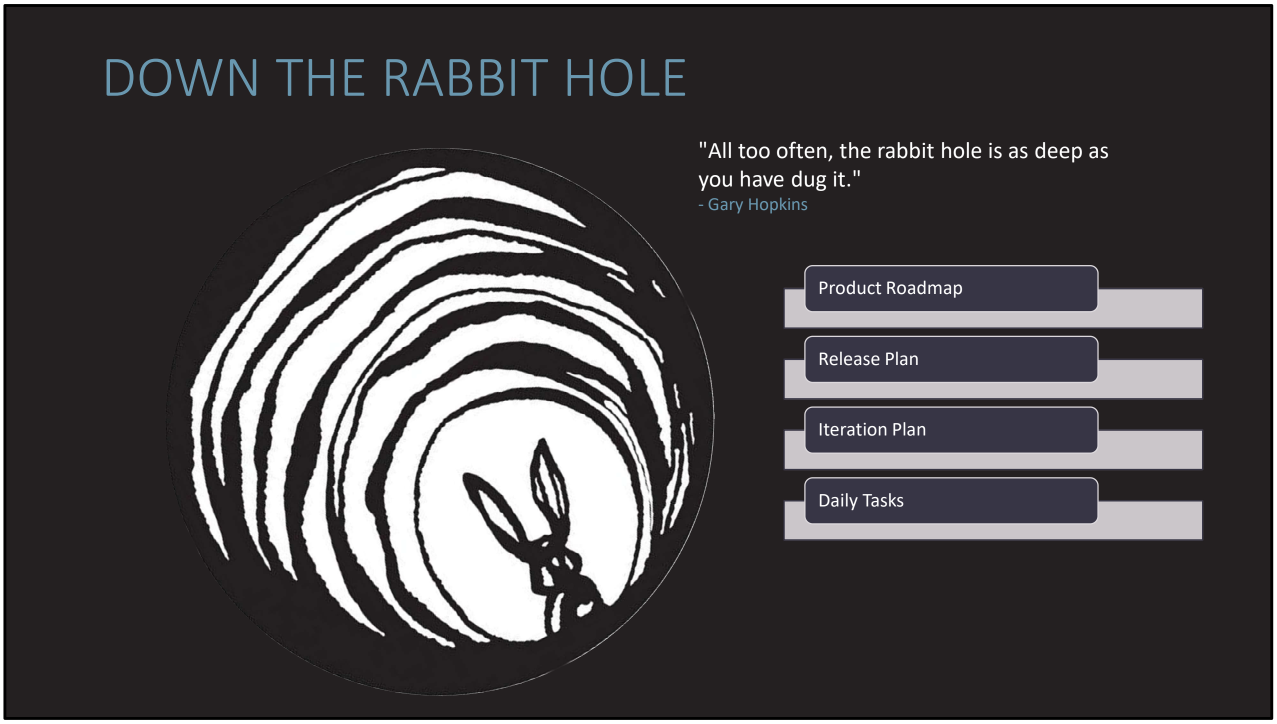
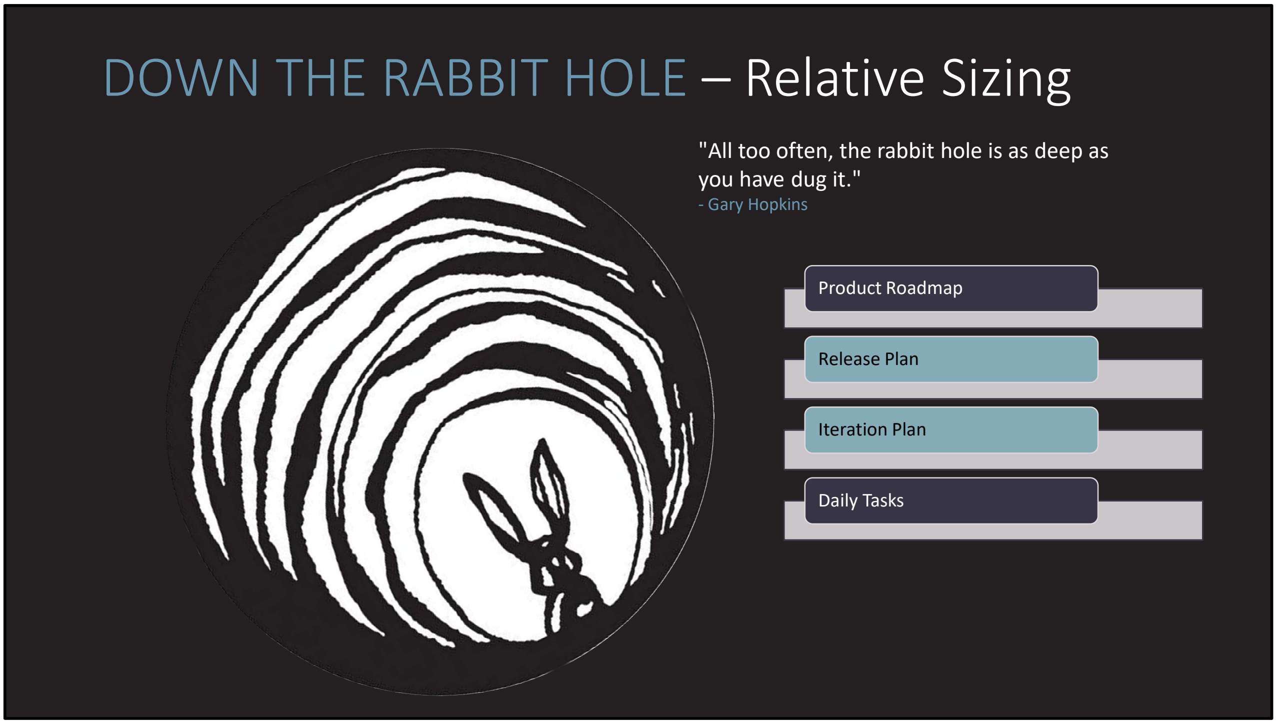


DOWN THE RABBIT HOLE

"All too often, the rabbit hole is as deep as you have dug it."
- Gary Hopkins

- Product Roadmap
- Release Plan
- Iteration Plan
- Daily Tasks

“Down the Rabbit Hole” is a series of estimation seminars covering agile estimation and planning from a very high level (Epics and Features), all the way down to a very granular level (Sub-Tasks).



DOWN THE RABBIT HOLE – Relative Sizing

"All too often, the rabbit hole is as deep as you have dug it."
- Gary Hopkins

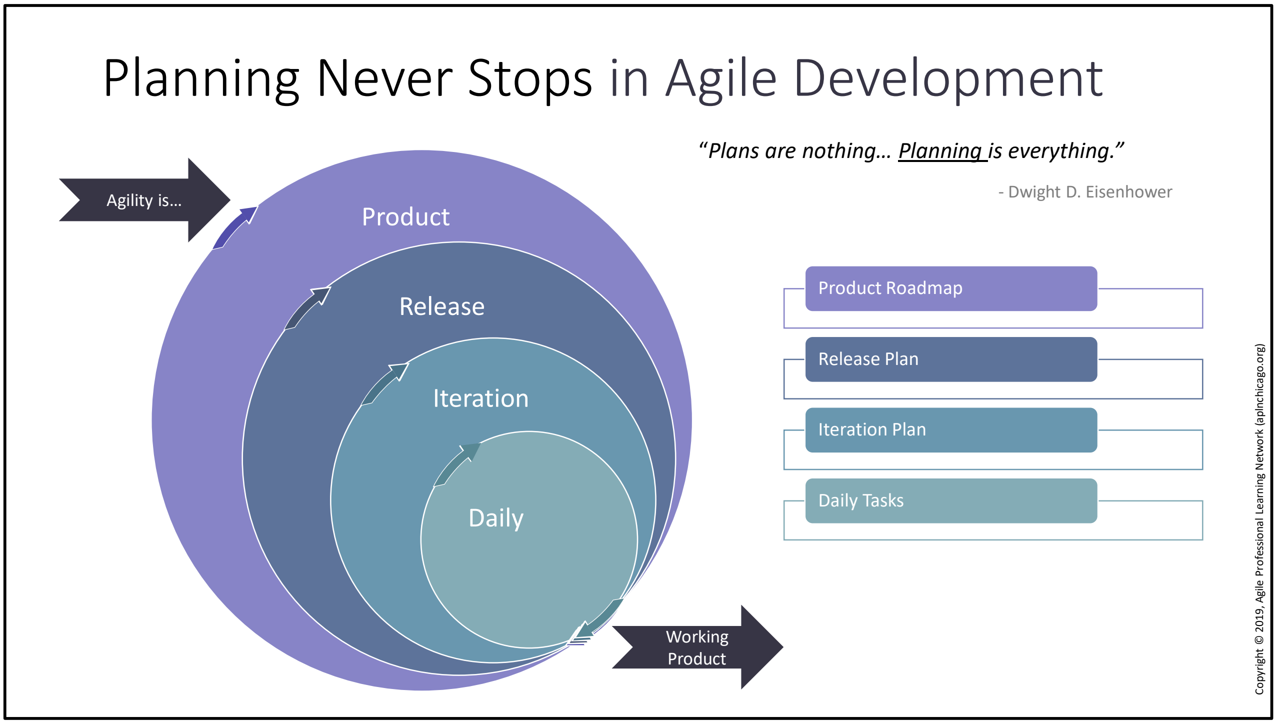
Product Roadmap

Release Plan

Iteration Plan

Daily Tasks

This section focuses on something that is fundamental to the discussions of Release and Iteration plans – estimation of work items. Especially SHARED work items.

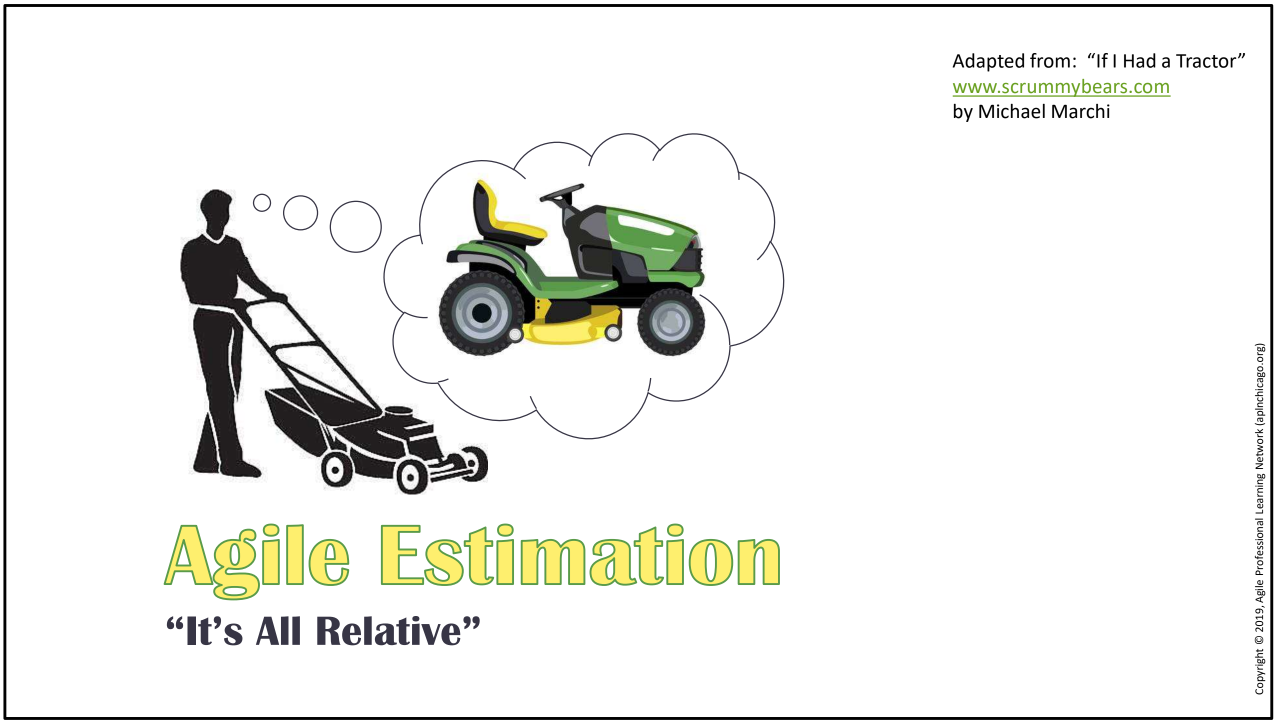


Planning Never Stops in Agile Development

"Plans are nothing… Planning is everything."

- Dwight D. Eisenhower

They say that planning never stops in Agile …



Adapted from: “If I Had a Tractor”
www.scrummybears.com
by Michael Marchi

Agile Estimation

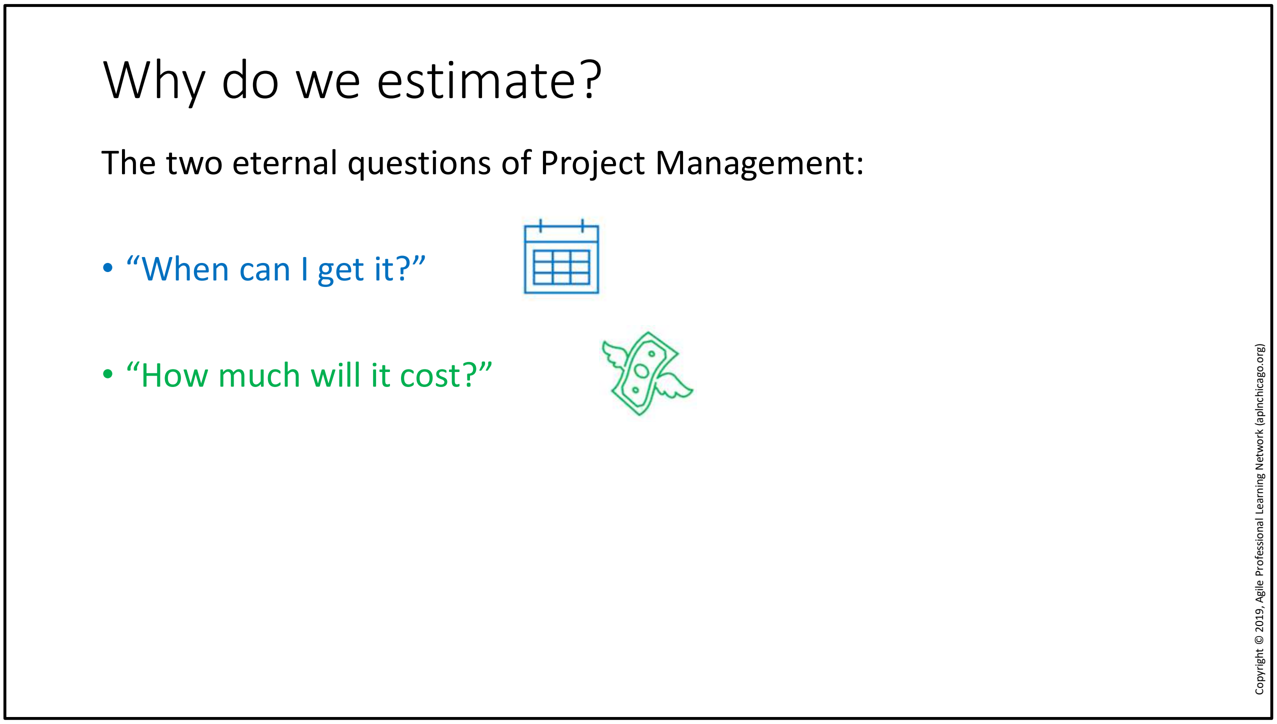
“It’s All Relative”

Being able to estimate individual backlog items (usually User Stories) is a key skill that enables us to perform Release and Sprint Planning.

This session focuses on explaining relative sizing and estimation through metaphor – the principal metaphor is that of a Lawn Mowing Service.

This material is derived from Michael Marchi’s series of blog articles on Agile Estimation and Planning – in particular “If I Had a Tractor” and the “Agile Estimation Primer” found on his agile blog at https://www.scrummybears.com

The material was developed and presented at the Agile Professional Learning Network (aplnchicago.org) monthly meetups



Why do we estimate?

The two eternal questions of Project Management:

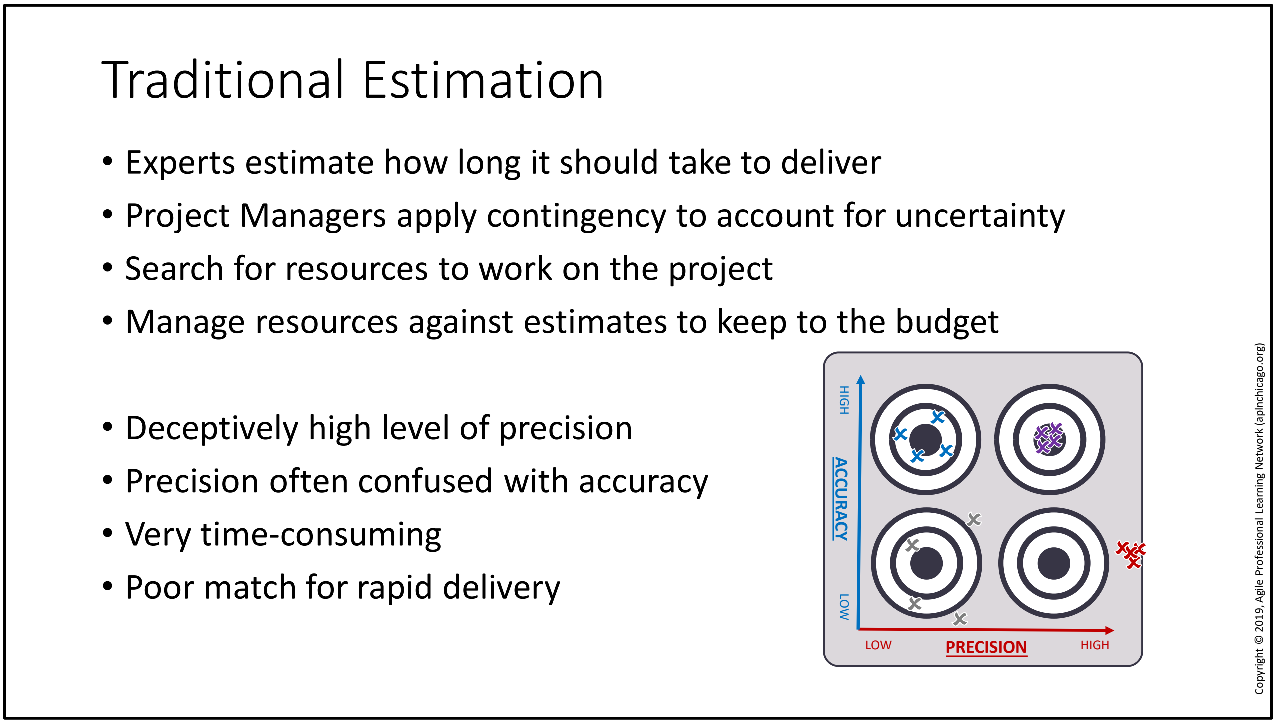
- "When can I get it?"
- "How much will it cost?"

When presented with a new work item, Project Managers are frequently asked for two key pieces of information. When can I get it, and how much will it cost me?

The two concepts are intimately intertwined. As the saying goes, "Time is Money".

What they're really asking you to do, is to predict the future.

The question is, *"How close to reality does the prediction have to be?"*



Traditional Estimation

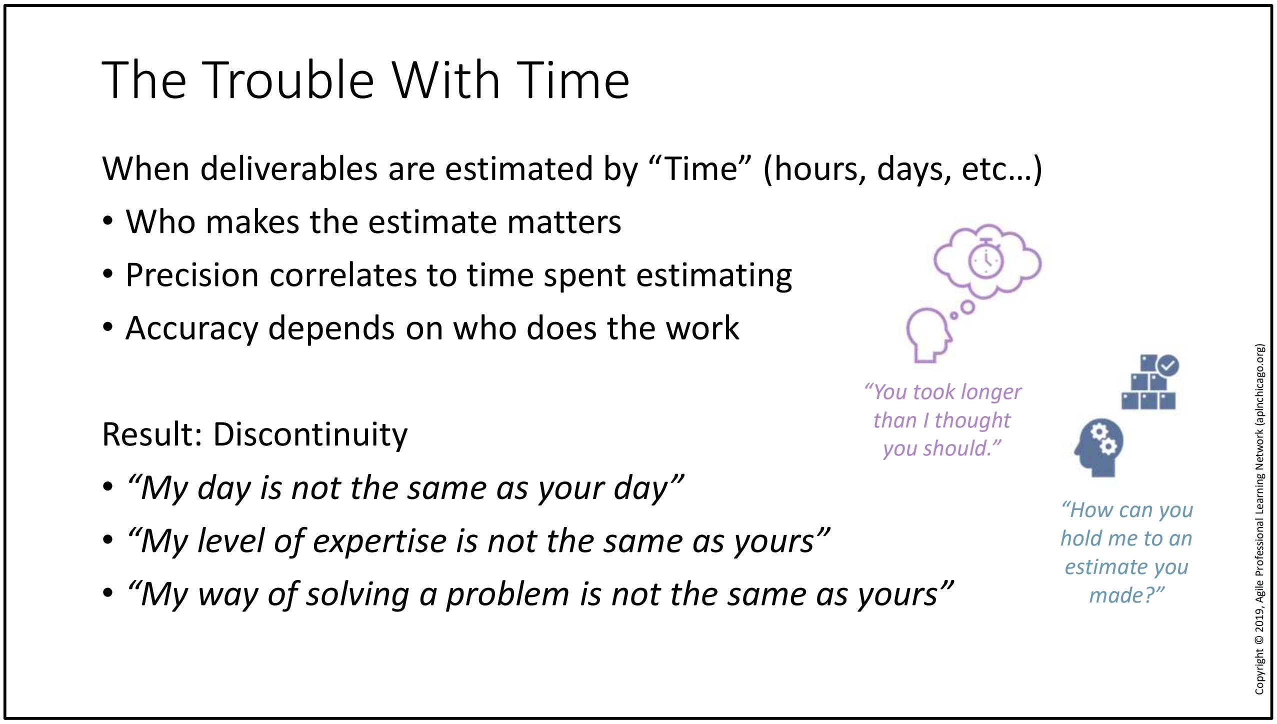
- Experts estimate how long it should take to deliver
- Project Managers apply contingency to account for uncertainty
- Search for resources to work on the project
- Manage resources against estimates to keep to the budget

- Deceptively high level of precision
- Precision often confused with accuracy
- Very time-consuming
- Poor match for rapid delivery

Many organizations attempt to create an extremely high level of precision to their estimates.

No lie, I once saw the estimate for a relatively SMALL project come in at 972 hours. Think about that. That's a lot of hours, isn't it? 972. I could argue that number is very precise. But I question whether it is accurate. The mere fact there is a high degree of precision would imply that a lot of thought went into it, and therefore it has a very small degree of uncertainty.

I just can't help but wonder, could they have invested less time in the estimate, and just called it 1000 hours? That may be less precise, but I would argue it's just as accurate. The two numbers are within 3% of each other, after all!



The Trouble With Time

When deliverables are estimated by “Time” (hours, days, etc…)

- Who makes the estimate matters
- Precision correlates to time spent estimating
- Accuracy depends on who does the work

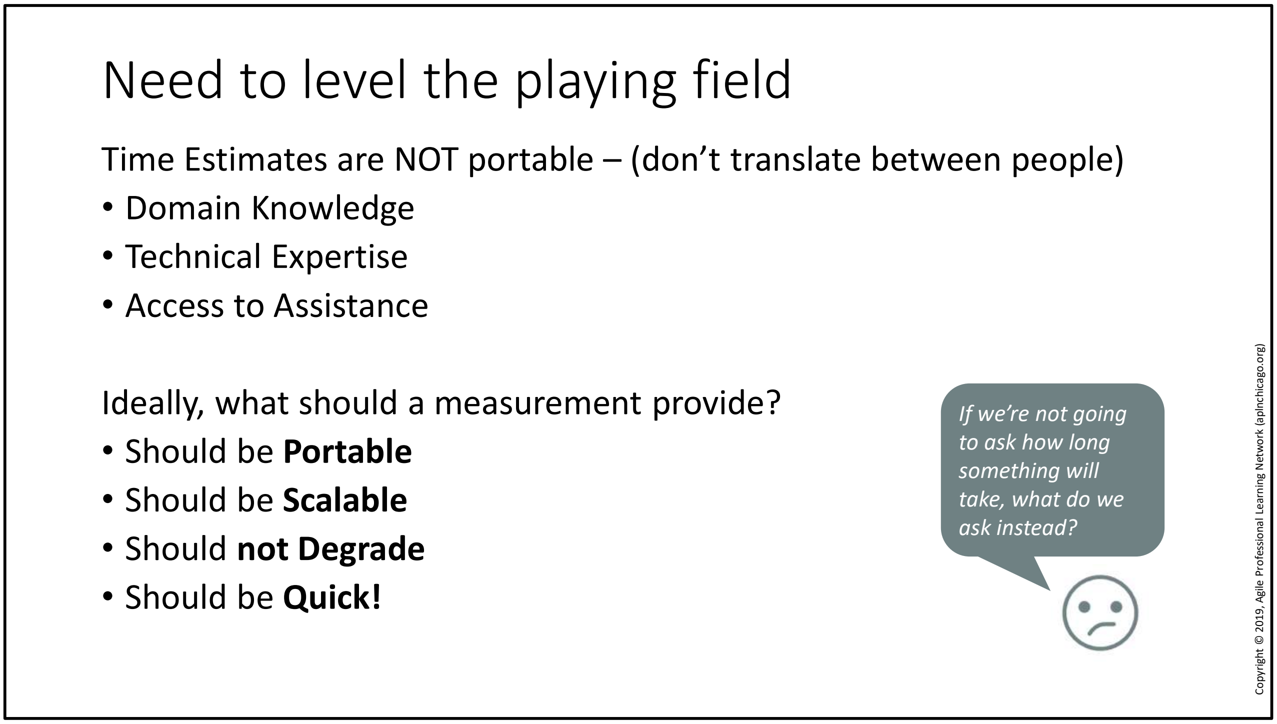
Result: Discontinuity

- *“My day is not the same as your day”*
- *“My level of expertise is not the same as yours”*
- *“My way of solving a problem is not the same as yours”*

“You took longer than I thought you should.”

“How can you hold me to an estimate you made?”

Estimation in time is a very personal thing, affected all sorts of factors. Who makes the estimate? What is their level of expertise? Are they the person who will do it? Are they distracted? Are they having a good day, or a bad day?



Need to level the playing field

Time Estimates are NOT portable – (don’t translate between people)

- Domain Knowledge
- Technical Expertise
- Access to Assistance

Ideally, what should a measurement provide?

- Should be **Portable**
- Should be **Scalable**
- Should **not Degrade**
- Should be **Quick!**

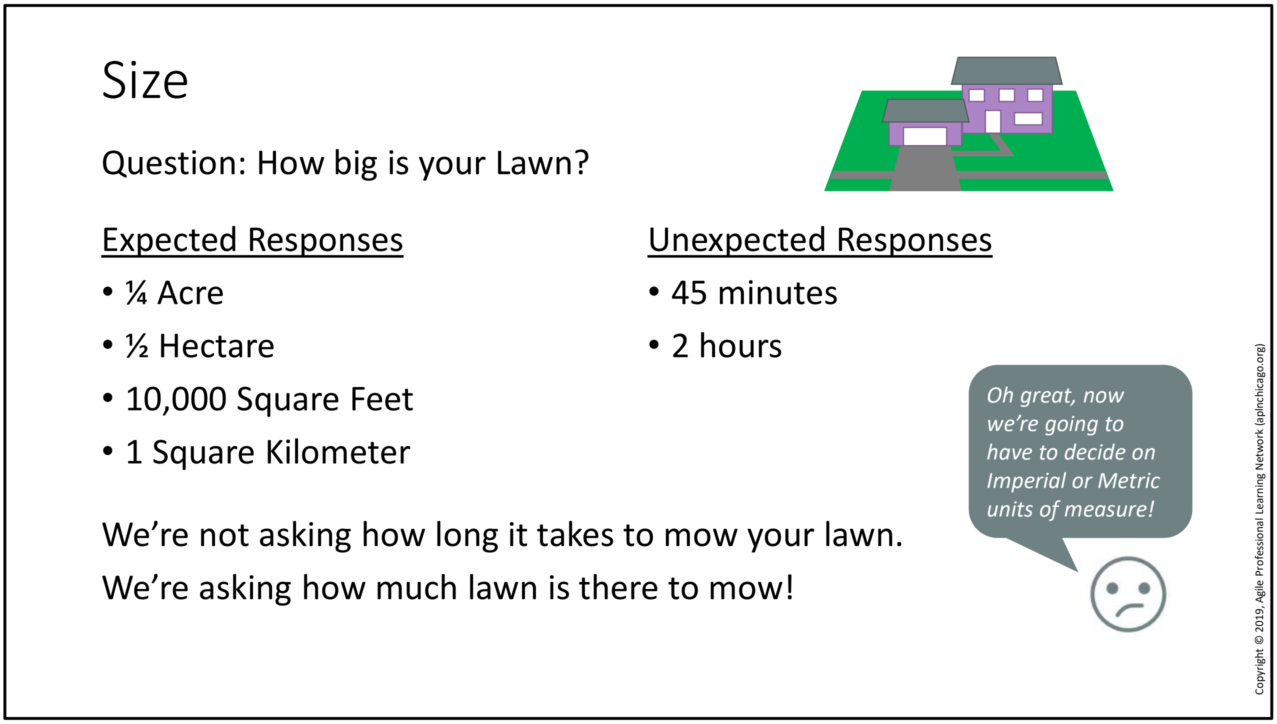
If we’re not going to ask how long something will take, what do we ask instead?

If I were to give two different people the same work item to complete, I will wager that they would not take the exact same amount of time to deliver that work item. I’d even go so far as to say the variance will likely be way more than 3%.

Another thing to consider is that an estimate you make today, is limited by what you know today, not necessarily what you’ll know by the time you get to it. We learn and grow constantly.

So even if I was very diligent in my precise estimate of my own future performance, that estimate is inherently … well … it’s an estimate. It’s not an actual.

If we’re going to recognize the difficulty inherent in a Precise estimate, maybe we can get just about as good a result by shooting for getting in the right ballpark. A value that is Accurate, but with a lower level of fidelity.



Size

Question: How big is your Lawn?

Expected Responses
- ¼ Acre
- ½ Hectare
- 10,000 Square Feet
- 1 Square Kilometer

Unexpected Responses
- 45 minutes
- 2 hours

Oh great, now we're going to have to decide on Imperial or Metric units of measure!

We're not asking how long it takes to mow your lawn.

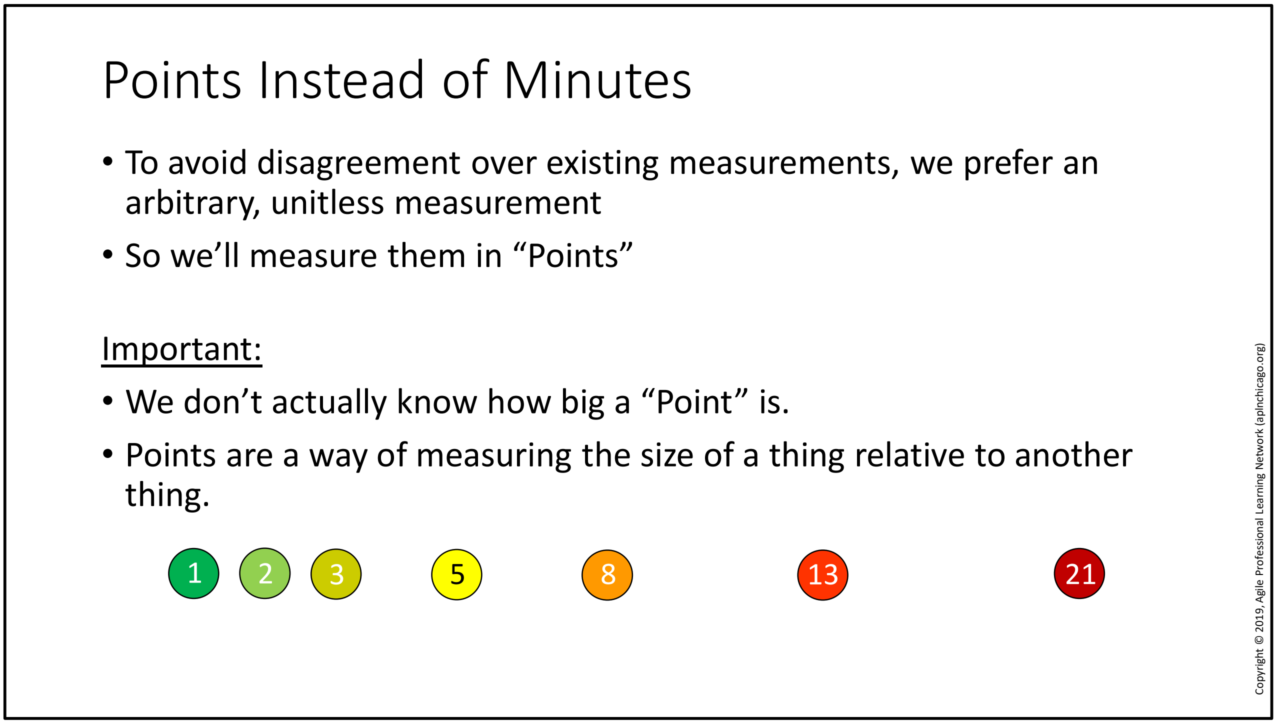
We're asking how much lawn is there to mow!

If I were to ask you: "How big is your lawn?", I would get a variety of responses. Many common answers are

- 1/4 acre
- 1/2 acre
- 10000 square feet
- 900 square meters

The answer I would not expect to get is "45 minutes".

Why is that? For one thing, I asked for a size estimate, not a time estimate. Yet, all over the globe, if I ask a software developer or project manager to estimate the size of a feature, they will generally answer in a number of hours, days, weeks, months, etc.



Points Instead of Minutes

- To avoid disagreement over existing measurements, we prefer an arbitrary, unitless measurement
- So we'll measure them in "Points"

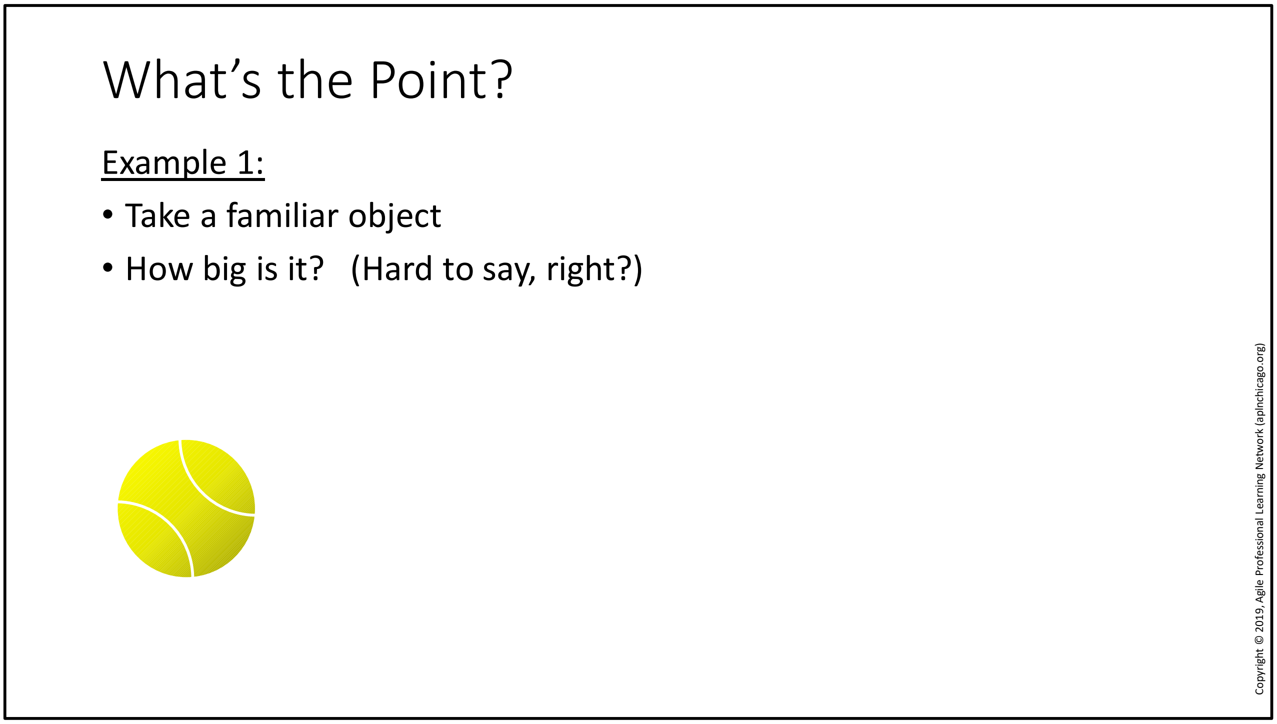
<u>Important:</u>

- We don't actually know how big a "Point" is.
- Points are a way of measuring the size of a thing relative to another thing.

1 2 3 5 8 13 21

One popular measurement technique is to apply a modified Fibonacci sequence to estimate sizes.

Each number in the Fibonacci sequence is equal to the sum of the two previous values, creating larger and larger gaps between values as you proceed. (We'll explain why in a minute)

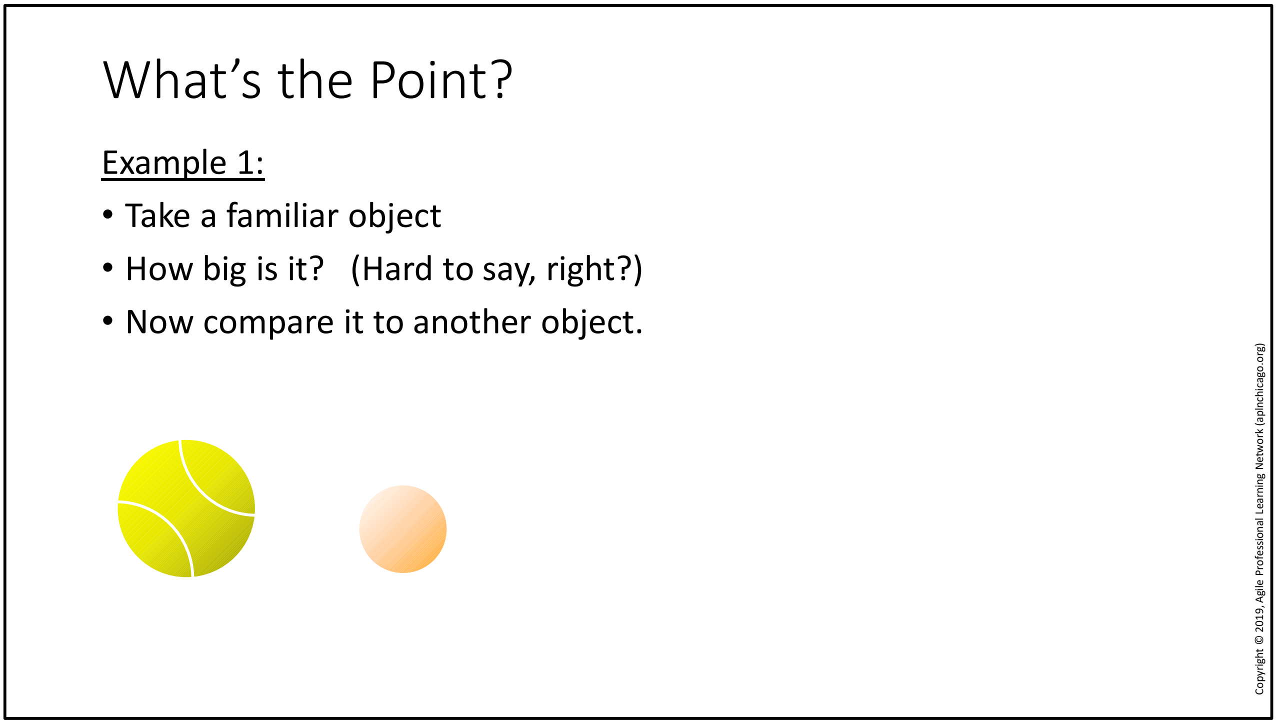


What’s the Point?

Example 1:

- Take a familiar object
- How big is it?   (Hard to say, right?)

What is a thing’s relative size?

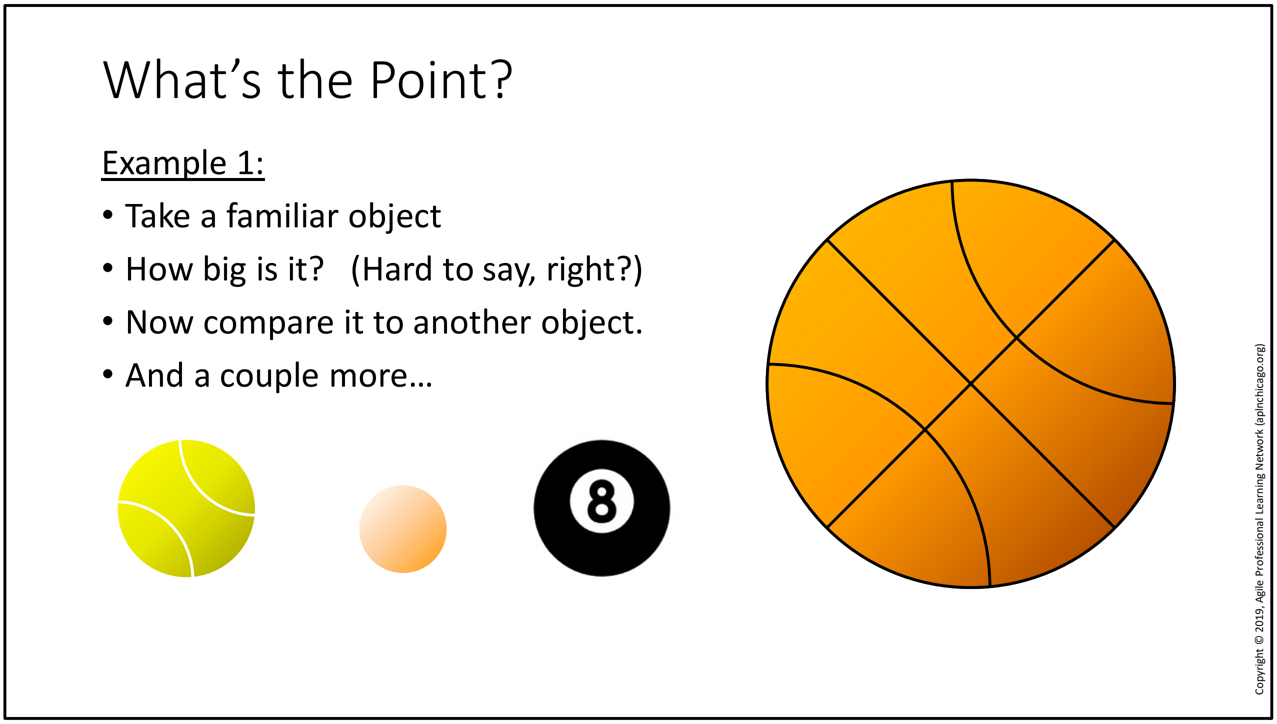


What’s the Point?

Example 1:

- Take a familiar object
- How big is it?   (Hard to say, right?)
- Now compare it to another object.

When you do relative sizing, you are comparing things to each other. You don’t need to know an exact value for the size of a thing to be able to say whether it is bigger, smaller, or about the same size as another thing.

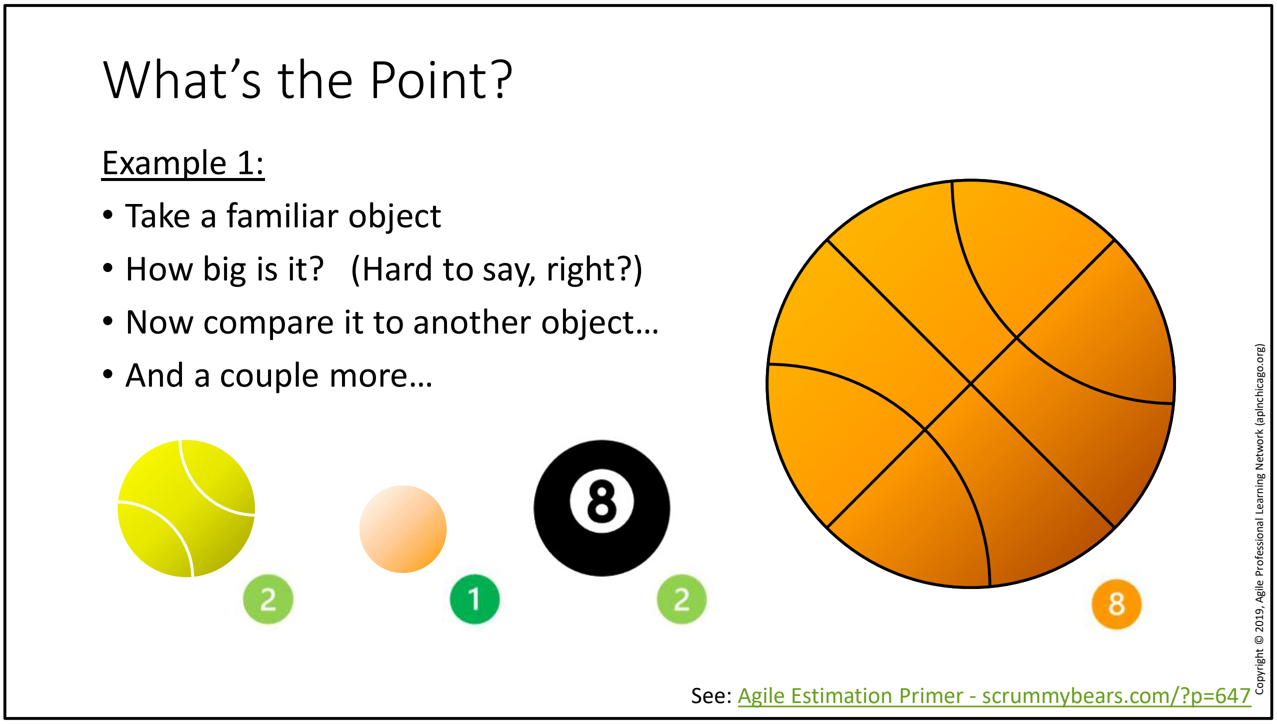


What’s the Point?

Example 1:

- Take a familiar object
- How big is it?   (Hard to say, right?)
- Now compare it to another object.
- And a couple more…

On this slide you can tell the ping-pong ball is smaller than the tennis ball. You can tell the billiard ball is bigger than the ping-pong ball, and the basketball is bigger still. But if you look a little closer, you can also probably say that the billiard ball and the tennis ball are very similar in size. Maybe not PRECISELY the same, but pretty darn close!



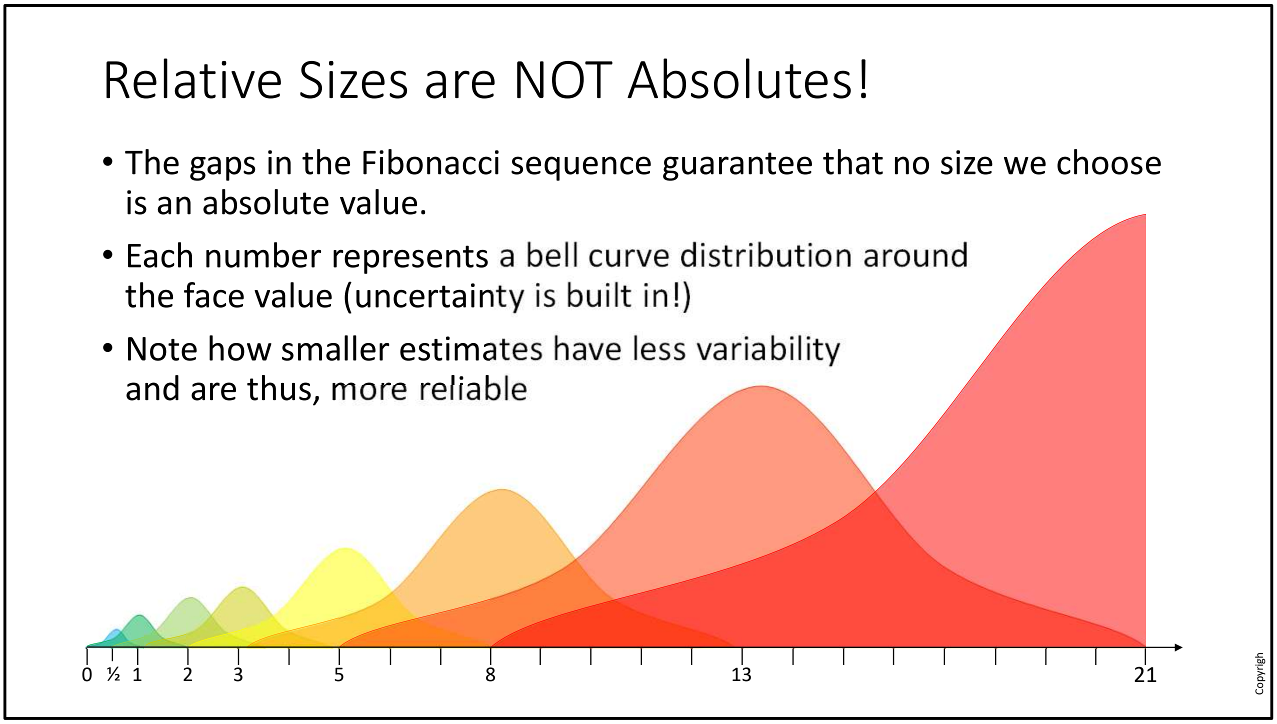
What's the Point?

Example 1:

- Take a familiar object
- How big is it?   (Hard to say, right?)
- Now compare it to another object…
- And a couple more…

See: Agile Estimation Primer - scrummybears.com/?p=647

Since the ping-pong ball is the smallest of the items, you could arbitrarily assign it to represent "1 point". So now look at the tennis ball. Is it bigger than the ping-pong ball? How much bigger? Maybe twice as big? Therefore, the Tennis and Billiard balls are both "2 points". And the basketball looks to be at least 4 times the diameter of the tennis ball, so let's call it "8 points".



Relative Sizes are NOT Absolutes!

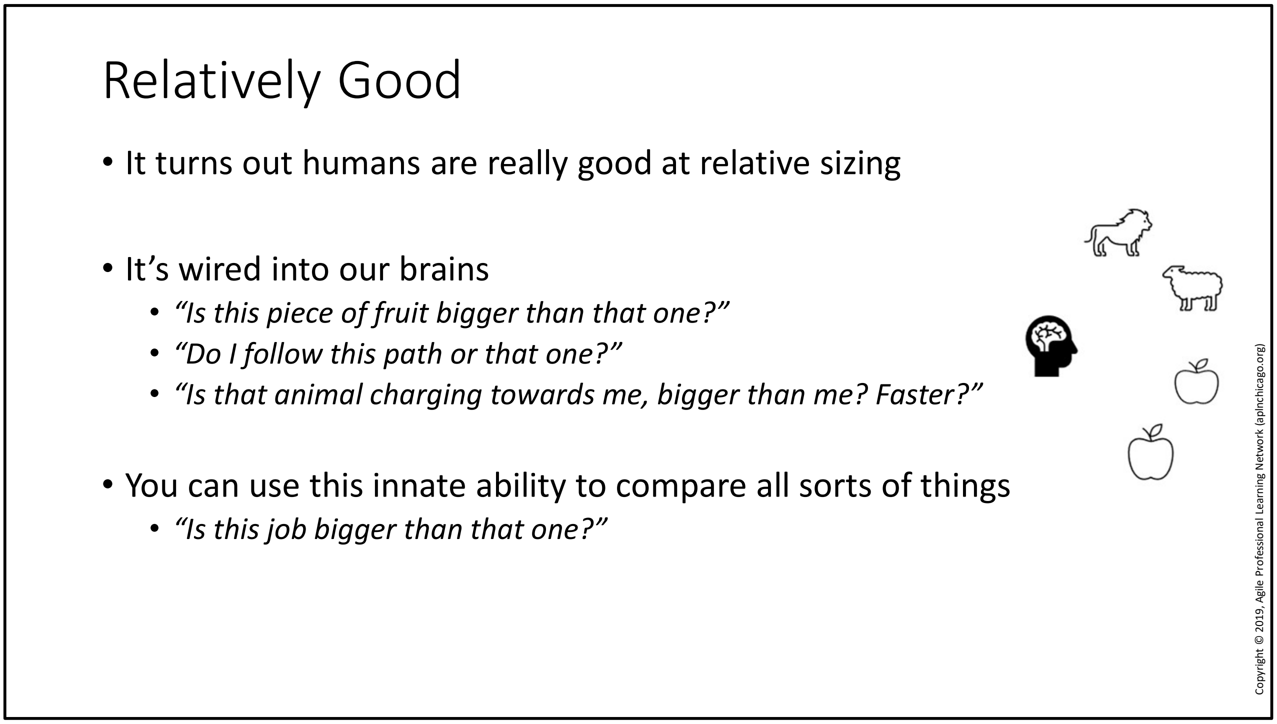
- The gaps in the Fibonacci sequence guarantee that no size we choose is an absolute value.
- Each number represents a bell curve distribution around the face value (uncertainty is built in!)
- Note how smaller estimates have less variability and are thus, more reliable

When we say that something is around the same size as a ONE, or something else is twice that size, we're not talking about an absolute valuation. Each Fibonacci estimate actually represents a range of values, with the bulk of things estimated at that value congregating around a certain dimension.

This should help ease your mind about relative sizing. It's okay (and even expected) that you may be wrong, and the bigger your estimate, the more wrong you are likely to be!

Teams that have been estimating for a while, start to realize they will generally (comfortably) deliver stories below a certain point value, and at the same time tend to struggle to reliably deliver stories above another point value. This information can be used to calibrate estimation!

For example: The fact that there is a bell of uncertainty around any value means that even if you sometimes succeed at delivering a "13" in a sprint, you probably fail an equal amount of the time. There would then be a strong argument to be made that a 13 is the absolute upper limit of any story you COULD take into a sprint, but because of the danger of not delivering, you would NEVER take on that risk, so you would cap the largest story you would consider bringing into a sprint at 8 points.



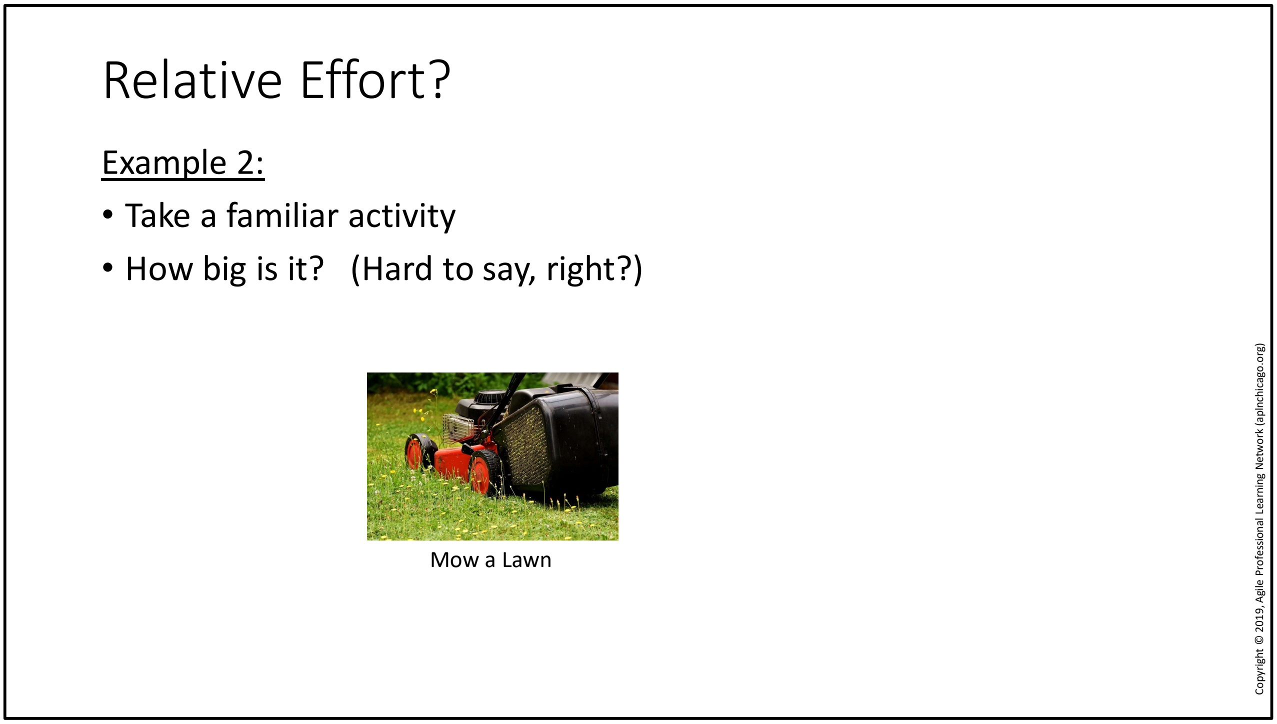
Relatively Good

- It turns out humans are really good at relative sizing
- It's wired into our brains
  - *"Is this piece of fruit bigger than that one?"*
  - *"Do I follow this path or that one?"*
  - *"Is that animal charging towards me, bigger than me? Faster?"*
- You can use this innate ability to compare all sorts of things
  - *"Is this job bigger than that one?"*

We are actually, really good at relative sizing. You've been doing it since you were born. It's an inherent survival instinct.

Anyone who has siblings can remember arguments about whose piece of cake was bigger than whose.

Let's leverage that instinct.



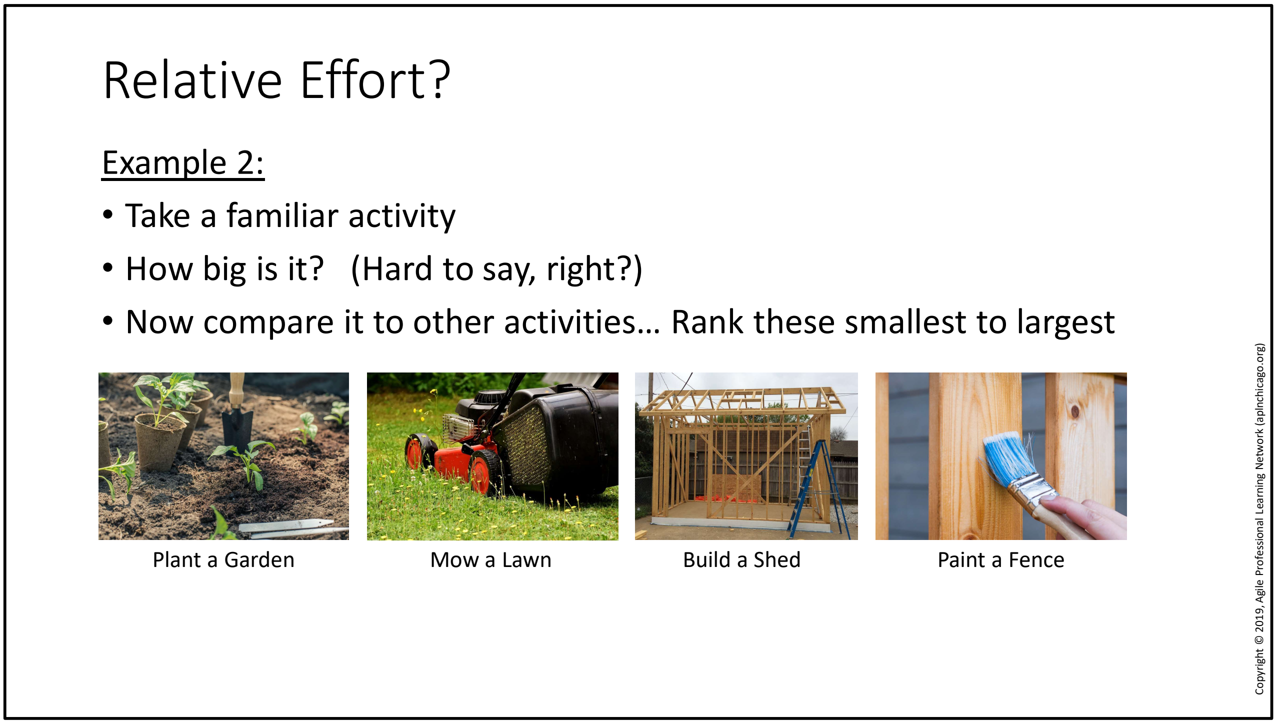
Relative Effort?

Example 2:

- Take a familiar activity
- How big is it?   (Hard to say, right?)

Mow a Lawn

The work we do is rarely as simple as the ball-sizing example would imply. Even the simplest of household chores have multiple layers of activity and complexity.



Relative Effort?

Example 2:

- Take a familiar activity
- How big is it?   (Hard to say, right?)
- Now compare it to other activities… Rank these smallest to largest

Plant a Garden

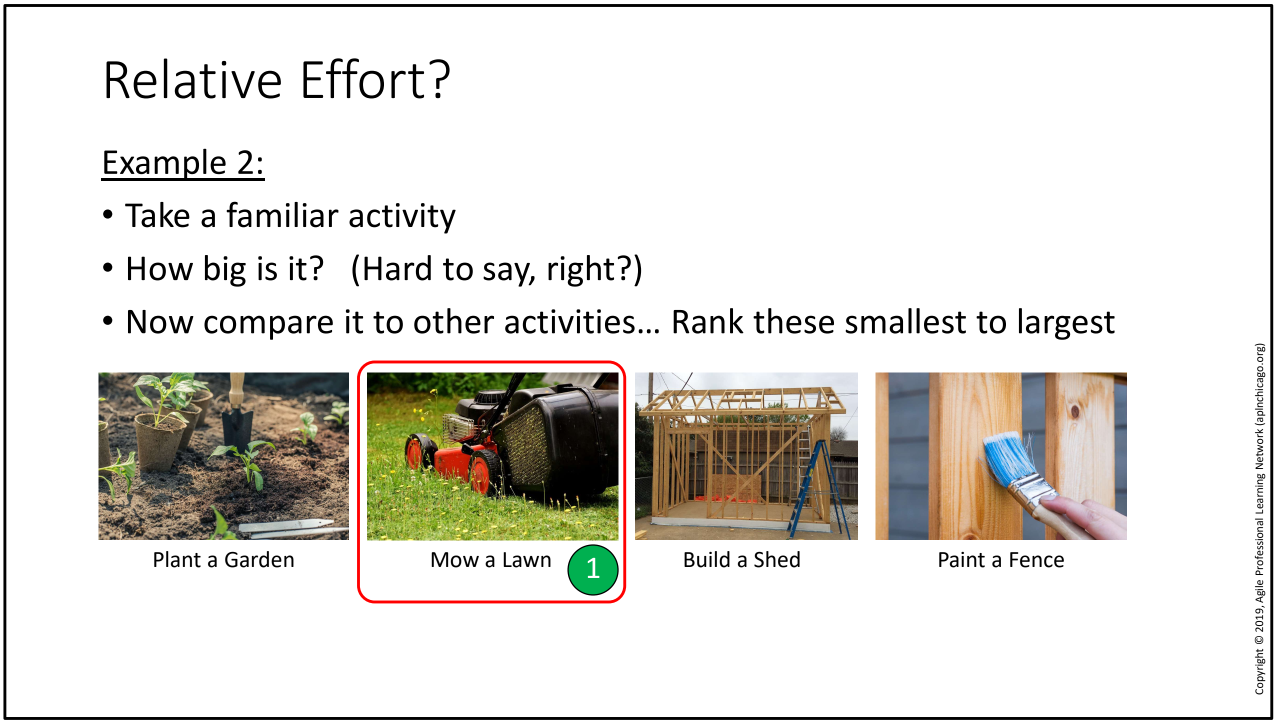
Mow a Lawn

Build a Shed

Paint a Fence

Look at these more complex examples. First, start by ranking them in effort and complexity from the simplest to the hardest.

Then consider how big they are RELATIVE to each other.



Relative Effort?

Example 2:

- Take a familiar activity
- How big is it? (Hard to say, right?)
- Now compare it to other activities… Rank these smallest to largest

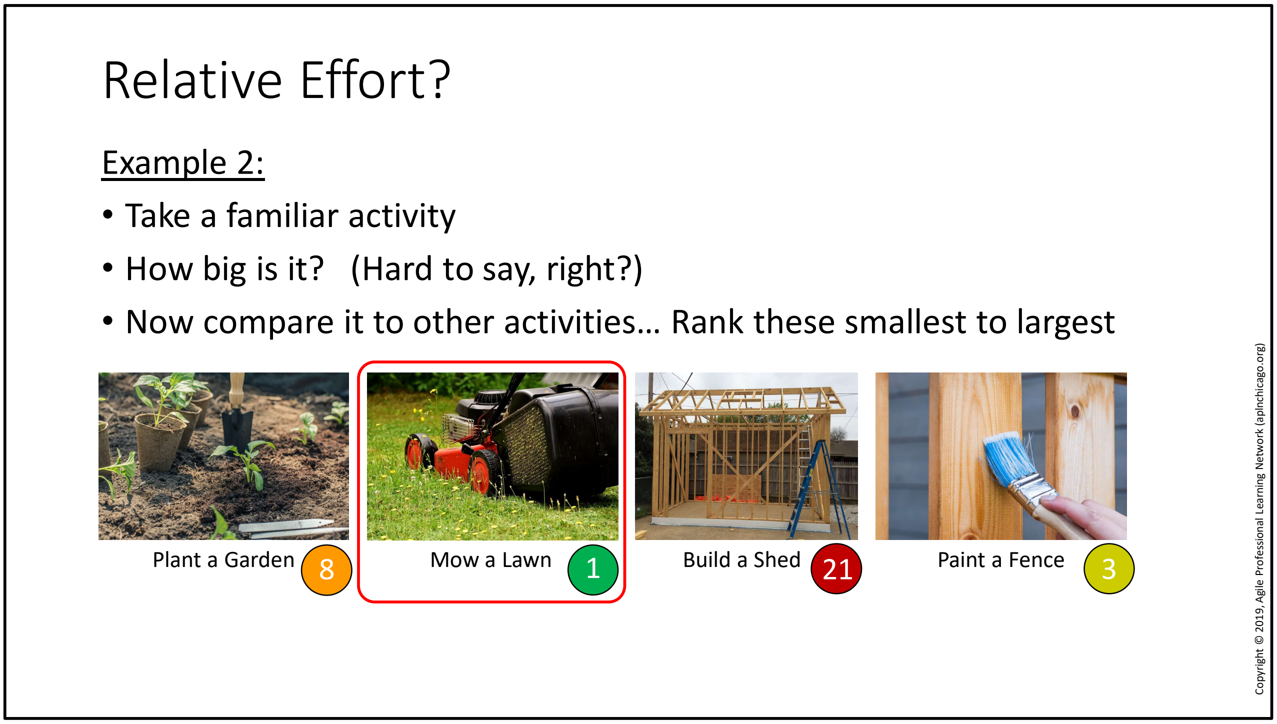
Plant a Garden

Mow a Lawn 1

Build a Shed

Paint a Fence

As with the ball example, we need a frame of reference. So let’s take the smallest one and arbitrarily declare it to be our baseline (1-point) activity.



Relative Effort?

Example 2:

- Take a familiar activity
- How big is it? (Hard to say, right?)
- Now compare it to other activities… Rank these smallest to largest

Plant a Garden 8

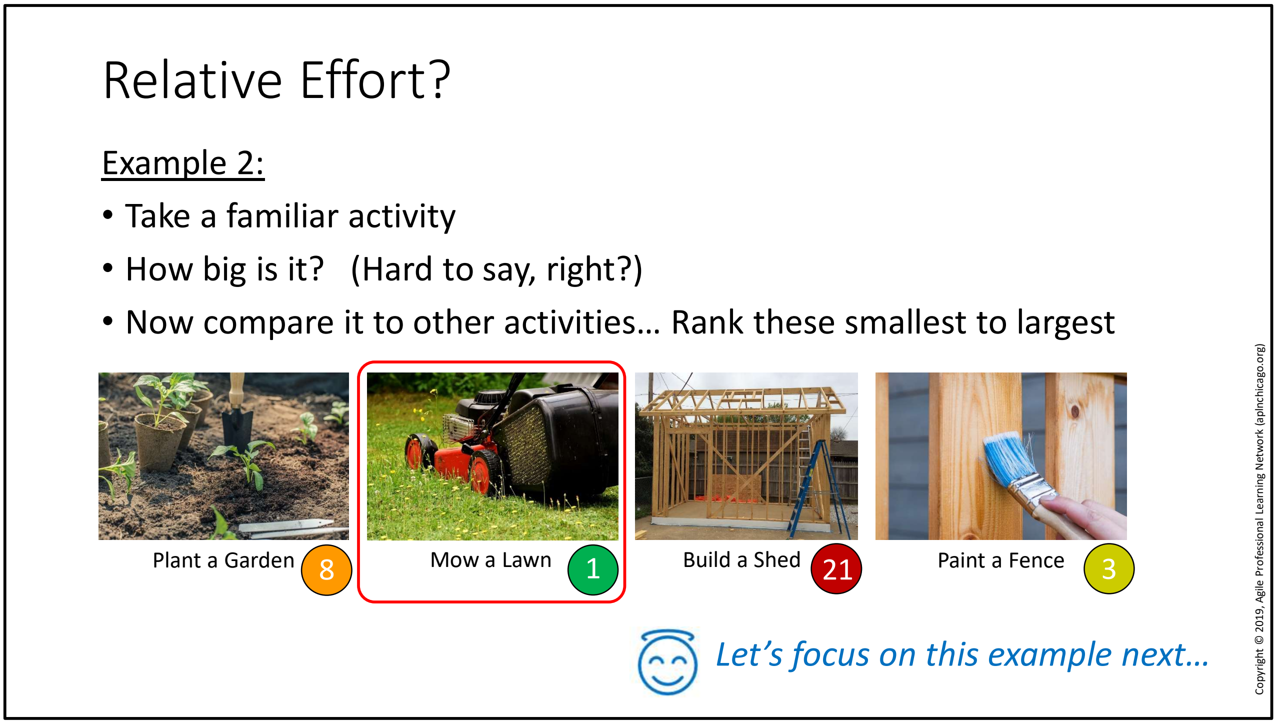
Mow a Lawn 1

Build a Shed 21

Paint a Fence 3

Remember, don’t think in terms of time. Consider the work involved. The complexity of the tasks necessary to complete it. The degree of care you need to employ to do a good QUALITY job.

This is where that Fibonacci sequence is going to come in handy!



Relative Effort?

Example 2:

- Take a familiar activity
- How big is it? (Hard to say, right?)
- Now compare it to other activities… Rank these smallest to largest

Plant a Garden 8

Mow a Lawn 1

Build a Shed 21

Paint a Fence 3

Let's focus on this example next…

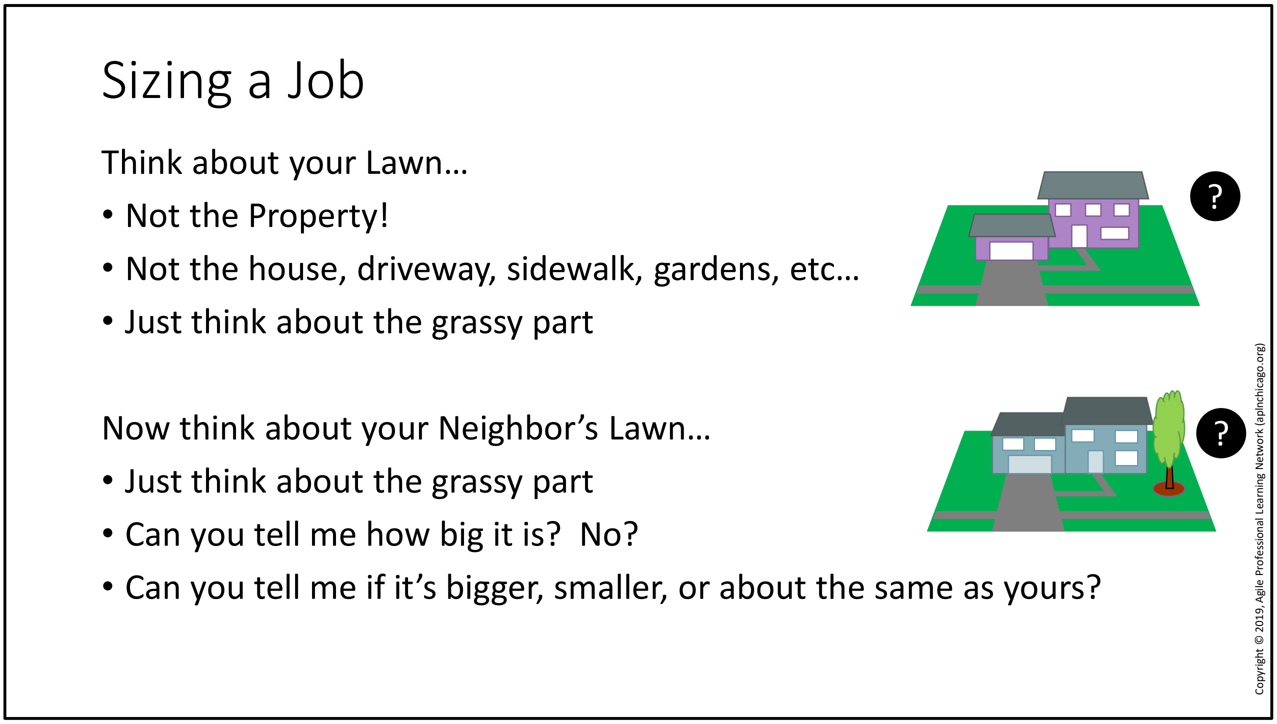
[animated] (replaces the 4 previous slides)

The work we do is rarely as simple as the ball-sizing example would imply. Even the simplest of household chores have multiple layers of activity and complexity.

Look at these more complex examples. First, start by ranking them in effort and complexity from the simplest to the hardest.
Then consider how big they are RELATIVE to each other. As with the ball example, we need a frame of reference. So, let's take the smallest one and arbitrarily declare it to be our baseline (1-point) activity.

Remember, don't think in terms of time. Consider the work involved. The complexity of the tasks necessary to complete it. The degree of care you need to employ to do a good QUALITY job.

This is where that Fibonacci sequence is going to come in handy!



Sizing a Job

Think about your Lawn…

- Not the Property!
- Not the house, driveway, sidewalk, gardens, etc…
- Just think about the grassy part

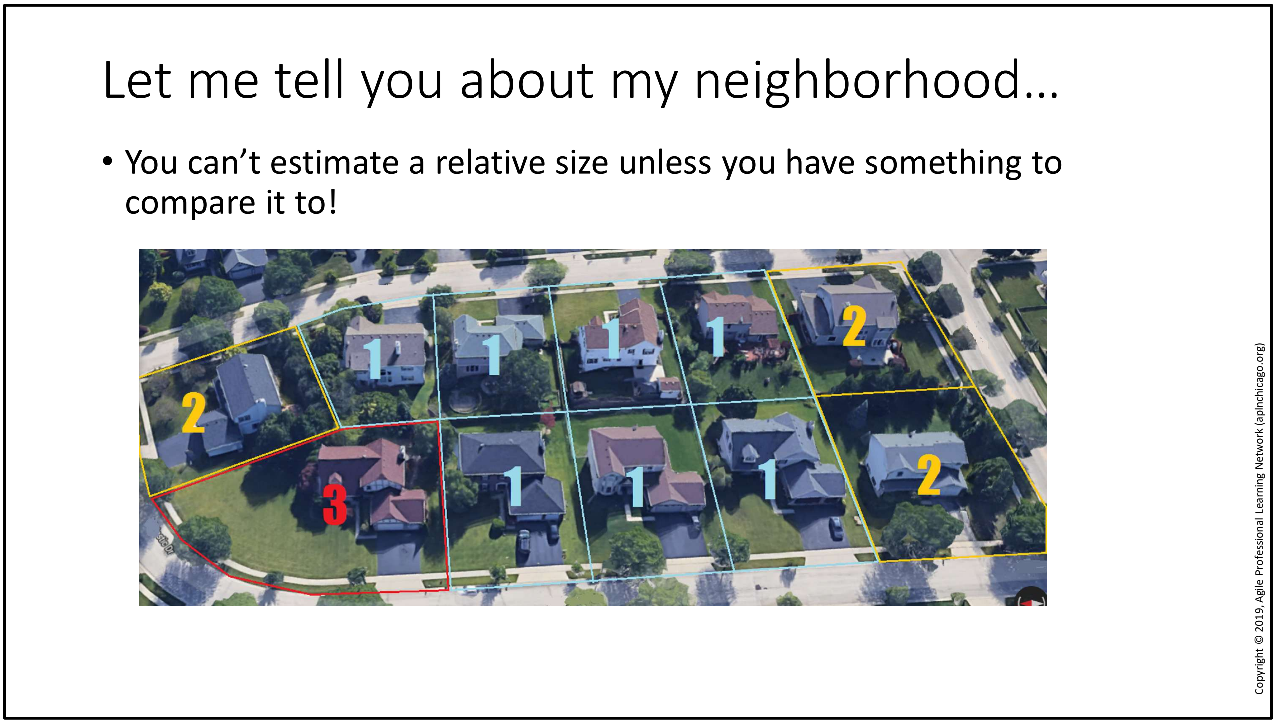
Now think about your Neighbor’s Lawn…

- Just think about the grassy part
- Can you tell me how big it is? No?
- Can you tell me if it’s bigger, smaller, or about the same as yours?

This is a huge leap. Can you visualize two complex objects and give me an answer?

The truth is, I’m playing the odds here. Many of you are probably in neighborhoods where the lots are uniformly sized. If you compare relative sizes, and they come up pretty equal, that’s not helpful. The devil is in the differences.

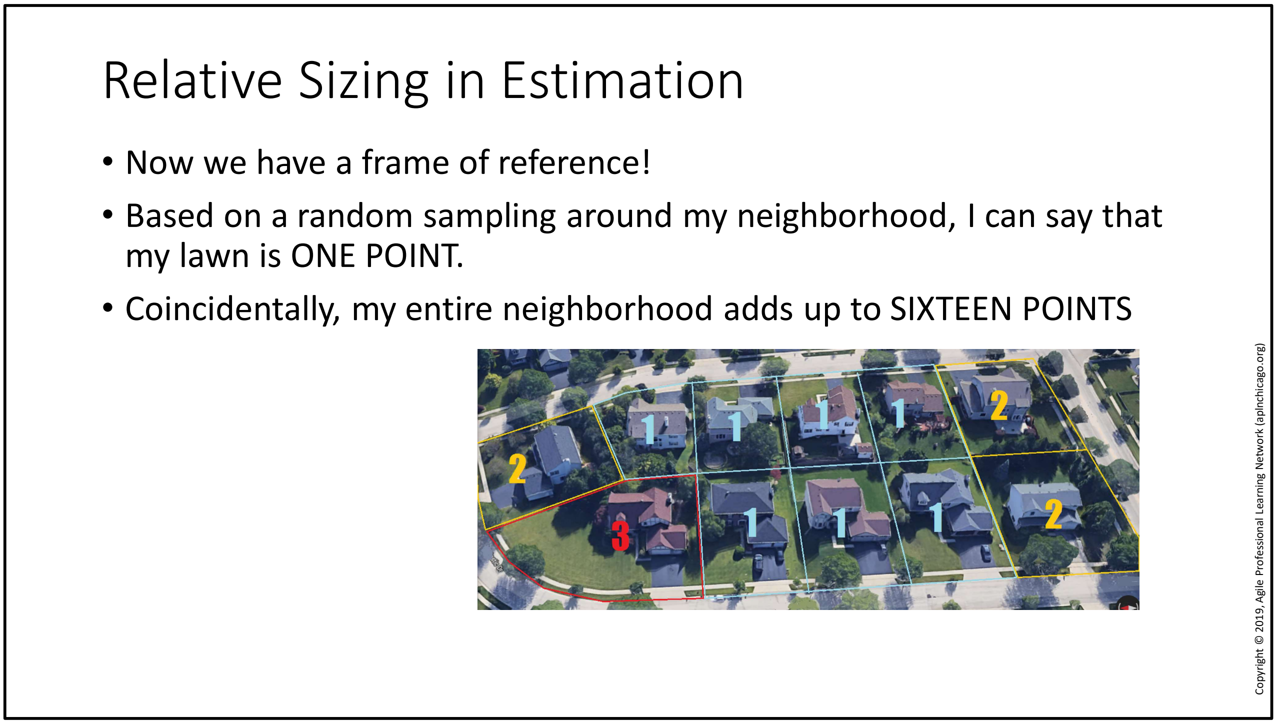
Of course, if all the things you need to estimate are really similar, then relative size is unimportant. Call them all “Ones”.



Let me tell you about my neighborhood…

- You can’t estimate a relative size unless you have something to compare it to!

I’m going to establish an arbitrary starting point now. Based on my understanding so far, my yard and my neighbors’ are all 1 point yards, the corner houses have 2-point yards, and the odd-shaped yard a 3-point yard. The chief advantage at this point, is this measurement was FAST. Not a lot of time spent agonizing over it. It was a quick 1,2, 3 and we’re ready to move on.



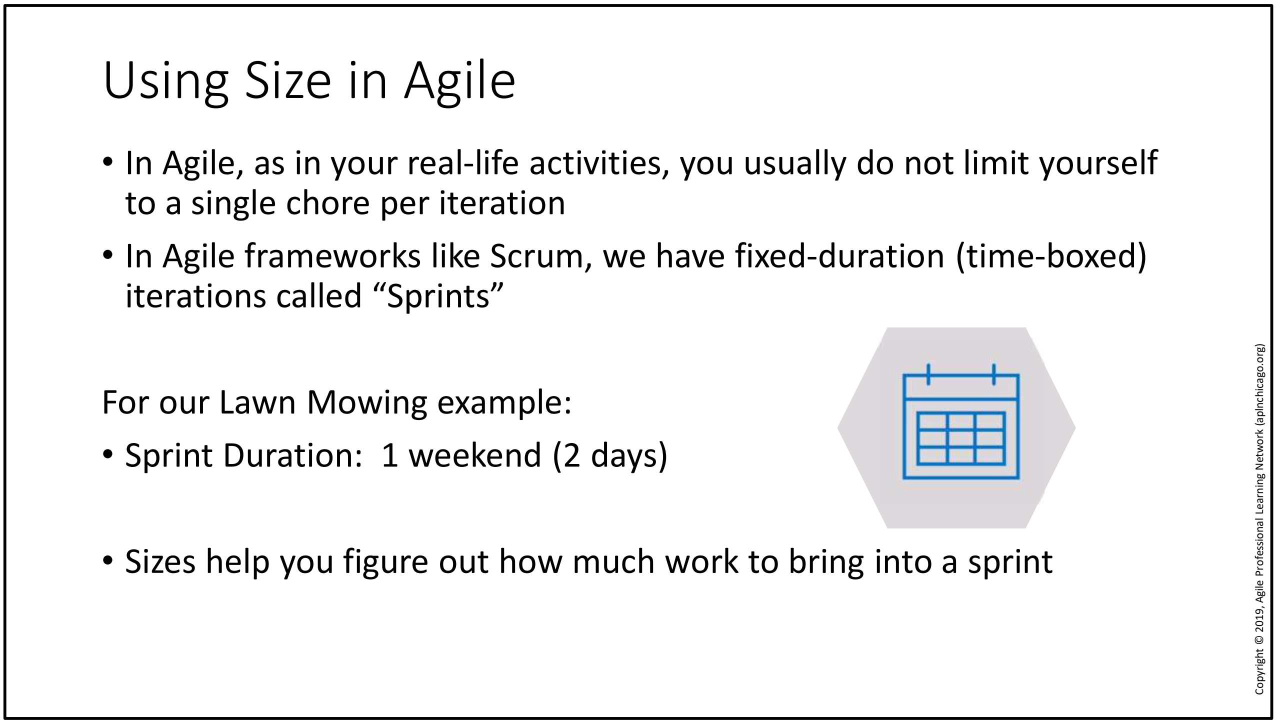
Relative Sizing in Estimation

- Now we have a frame of reference!
- Based on a random sampling around my neighborhood, I can say that my lawn is ONE POINT.
- Coincidentally, my entire neighborhood adds up to SIXTEEN POINTS

1
1
1
1
2
2
3
1
1
1
2

All of the lawns on my block add up to 16 total points. What does that mean?

You don't have enough information to ascribe meaning to that … yet.



Using Size in Agile

- In Agile, as in your real-life activities, you usually do not limit yourself to a single chore per iteration
- In Agile frameworks like Scrum, we have fixed-duration (time-boxed) iterations called “Sprints”

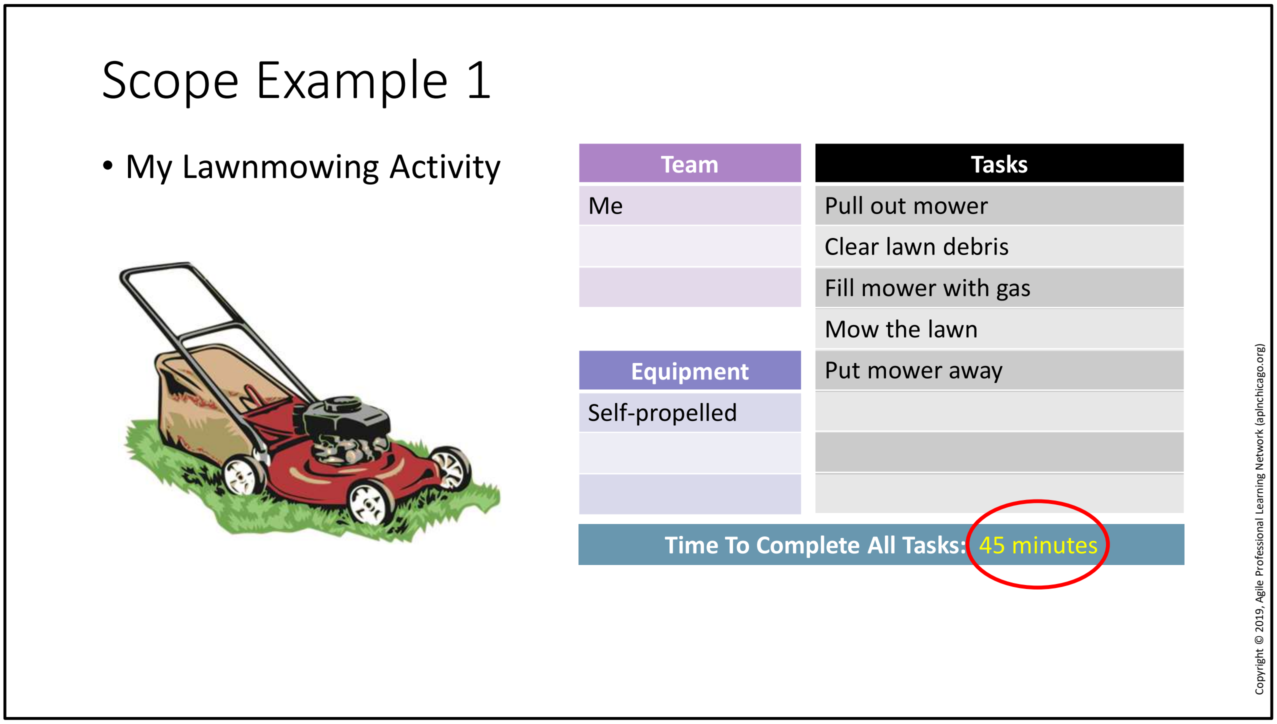
For our Lawn Mowing example:

- Sprint Duration: 1 weekend (2 days)

- Sizes help you figure out how much work to bring into a sprint

In Scrum, we allocate timeboxes of activity called “Sprints”, and try to forecast how much work we can accomplish within that timebox. Estimation of relative Effort can help us figure out how much work to bring into a sprint, by looking at how much we’ve accomplished in past iterations.

For the ensuing examples, just because we’re homeowners with other jobs, we can only work on maintaining our homes on the weekend. So Let’s say that each Sprint in our homeowner backlog is 2 days long (Saturday and Sunday).



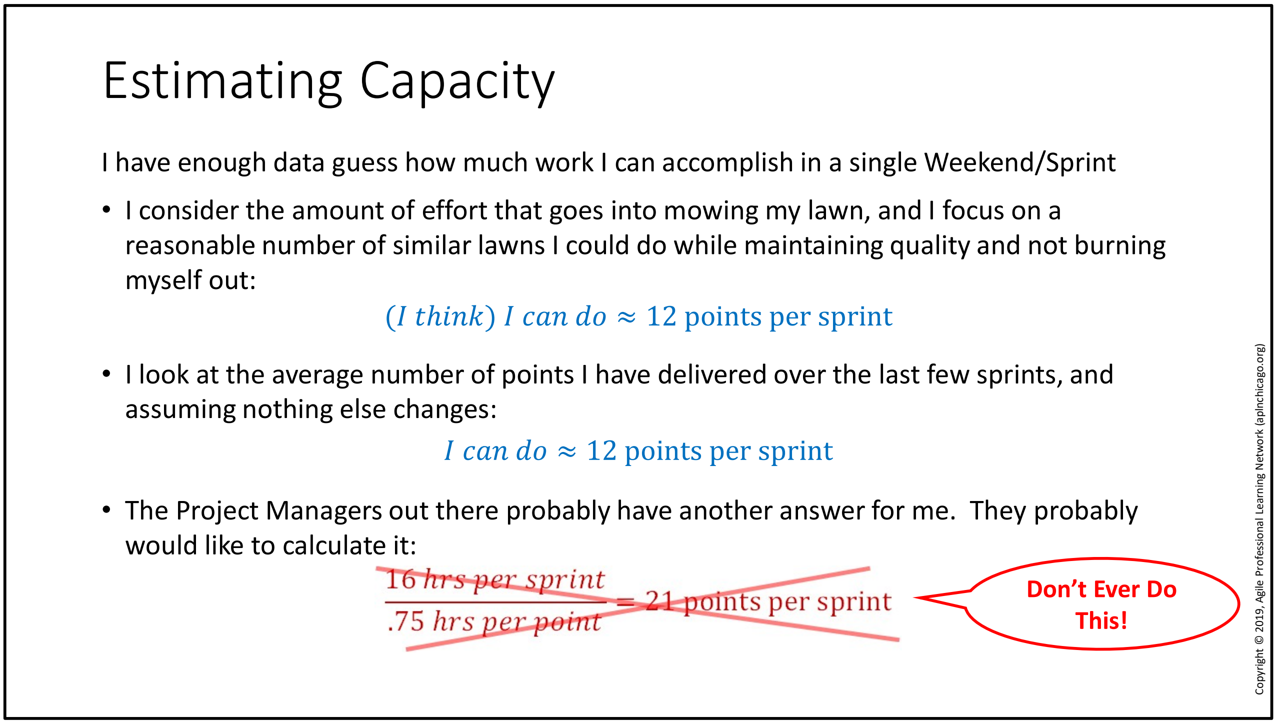
Scope Example 1

- My Lawnmowing Activity

Team	Tasks
Me	Pull out mower
	Clear lawn debris
	Fill mower with gas
	Mow the lawn
Equipment	Put mower away
Self-propelled	
Time To Complete All Tasks: 45 minutes	

Meet my team. Er… Me. I have a relatively simple, self-propelled lawn mower. For me, mowing the lawn had a relatively simple definition of done: “All the grass is cut to a uniform length.” No frills. No drama. Simple.

Remember, I have decided that I have a one-point lawn. It HAPPENS to take me 45 minutes to mow the lawn…



Estimating Capacity

I have enough data guess how much work I can accomplish in a single Weekend/Sprint

- I consider the amount of effort that goes into mowing my lawn, and I focus on a reasonable number of similar lawns I could do while maintaining quality and not burning myself out:

$$(I\ think)\ I\ can\ do \approx 12 \text{ points per sprint}$$

- I look at the average number of points I have delivered over the last few sprints, and assuming nothing else changes:

$$I\ can\ do \approx 12 \text{ points per sprint}$$

- The Project Managers out there probably have another answer for me. They probably would like to calculate it:

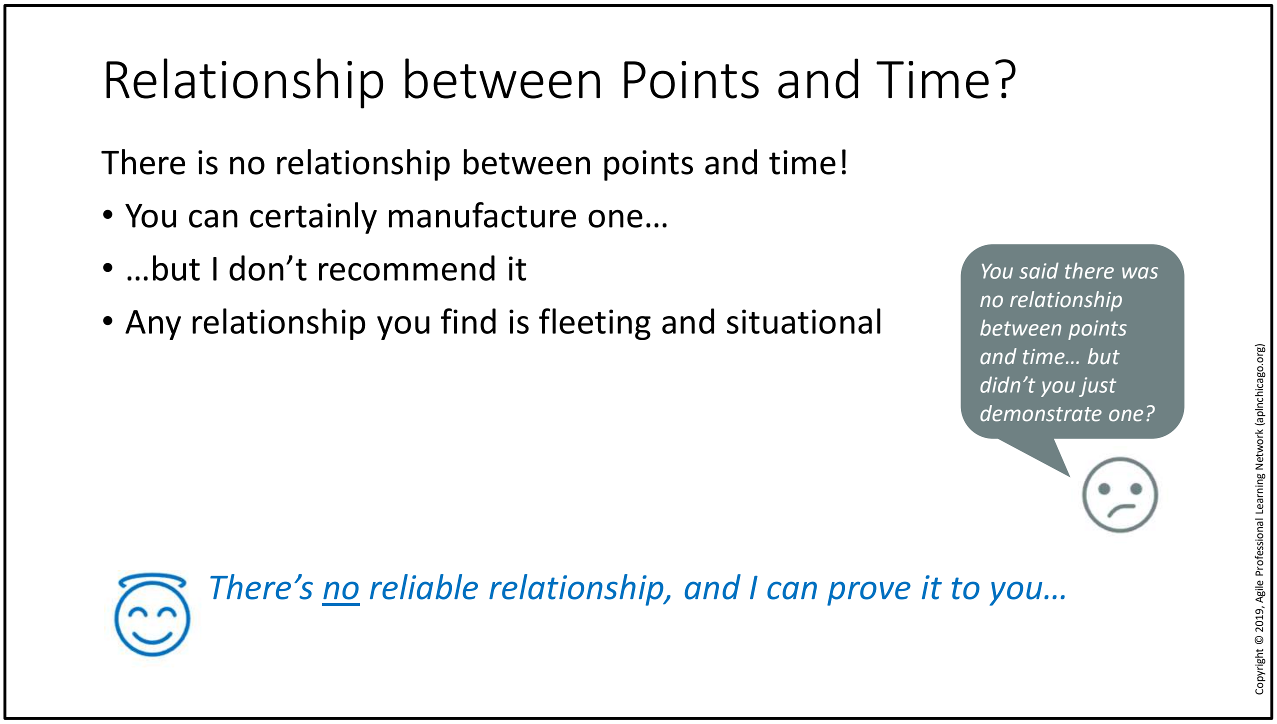
$$\cancel{\frac{16\ hrs\ per\ sprint}{.75\ hrs\ per\ point} = 21 \text{ points per sprint}}$$

In the agile world, Capacity has a slightly different meaning than it does in traditional Project Management.

A team will forecast their capacity (their ability to do quality work) based on a number of factors. We want this to be something that they have a hand in defining.

Traditional Project Management would tell the team members how much time they have available to perform their work and set the expectation that the team members will fill that time 'working'.

Given what we know so far, can you think of why this is a dangerous practice?



Relationship between Points and Time?

There is no relationship between points and time!

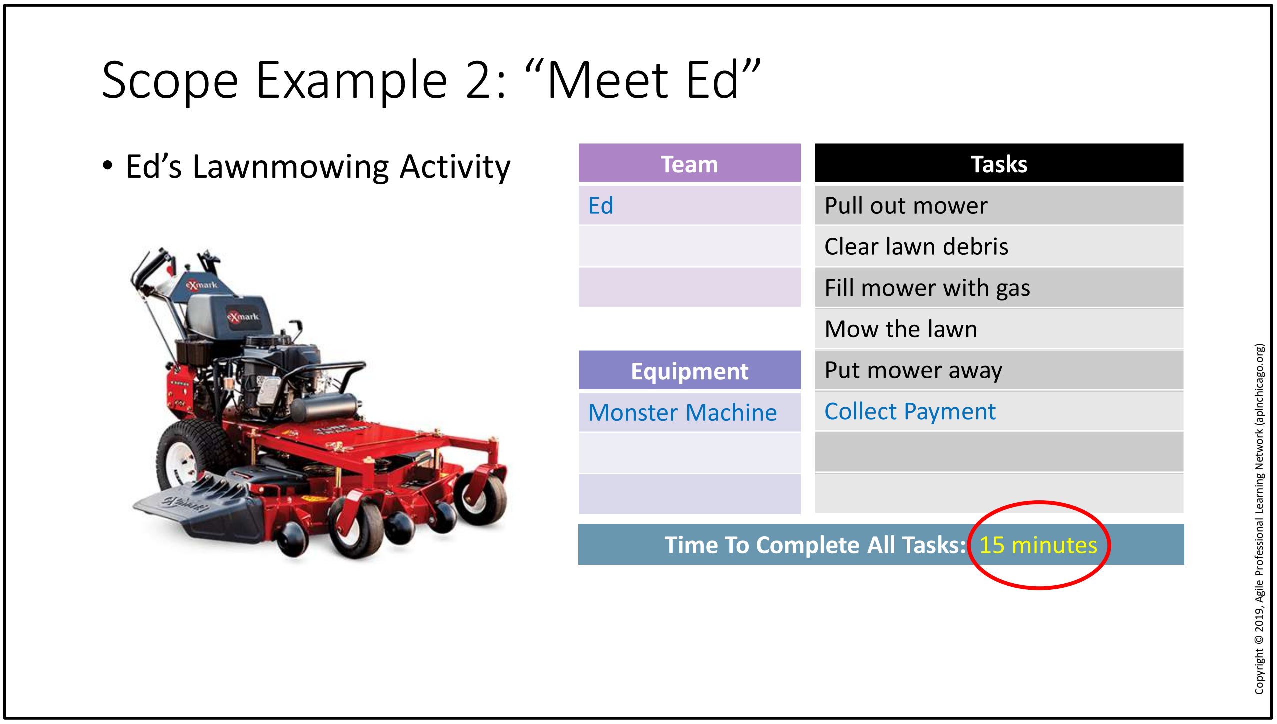
- You can certainly manufacture one…
- …but I don't recommend it
- Any relationship you find is fleeting and situational

You said there was no relationship between points and time… but didn't you just demonstrate one?

There's <u>no</u> reliable relationship, and I can prove it to you…

The skeptics among you are going to find this statement to be difficult. I pretty much just told you it takes me 45 minutes to mow my 1-point lawn. So, why is it wrong to look at available time as a way of estimating capacity?

Trust me, it's a bad practice. And I'll prove it to you.



Scope Example 2: “Meet Ed”

- Ed’s Lawnmowing Activity

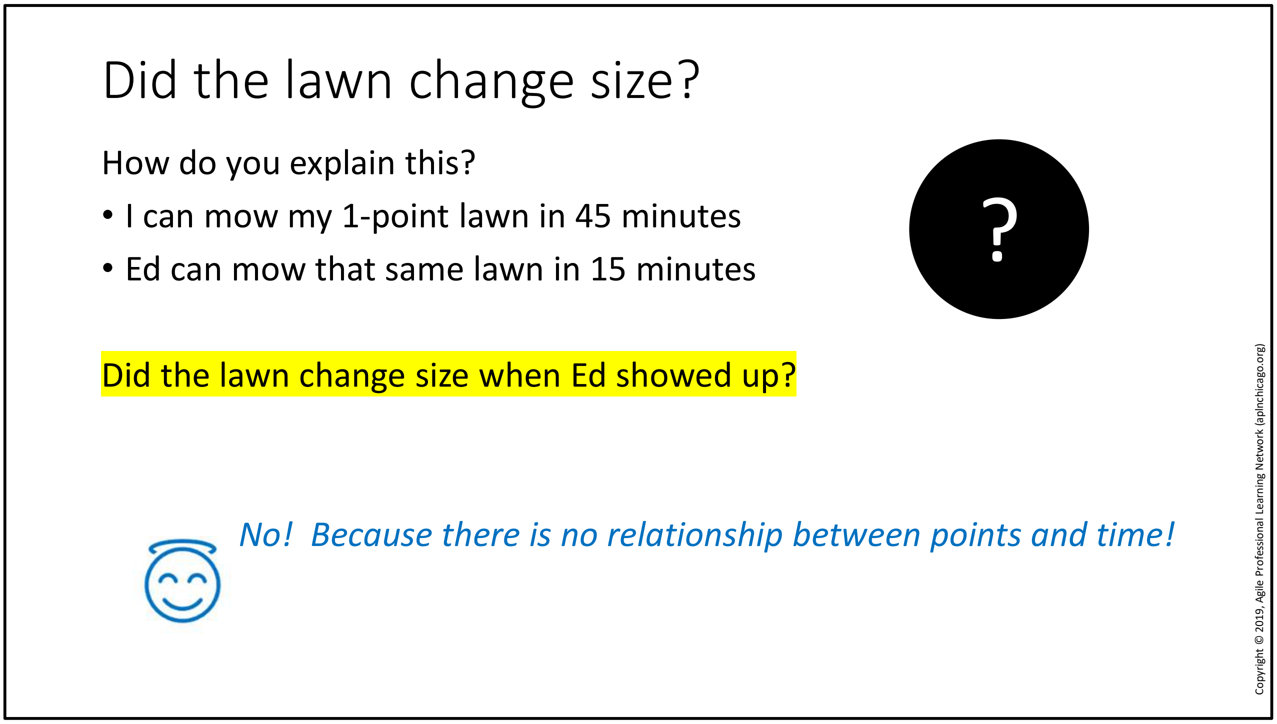
Team	Tasks
Ed	Pull out mower
	Clear lawn debris
	Fill mower with gas
	Mow the lawn
Equipment	Put mower away
Monster Machine	Collect Payment

Time To Complete All Tasks: 15 minutes

A few years ago, I hired someone to mow my lawn for me. Don’t judge. It was a rough summer, and I just needed a little help. Besides, once you hire these guys, they keep showing up to do their job. (weird)

Anyway, “Ed” has one of those big, zero-turn radius monster machines. The kind with a little wheeled step-pad on the back, so you can glide along behind the mower while it does all the work.

Get this: Ed pulled that bad-boy off his trailer, mounted up and mowed my entire lawn in only TEN MINUTES. He then loaded it back onto the trailer, came to the door, and collected his payment. From arrival to departure, he was done in 15 total minutes.



Did the lawn change size?

How do you explain this?

- I can mow my 1-point lawn in 45 minutes
- Ed can mow that same lawn in 15 minutes

?

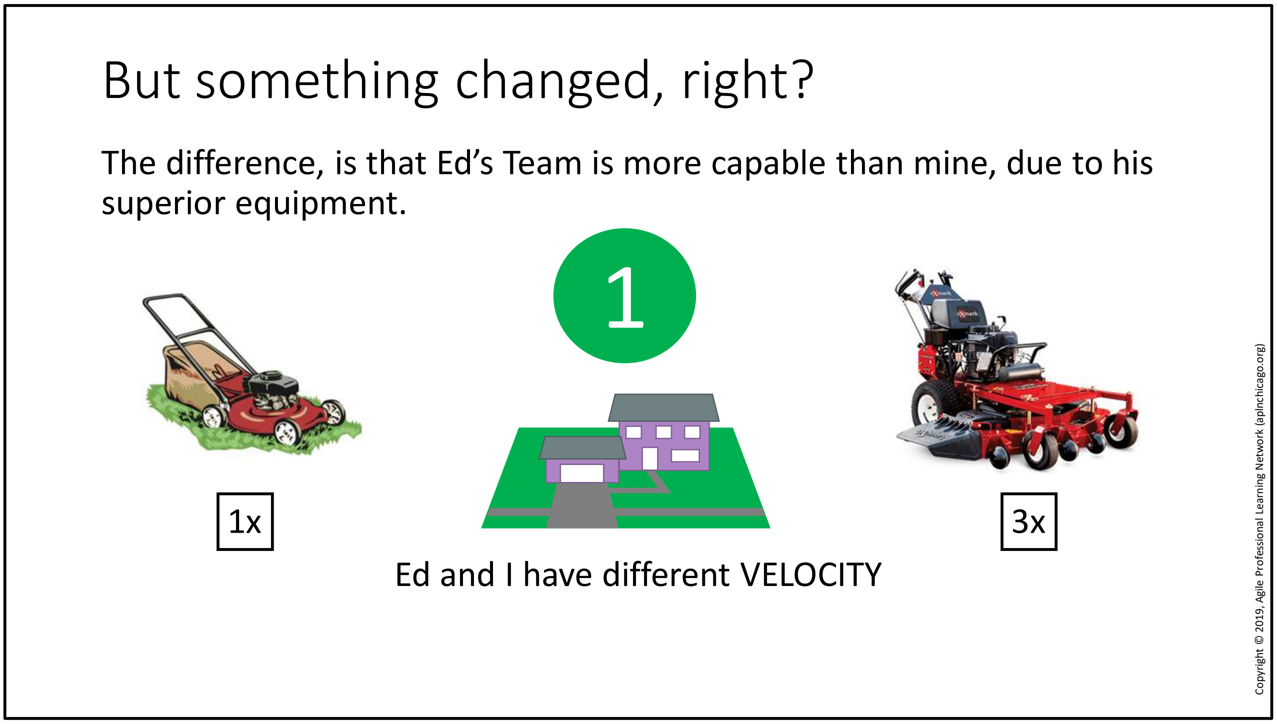
Did the lawn change size when Ed showed up?

No! Because there is no relationship between points and time!

Which begs the question: Did my lawn change size?

No, of course not!

As we've already established. Our estimation model using Points is designed to reinforce that there is not a relationship between Points and Time.



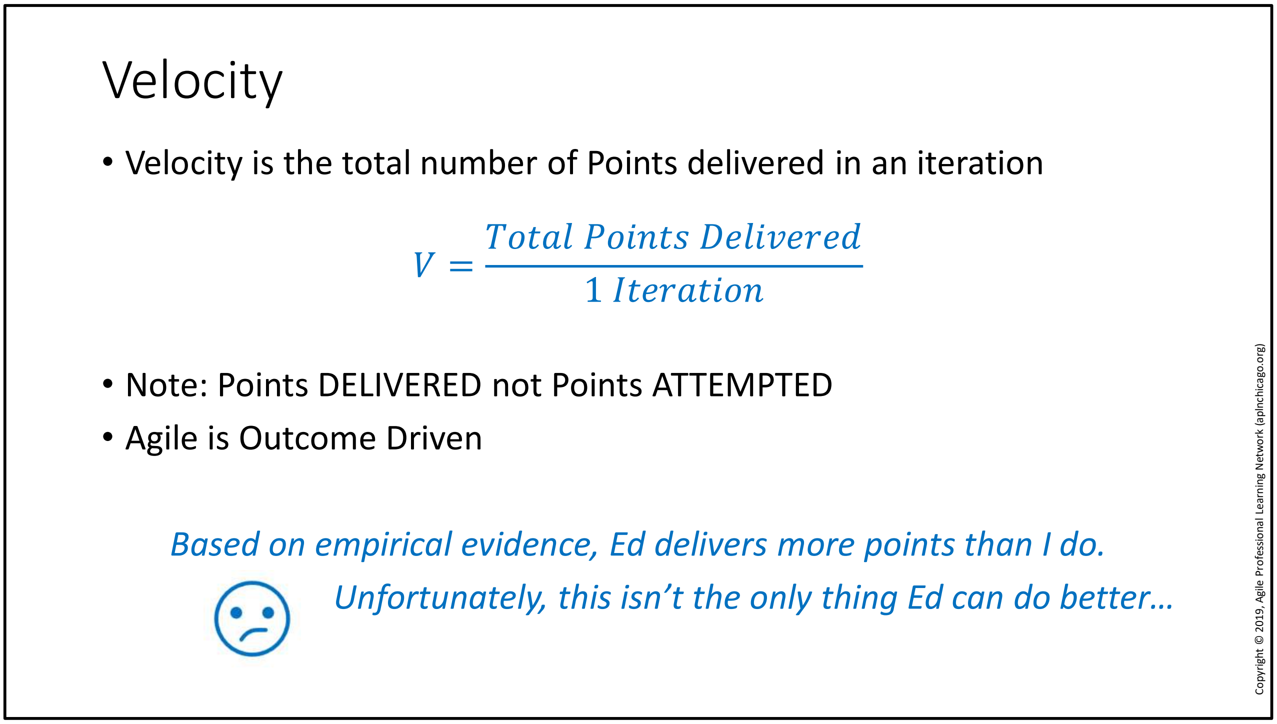
But something changed, right?

The difference, is that Ed's Team is more capable than mine, due to his superior equipment.

Ed and I have different VELOCITY

Just because Ed can finish the lawn faster than I can, doesn't mean the lawn changes size. It is in fact, still the same 1-point postage stamp it always has been.

The difference is that Ed is capable of doing more lawns like mine in the same time it takes me to do mine. Almost 3 times as many!



Velocity

- Velocity is the total number of Points delivered in an iteration

$$V = \frac{Total\ Points\ Delivered}{1\ Iteration}$$

- Note: Points DELIVERED not Points ATTEMPTED
- Agile is Outcome Driven

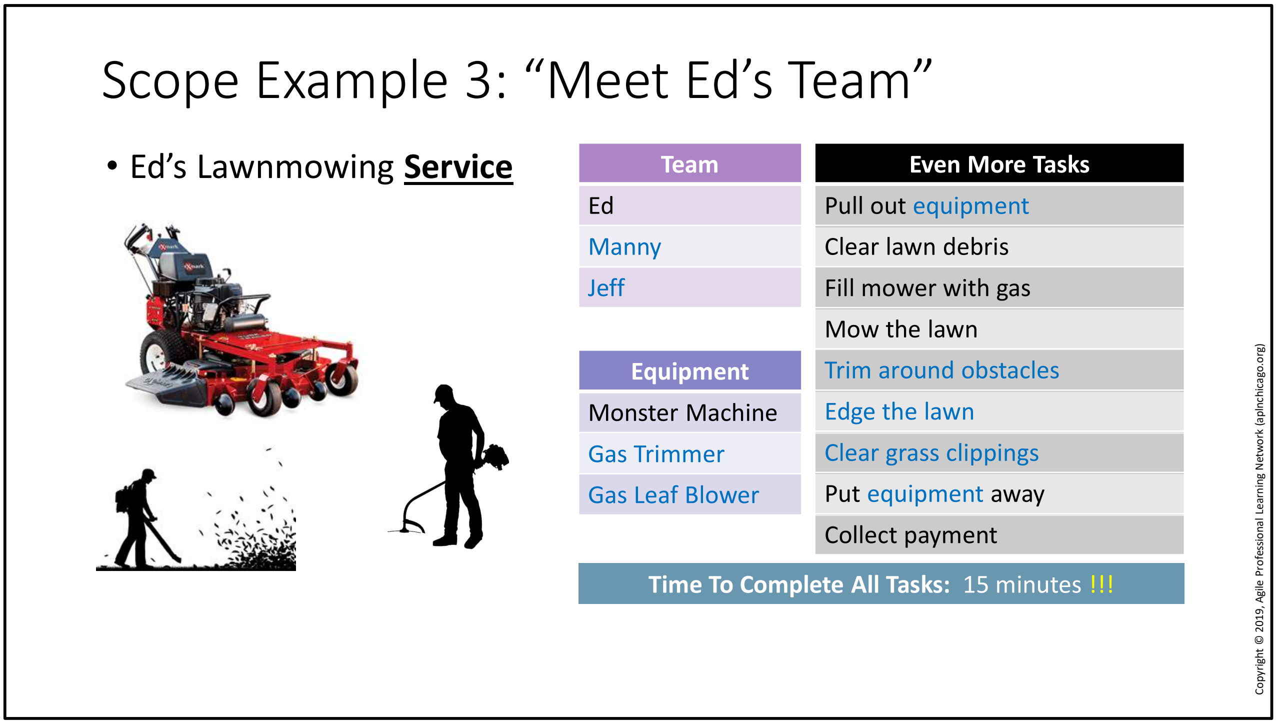
Based on empirical evidence, Ed delivers more points than I do.

😕 *Unfortunately, this isn't the only thing Ed can do better…*

Capacity was a forecast of capability. It represents how much work a team thinks it can accomplish in the next iteration.

Velocity is more meaningful. It represents how much work the team actually accomplished during that iteration. It is a lagging indicator based on OUTCOMES.

In terms of mowing, Ed DELIVERS far more work than I do. That's bad enough. But unfortunately for me, it's not the only thing Ed does better…



Scope Example 3: “Meet Ed’s Team”

- Ed’s Lawnmowing **Service**

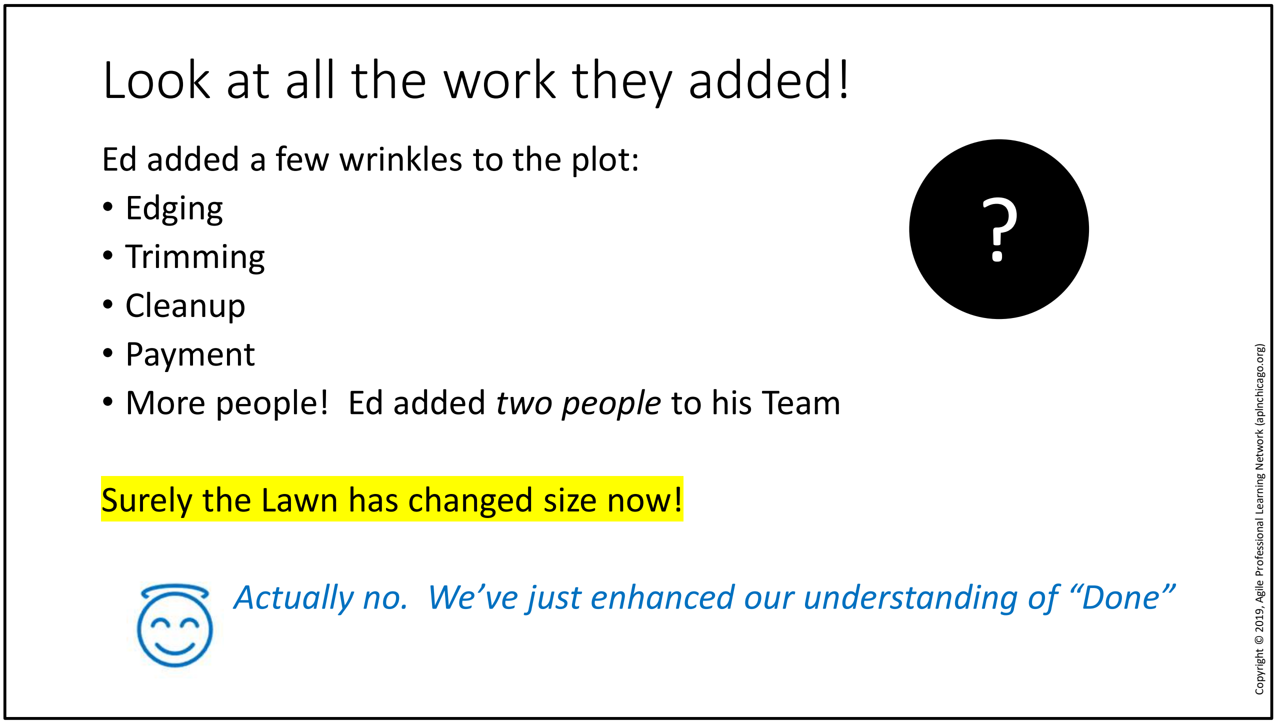
Team
Ed
Manny
Jeff

Equipment
Monster Machine
Gas Trimmer
Gas Leaf Blower

Even More Tasks
Pull out equipment
Clear lawn debris
Fill mower with gas
Mow the lawn
Trim around obstacles
Edge the lawn
Clear grass clippings
Put equipment away
Collect payment

Time To Complete All Tasks: 15 minutes !!!

Ed didn’t come alone. Ed brought friends. While Ed is out Mowing, Manny and Jeff are out trimming the edges of the lawn and cleaning up debris. They are adding additional services that I haven’t been doing. And despite the extra work, Ed’s team still completes the entire exercise in only 15 minutes of clock time.



Look at all the work they added!

Ed added a few wrinkles to the plot:

- Edging
- Trimming
- Cleanup
- Payment
- More people! Ed added *two people* to his Team

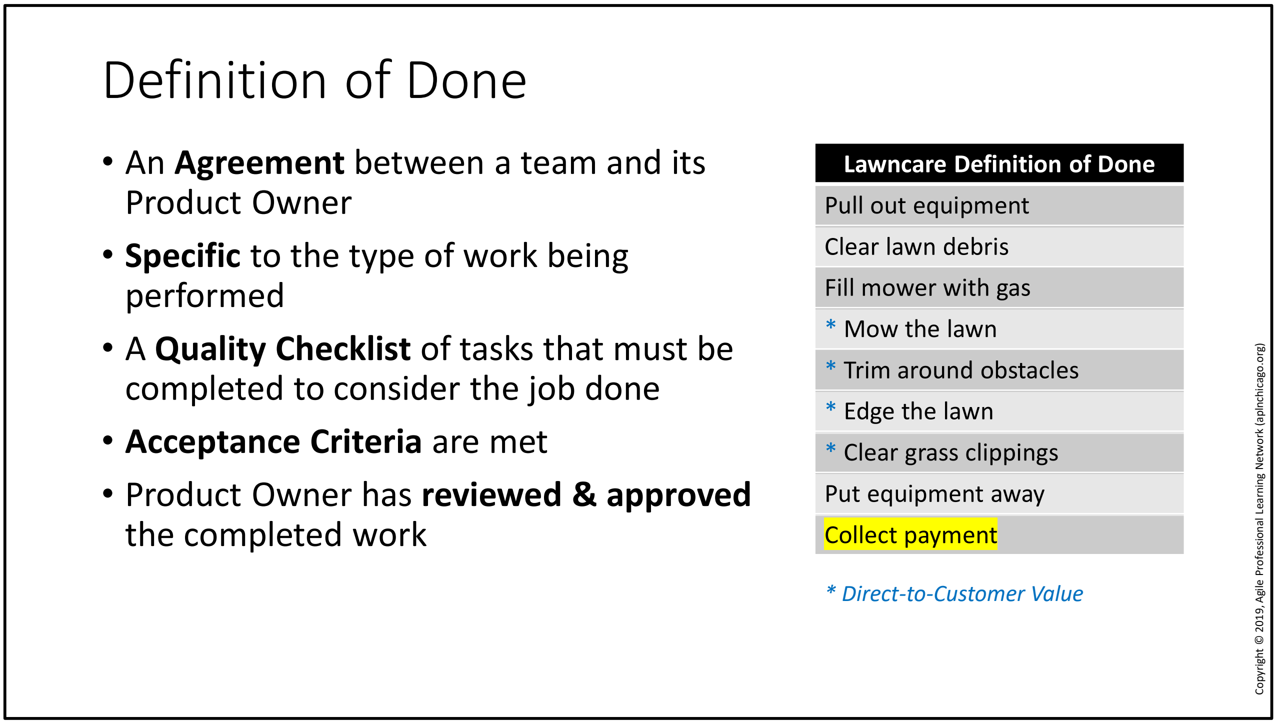
?

Surely the Lawn has changed size now!

😇 *Actually no. We've just enhanced our understanding of "Done"*

Given all this extra work, and considering there are more people necessary to complete it, surely the size of the lawn has changed now.

It has NOT changed size. Because Ed's team performs that same level of service for all the lawns they mow. This is a consistent level of effort, therefore the estimate does not need to change!



Definition of Done

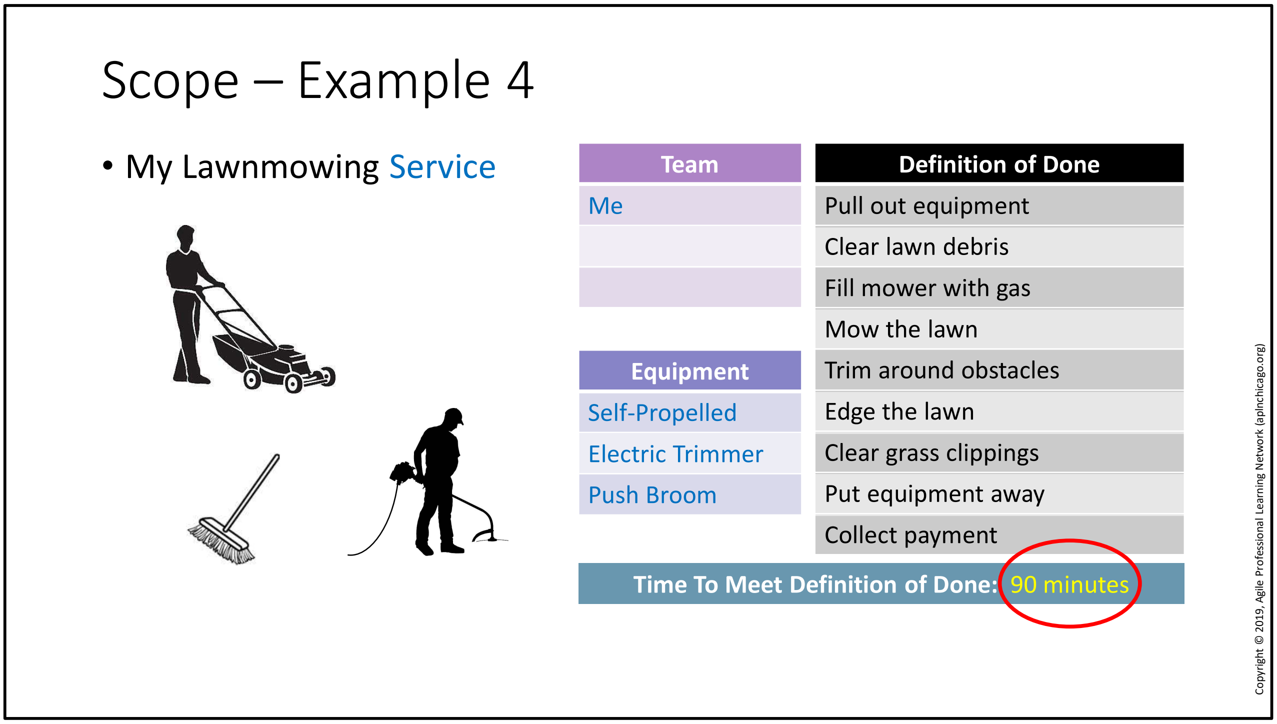
- An **Agreement** between a team and its Product Owner
- **Specific** to the type of work being performed
- A **Quality Checklist** of tasks that must be completed to consider the job done
- **Acceptance Criteria** are met
- Product Owner has **reviewed & approved** the completed work

Lawncare Definition of Done
Pull out equipment
Clear lawn debris
Fill mower with gas
* Mow the lawn
* Trim around obstacles
* Edge the lawn
* Clear grass clippings
Put equipment away
Collect payment

** Direct-to-Customer Value*

The introduction of Ed's team into the mix has enhanced our understanding of what it means to do a quality job. In agile, we enumerate these standard practices that apply to all work done by the team. We call that quality standard, "Definition of Done".

When an agile team declares work to be "Done", we can be assured that they have taken these standards into account.



Scope – Example 4

- My Lawnmowing Service

Team
Me

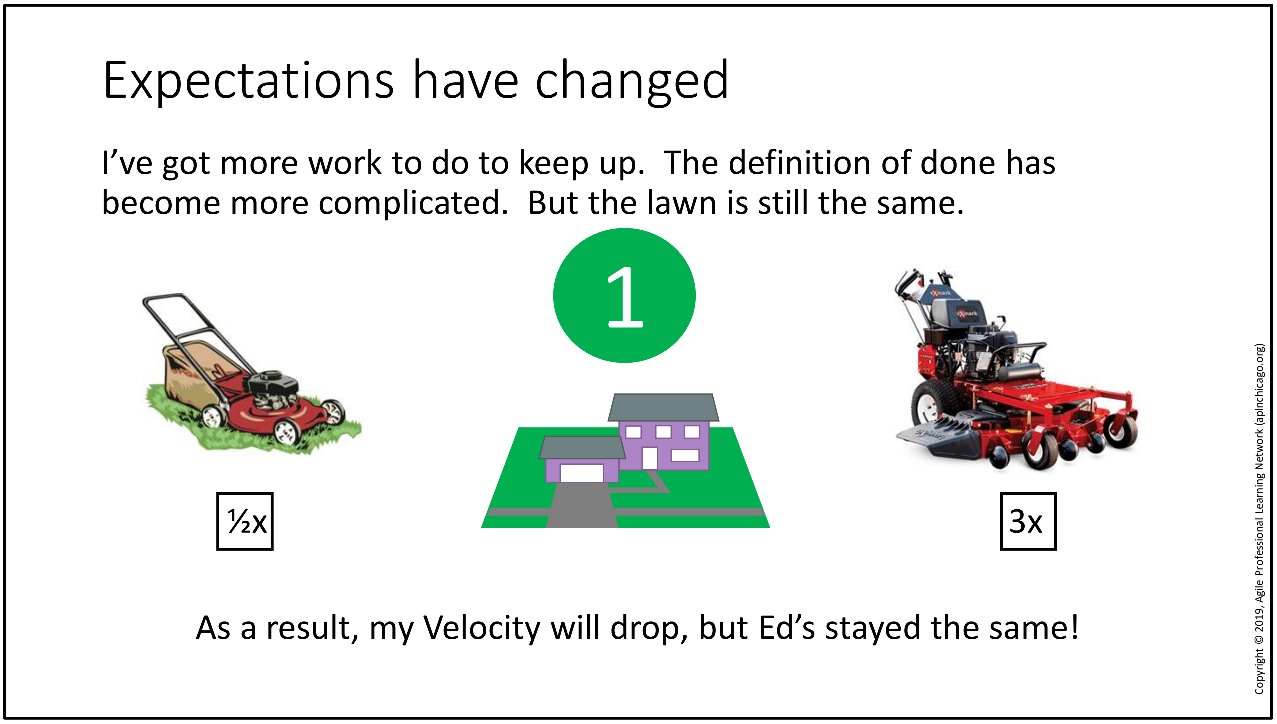
Equipment
Self-Propelled
Electric Trimmer
Push Broom

Definition of Done
Pull out equipment
Clear lawn debris
Fill mower with gas
Mow the lawn
Trim around obstacles
Edge the lawn
Clear grass clippings
Put equipment away
Collect payment

Time To Meet Definition of Done: 90 minutes

So now, I have a problem. The rules have changed. The definition of a QUALITY JOB has been enhanced. In order to stay competitive, I will need to meet the standards set by Ed's Definition of Done.

Unfortunately, my team is still just one person, and I don't have nearly as advanced tools as Ed's team has. It now takes me longer to mow my lawn (because mowing is just the first step, followed by trimming, followed by cleanup).

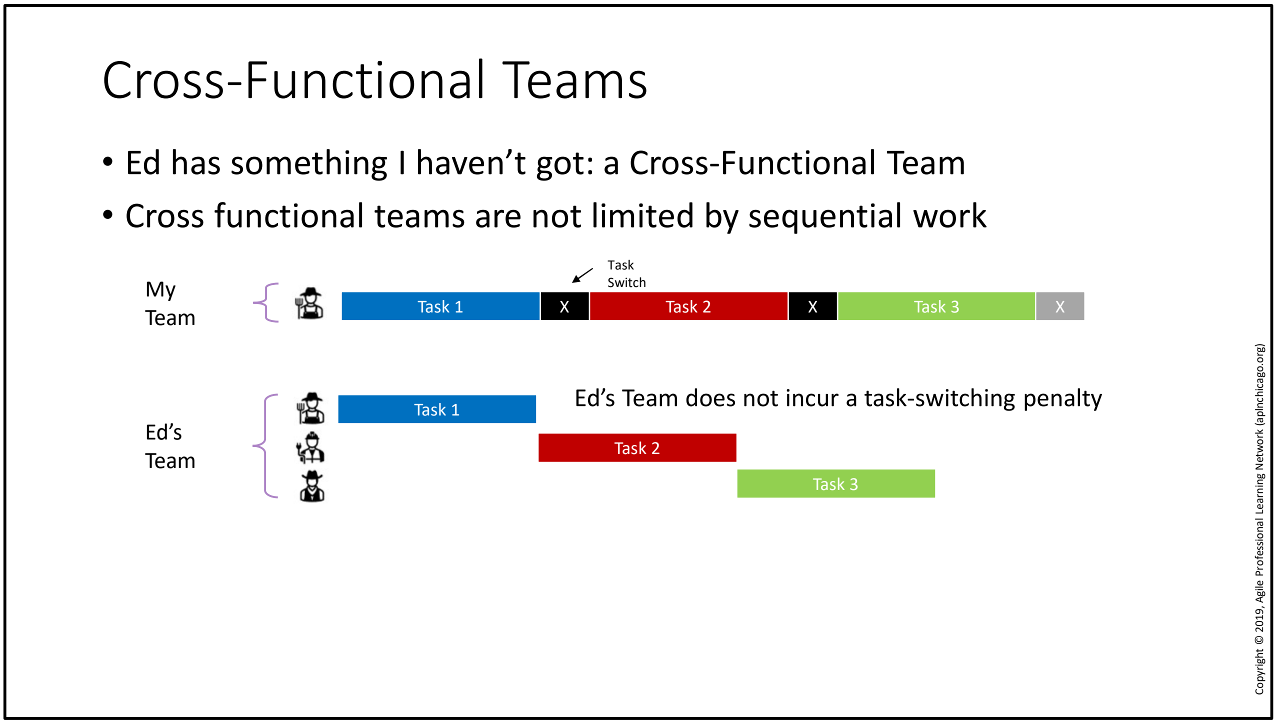


Expectations have changed

I've got more work to do to keep up. The definition of done has become more complicated. But the lawn is still the same.

As a result, my Velocity will drop, but Ed's stayed the same!

The lawn hasn't changed size. The expectations of what constitutes a quality job has changed. I therefore will get fewer lawns completed in each sprint. My VELOCITY will change. It will go down.



Cross-Functional Teams

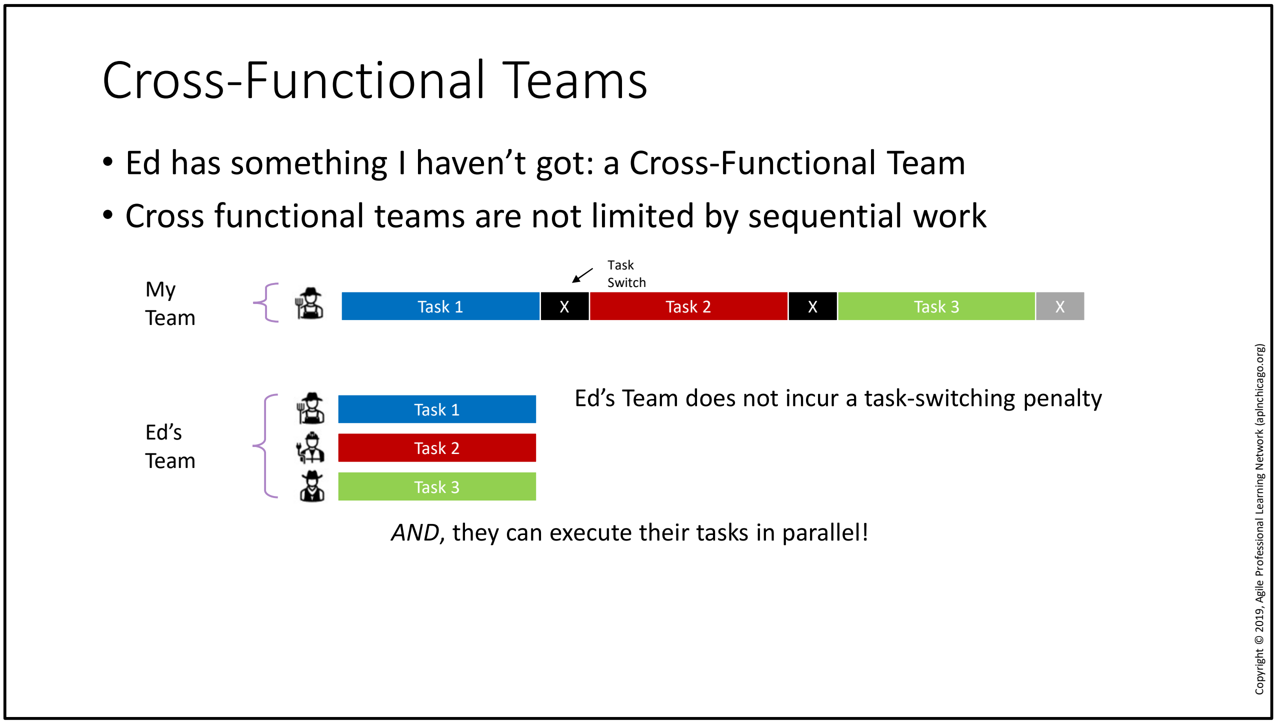
- Ed has something I haven't got: a Cross-Functional Team
- Cross functional teams are not limited by sequential work

Ed's Team does not incur a task-switching penalty

The more complex the Definition of Done gets, the more benefit you see from a cross-functional team.

Look at the impact all this work (plus the task-switching penalty) has on my performance. I'm the only one on the team, so I have to do it all.

Whereas Ed, Manny and Jeff divide and conquer, and don't even have to switch tasks so they don't pay a task-switching penalty!



Cross-Functional Teams

- Ed has something I haven't got: a Cross-Functional Team
- Cross functional teams are not limited by sequential work

My Team

Task Switch

Task 1 | X | Task 2 | X | Task 3 | X

Ed's Team

Task 1

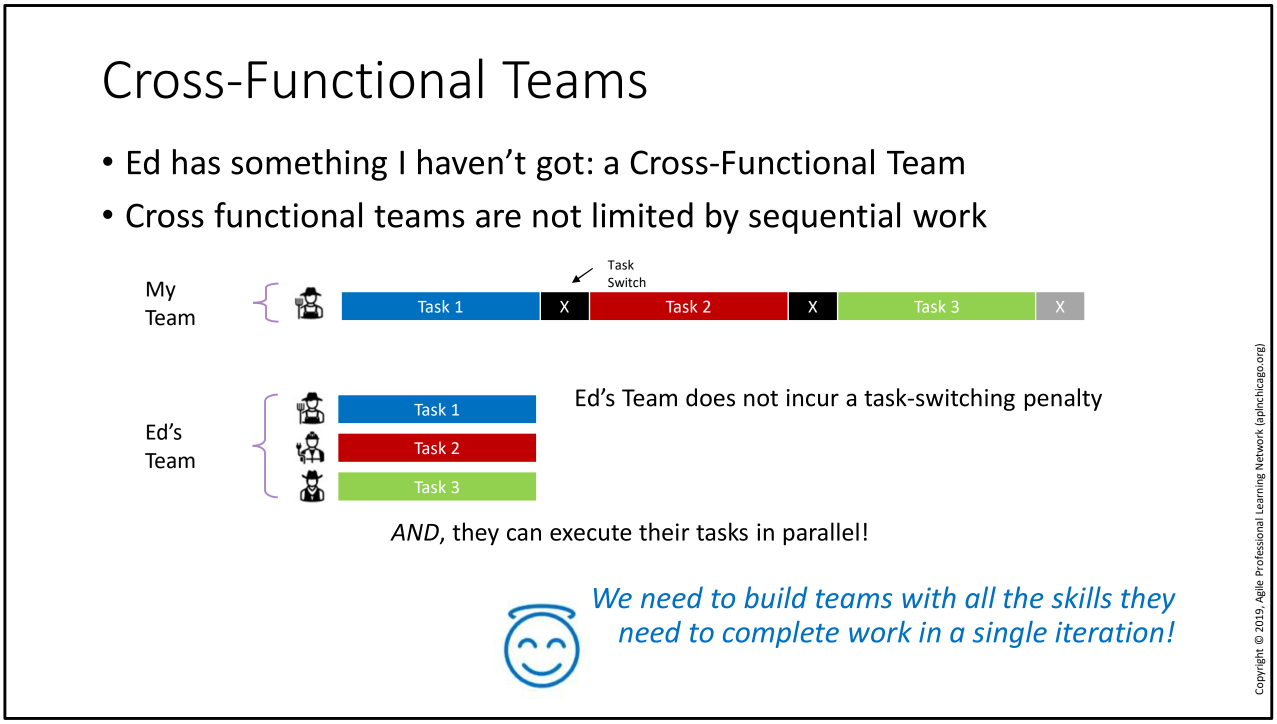
Task 2

Task 3

Ed's Team does not incur a task-switching penalty

AND, they can execute their tasks in parallel!

…and even better: Because he has three people, they have the opportunity to execute their individual tasks in parallel. Something I cannot do myself – *at least not without hurting myself.*



Cross-Functional Teams

- Ed has something I haven't got: a Cross-Functional Team
- Cross functional teams are not limited by sequential work

My Team

Task Switch

Task 1 | X | Task 2 | X | Task 3 | X

Ed's Team

Task 1

Task 2

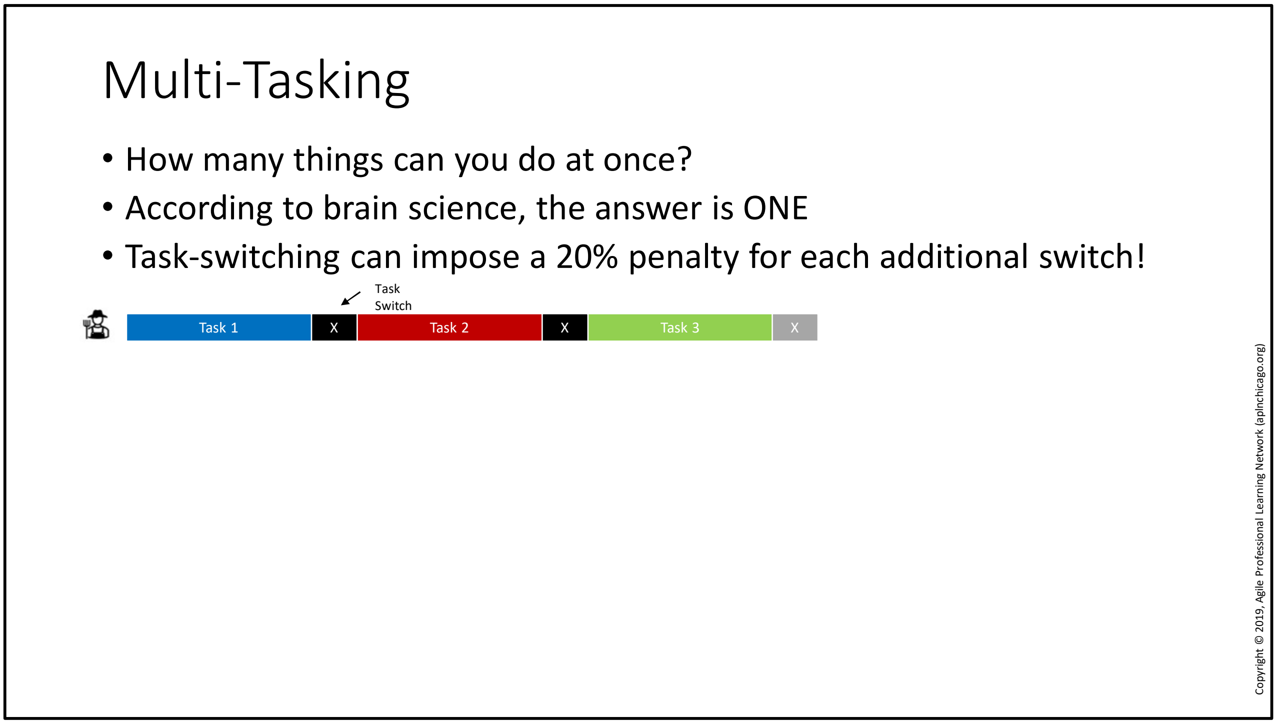
Task 3

Ed's Team does not incur a task-switching penalty

AND, they can execute their tasks in parallel!

We need to build teams with all the skills they need to complete work in a single iteration!

This is a great argument for working more efficiently by building cross-functional teams!



Multi-Tasking

- How many things can you do at once?
- According to brain science, the answer is ONE
- Task-switching can impose a 20% penalty for each additional switch!

Task Switch

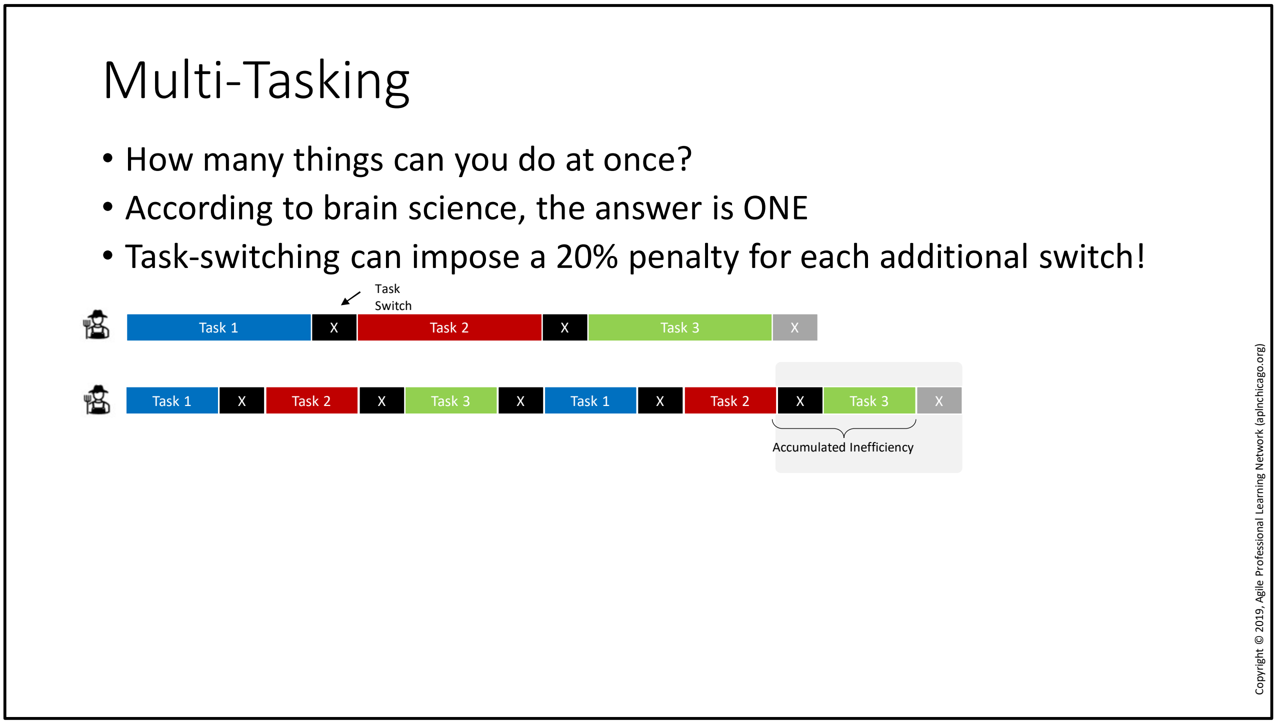
Task 1	X	Task 2	X	Task 3	X

Multi-tasking is a myth.

Your brain can only focus on ONE complex task at a time. In order to present the illusion of multi-tasking, your brain has to rapidly switch between tasks.

Task switching is a terrible thing.

It's not free. There is a cost in productivity that you pay every time your brain saves and unloads one complex task, then loads the next. Conventional wisdom says task-switching can rob us of 20% of our productive time. It's simply lost. The more you switch back and forth, the worse it gets.



Multi-Tasking

- How many things can you do at once?
- According to brain science, the answer is ONE
- Task-switching can impose a 20% penalty for each additional switch!

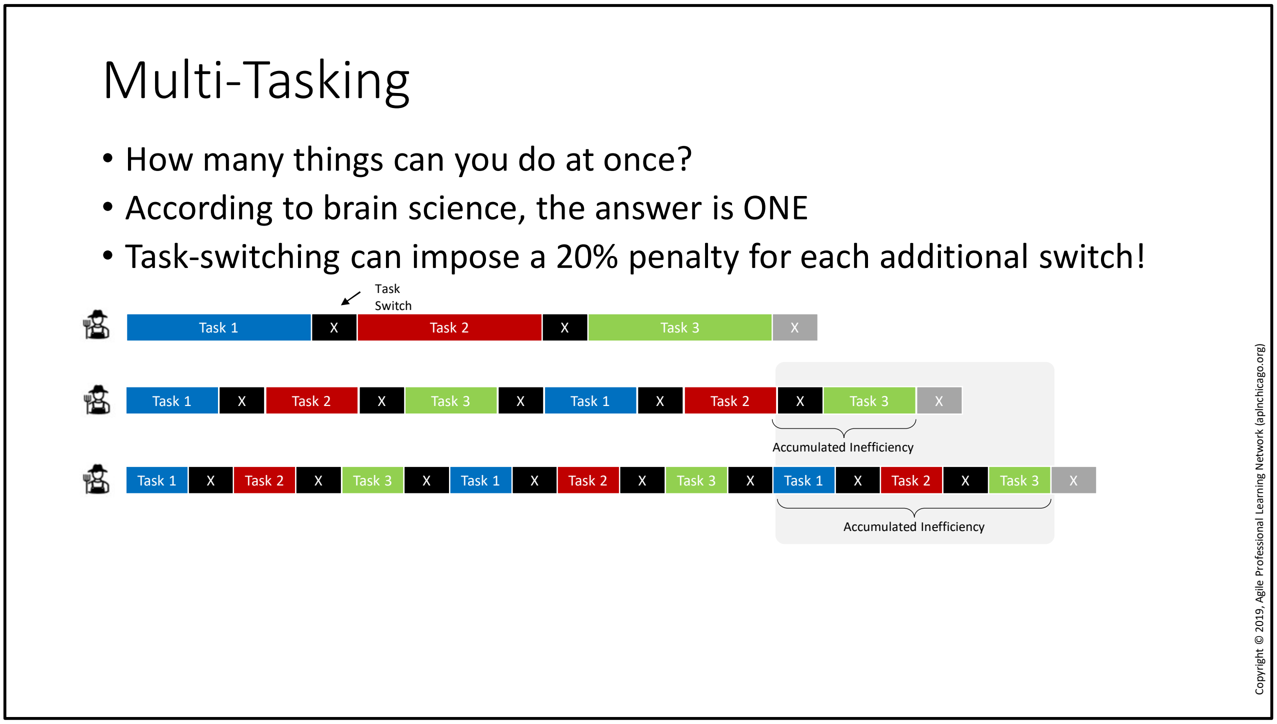
Task Switch

Task 1 | X | Task 2 | X | Task 3 | X

Task 1 | X | Task 2 | X | Task 3 | X | Task 1 | X | Task 2 | X | Task 3 | X

Accumulated Inefficiency

For instance, if I were to multi-task by dividing my lawn-mowing activities between the front yard and the back yard. That is, finish the entire front yard mowing, trimming and cleanup before moving to the backyard. With the task switching, I'll incur a penalty – the second way will take longer than the first way.

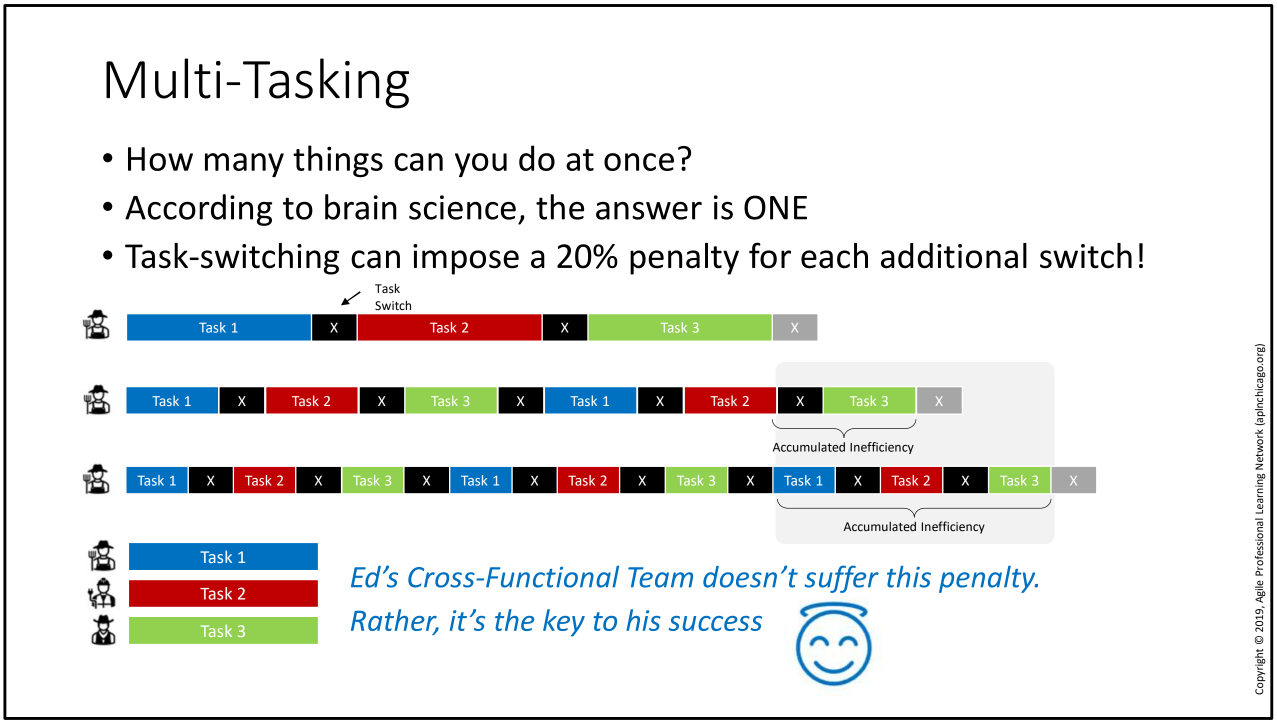


Multi-Tasking

- How many things can you do at once?
- According to brain science, the answer is ONE
- Task-switching can impose a 20% penalty for each additional switch!

Task Switch
Task 1 | X | Task 2 | X | Task 3 | X
Task 1 | X | Task 2 | X | Task 3 | X | Task 1 | X | Task 2 | X | Task 3 | X
Accumulated Inefficiency
Task 1 | X | Task 2 | X | Task 3 | X | Task 1 | X | Task 2 | X | Task 3 | X | Task 1 | X | Task 2 | X | Task 3 | X
Accumulated Inefficiency

It gets even worse if you try to break it up more. I might feel better while I'm working the third way, because I can see all the tasks progressing a little bit throughout the job, but I'm paying that penalty over and over and over again. That is wasteful. It's not efficient!



Multi-Tasking

- How many things can you do at once?
- According to brain science, the answer is ONE
- Task-switching can impose a 20% penalty for each additional switch!

Task Switch

Task 1 | X | Task 2 | X | Task 3 | X

Task 1 | X | Task 2 | X | Task 3 | X | Task 1 | X | Task 2 | X | Task 3 | X

Accumulated Inefficiency

Task 1 | X | Task 2 | X | Task 3 | X | Task 1 | X | Task 2 | X | Task 3 | X | Task 1 | X | Task 2 | X | Task 3 | X

Accumulated Inefficiency

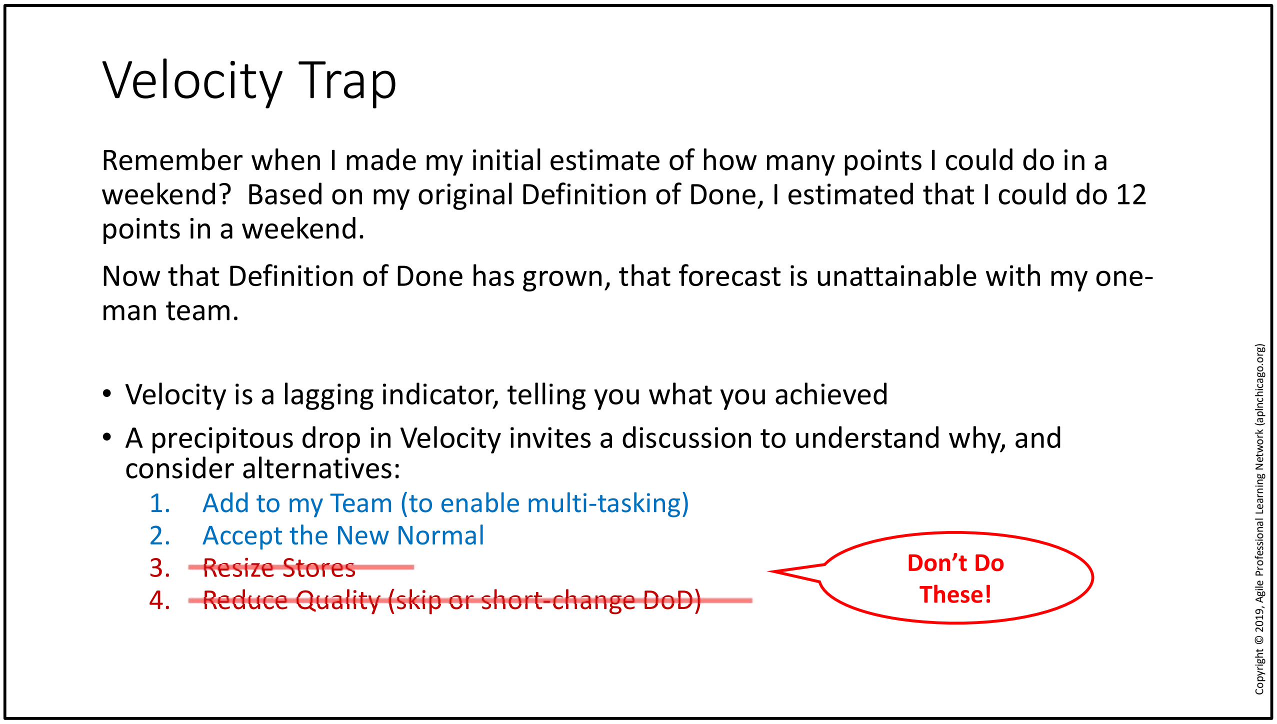
Task 1

Task 2

Task 3

Ed's Cross-Functional Team doesn't suffer this penalty.
Rather, it's the key to his success

Meanwhile, Ed's cross-functional team doesn't suffer any penalty at all. Now we can clearly see why Ed can do the same 1-point job in so much less time. And we can understand why we don't have to change the point estimate based on which team is executing the work. The 1-point job isn't finished until all the tasks are complete.



Velocity Trap

Remember when I made my initial estimate of how many points I could do in a weekend? Based on my original Definition of Done, I estimated that I could do 12 points in a weekend.

Now that Definition of Done has grown, that forecast is unattainable with my one-man team.

- Velocity is a lagging indicator, telling you what you achieved
- A precipitous drop in Velocity invites a discussion to understand why, and consider alternatives:
  1. Add to my Team (to enable multi-tasking)
  2. Accept the New Normal
  3. ~~Resize Stores~~
  4. ~~Reduce Quality (skip or short-change DoD)~~

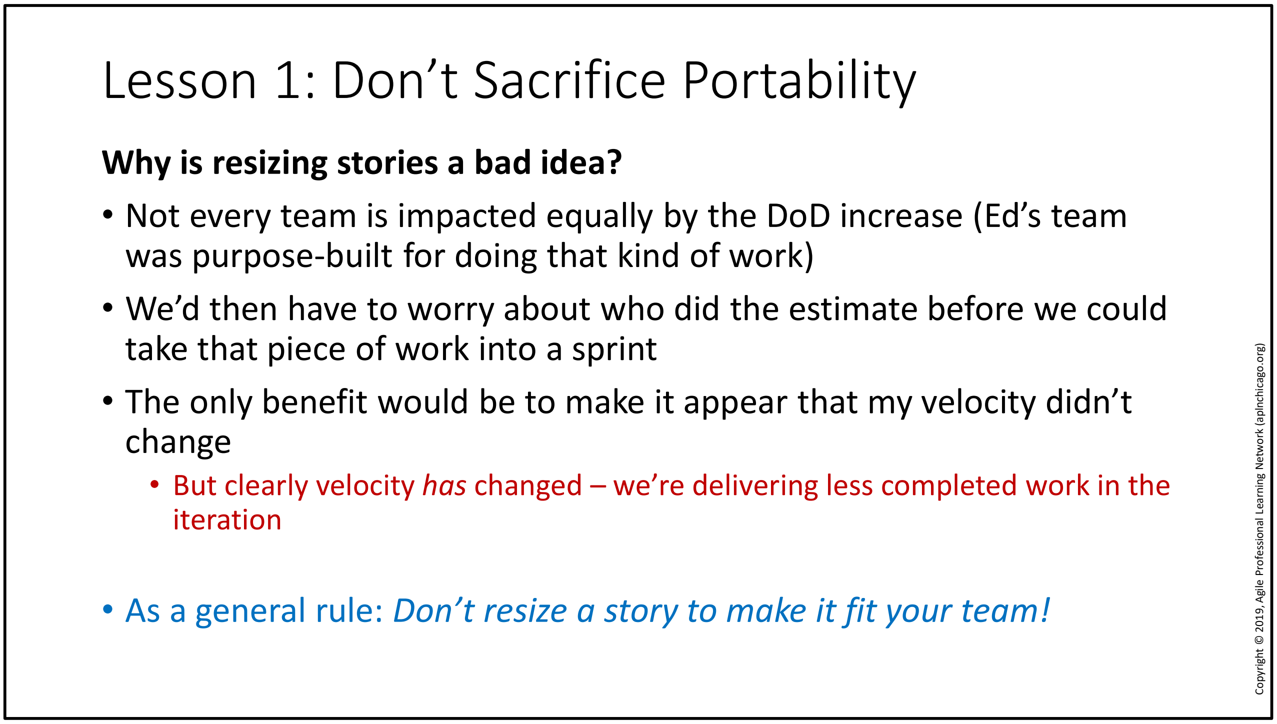
Don’t Do These!

A velocity trap occurs when you decide that Velocity is a metric that must be met, rather than an indicator of how things are going.

The urge to adjust estimates or skip steps in the Definition of Done in order to maintain velocity can be strong – especially in teams/organizations new to agile.

If something happens that causes a drop in Velocity (or an increase!), we want to see that so we can talk about it, and decide what (if anything) we can do about it.

We do NOT want to disguise it.



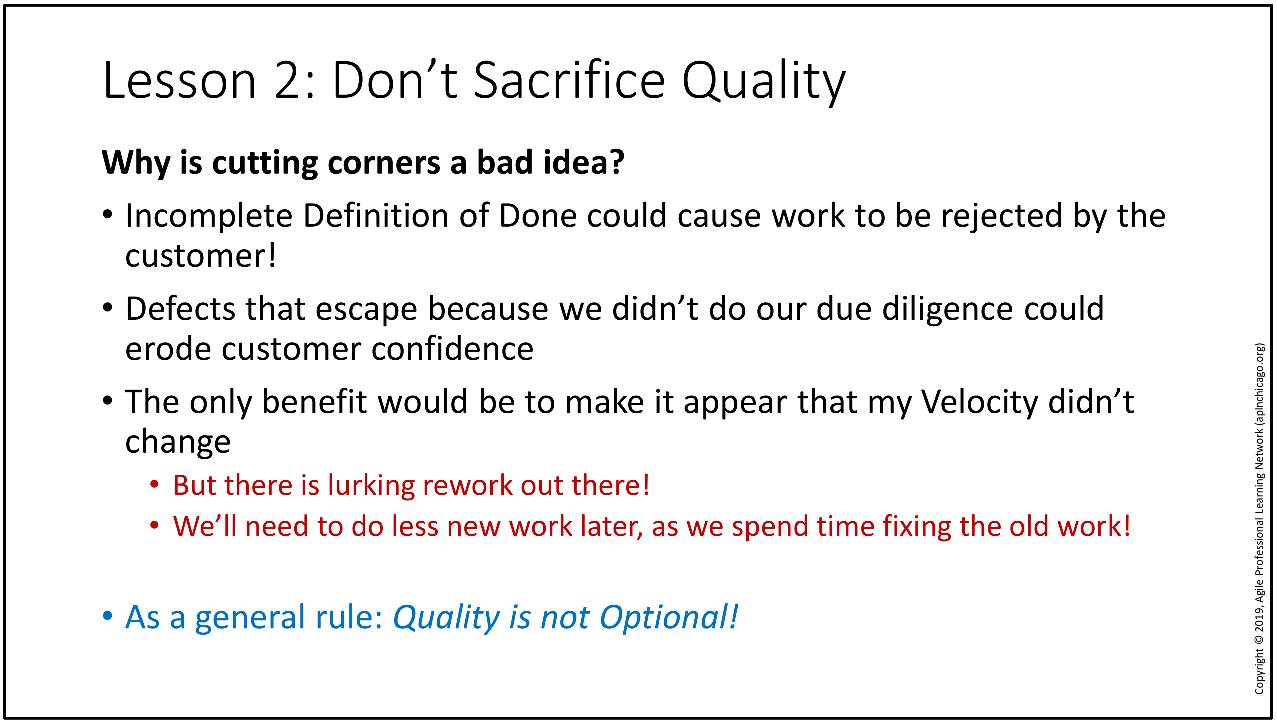
Lesson 1: Don’t Sacrifice Portability

Why is resizing stories a bad idea?

- Not every team is impacted equally by the DoD increase (Ed’s team was purpose-built for doing that kind of work)
- We’d then have to worry about who did the estimate before we could take that piece of work into a sprint
- The only benefit would be to make it appear that my velocity didn’t change
  - But clearly velocity *has* changed – we’re delivering less completed work in the iteration

- As a general rule: *Don’t resize a story to make it fit your team!*

When I made the estimates of the relative size of the lawns on my block, I based that on the relative effort of each job compared to the effort of the others. Yes, there is more to do, but there is more to do across ALL THE LAWNS. The scope of the work increased, but the ratio of the amount of work is still relatively constant.



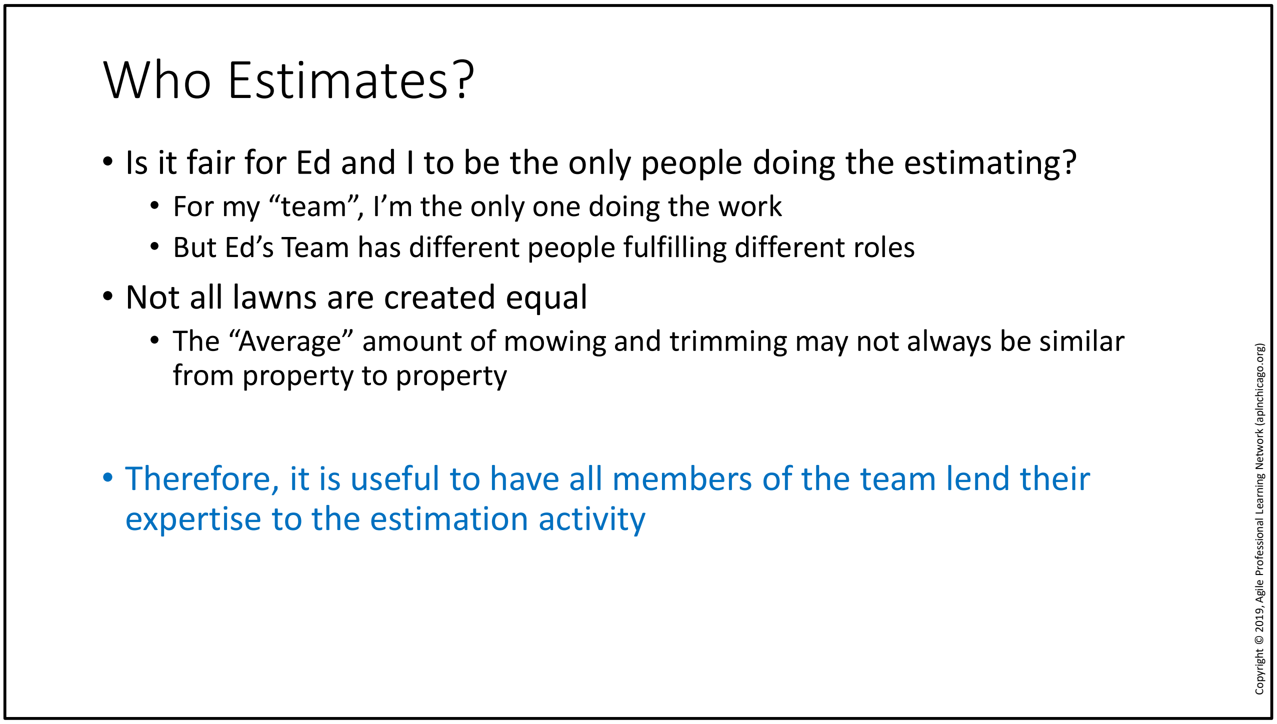
Lesson 2: Don’t Sacrifice Quality

Why is cutting corners a bad idea?

- Incomplete Definition of Done could cause work to be rejected by the customer!
- Defects that escape because we didn’t do our due diligence could erode customer confidence
- The only benefit would be to make it appear that my Velocity didn’t change
  - But there is lurking rework out there!
  - We’ll need to do less new work later, as we spend time fixing the old work!

- As a general rule: *Quality is not Optional!*

Skimping on the new Definition of Done steps is a terrible idea. Quality is not an option. To satisfy our customer it is a necessity.



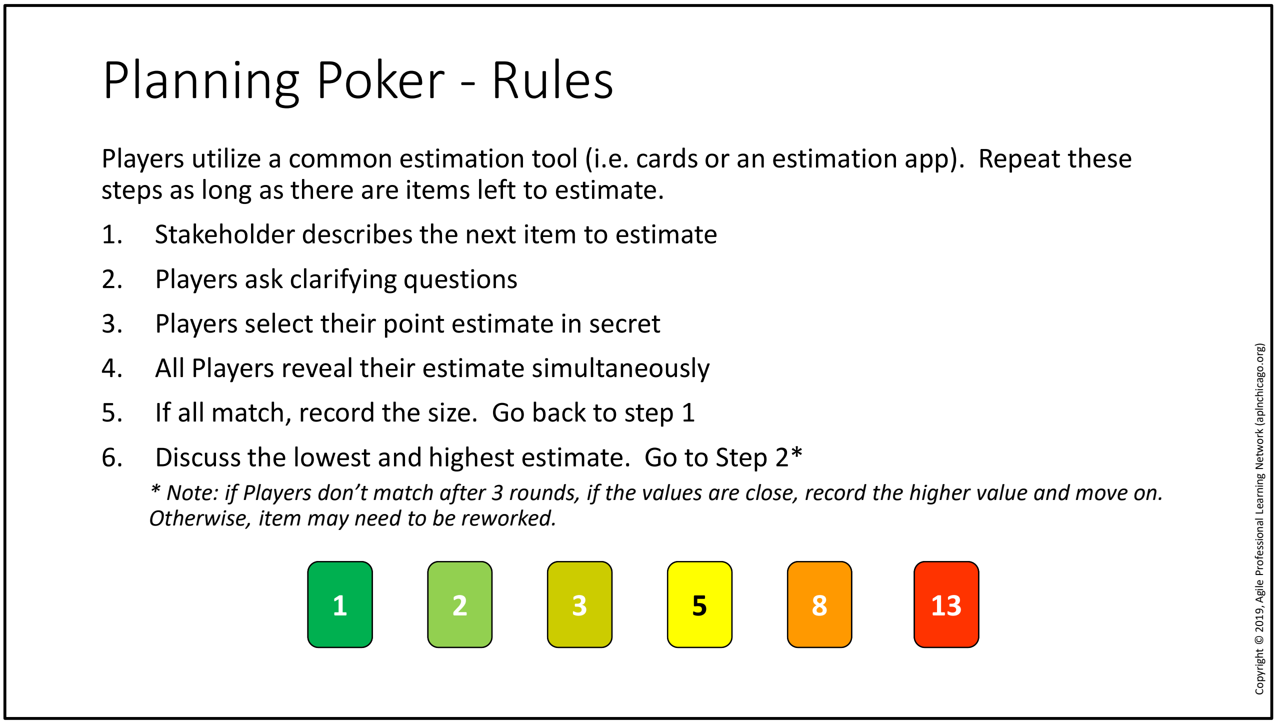
Who Estimates?

- Is it fair for Ed and I to be the only people doing the estimating?
  - For my "team", I'm the only one doing the work
  - But Ed's Team has different people fulfilling different roles
- Not all lawns are created equal
  - The "Average" amount of mowing and trimming may not always be similar from property to property
- Therefore, it is useful to have all members of the team lend their expertise to the estimation activity

Now that we've seen the benefit of creating a cross-functional team (Ed's team performs better than mine, after all), who should have a hand in the estimation of work for that cross-functional team?

The answer? All of them! They all have a hand, and a unique perspective on the work that needs to be done.

How do we take different points of view into account? There's a fun tool called "Planning Poker"…



Planning Poker - Rules

Players utilize a common estimation tool (i.e. cards or an estimation app). Repeat these steps as long as there are items left to estimate.

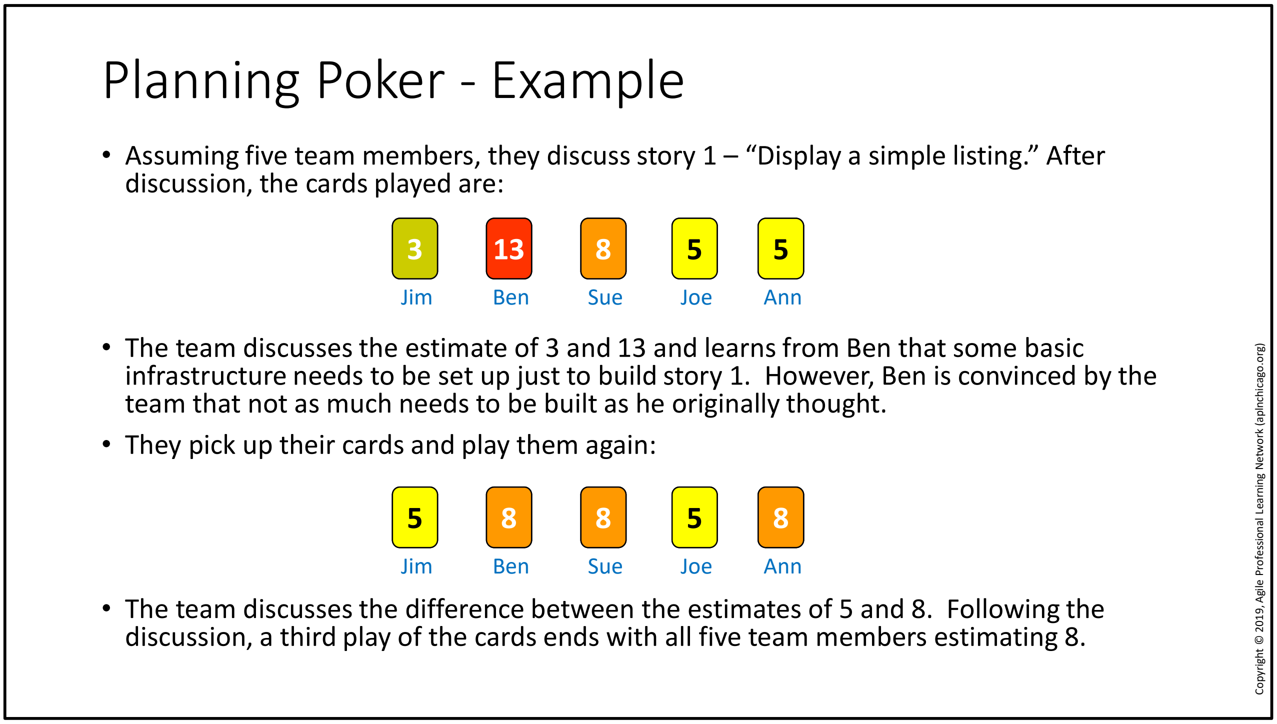
1. Stakeholder describes the next item to estimate
2. Players ask clarifying questions
3. Players select their point estimate in secret
4. All Players reveal their estimate simultaneously
5. If all match, record the size. Go back to step 1
6. Discuss the lowest and highest estimate. Go to Step 2*

** Note: if Players don't match after 3 rounds, if the values are close, record the higher value and move on. Otherwise, item may need to be reworked.*

1 2 3 5 8 13

During Backlog Refinement, for as long as there are items in the backlog yet to be sized, and we still have time left in the ceremony timebox…

The product owner presents a work item. The team (Players) ask questions until they understand the ask well enough to offer an estimate. The players do NOT discuss the numbers prior to revealing their estimates all at the same time. If the numbers are different, the highest and lowers estimates are discussed. Then they vote again (adjusting estimates to reflect the outcome of the discussion). We are looking for consensus. If they can't agree after a few rounds of voting, decide on a way to proceed.



Planning Poker - Example

- Assuming five team members, they discuss story 1 – "Display a simple listing." After discussion, the cards played are:

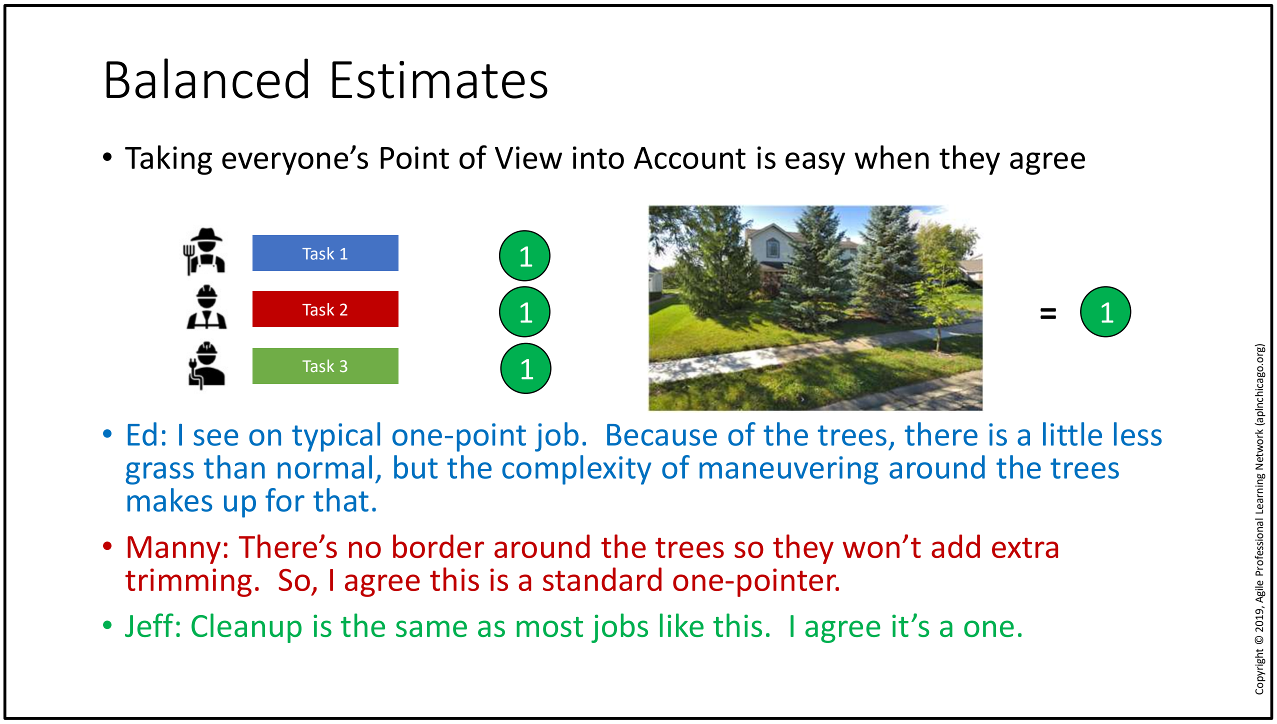
3	13	8	5	5
Jim	Ben	Sue	Joe	Ann

- The team discusses the estimate of 3 and 13 and learns from Ben that some basic infrastructure needs to be set up just to build story 1. However, Ben is convinced by the team that not as much needs to be built as he originally thought.
- They pick up their cards and play them again:

5	8	8	5	8
Jim	Ben	Sue	Joe	Ann

- The team discusses the difference between the estimates of 5 and 8. Following the discussion, a third play of the cards ends with all five team members estimating 8.

In this example we can see the impact of diverse interpretations of a user story. This is the whole point of having a story and engaging the team in the estimation. How will we know we've unified the understanding of the work that needs to be done without a conversation?



Balanced Estimates

- Taking everyone's Point of View into Account is easy when they agree

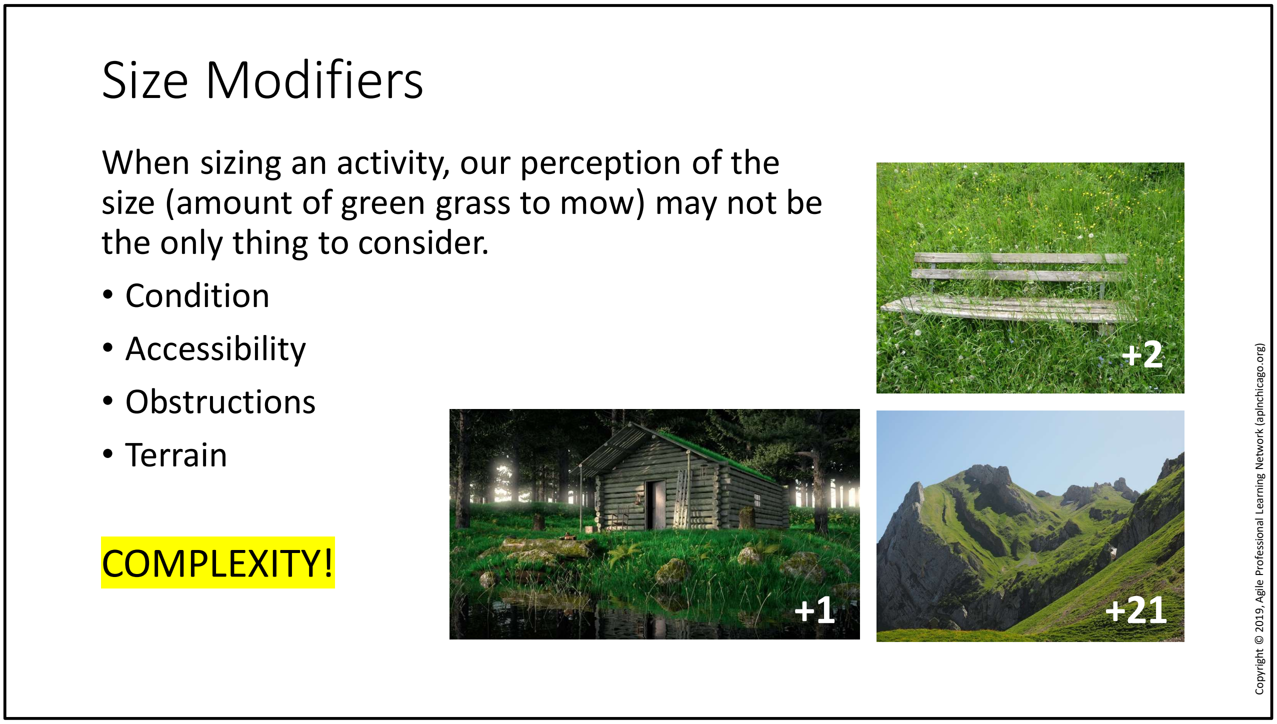
Task 1 1
Task 2 1
Task 3 1
= 1

- Ed: I see on typical one-point job. Because of the trees, there is a little less grass than normal, but the complexity of maneuvering around the trees makes up for that.
- Manny: There's no border around the trees so they won't add extra trimming. So, I agree this is a standard one-pointer.
- Jeff: Cleanup is the same as most jobs like this. I agree it's a one.

In this example, our three team members look at the job, and they all bring their unique point of view into the estimation.

Note, they are not saying how long each of the steps they need to do will take. They are talking about a Normal amount of effort that goes into work of this type.

That is, *"Compared to other one-point jobs we've done, this looks like just about the same amount of effort."*



Size Modifiers

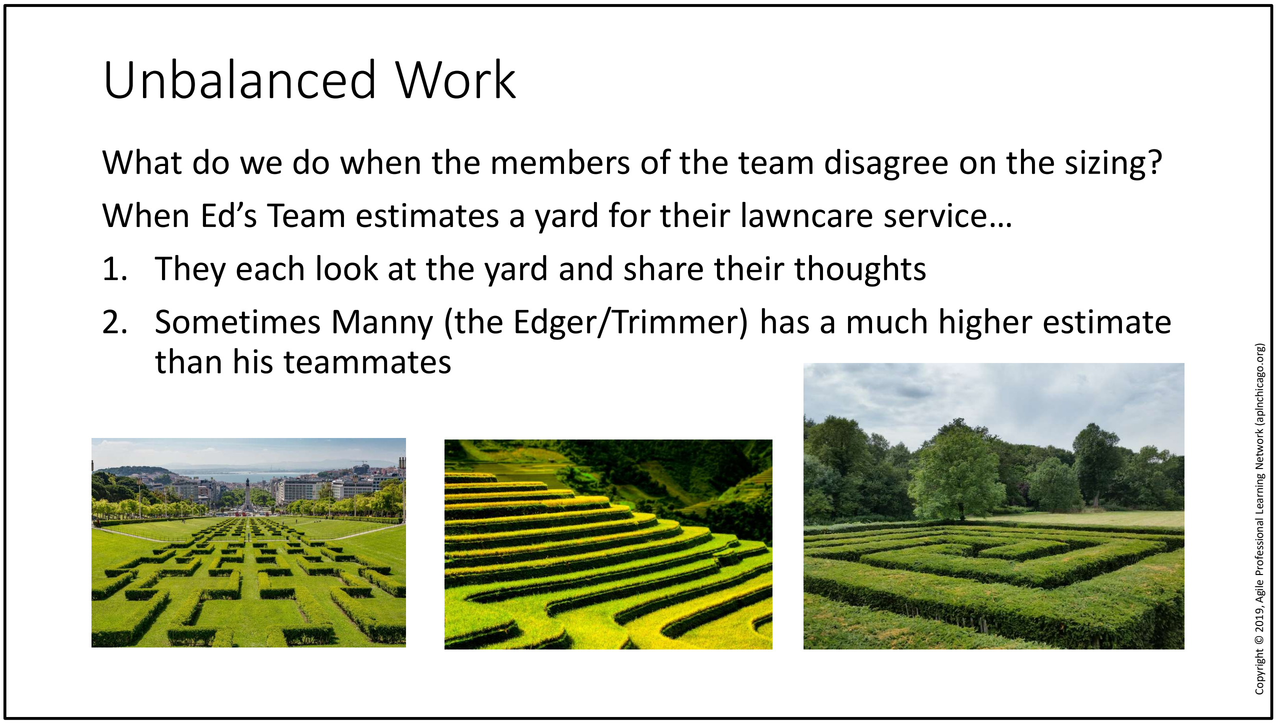
When sizing an activity, our perception of the size (amount of green grass to mow) may not be the only thing to consider.

- Condition
- Accessibility
- Obstructions
- Terrain

COMPLEXITY!

Some of the factors that can influence a lawn estimate are probably not visible from an arial shot. Unusually long grass that will clog your mower, a lot of rocks that could damage the machine, or impossibly steep terrain that could send you and your mower tumbling!

Sometimes you just have to eyeball the situation yourself to get a really good idea, and you may, based on that observation, apply a modifier to your relatively quick thumbnail guess.



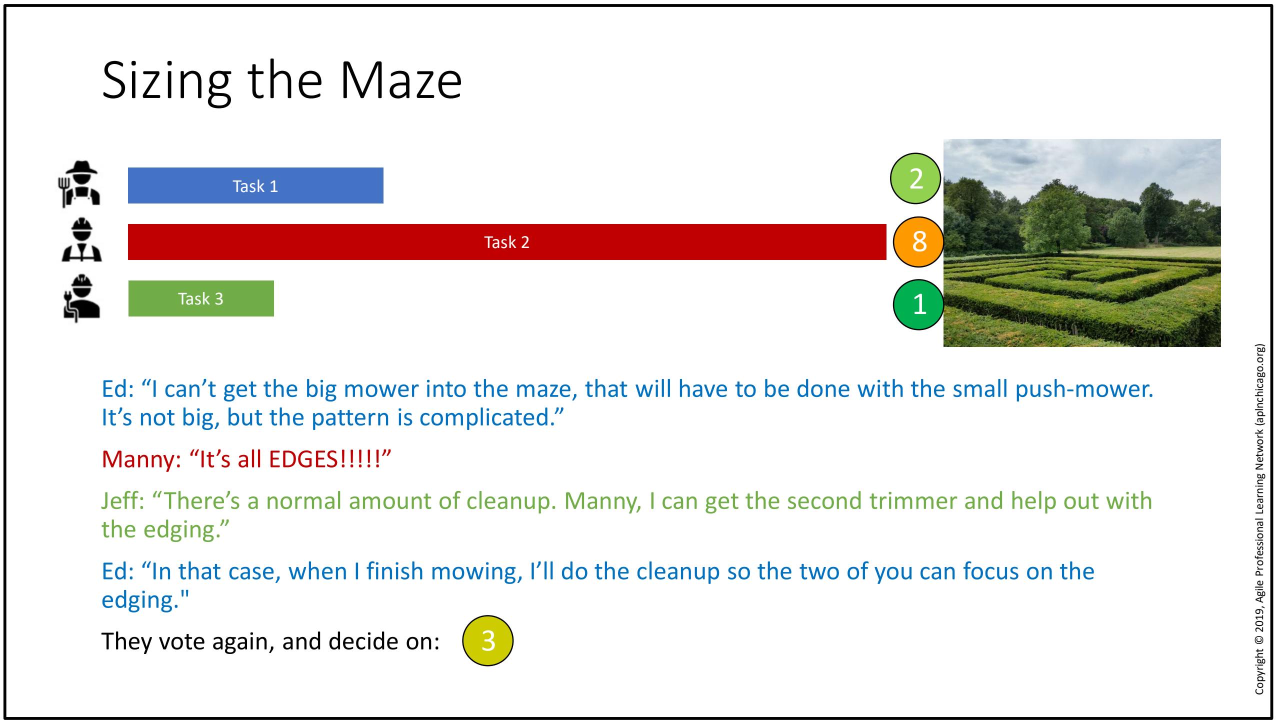
Unbalanced Work

What do we do when the members of the team disagree on the sizing?

When Ed’s Team estimates a yard for their lawncare service…

1. They each look at the yard and share their thoughts
2. Sometimes Manny (the Edger/Trimmer) has a much higher estimate than his teammates

Here we present some examples that offer wildly different levels of complexity for the different roles involved. Understanding WHY the estimates are different is the first step.



Sizing the Maze

Task 1
2
Task 2
8
Task 3
1

Ed: “I can’t get the big mower into the maze, that will have to be done with the small push-mower. It’s not big, but the pattern is complicated.”

Manny: “It’s all EDGES!!!!!”

Jeff: “There’s a normal amount of cleanup. Manny, I can get the second trimmer and help out with the edging.”

Ed: “In that case, when I finish mowing, I’ll do the cleanup so the two of you can focus on the edging."

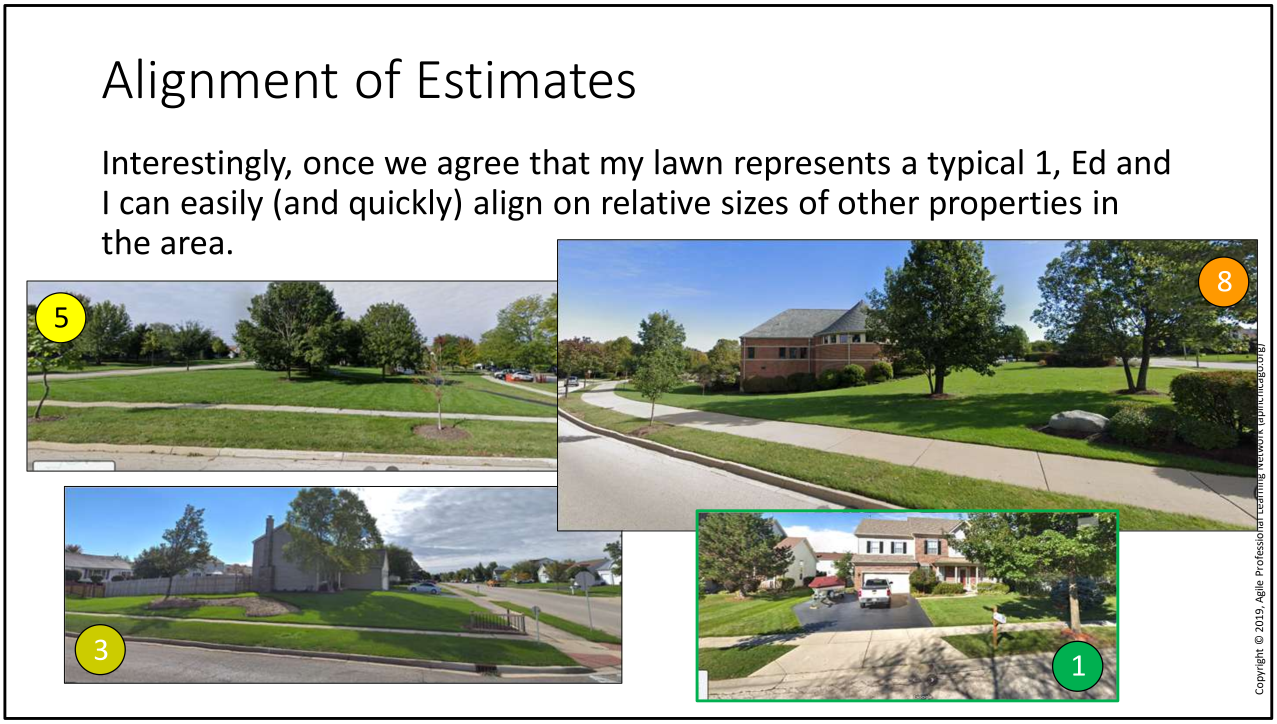
They vote again, and decide on: 3

Manny has a lot more work to do than the others, and that work is wildly out of proportion.

Some groups of individuals will shrug their shoulders at that, and say, “Too bad, dude. I guess we’ll have to call it an 8.

NOTE: It would be an 8, not an 11. You don’t add them all together, you take the one that’s the biggest.

But a mature team will look for ways to bring that work into alignment across the board. What if Manny wasn’t the only one who was doing trimming? The team should discuss the work and determine how they could work together in delivery of that work item, and see if it affects the overall estimate.



Alignment of Estimates

Interestingly, once we agree that my lawn represents a typical 1, Ed and I can easily (and quickly) align on relative sizes of other properties in the area.

Of course, there are more properties around my house than just the ones on my block. Way more than any one lawn mowing service (even Ed's) could do in a single weekend. We'll need to have more than one team… and if we're going to have more than one team working together.

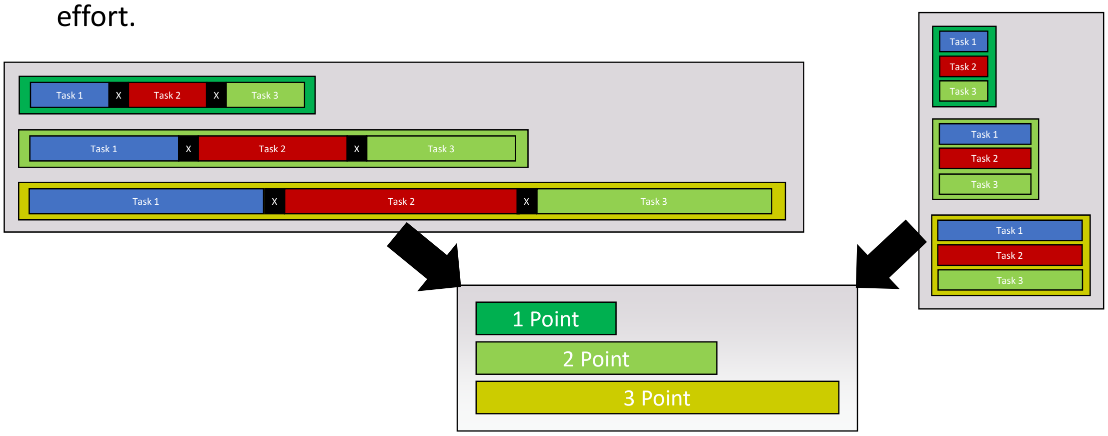
When this happens, we'll need to find a way to unify our estimates – that is, to bring them into relative parity with each other. You do this by having the two groups all estimate the same story on their own unique scale, then decide which estimate will become the benchmark. The other estimates are scaled accordingly.

Note, this can be disruptive after the fact, so I find it's better to align the teams through training so they're already in the same ballpark of estimation.

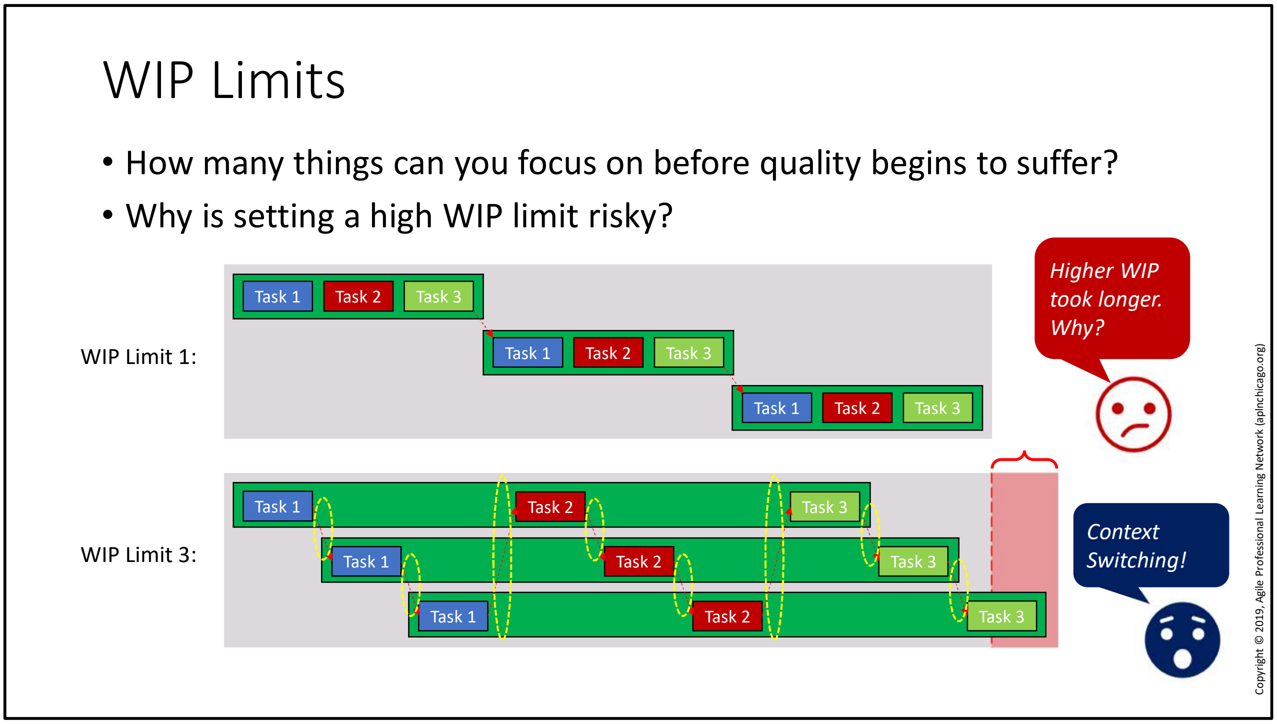
A one is a one is a one.

Abstracting to a Common Backlog

- Once Ed's Team and mine align on our relative sizes, the backlogs become interchangeable.
- We can stop worrying about individual contributions and focus on collective effort.



Now all the work can be combined into a single backlog, and we can distribute work to the different teams. Note: Each team will have a unique Velocity. We've already seen why. That's not a bad thing. It's an accurate thing.



WIP Limits

- How many things can you focus on before quality begins to suffer?
- Why is setting a high WIP limit risky?

WIP Limit 1:

Task 1 Task 2 Task 3

Task 1 Task 2 Task 3

Task 1 Task 2 Task 3

WIP Limit 3:

Task 1 Task 2 Task 3

Task 1 Task 2 Task 3

Task 1 Task 2 Task 3

Higher WIP took longer. Why?

Context Switching!

Getting stuff done.

Remember, the whole point of doing this work, is to get paid. We can only get paid by our customers if we've satisfied the Definition of Done on each property.

What would happen if we didn't do one lawn at a time, but instead had each member of the team run out and just do their own thing across all items on the backlog?

Let's say it started raining halfway through our sprint, and we had to stop work on all the lawns in the backlog. We would only get paid for the lawns we actually finished!

It's to our advantage therefore to get as many work items to "Done" as soon as possible (to fend off the risk of a disaster impacting our delivery).

What we’ve learned so far

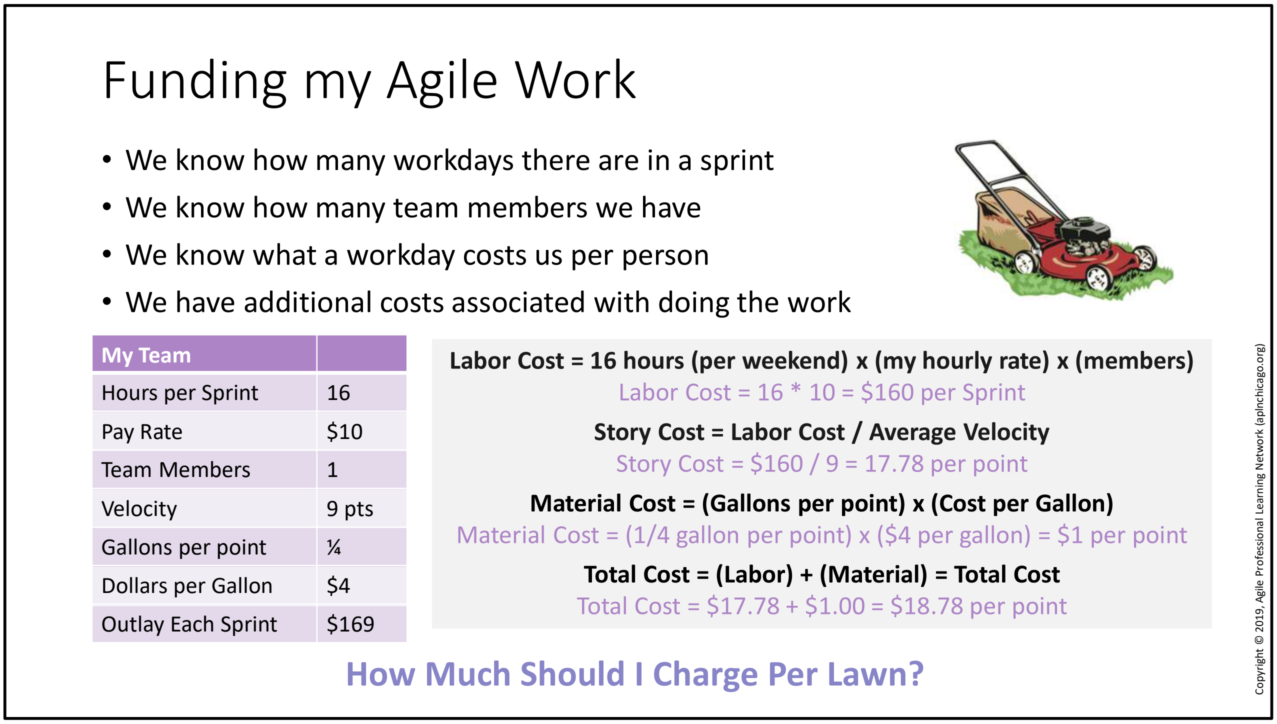
- Relative Sizing can eliminate ownership bias
- Teams should be assembled based on the work they do
- Controlling WIP and focusing on completed delivery is efficient
- Velocity ties our estimates to delivery of value over time

☑ We can now answer that first question

- “When can I get it?”

□ Now let’s consider the second question

- “How much will it cost?”



Funding my Agile Work

- We know how many workdays there are in a sprint
- We know how many team members we have
- We know what a workday costs us per person
- We have additional costs associated with doing the work

My Team	
Hours per Sprint	16
Pay Rate	$10
Team Members	1
Velocity	9 pts
Gallons per point	¼
Dollars per Gallon	$4
Outlay Each Sprint	$169

Labor Cost = 16 hours (per weekend) x (my hourly rate) x (members)

Labor Cost = 16 * 10 = $160 per Sprint

Story Cost = Labor Cost / Average Velocity

Story Cost = $160 / 9 = 17.78 per point

Material Cost = (Gallons per point) x (Cost per Gallon)

Material Cost = (1/4 gallon per point) x ($4 per gallon) = $1 per point

Total Cost = (Labor) + (Material) = Total Cost

Total Cost = $17.78 + $1.00 = $18.78 per point

How Much Should I Charge Per Lawn?

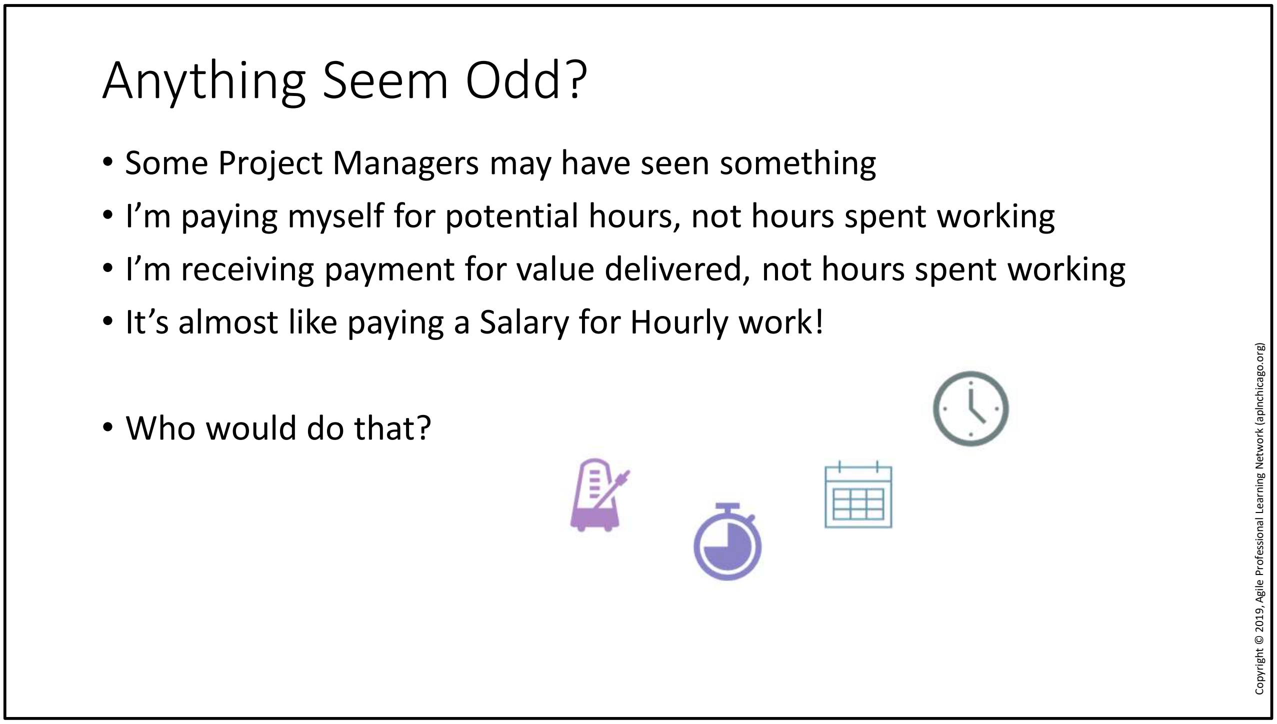
Traditional funding is based on time spent and pay rate. But Project Managers took that to a very precise level by counting (and charging) projects based on hours spent by individuals.

In agile models, we simplify that approach by funding team iterations.

You know how many hours are in a sprint. You know the pay rate for every member of the team. Therefore, you can calculate what that team costs you to exist.

If you divide the Cost of a Sprint by the number of Story Points delivered, you could calculate a Per-Story-Point abstract cost.

If I deliver 9 story points in a sprint, then those 3-SP lawns burned 1/3 of the cost of the sprint!



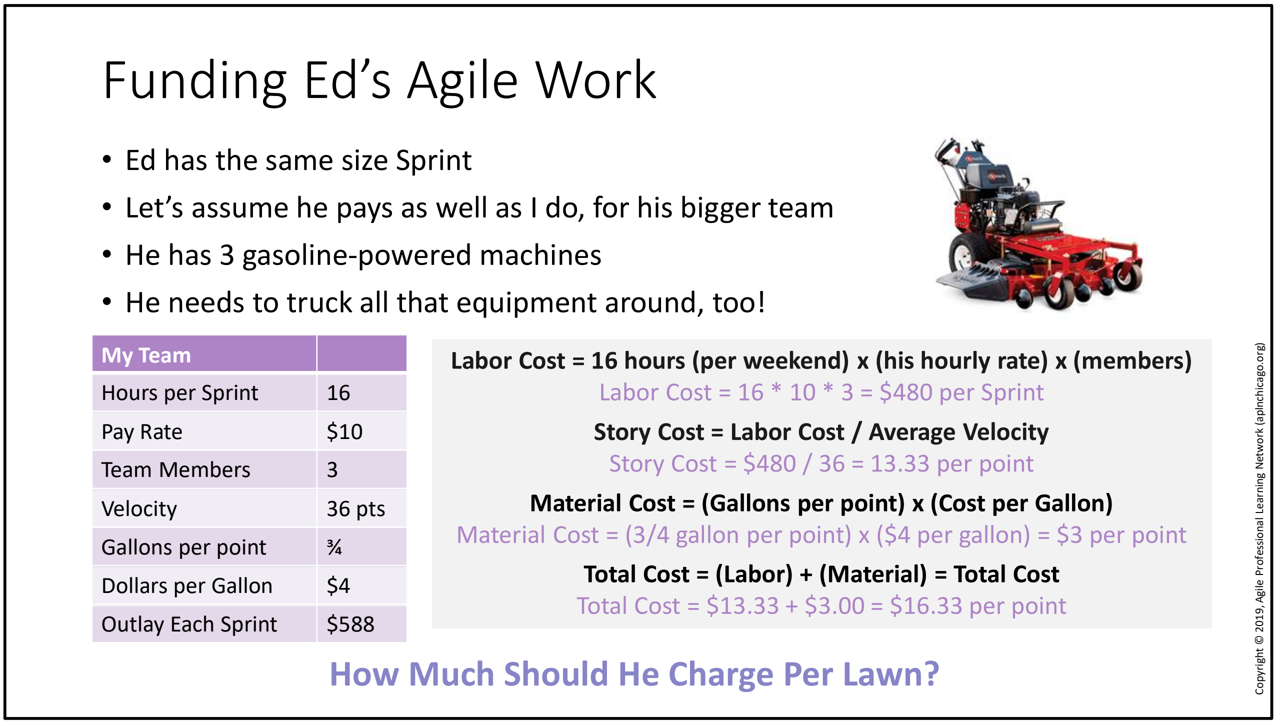
Anything Seem Odd?

- Some Project Managers may have seen something
- I’m paying myself for potential hours, not hours spent working
- I’m receiving payment for value delivered, not hours spent working
- It’s almost like paying a Salary for Hourly work!

- Who would do that?

This opened the door for something strange. We’d be charging based on the potential of the team, not on actual time spent. That’s a little like paying a Salary, isn’t it? That’s similar to funding fixed-bid projects. In effect, story pts are your bid for a given job.

Yes, my friend. <u>*You*</u> are that crazy, wonderful person who pays a fixed salary to your people, regardless of how much work they do! And let me tell you, it’s the reason everyone gets so wrapped around the axle on budget. Good grief, listen to them: *“Why am I paying for them when they’re not working? That’s not fair! I should only pay for the time they’re doing work! Well, I’ll show you! I’m going to pay someone to make sure everyone is busy every second of the day so I don’t waste any money on idle time ever again! Yeah, that’ll do it! And because I’m so good at getting project managers to get workers to work, someone should pay me too*! *Which means I need to be busy, every minute of the day… uh…* ***micro****-managing them into a frenzy of performance…”*



Funding Ed’s Agile Work

- Ed has the same size Sprint
- Let’s assume he pays as well as I do, for his bigger team
- He has 3 gasoline-powered machines
- He needs to truck all that equipment around, too!

My Team	
Hours per Sprint	16
Pay Rate	$10
Team Members	3
Velocity	36 pts
Gallons per point	¾
Dollars per Gallon	$4
Outlay Each Sprint	$588

Labor Cost = 16 hours (per weekend) x (his hourly rate) x (members)
Labor Cost = 16 * 10 * 3 = $480 per Sprint

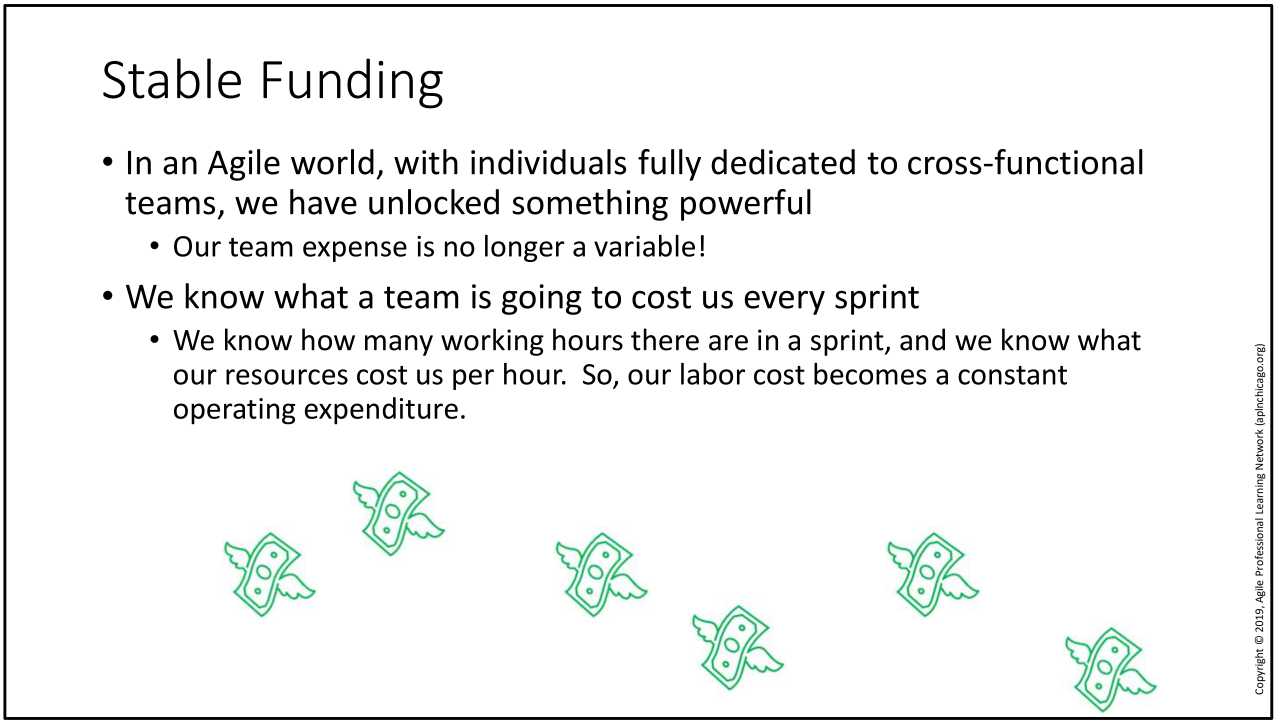
Story Cost = Labor Cost / Average Velocity
Story Cost = $480 / 36 = 13.33 per point

Material Cost = (Gallons per point) x (Cost per Gallon)
Material Cost = (3/4 gallon per point) x ($4 per gallon) = $3 per point

Total Cost = (Labor) + (Material) = Total Cost
Total Cost = $13.33 + $3.00 = $16.33 per point

How Much Should He Charge Per Lawn?

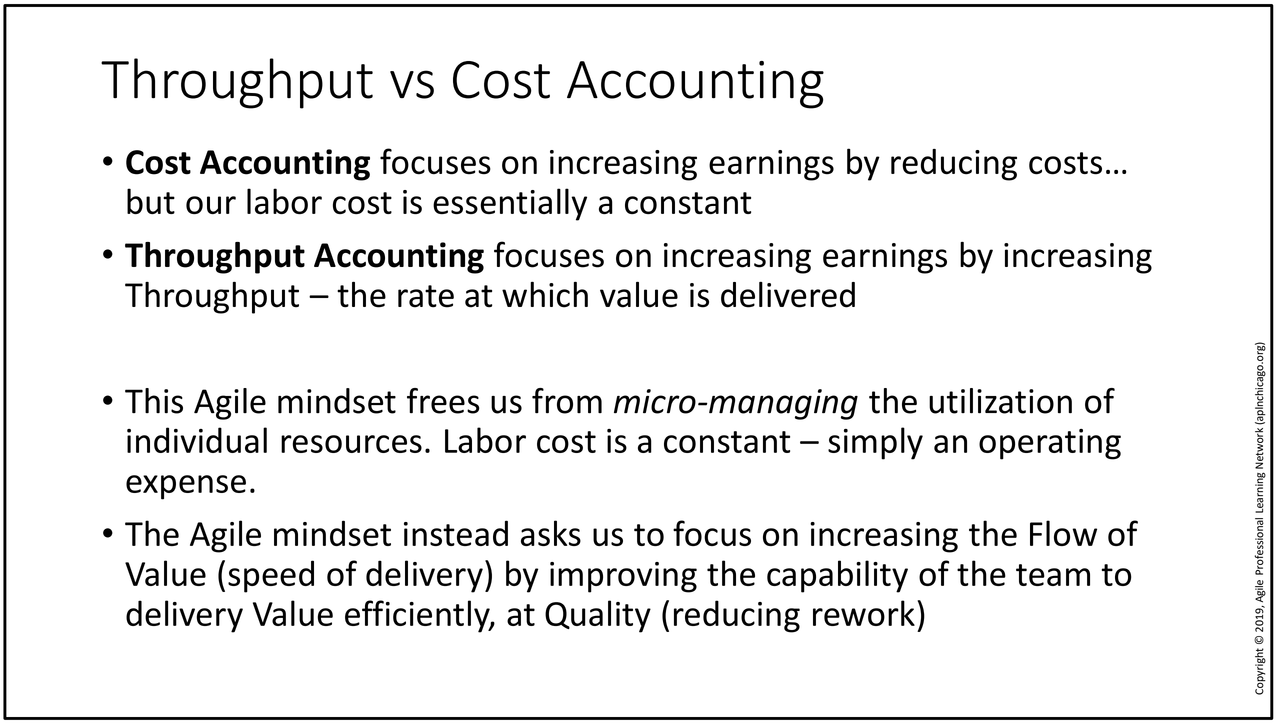
Oddly, Ed’s efficiency means his outlay per point is even lower than mine, even though he has a bigger team! It’s not all roses for Ed. He still has to truck all that equipment around, so he needs to add additional travel cost as well. That’s why he loves getting groups of customers in the same area.



Stable Funding

- In an Agile world, with individuals fully dedicated to cross-functional teams, we have unlocked something powerful
  - Our team expense is no longer a variable!
- We know what a team is going to cost us every sprint
  - We know how many working hours there are in a sprint, and we know what our resources cost us per hour. So, our labor cost becomes a constant operating expenditure.

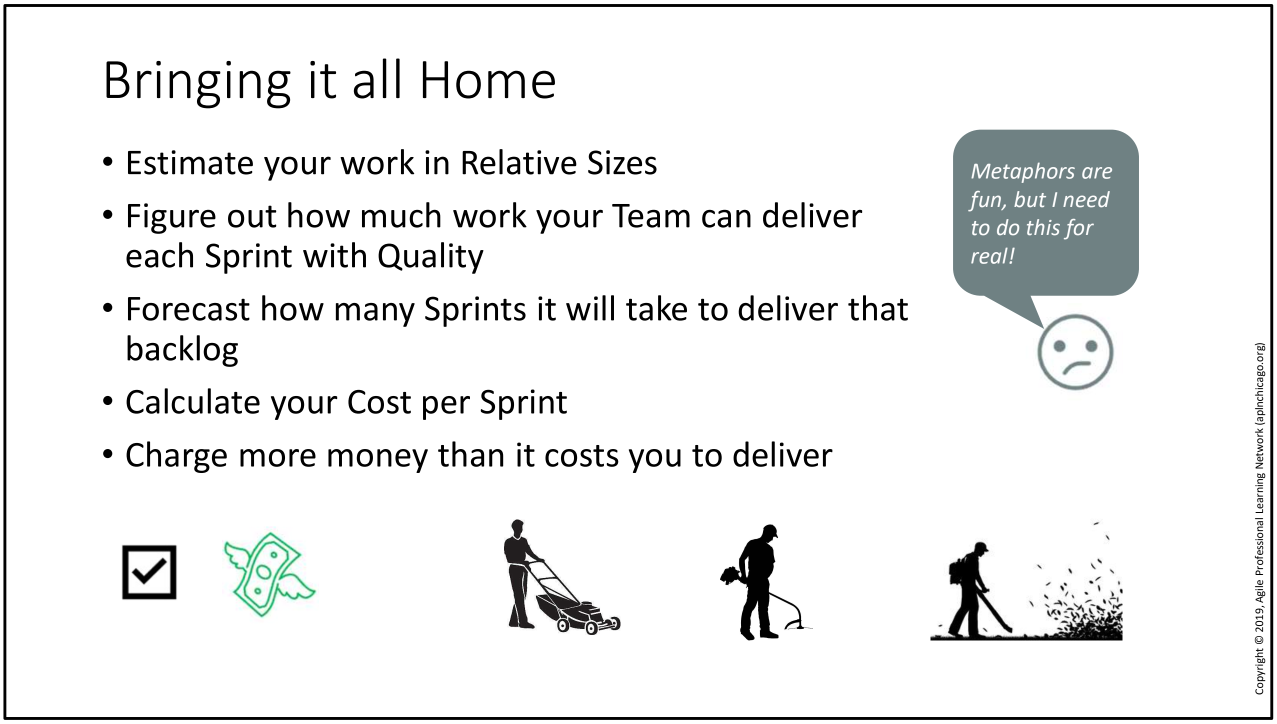
If we are asked to forecast delivery of a feature, we could estimate the stories that describe that feature and divide the total points by the average velocity of the team --- giving us a rough number of sprints to deliver that feature. Multiply that number of sprints by the cost of a sprint, et viola, you have both a cost estimate for the feature, and a projected duration for delivery if that was all they focused on.



Throughput vs Cost Accounting

- **Cost Accounting** focuses on increasing earnings by reducing costs… but our labor cost is essentially a constant
- **Throughput Accounting** focuses on increasing earnings by increasing Throughput – the rate at which value is delivered

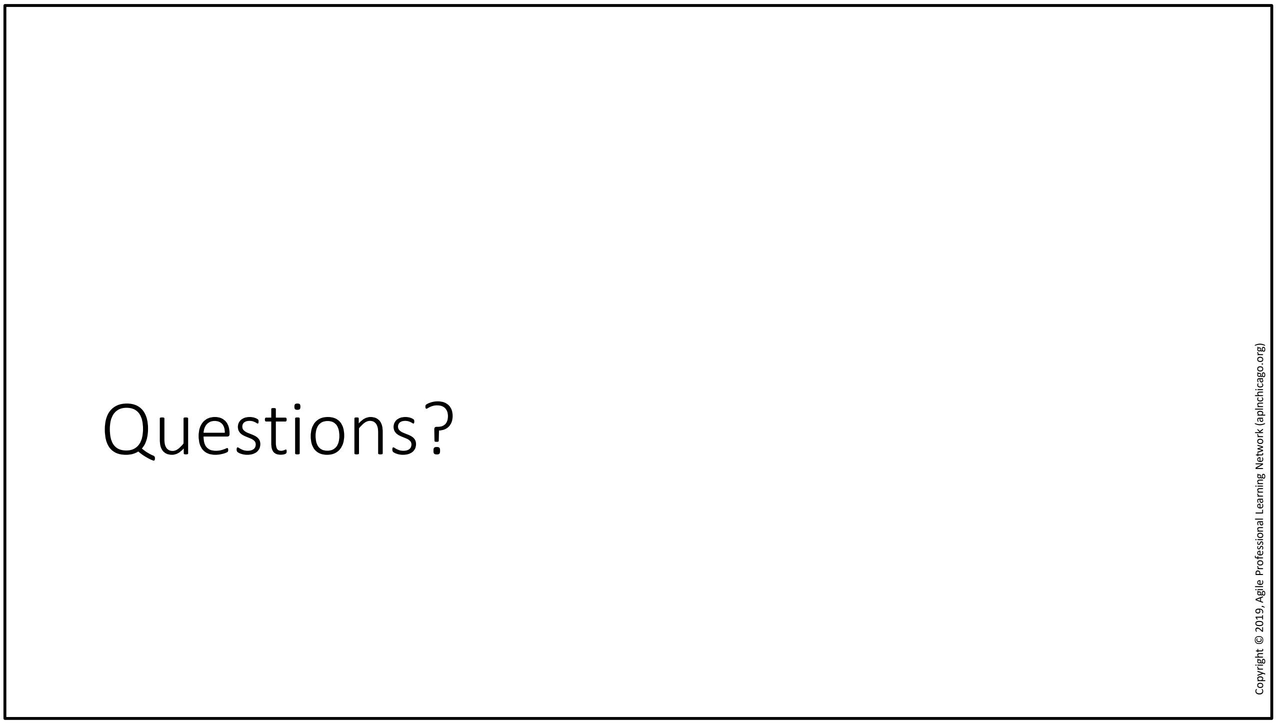
- This Agile mindset frees us from *micro-managing* the utilization of individual resources. Labor cost is a constant – simply an operating expense.
- The Agile mindset instead asks us to focus on increasing the Flow of Value (speed of delivery) by improving the capability of the team to delivery Value efficiently, at Quality (reducing rework)



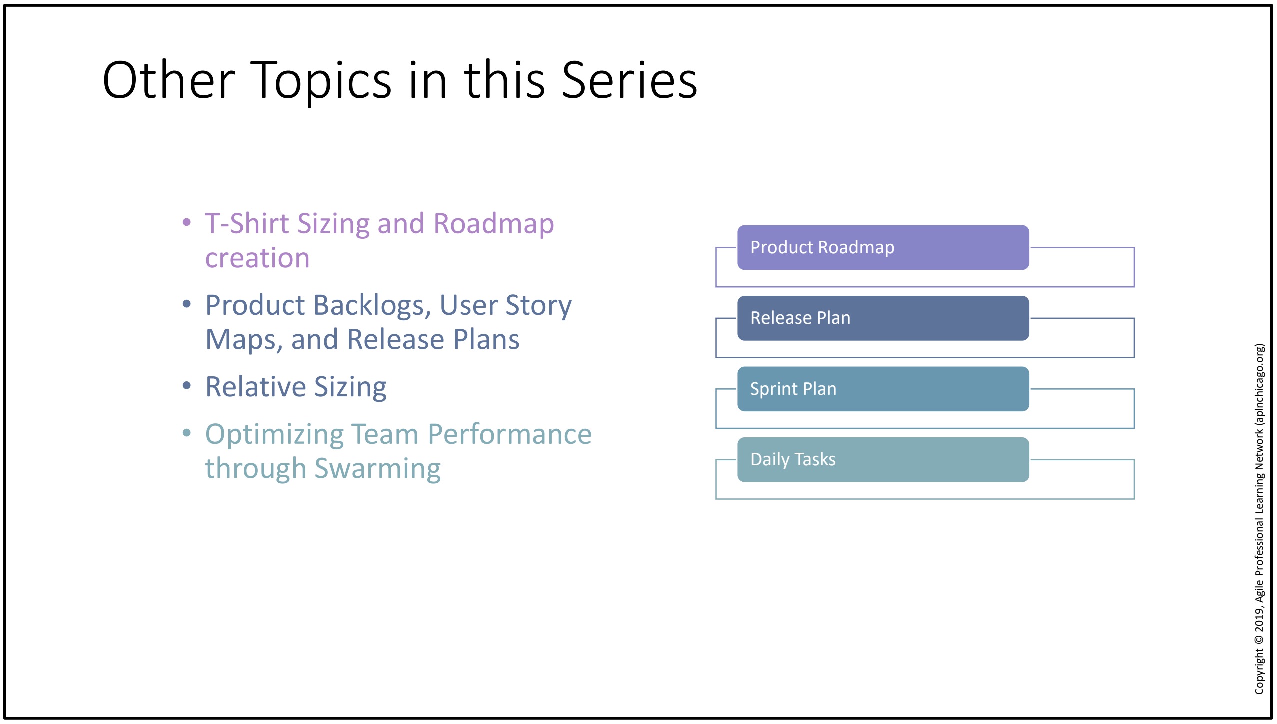
Bringing it all Home

- Estimate your work in Relative Sizes
- Figure out how much work your Team can deliver each Sprint with Quality
- Forecast how many Sprints it will take to deliver that backlog
- Calculate your Cost per Sprint
- Charge more money than it costs you to deliver

Metaphors are fun, but I need to do this for real!



Questions?



Other Topics in this Series

- T-Shirt Sizing and Roadmap creation
- Product Backlogs, User Story Maps, and Release Plans
- Relative Sizing
- Optimizing Team Performance through Swarming

Product Roadmap

Release Plan

Sprint Plan

Daily Tasks

Thank You!