Table of Contents

Simulating Agile Execution with the Ball Point Game

I have no idea where the "Ball Point" game came from but I learned about it from Bob Schatz. Since learning about it I have found a great game to help people understand (by doing) what it means to execute in an agile (Scrum) way.

Setup

I introduce the game by saying "Whether you knew this or not, when you came to work today you actually came to work in our 'Ball Point' factory. Your 'ball point' team is everyone in the room. Work consist of passing the ball around from person to person, passing through each person until the ball gets back to the start point. When this happens, you record a 'ball point'. Your purpose in life is to generate as many 'ball points' as you can."

I then go through the set of "rules of the game" per the slide:

Exercise – The Ball Passing Game

- In 2 minutes come up with an estimate of how many balls you will be able to move through your "factory" in a 60 second period subject to following rules:
    - One big team (i.e., everyone)
    - Must pass the ball
    - Ball must have "air time" (i.e., cannot just hand it to someone)
    - Ball cannot be passed directly to a neighbor
    - Start point = end point
    - When a ball goes through each player and comes back to the start point it means the team records "1 ball point"
    - Quality control – dropped ball has to go back to the beginning

And explain the basic mechanics of the game.

"You will be given:

- 2 mins initially to put together your plan
- At the end of the planning period you need to provide me with an estimate of the number of 'ball

points' you think you'll get done in a 60 sec period
- We will then run an iteration of 60 secs where you'll do the work
- You will then be given 2 mins to plan your next set of work and provide me with another estimate of how many ball points you'll think you'll get done in the next 60 sec period
- We are go to do 5 60 second periods
- We have "quality control" - if you drop a ball that ball has to go back to the beginning if you want to have that ball count

Does everyone understand. OK go - 2 minutes until you have to provide me with your first estimate.

Oh, and by the way, 'no tools' is another rule. You cannot just put all the balls in a bag and pass the balls around that way."

What You'll Need

An area big enough for everyone in the room to participate in a rough circle.

A box of about 20 balls.

Something to time each 60 sec and 2 minute period

A flip chart / whiteboard as follows:

Sprint	Estimate	Actual
1		
2		
3		
4		
5		

Facilitation

I find it easiest to be the scribe and time-keeper for the game. I just use a countdown timer on my phone and write results up on the board as we go. I also take notes as the game progresses based on what I see and hear.

Things will seem chaotic at first. Just remind them of the 2 minute time box for an estimate at the appropriate time. Then force the estimate, and get the group to do the first iteration.

People will try to second guess themselves and you. You will be asked things like "is someone opposite me a neighbor" and so on. Answer the game ones as quickly as you can, and as often as not by pointing to the rules of the game with a comment "its your process …"

Try to make sure that the estimate provided is a "team" view, not just whoever spoke first or the loudest. This will help with team accept the results.

In the first sprint watch for teams not counting completion correctly, or failing to count at all. You don't want to have an argument so if they have a number jot it down, and then put what you thought the number was beside. Another area that might be useful for debrief.

After the 3 sprint I typically make the following statement "You know when I did this with another group (perhaps name something) they were able to produce <a big number of ball points> … just saying" I calculate the <big number> by multiplying the current "actual" by 2.5 or 3 and a bit (don't want to make it too obvious). Be careful how to phrase this - you don't want to push a "stretch goal", just provide the information.

Debrief

One of the interesting things about this simulation is that there is a lot can be shown and talked about as a result. This is what makes a great exercise, but it also means that you should be selective about what is highlighted, partially lead by what the group raises up but also, as a coach / trainer, you might have a couple of ideas in mind. In particular, do not try to do all these messages, just focus on 2 or 3 key ideas.

To start the conversation you might want to start with something open ended:

- What happened?
- What did you observe?
- What surprised you?

You can then guide the remainder of the debrief around the remaining points depending on what you noticed, team interest, and time.

Let's look at a typical set of results that you will get out of this game and look at things we might see.

Sprint	Estimate	Actual
1	5	1
2	7	10
3	15	13
4	26	9
5	26	23

What do we see?

1. The first sprint was way off in terms of the estimate. Ask the team how they came up with the estimate and why they gave me a 5. You will find a lot of "reasons" but we need to point out that 0 is a perfectly good estimate in the absence of other information (after all, how would we know, since we had never passed balls this way before). Also ask that if they feel compelled to provide an estimate why they didn't just try out passing some balls around to test their ideas in the two minute period (I've had a couple of teams that have done this, but most do not).

2. We have a 23X improvement in "productivity" over a 5 sprint period. Most teams do a pretty dramatic improvement. And you can suggest that when teams start operating using an agile approach it is not uncommon to see this.

3. How did the improvement come about? It came about because we stopped and talked about our "work" and figured out a better plan.

4. Where did all the good ideas come from? What you will find is that a number of people will contribute ideas. This leads to a discussion about the wisdom of the crowds and why top down planning is wasteful as it doesn't fully leverage the brains of all the people on the team.

5. The previous point also leads to a conversation about leadership. Who was the boss? In most cases there was no boss. People became leaders as their ideas were implemented. This is part of the dynamic of a true team and one of the reason they accelerate. We say that leadership "emerges" and, it should noted, the role of leader might flip from person to person depending on context.

6. What was the effect of quality control (dropping the ball)? Not only was it descriptive in a general sense ("oh someone dropped the ball") but it also took time to recover as people got back into a work rhythm. Think of defects in a similar light? Even if we don't care about the customer and their reaction to a defect, we need to stop producing defects (technical debt reduction, invest in automation) so we can focus on providing value.

7. Most teams start by throwing balls around, whether or not the person on the receiving end is ready. Most teams evolve to have a rule to "wait until you have eye contact with person catching the ball before you throw". This is an example of a pull system. This allows you to talk about the improvement in productivity as a result of the smoother flow you see as a result of pull systems.

8. Ask about the effect of the of knowing what "another team" did. Often you will see teams say "we need to get at least as much as that team". You can point out that you lied and just made the number up, but then ask "why would it matter what the other team could do?" The only thing we can control is what we do. We don't understand the other teams context and so we shouldn't assume we can use the same approach. Perhaps all the entire team was ex-basketball players … There is nothing wrong with trying to learn from others. But we shouldn't say "we can do that amount" until we have actually done it because we "have to be as good as them". You just set false expectations that way.

9. You will often see a dramatic increase in productivity from one step to the next (e.g. 9 to 23 in the example above). Ask "was this because we all started throwing balls faster?" Most of the time it happens because the team makes a significant change in their process. Message is that it is only through changing the system that we can improve our productivity - working "harder" doesn't help.

10. Sometimes experiments "fail" (we went from 13 to 9 in the example). As long as we learn from it, we are all good. Ask "could we have got to the next level without the failure?" Point is to try things.

11. Where were constraints in the system and what did the team do to address these? You will often see a team member who has trouble catching a ball, or you'll see a long throw from the last person in the sequence to the first person in the sequence so they can collect a "ball point". And typically teams address these as part of the process. Ask "if we had sped up any other part of the system (e.g. Throw more balls) would it if have us produce more ball points?" This is an example of the theory of constraints, the idea that a system can produce no faster the slowest choke point, you need to focus on the constraint and, once addressed something else will become the constraint.

12. Point out the power of stepping back to see what is going on - the retrospective. Sometimes teams will not use the full 2 minutes. Ask "why"? Again the reason will be all over the place (want to get on with doing, etc). Ask is there anything stopping you from running a quick experiment during the retrospective time to see what you can learn? Point is that we have a tendency to work over

thinking about the work.
13. Most teams will report that they had fun during the exercise. This helps you talk about common goal and engagement etc.

Other things (perhaps a little more subtle) that can be talked about include:

Effect of Cadence

What we set up with this exercise is a cadence. When people started the exercise it felt chaotic. How did we become less chaotic? Well we knew we had to do something in 2 mins. And then we knew it was 60 seconds like, then another 2 minutes and so on. In other words the cadence functioned as an organizing principle for the group. This is useful to remember no matter what flavor agile you use. For example, Kanban can benefit from regularly scheduled planning, demos and retrospectives, even though this is not a formal part of Kanban.

Managing Expectations

In most organizations teams will say "management have unrealistic expectations on the plan". Management will say "but we asked the team for their estimates …" Agile tells us that the traditional estimating approach has a number of problems but the people at the team level are also responsible for some of this.

If you look at the chart of results from the expectation of a manger watching the team what we see sprint by sprint is

Sprint	Estimate	Actual	Met Commitments
1	5	1	No
2	7	10	Yes 🙂
3	15	13	No
4	26	9	No
5	26	23	No

In other words, in 5 sprints, we only made and met commitments once. How is a manager to view this. What is really sad is that we had a 23X improvement in productivity, but from a planning perspective we have set bad expectations.

Where did these expectations come from? The team. Disregarding the first estimate at the moment, look at the behavior for the remaining estimates. After the first sprint we know we can produce 1 ball point. But what did we tell management (me) we could do? 7! After 10, we said 15. Why wouldn't we just say the estimate is "whatever we produced last time"? If we do this and we get a better than expected result, this is usually seen as great news. This type of estimating is called "yesterday's weather" - in the absence of any other facts, best guess for today's whether is whatever it was yesterday:

Sprint	Estimate	Actual	Met Commitments
1	5	1	No
2	1	10	Yes ☐
3	10	13	Yes ☐
4	13	9	No
5	13	23	Yes ☐

This helps set expectations based on reality, not fantasy.

Applying Agile Principles

Even though we understand agile principles of feedback, experiments, incremental approaches, etc conceptually we often have trouble recognizing when we should apply them. For example, in the ball game an "experiment" to find out how many balls we could do before providing an estimate might have been useful. And most teams don't understand how rapidly incremental improvements build to really impressive results (23X improvement in productivity) as it is often hard to see unless you step back a little.

Scientific Method / Deming Plan-Do-Check-Act (Adjust) (PDCA)

The game shows the application of the scientific method or Demings PDCA cycles - you put a plan or experiment together with an expected set of results, you then do something, you check the results and make adjustments (or determine the next experiment to). All agile approaches incorporate these cycles.

Face-to-Face Communication

A little difficult given that the only way to play this game is when everyone is in the room, but perhaps a question "how would you have done these improvements if your retrospective was done online, or across multiple time zones". Often teams have remote people, so this is just a reminder that most retrospectives with remote people should include a discussion about how to improve team communication.

T-Shaped People

This is the opposite of the theory of constraints. Ask "what if we did have a professional basketball player on this team, someone who is way better than anyone else. Would it have helped us improve productivity." Since the whole system determines the result, the answer is "no" unless we had the team learn from the expert and thereby improve the capability of the whole team.

Improvement Requires Risk

Most teams settle down on a process and eventually that process does not improve as rapidly as it once did. In other words, for this way of working, there is going to be a basic ability to "produce" a kind of "natural velocity" for the process as it stands. Often you will find teams will decided to do something very different in order to improve (sometimes as a result of hearing that some other team is "better"). Most of the time the experiment does not work as well as anticipating resulting in the same or less ball points than they had before, but most teams decide they want to continue with their new approach.

The lessons here are:

- To improve dramatically you will probably need to take a big risk where you might not get a short term result.
- The reason that people will continue working their new approach rather than reverting back to the old is that they see significant improvement is possible and with their recent experience (learning) they know they can make the new approach better than the old.

Finish

Close the conversation by asking about application to their work - what might they change?

And then I usually finish the discussion with a "Do you know what you have done? You have just done agile - all the rest is details:-)"

Complexity

A slightly meta point, but may be interesting for some, this is a good example of emergent behavior as a result of complex systems. You will find that, if you run this a number of times, that there won't be the same soliton coming out for any group although there will be a lot of similar patterns. The emergent behavior is a result of constraints, boundaries, and focusing.

Want To Know More?

- http://dpwhelan.com/blog/uncategorized/learning-scrum-through-the-ball-point-game/ has a number of variations as well as more discussion about how to run the game and the things you can debrief.

games, simulations, learning, agile, estimation

From:

Permanent link:

https://www.hanssamios.com/dokuwiki/simulating_agile_execution_with_the_ball_point_game

Last update: **2020/06/04 12:01**