Webinar: Velocity and Story Points Ver 6

LeanAgileTraining.com
May 18, 2026
© 2026 Joseph Little

Reminder

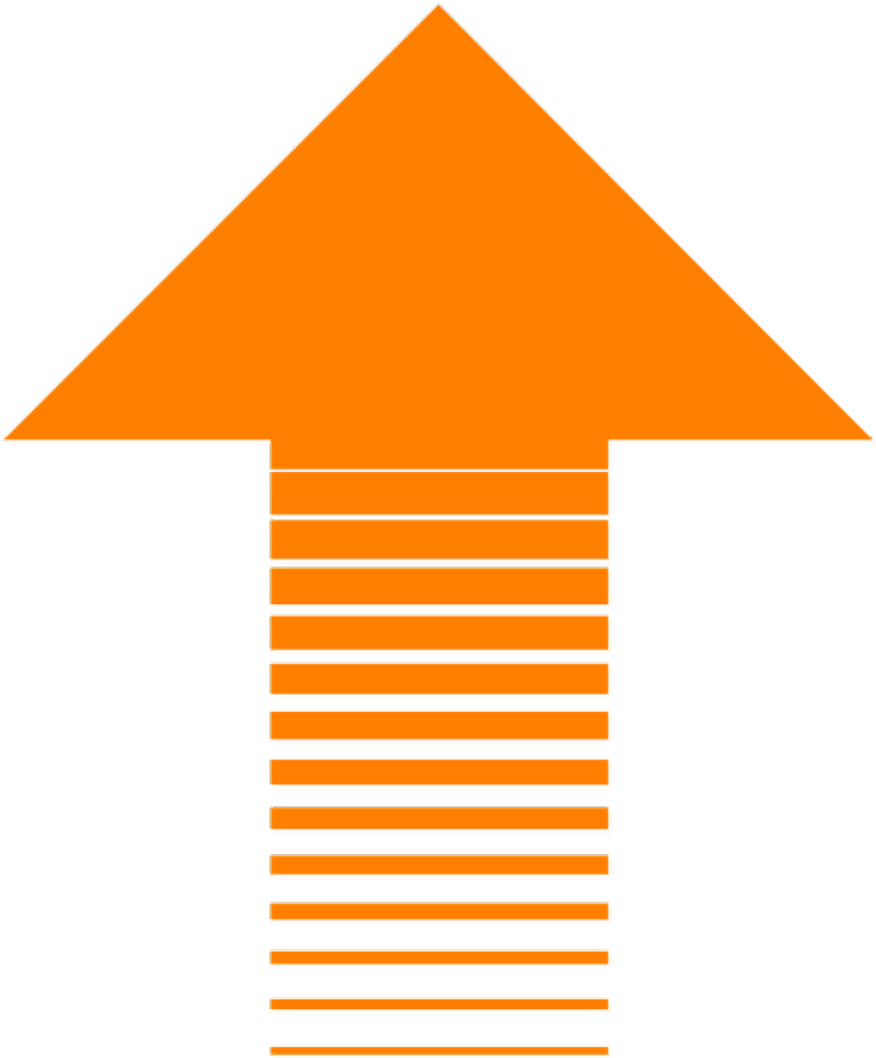
To get the PDF of the slide deck:

- I need your email attached to this webinar
- Send an email to info@LeanAgileTraining.com



Scope of this discussion

1. Why?
2. Scrum is a Game
3. What are Story Points? How do we use them?
4. Planning Poker is a Game.
5. Winning or ~~Losing~~ Learning.
6. Best we can cover in 1 hour.



Questions

Please ask

Along the way or later

Out loud or in chat



Some Emotion

- This is a topic with some controversy.
- The “no estimates” guys don’t want to do any estimating or planning.
- There is, I think, another group who just hate Story Points and/or Velocity.
- I do think there are some reasonable concerns that explain why they have these positions.
- So, for those people, I am to some degree trying to make a case: please think about all this a different way. Take another look.
- Also: I know I will never convince some of them. Maybe no one would.



AI means we start to talk differently

- It used to be: “the Velocity of the Whole Team”
- Now: “It is the Velocity of the Whole Team, including particularly whatever extra power they can get, on the Business side or the Developer side, from AI (AI agent or any other kind of AI).”



First: A Game

- Scrum is a game.
- So important, so often forgotten.
- If you have a game, you have to keep score.
- Keeping score helps many ways, but of course you have to be reasonable about it.
- And you have to help the Team win. Mostly win. (51%?)



Issues, we have issues!

1. Team members are uncomfortable with Estimating
2. Estimates for the Sprint can be notably inaccurate
3. Some people hate Story Points
4. Some people hate Velocity
5. Some people have misused Velocity and Story Points
6. Over-Stress (no fun)
7. Unaware of Learning!



Key Ideas

1. The Scrum Team wins (or learns) together

2. Winning is good! (“We are the Champions!”)

3. Losing is Good! (“You live, you learn!” Alanis.)

4. People are engaged more with Games.

5. Learning is important for Knowledge Workers!



What is Velocity

- Sprint 1= 18 SP
- Sprint 2= 22 SP
- Sprint 3= 20SP
- AVERAGE = 20SP
- What people should mainly talk about is: The average Velocity over the last 3 (or 4) Sprints.
- And that is the start to identifying the Capacity for Sprint 4. Ex: Might be we agree 21 SP is our Sprint 4 Expected Capacity.



What is the Velocity this Sprint?

- Story 1, 3 SP, done
- Story 2, 3 SP, done
- Story 3, 3 SP, done
- Story 4, 3 SP, done
- Story 5, 2SP, done
- Story 6, 2SP, done
- Story 7, 2SP, done
- Story 8, 2SP, NOT done-done
- Expected Velocity: 20 SP
- Actual Velocity: 18 SP
- Rule: The stories done-done count toward your real score.
- No credit for partially done.



So what? Why?

- Scrum is a Game (see cover of Scrum Guide 2020).
- Each Sprint is a Game.
- The TEAMS sets a goal for the Game: the number of SPs the team expects to complete.
- Then we measure how we did!
- Then: we win! (Yippee!) or we learn! (Great!)



One key benefit

- Less stress. LESS stress!
- Why?
  - We (the “Developers”) get to set the goal
  - We only commit to what we think we are reasonably likely to complete - fits in “capacity”
  - We get to demand “good enough” information
  - We *adjust things* to “win” between 50-60% of Sprints
  - We KNOW over-stressing is bad for knowledge workers



Velocity enables the benefits of the Game

- If we win, it builds the Team. (We sing “We are the Champions”)
- If we don’t win, then we have an incentive to improve, so we can win more later.
- We get transparency that we need to improve.
- We only have to look at the elements in the “losing” Sprint to see what to improve on. Again, this improves transparency.



How do we use it?

- To define our expected capacity (to decide how much to commit to)
- We start with the Velocity number
- We (the Team) can adjust it (for the next sprint), up or down, based on fair reasons.
- And then we commit (eg, to 20 SP of work). (Committing is more than just the SPs.)



Why might we adjust from the Velocity?

- People are OOO more (or less)
- An impediment was fixed
- We will collaborate more or better
- The PO will answer questions faster or better (or vice versa)
- The SM will get blockers fixed faster or better
- Or WORSE. (S### happens!)
- Etc



The Devs decide how much to take in the Sprint

1. The Team accepts stories
2. The Team insists that stories have “good enough” details
3. The Team adds up all the SPs, and compares with expected capacity
4. The Team also looks at the sum of the hours on the tasks, and compares to expected capacity. Maybe adjusts.
5. The Team then commits.
   Ex: Cap = 20SP, so commit to 20SP.



It’s a Game

1. We plan and set a Sprint Goal (focus now: the 20SP)

2. We work and measure, and see if we can score enough (ie, SPs). Yes, life is rough.

3. We get a small win! One Sprint. We celebrate.

4. Or we learn. Have an incentive to improve. Become a bit less imperfect. Like every ball club.

5. Recommend: Try to win 50-60% of games (Sprints).



Understand Deviation

- Probably improve if we have 8+ small stories
- We need to identify and mitigate things that cause large Velocity deviations (compared to expected). Ex: over-committing, poor estimating, etc.
- Big stories cause deviation.
- The Developers get to estimate and commit. Average the estimates of each person.
- The visibility is better and the climb starts in a better way



Scrum is fun!

- Commit so you can win
- Work smarter, not harder
- Build high motivation (looking at the PO!)
- Celebrate small wins
- Enjoy game of Planning Poker
- Use planning process to attain eu-Stress
    - And to increase learning, eg, we become more effective as we learn



About Estimating Stories

How to do Story Pointing

- The Developers (coders, testers, etc) vote
- Reference Story = 1 SP
- **Wide-band Delphi expert estimation (Rand Corp)**
- Use Fibonacci cards
- Often several rounds (3 or 4?)
- Within 3 consecutive Fibonacci cards
- THEN: Average (to nearest integer)



Example

1. Three voters (prefer 5 voters).
2. Initial votes are 21-8-2
3. Extremes share knowledge
4. Next vote: 5-3-2. Three consecutive Fibonacci numbers.
5. Vote: 5+3+2 = 10. 10 / 3 voters = 3.3 SP
6. But we round to 3 SP.



Breaking Up is good to do

1. Learn 20 Ways to break up stories (xp123.com). Build examples. Practice.

2. https://leanagiletraining.com/key-problems/breaking-down-user-stories-some-resources/

3. Feels hard at first. Becomes easy.

4. Estimates are better for smaller stories

5. Helps that all the stories in a sprint are small, about the same size.

6. Different win: 80-20 Rule!!!



More Tips

- You never have complete knowledge - always learning.
- So, estimate the best you can with the knowledge you have now.
- Coders and testers both estimate for the total “work” by the Team. (Yes, a learning process)
- Estimating and reviewing helps identify knowledge you want. Add knowledge and re-estimate.



More Tips - 2

- Share your knowledge (share the learning)



Planning Poker Game

1. Makes it fun.
2. Fun brings engagement
3. Engagement means more learning, more sharing, more retention of knowledge
4. Everyone brings something to the party
5. Small “mistakes” are forgotten (especially if we win often)
6. Learn from the bigger mistakes



Key Win

- The knowledge the Team gains helps estimating and planning.
- And helps them **build** a better product. Better execution.



Human Psychology

I hope you see how this all builds on some basics of how people work.

How they think and feel.



And LEARN

1. **Re-estimate multiple times**
2. Re-estimate a given story when you gain new knowledge (more detail, other info)
3. Developers get “all” the information they need, eventually
4. Voting (and doing) helps them learn what info they need most
5. “The Season is more important than one game”
6. LESS pressure.
    - They do not know at first what the Velocity will be



Two wins from Planning Poker

1. **The number on the card gives visibility to what our estimate is. This enables learning.**

2. This visibility allows us to “prioritize our stupidity”, and learn.

3. We learn about each other (eg, within the Developers). This will help.

4. **We share the most useful knowledge quickly**. In about 60-75 mins, initially. Awesome!

5. “Everyone knows something.”

6. Then: Everyone knows what all of us know.



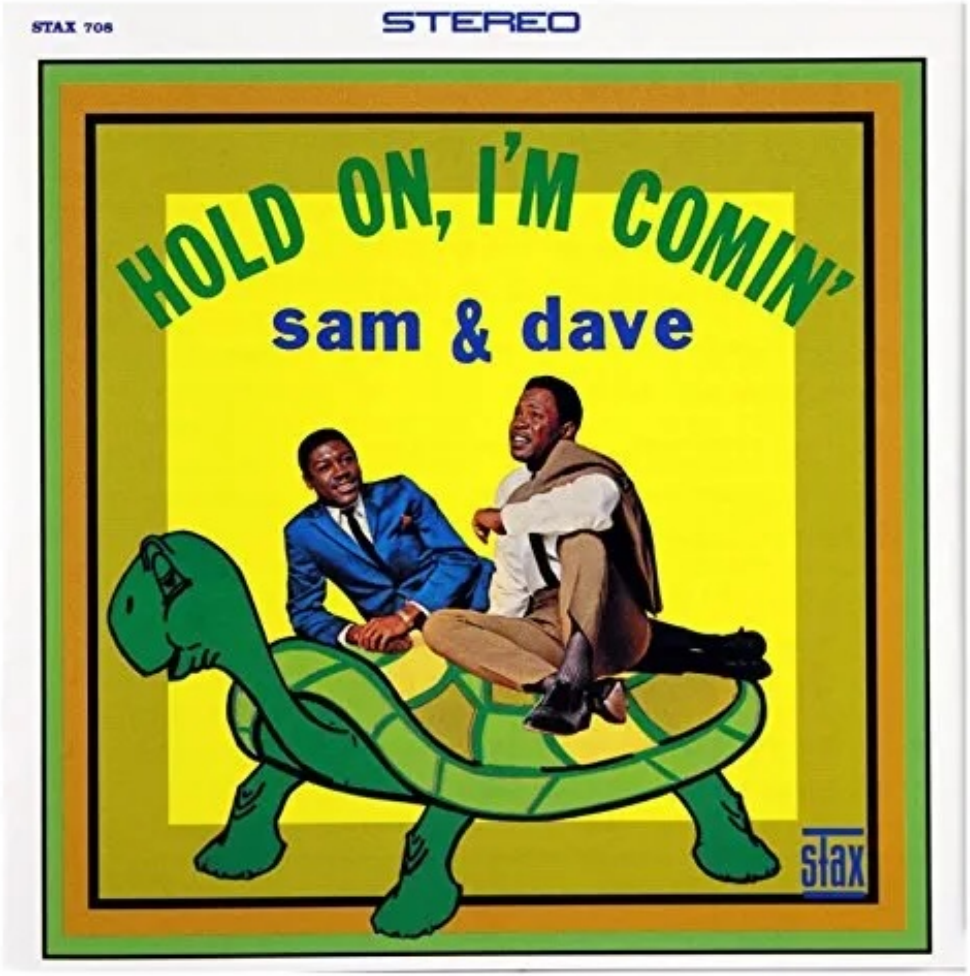
Eternal problems

1. Humans avoid the truth
2. Human illusion (I am Superman!)
3. Customer wants and needs are unclear
4. “Impossible to see, the future is”
5. Unclear Requirements
6. Stuff happens
7. “What was it you wanted?”
8. Other distractions
9. Dementors
10. We wait for perfection
11. “Oops, I did it again”



“Hold on, I’m Coming”

- When in trouble with the Sprint Commitment:
- A firefighter has a simple plan (Emergency Procedure)
- Collaborate more or better!
- Fix impediments!
- Get answers quicker!
- Do something unexpected! (Get creative team!)
- Drop the lowest priority story!
- Try to fulfill the Sprint Goal, at least.
- Learn!



Later

1. **Again, if you “win” that Sprint, “we are the champions"**

2. If we lose, we see clearly that we did.

3. If it was special stuff hitting the fan, well, stuff happens

4. If we can learn, then we actively try to learn. (It HELPS to “lose” some. And learning is not losing.)

5. ALWAYS: We want to fix impediments. And the Team actively tries to ID -> what is our biggest impediment now.

    - What is the best thing to work on to improve ourselves.

6. Velocity is one way to see if work on impediments actually helped. (Sometimes not.)



Questions

Please ask

Now or later



Other webinars

1. Two types:
   - About courses and workshops (30 mins)
   - About agile ideas (60 mins)

2. We welcome your ideas!

3. Sign up at our website or at MeetUp (“Lean Agile Training webinars”)

4. Don’t forget our courses and workshops!



Contact us

- Please contact us to discuss further. Joseph Little
- info@leanagiletraining.com
- Happy to discuss.

