many jelly beans are there on the left as compared to right ?





Relative Estimation

We're pretty good at estimating relatively

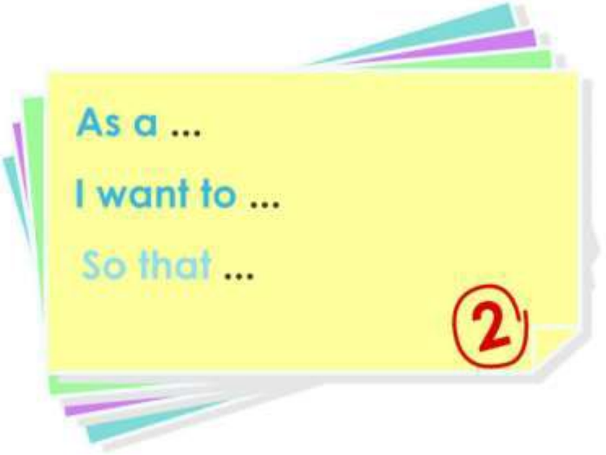
This
looks x2 as big
as that
Create new hockey pool
4 days
Update existing hockey pool
2 days

Estimation

What is estimation in Agile

Estimation in Agile is a method of measuring a user story.

Always relative estimation



Agile teams prefer to express estimates in units other than time-honored "man--day" or "man--hour".
The most widespread unit is "**story points**"

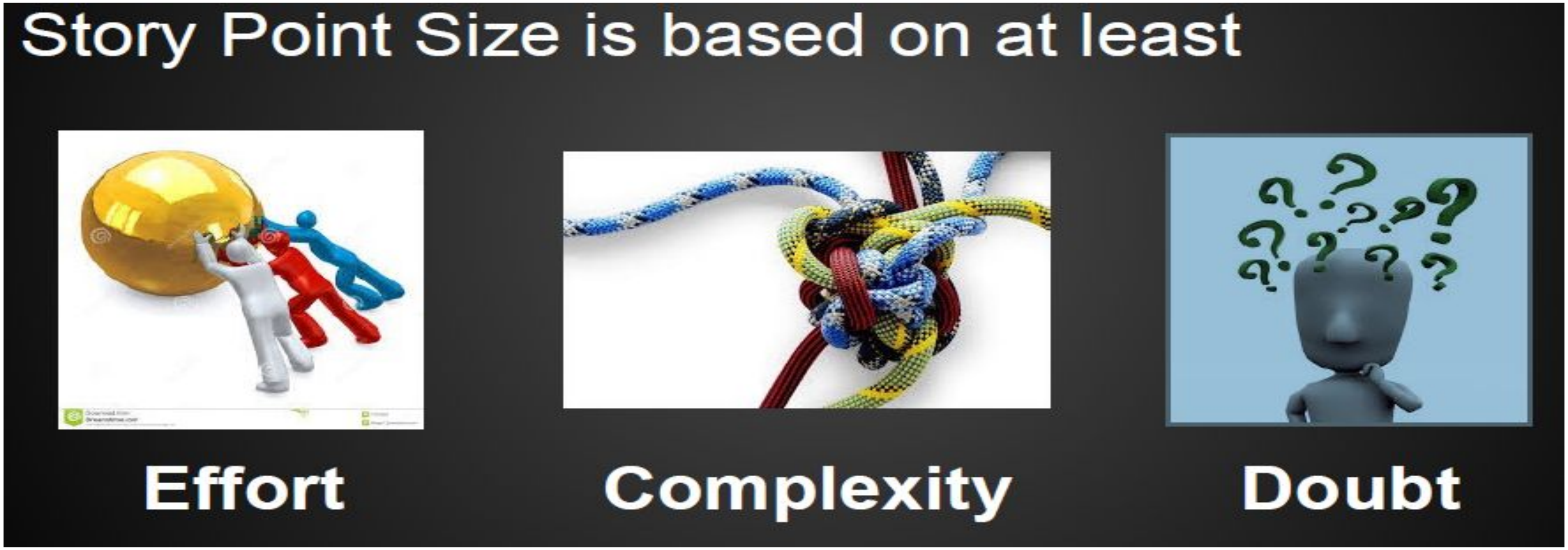
Relative Sizing Advantages

- Less Stress: Team doesn’t worry if estimates are not spot-on
- Meeting sprint commitments starts to improve each sprint
- We can have historical velocity that can be used towards future planning

Size is not based on effort alone

Unlike Waterfall, when we size stories, we do NOT base it on effort alone!

COMPLEXITY – How difficult it seems?
EFFORT - How much of it there is?
RISK/DOUBTS/UNCERTAINITY - Current knowledge (uncertainty)



Story Point Size is based on at least

Effort

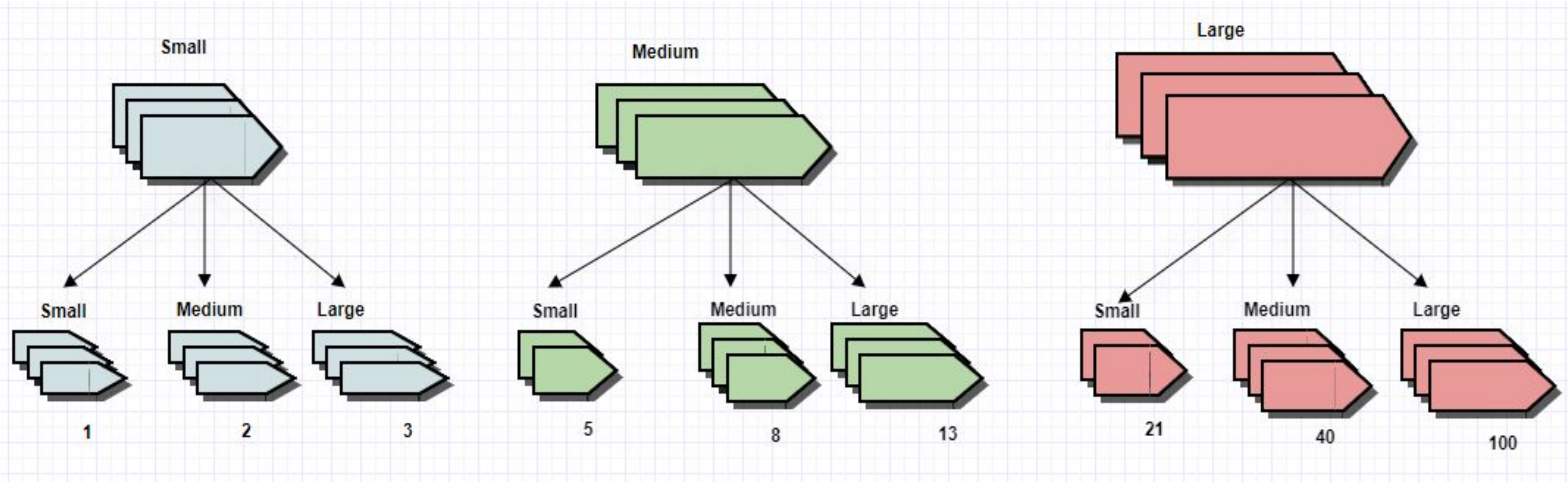
Complexity

Doubt

Slice and View (when no baseline is available)

Print all stories on small paper format.
Ask the Team to sort them by Small, Medium, Large. Like that, obviously you get 3 stacks.
Take the first stack (Small) and apply the same technique: Sort all in that stack upon Small, Medium, Large.
Take the second stack (Medium) and apply the same technique: Sort all in that stack upon Small, Medium, Large.
Take the third stack (Large) and apply the same technique: Sort all in that stack upon Small, Medium, Large.
Like that you have them nicely ordered from very small to the biggest at the other side of the line.
You did the comparison of each stack. After that you assign the Fibonacci sequence to them.   It is that simple.   Your baseline of estimations are created.

Slice and View (when no baseline is available)



Planning Poker

Planning Poker is an agile estimating and planning technique that is consensus based. To start a poker planning session, the product owner or customer reads an agile user story or describes a feature to the agile team

Each estimator is holding a deck of Planning Poker cards with values like 0, 1, 2, 3, 5, 8, 13, 20, 40 and 100, which is the sequence that most recommend. The values represent the number of story points in which the team estimates.

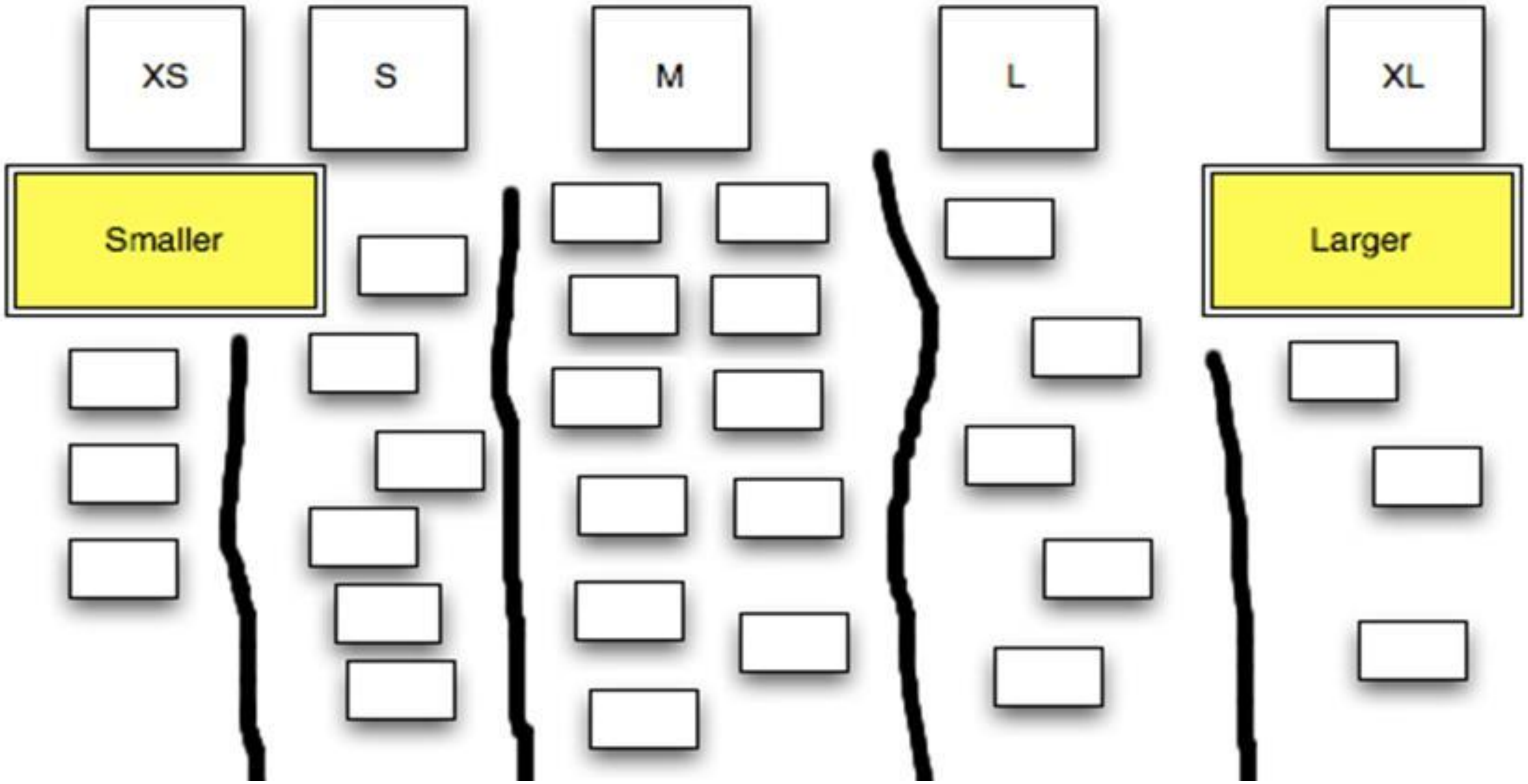
The team discuss the feature, asking questions of the product owner as needed. When the feature has been fully discussed, each person privately selects one card to represent his or her estimate. All cards are then revealed at the same time.

If all estimators selected the same value, that becomes the estimate. If not, the estimators discuss their estimates. The high and low estimators should especially share their reasons. After further discussion, each estimator reselects an estimate card, and all cards are again revealed at the same time.

The poker planning process is repeated until consensus is achieved or until the estimators decide that agile estimating and planning of a particular item needs to be deferred until additional information can be acquired.

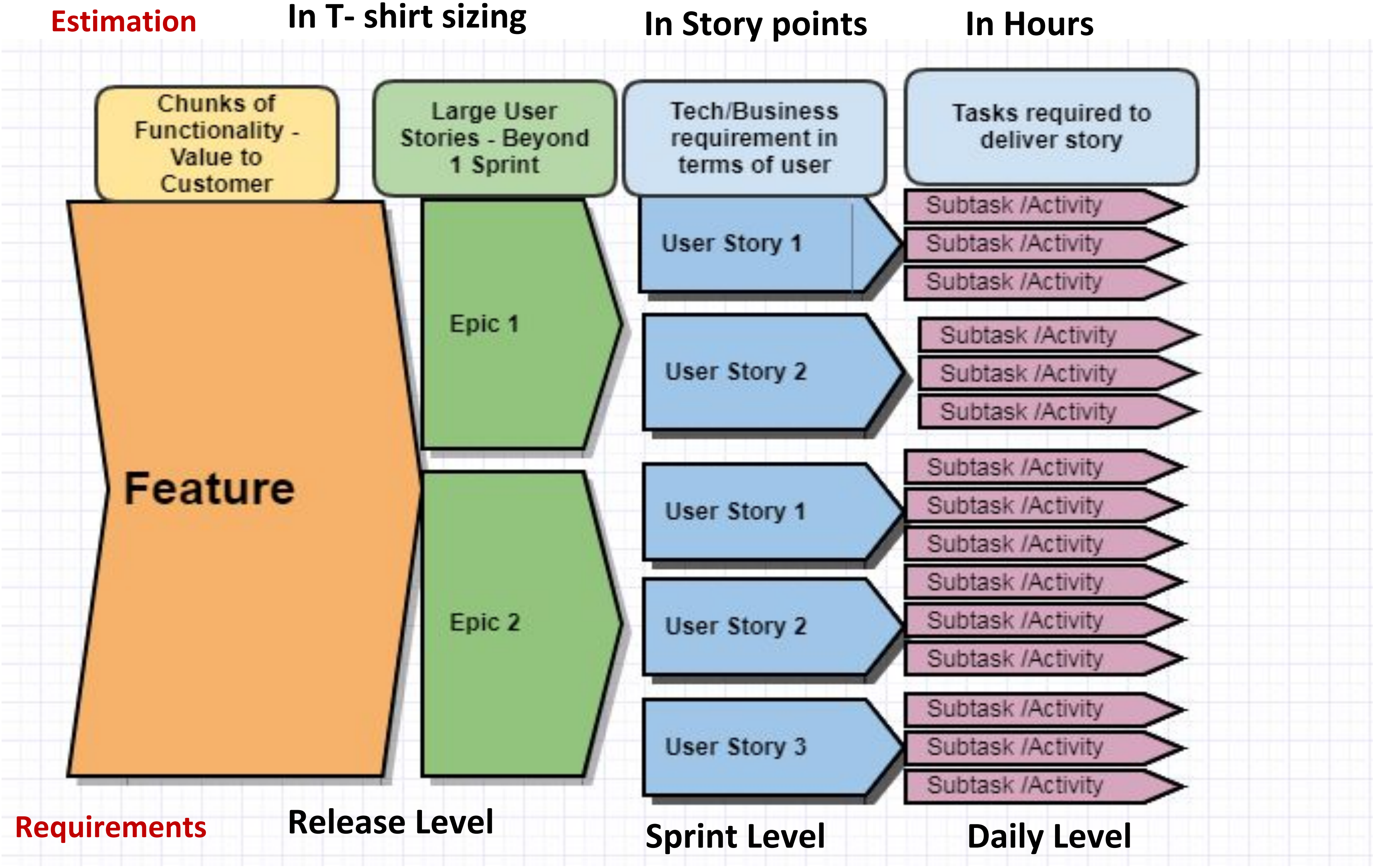
Affinity Estimation

Step 1: Silent Relative Sizing
Step 2: Wikipedia-like Editing of Wall
Step 3: Place Items into Relative Sizing Buckets
Step 4: Product Owner "Challenge"
Step 5: Get it into an Electronic Tool



Use any Relative measure

Industry practice



Estimation

In T- shirt sizing

In Story points

In Hours

Chunks of Functionality - Value to Customer

Large User Stories - Beyond 1 Sprint

Tech/Business requirement in terms of user

Tasks required to deliver story

Feature

Epic 1

User Story 1

Subtask /Activity

Subtask /Activity

Subtask /Activity

User Story 2

Subtask /Activity

Subtask /Activity

Subtask /Activity

Epic 2

User Story 1

Subtask /Activity

Subtask /Activity

Subtask /Activity

User Story 2

Subtask /Activity

Subtask /Activity

Subtask /Activity

User Story 3

Subtask /Activity

Subtask /Activity

Subtask /Activity

Requirements

Release Level

Sprint Level

Daily Level