Naked Planning

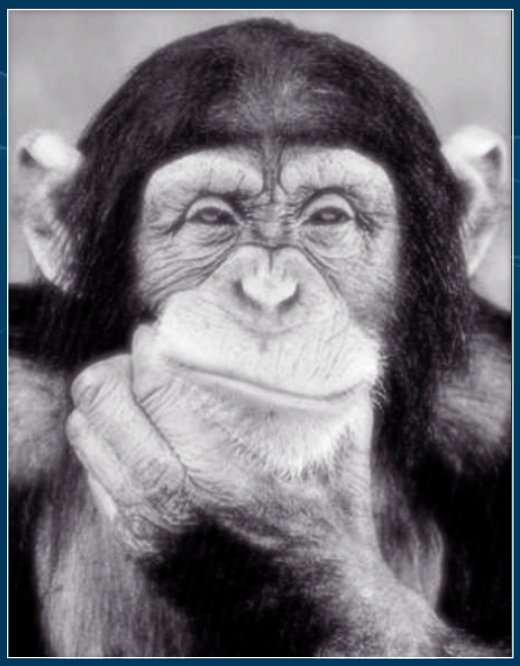
Estimation to Agile Estimation to No Estimation

improving

How long will it take? How much will it cost?

We're not good at it.

Software Estimation



+

- Wisdom of crowds
- Experience
- Statistics

improving

Try to account for that in three key ways.

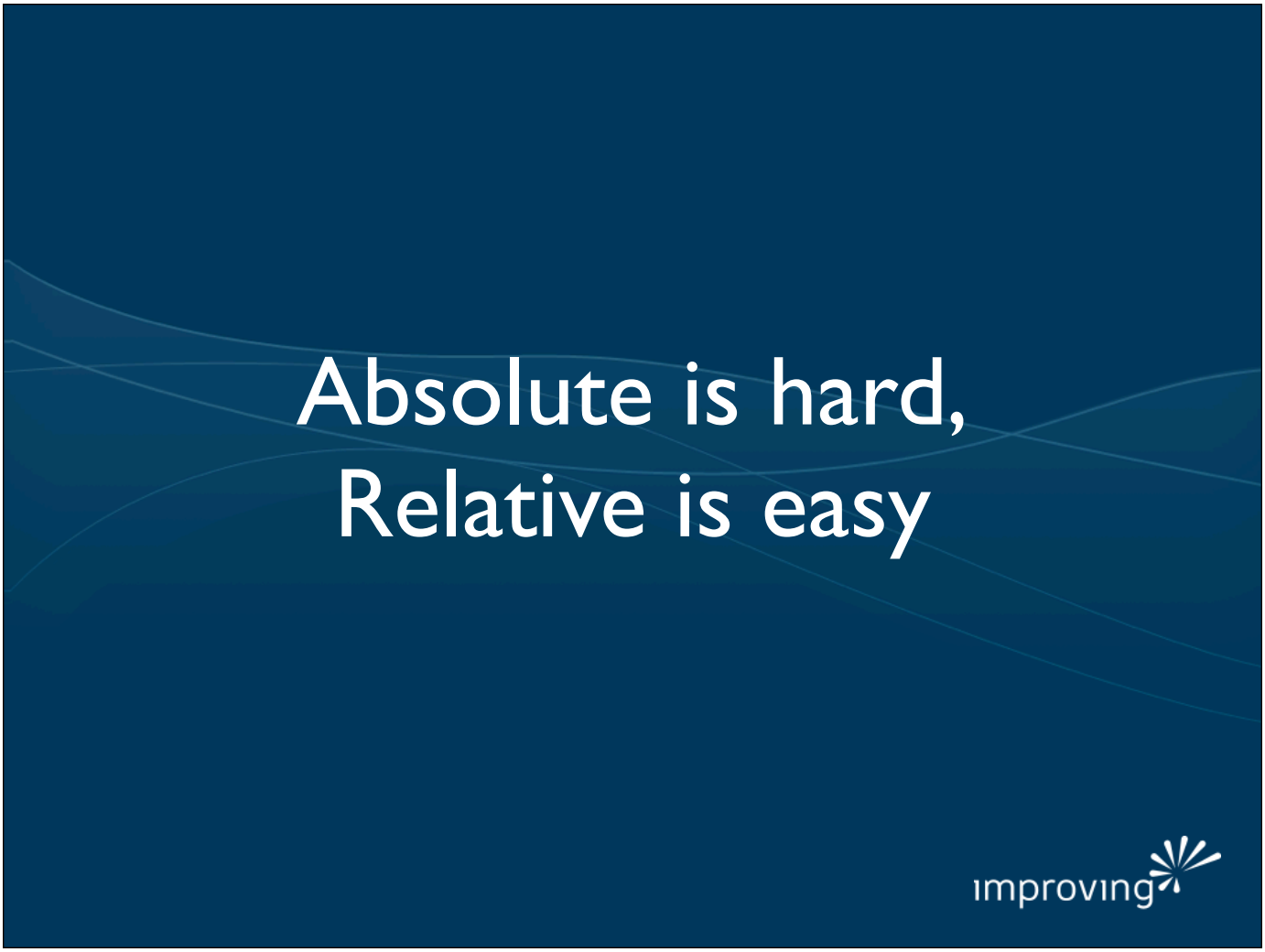
Software Estimation

Wisdom of crowds		Groupthink
Experience	⇒	Things change
Statistics		Variance

improving

Those solutions have their own problems.

But why is it so hard in the first place?



Example: How wide is this room? Not easy to say. Is this room wider than it is long? Easier.

Why? Presumably a survival trait...

We don’t need to know the bear is 50 feet away. Just that Joe is closer to the bear than we are.

Agile Estimation

Makes use of strengths
and minimizes worry

improving

Uses relative estimation.

Incorporates (or makes clear) the need for experience by using evidence.

Doesn’t ignore variance, but avoids worrying about it at points where nothing can be done.

Typical Agile Process

Take something you have done or is understood well.

Compare new things to it.

improving

While doing this, create discrete buckets using fibonnaci to assign “points”. Everything must fit in one of the buckets. Once everyone understands the buckets you move to...

Planning Poker

Incorporates wide-band delphi.

Wide-Band Delphi

1. Group of experts estimate.
2. Review estimates.
3. Discuss what is being estimated.
4. Repeat as needed.

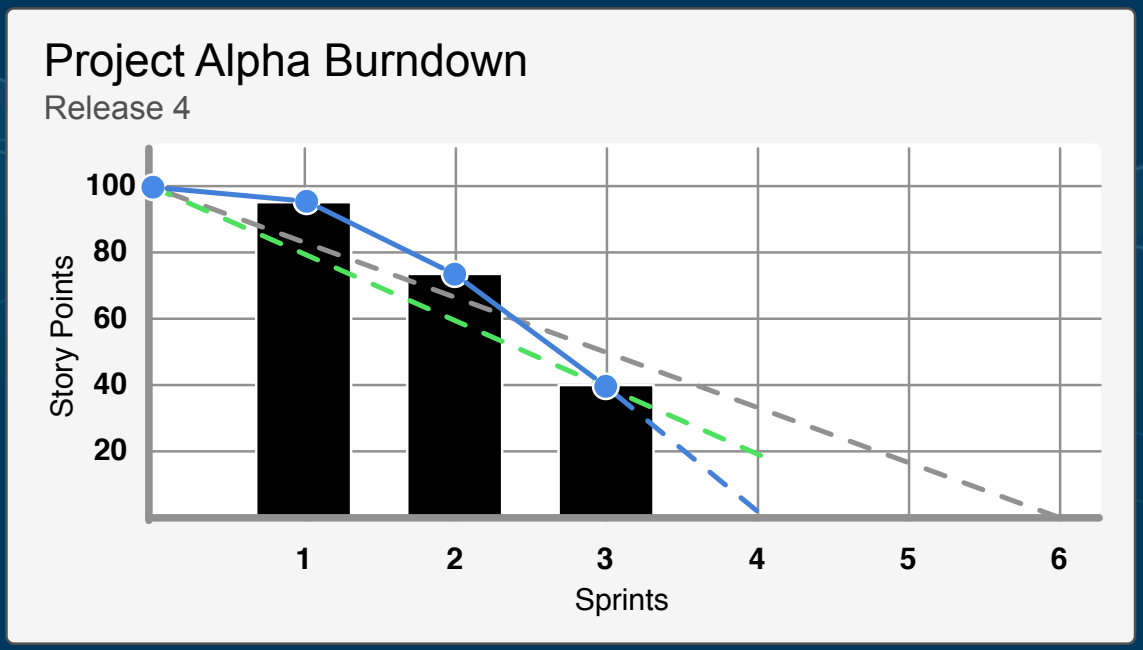
improving

Key feature is that no one shows their estimate before anyone else. In planning poker, the team uses cards to do this. Cards have numbers from fibonacci. Usually range from 1 to 21. Also have 100, “?”, and “Break!”

Everyone picks a card from their hand. Once all are picked, the whole group reveals at once.

If consensus is something is “too big”, it must be split.

Build and Use Evidence



Use rolling average.

Estimates Don't Matter

improving

We've spent most of our time talking about estimates... now here's why you shouldn't do them at all.

Over time on projects, as the team (developers and story writers) gain experience, everything tends to gravitate toward a smaller set of buckets. Usually 3, 5, and 8, sometimes a little different. This happens naturally.

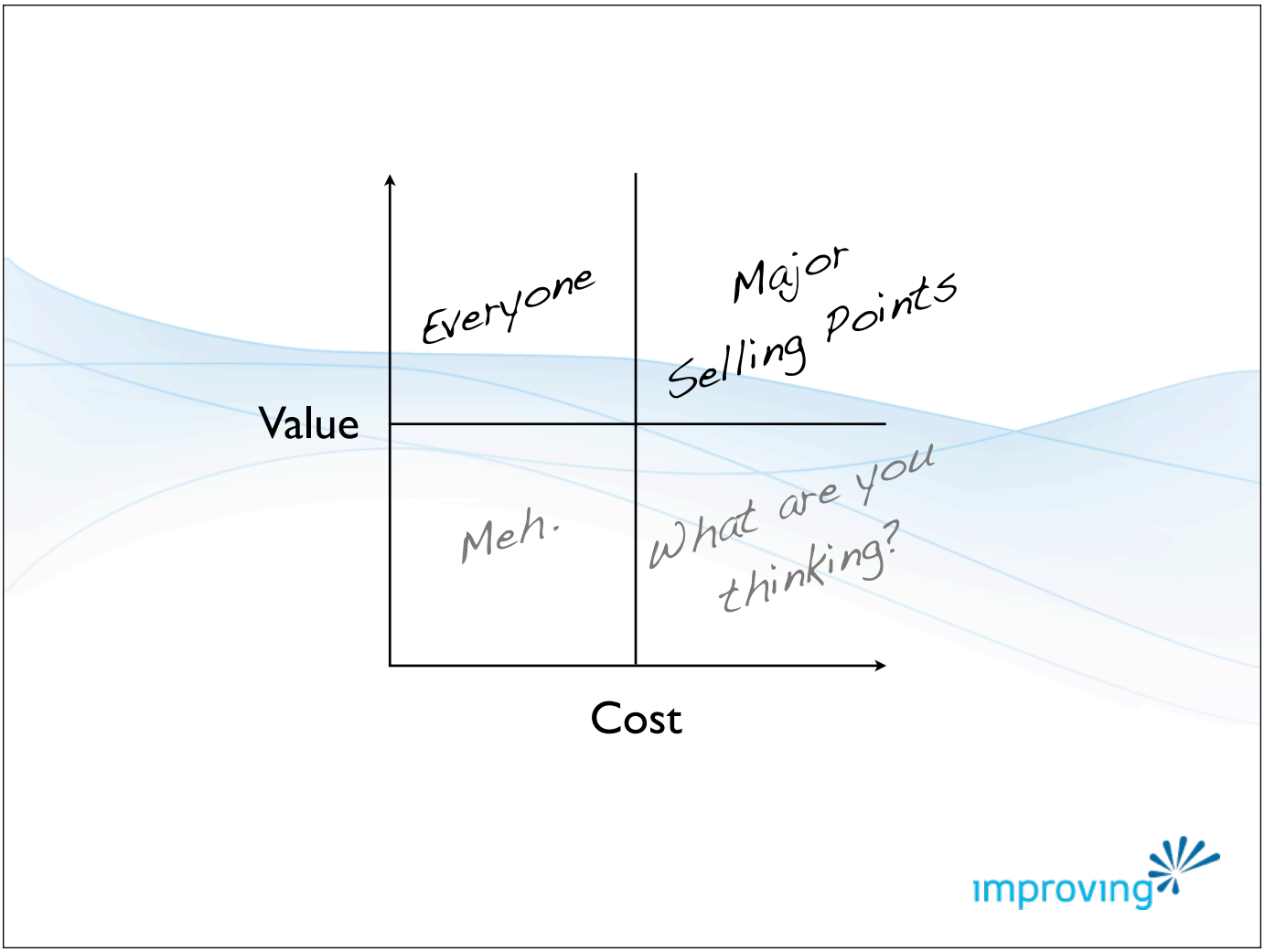
If that happens naturally, why not put in a little work and make everything fit in one bucket? That would mean you don't need to estimate; everything's a 5.

Naked Planning

improving

Arlo Belshee invented it. Basically means a stripping down of project planning to its bare essentials. Similar to the Kanban method of David Anderson.

Just like agile doesn't mean ad hoc or "cowboy coding", neither does this.



Working on high-cost and low-value is just stupid.

Working on low-cost and low-value won't win you customers. It's something of a death march or race to the bottom at that point.

Low-cost and high-value features are attractive to the CFO, but everyone can quickly replicate them.

High-cost and high-value features are where you really differentiate your product. They have long-term value.

Focus on Value

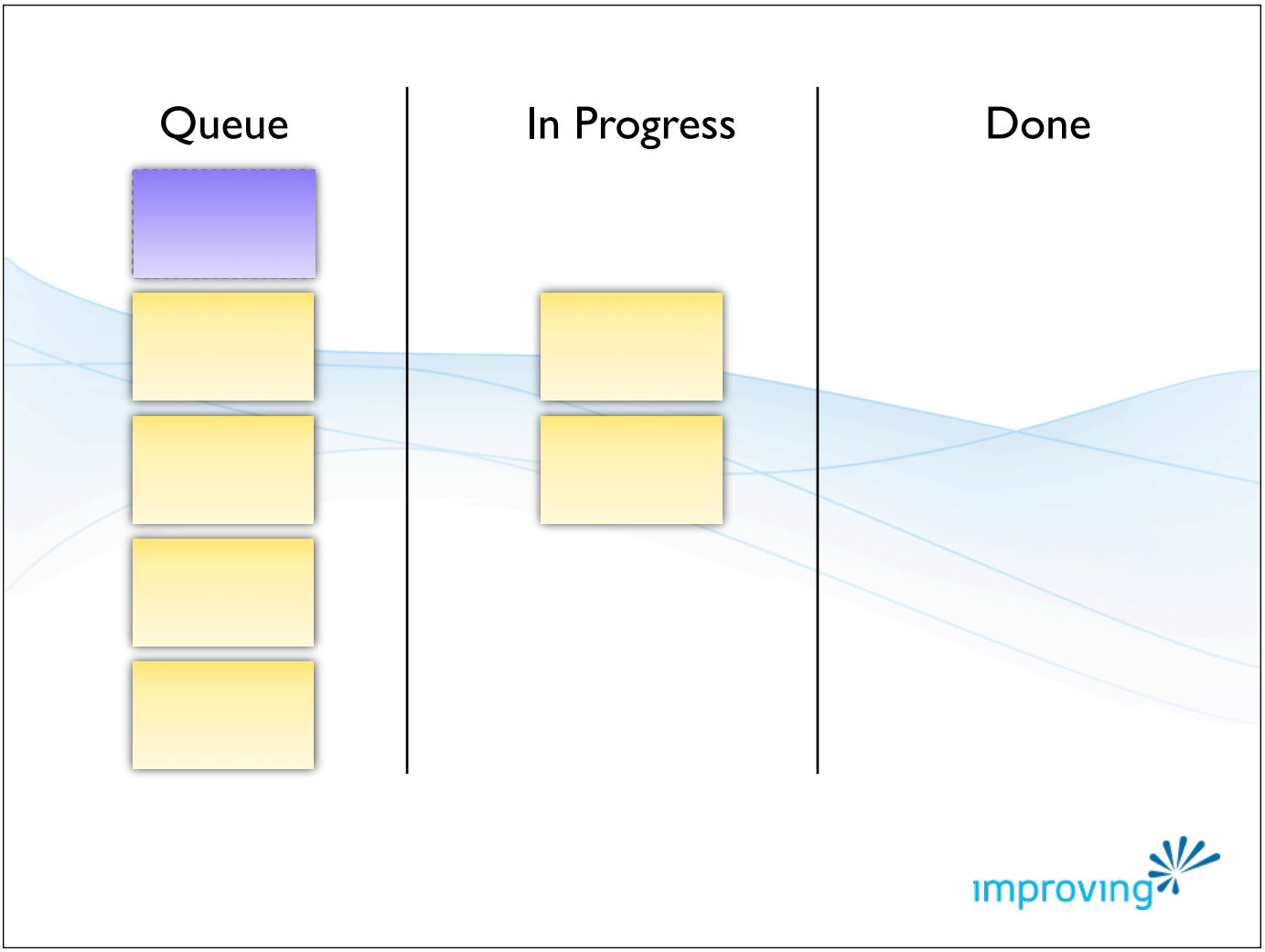
improving

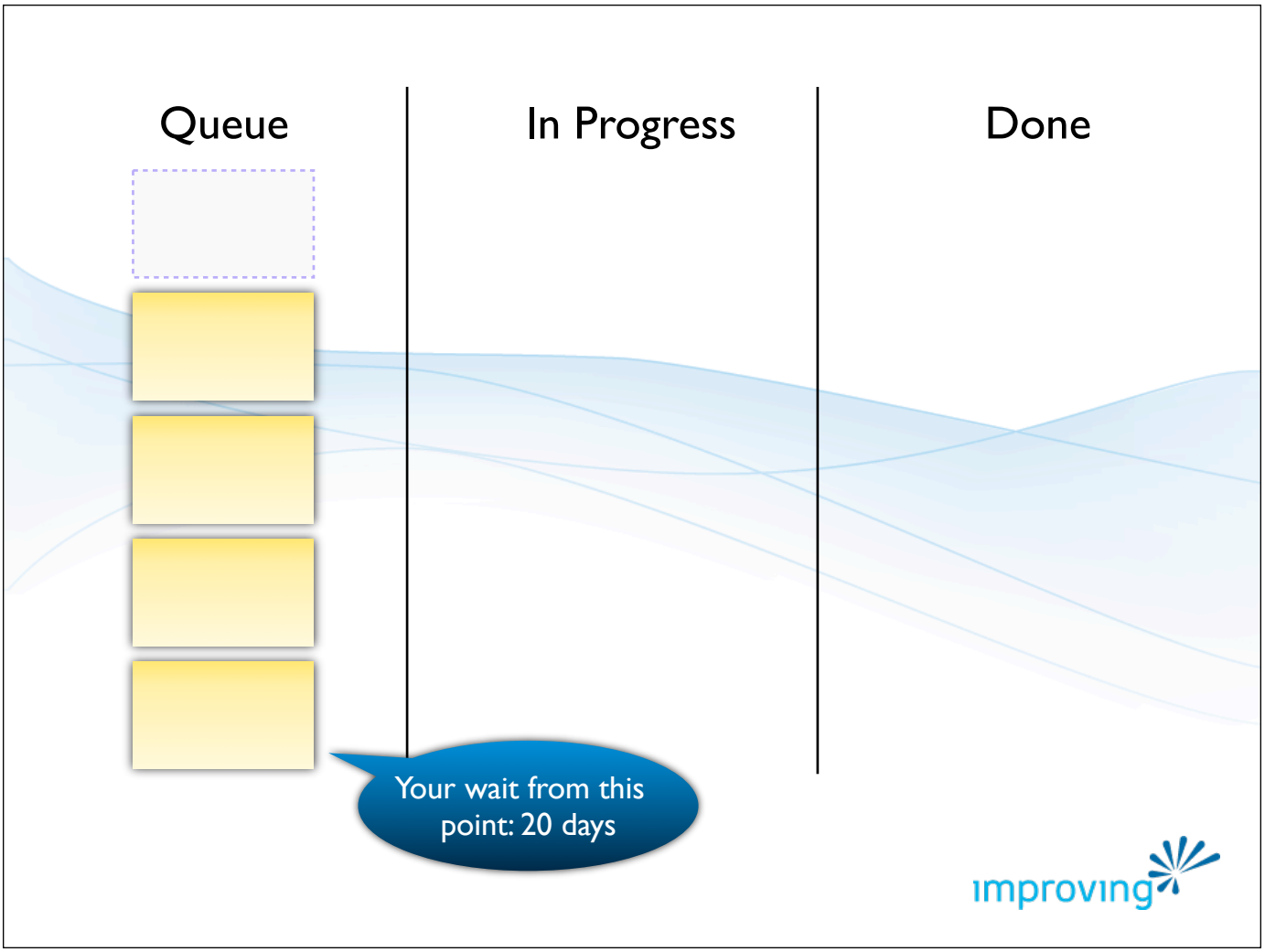
We already said cost of features/stories will end up relatively the same. Value doesn't work that way. So focus on the value and let the costs work themselves out.

That doesn't mean work only on high-cost and high-value. If you do, your competition will appear more reactive and thus better. You need a combination.

Pull and Flow

Naked planning doesn't use iterations. When capacity is available, the team pulls work in. This leads to a smooth flow of work through the process from backlog to production. This requires a lot of discipline because the team must be able to work on each story (minimal marketable feature) and release it separately from the others.


Queue
In Progress
Done
improving



Team pulls something in, the stopwatch starts. Once it is done, the stopwatch stops. If it is five days per feature, and my feature is 4th in line that means that it will be approximately 20 days before my feature is live.

At this point, you can have some fun with statistics. A key is to monitor standard deviation of that time in queue to keep the deviation as small as possible.

Don't go out and do what I've just said Naked Planning is. This form is my interpretation and it is designed to work in specific circumstances. A dedicated, solid team working on a feature-by-feature basis on the same product for a long time.

Even if that describes your situation, still don't just do this. Instead do a search for "Naked Planning" and read up on it elsewhere. Compare it to David Anderson's Kanban process. Understand *why* they are the way they are. Then apply what makes sense to your projects.

Credits

- Arlo Belshee - Naked Planning
- David Anderson - Kanban
- Ben Rady - Houston TechFest talk and Improving Podcasts on Naked Planning

improving