Teaching Lean Principles with

The getKanban Board Game

Facilitator's Guide

getKanban version 2.0 by getKanban.com 2010
Revised and adapted by @bsktcase 2015

Greetings facilitator!

Why would I mess with a great game? Good question.

Even more than other methodologies ("frameworks", whatever), Kanban requires deep understanding in order to bring about real change and measurable results. Too many teams struggle, even after they understand the "rules" of Kanban—failing to recognize visible problems on their own boards, or not knowing how to replace old bad habits with better ones.

I see the same struggle, in microcosm, when I train teams using getKanban Version 2.0. Even though the game urges teams to limit work in progress (WIP)—the linchpin of Lean and Kanban—I found that my players inevitably wanted to raise it. Over numerous classes and diverse groups of players, I tried hints, then rants, about how limiting WIP reduces cycle time. But there was a problem: play teams who took my advice and lowered their WIP weren't seeing better outcomes.

My colleague Andrew Clear and I play-tested the game on our own, and we ran into the same problem. Lowering our WIP *didn't* reduce our cycle time. At least, not right away: we realized the improvement was happening, but on a long delay. Just as our Control Chart started to look better, the game ended!

And that's when we figured it out. Queues! We teach it in class—*once formed, queues are impossible to eliminate by normal means*—but we hadn't detected it. getKanban Version 2.0 *starts out* with big queues, and way too much WIP (17) for the number of team member dice (7-8). We adjusted the gameplay to help teams recognize their queues first. There's nothing better as a trainer than to see that look of understanding on a student's face—the "a-ha" moment—when you know they really get it, and more importantly, they'll remember it. And sure enough, once they realized they had queues, we found them limiting WIP on their own, even more aggressively than the original game recommends, and getting great results!

I hope you enjoy our modified version of getKanban Version 2.0, and that you and your teams really *feel* a difference in your learning. Go attack the queues in your software pipeline after you play!

This guide was produced in collaboration with the great people at Northwest Cadence. Check them out at www.nwcadence.com.

As they say on Day 9: "Good luck!"

Cheryl M. Hammond
Seattle, Washington, USA
@bsktcase

Introduction

My colleagues were involved in development and play testing of the getKanban game starting with Version 1.0, and I have used the free Version 2.0 for Lean/Kanban and agile instruction for several years. This Facilitator's Guide represents the modifications we've made during that time to maximize players' learning, especially of the negative impact of queues and the positive results of limiting work in process (WIP).

Most of this guide is intended to be read once, to give you an overview of the basic rules and our changes. The Day 9 suggested script, which you might use more often, is formatted as a pull-out for quick reference, so you can print it separately if you wish.

Prerequisites

We use the getKanban game to reinforce through experiential learning *after* teaching Lean principles (based on Donald Reinertsen's *Principles of Product Development Flow*) for at least 1.5 hours, and usually also the basics of Kanban for at least 1 hour.

Preparing the Game

Kit(s)

Version 2.0 of getKanban is free. PDF game kits can be downloaded at bit.ly/v2-print. You can play with disposable **paper** kits, or **laminate** your pieces for re-use. We travel a lot, so to make the game kits more easily portable, we re-designed the board to fit 11x17"/A3 printer paper. This also requires printing the Story Cards at 90% to fit. You can download a complete game kit, including these mods, at github.com/bsktcase/getKanbanVersion2.

If you play with re-usable kits, you will need **fine-point dry-erase markers**. Make sure all game pieces are wiped clean at the end of each play session (especially red ink, which may stain), and check each kit for completeness before starting. The CFD is easier to fill out if you provide pens in at least 4 different **colors** (ideally red, blue, orange, and brown, but I've never been able to find a brown fine-point dry-erase marker).

You can get **six-sided dice** in quantity on Amazon. It's simplest to play with 2 red, 3 blue, and 3 orange dice to match the game instructions, but we've also been successful using bulk dice in sets of random colors—players figure it out quickly enough. Make sure each color in each kit is easy to tell apart from the others.

Hello Facilitator,

Thank you for your interest in The getKanban Board Game! I suggest that before reading this guide, you read through the Game Instructions, then look over the components of the game, particularly the board, the charts, the Financial Summaries sheet, the green Event Cards, and the Story Cards. After reviewing the components, and reading this guide, you might want to run through the game yourself.

The guide and instructions are very prescriptive to avoid ambiguity as you are starting out. If you facilitate the game often, you may wish to modify the startup procedure and the rules to your own taste.

I designed this game as a part of my Kanban training class. It proved to be such an effective way to teach the basics of Kanban, that I decided to make it more widely available. Please don't hesitate to contact me if you are interested in Lean and Kanban training or consulting, or if you have any questions about the game.

Kind regards,

Russell Healy
Principal Consultant
getKanban Limited
Wellington, New Zealand

Email: russell.healy@getkanban.com
Web: http://www.getkanban.com

Game Duration

For a getKanban game session with multiple play teams, allow at least 3 hours. Because of the constant complex decision-making, some teams can spiral into excessive analysis (which *doesn't* correlate with better outcomes). I haven't used the finish-first bonus; instead, I ask teams who finish Day 21 early to *keep playing* until they have completed all Story Cards or until everyone else is through Day 21. This allows them to learn more about the long-term impact of their earlier choices.

> "The game is designed to be played as a team. It involves constant decision making, and can therefore suffer from lengthy analysis. To counter this, it is best when multiple teams play against each other, with a bonus on offer for the first team to finish, as described in the Game Instructions. This also creates a positive tension and a more memorable experience."

Team Size

Ideal team size for first-time players is 4±1. This gives everyone something to do and tends to keep all players engaged.

Scoreboard

A little friendly competition (keep it friendly!) makes a good motivator. In getKanban, teams compete to maximize profit from subscribers. This helps reinforce the Lean principle of economic decision-making, and the coach (that's you, together with the **Financial Analyst** role, described below) can remind teams of this principle to guide their in-game choices.

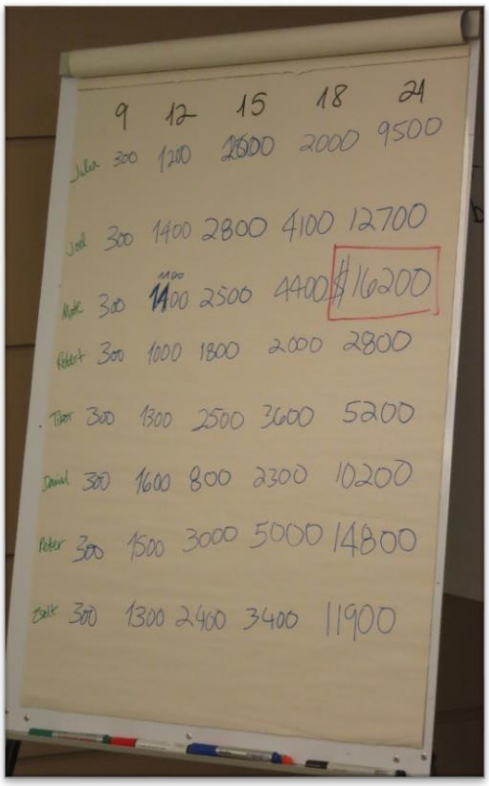
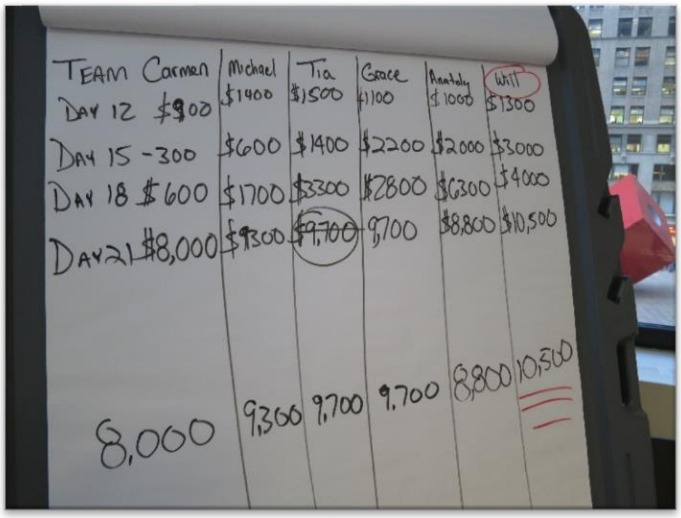


Figure 1: A few sample scoreboards

Draw a grid on a flip chart or whiteboard to track scores:



> "Throughout the game, the Financial Analysts will be telling you their team's gross profit. Post these values on the grid."

Getting Started

For large groups, make a short speech, like this. You'll need it:

> *"Everyone, the getKanban game is ridiculously complex, and it will take a little time for me to explain all the rules. We are also going to play one round all together before I release you to play on your own. You are going to want to start fiddling with the pieces and talking to each other, but I really, really need you to stay with me and listen to all the instructions carefully before you do that. Please don't make me yell."*

Offer bribes (candy, biscuits) if you think it's necessary. Otherwise you'll be hoarse tomorrow.

Now that you have their attention, use the slides to help teams get oriented to the rules of the game.

Organize and Train Teams

There are three specific roles, corresponding to the three reports kept during the game:

- **Control Chart Tracker**. This player updates the Control Chart every time a story is deployed. The Control Chart Tracker has the best visibility into the team's average cycle time, and should be proactive to ensure that the team is considering cycle time in its decisions.
- **CFD Tracker**. This player updates the Cumulative Flow Diagram at the end of each game day. The CFD is notoriously confusing to fill in, so a volunteer with lots of patience and attention to detail is the best choice for this role.

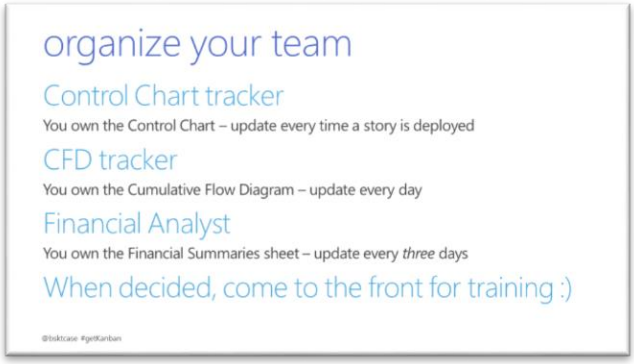


Figure 2: "Train" the players on their roles

- **Financial Analyst**. This player updates the Financial Report every three game days. The Financial Analyst has the best visibility into the monetary impact of the team's choices, and should continuously keep the team focused on making decisions that maximize profit.

When playing with multiple teams, it's easiest to bring each role to the front one at a time for instruction. Give them a quick overview of how to fill out their chart (detailed instructions are on the back of the Control Chart and CFD; the Financial Report has explicit instructions on the front). Especially impress upon the Control Chart tracker and Financial Analyst that they should be proactive about sharing their knowledge of cycle time and revenue throughout the game—they have information their team might not know to ask them for.

Set Up Game Board

Project this slide and ask teams to set up their board to match:

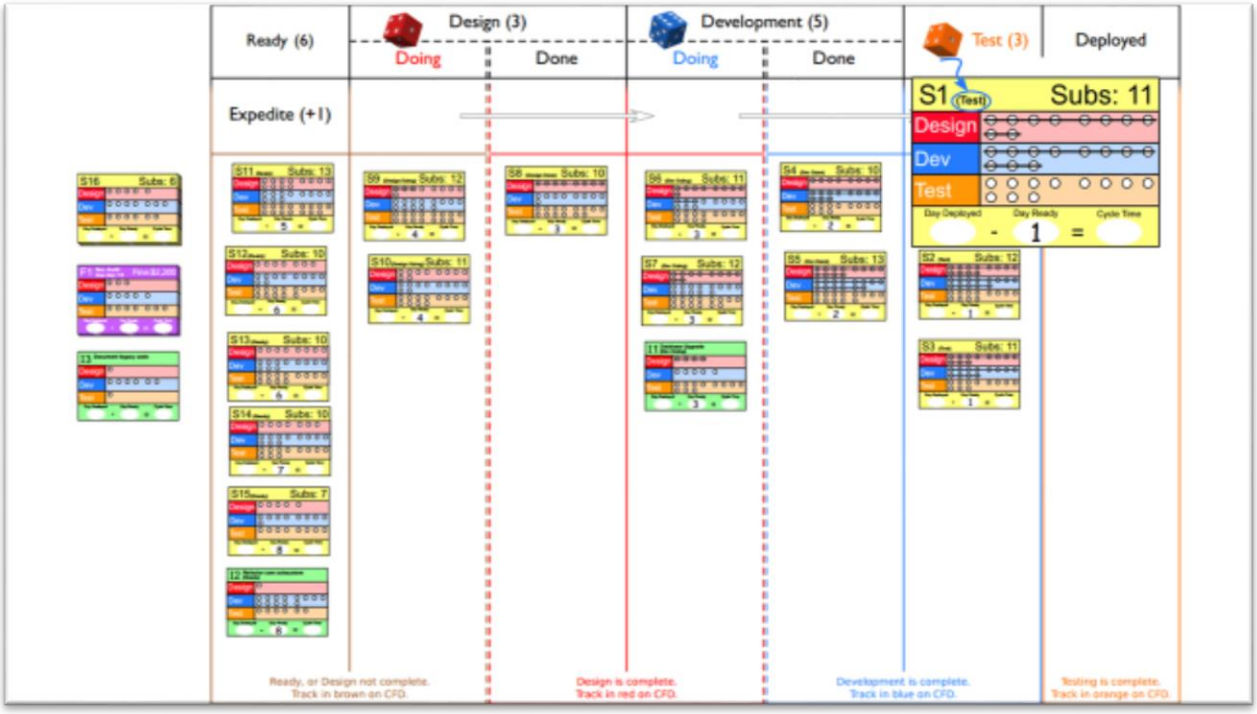


Figure 3: Initial setup for the getKanban board

Point out that the starting column is also marked on each Story Card. Before play, check each team's board to make sure it matches the layout shown.

> "The reason we prime the board in this way is to save time, and get straight into the learning opportunities. If we started with an empty board on day 1, it would take several days of playing before stories were completed, and cycle time could be calculated."

Introduce Game Concepts

Use the slides to provide an overview of key game concepts.

Story Cards and decision-making

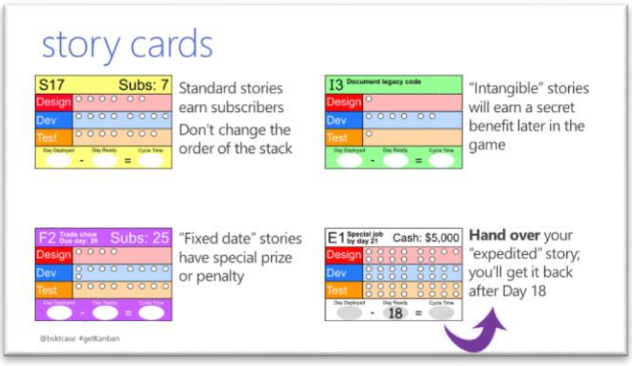


Figure 4: Talk through the different types of Story Cards

The four different types of Story Cards, with four different monetization models, are intended to make comparison and prioritization difficult, just like in real life.

The Intangible (green) stories are particularly apt for players who struggle with legacy code and technical debt—they provide an unknown amount of benefit at an unknown future time, which makes them tough to prioritize.

At this time, ask each team to locate their white "E1" Story Card and **give it to you**. Put it in

your pocket or set it aside. Tell them the game will instruct them when they can ask for it back. This prevents the very common error of teams playing E1 too early.

Subscribers and cycle time

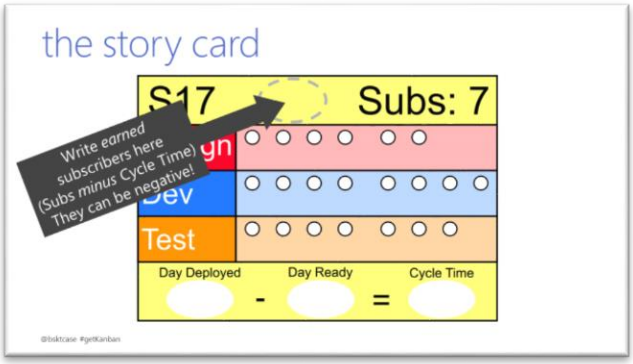


Figure 5: Explain the modified "earned subscribers" rule

"Earned subscribers" is the most significant of our rule changes. Instead of earning the number of subscribers printed on the yellow Story Cards, teams earn that number *minus* the cycle time of that story—including, if cycle time exceeds the printed subscriber number, *negative* (lost) subscribers. This intentionally makes the penalty for high average cycle time very, very severe.

We use the scenario that each team is a mobile app development company, and the "Subs" number on the card represents the number of users who have *requested* a particular feature (maximum possible subscribers). If we deliver quickly enough, they'll all buy our app (earned subscribers), but if we take too long, some of them will get tired of waiting (subtract cycle time) and will buy our competitor's app instead. And if we *really* take too long to deliver a feature, we can lose *existing* subscribers as well (maximum minus cycle time can be negative).

As you explain this rule using the Story Card slide, be aware that there's a secret hint on the slide: the maximum possible number of subscribers on the card shown is lower than their starting cycle time will be. We don't tell them this directly, but we do remind the Control Chart Tracker to compare their team's cycle time with the "Subs" value on upcoming stories as they play.

Dice and dots

Dice represent people. Dots on the cards represent the effort required to complete each type of work for each story.

Figure 6: Allocate dice and explain how they are scored

Instruct the team to sort the dice as follows:

- Locate the color that you have two of (or red, if you bought official-colored dice). These are your Designers. Place them above the Design column for now.
- You now have two different colors of three dice each. Pick either one of these colors (or blue). These are your Developers. Place them above the Development column.
- You have three remaining dice (orange). These are your Testers. **Give one of your Testers to the instructor**: you haven't hired them yet. (Put it in your pocket or set it aside.) The game will tell you when you get to hire your Tester. Place the remaining two Tester dice above the Test column.

As the game progresses, teams may object to the high variability in dice rolls. This variability is intentional, not only because it represents people's variable levels of productivity each day (due to meetings, distractions, hangovers, etc.), but also because in-game, players are able to see the total actual number of dots required for each story ahead of time—as if they had access to flawless estimation all the time, which isn't like real life at all! The extra variability in the dice rolls creates necessary uncertainty about when any given story can be finished.

A Few Semi-Arbitrary Rules

Some teams will want to ease the pain of their queues by "gaming" the game (rather than addressing the queues), and we use these rules to prevent that. We cheerfully disclose that these are "because I said so":

- Teams can *only* reprioritize stories at the beginning of the game day ("Daily Standup" on the Daily Steps chart). Teams *must* work on (mark dots off of) stories in priority order (top to bottom).
- Teams can *only* assign dice to columns at the beginning of the game day ("Daily Standup" on the Daily Steps chart).
- Teams can freely choose whether to play a yellow, a purple, or a green story, and they can play purple stories in either order. However, they *must* play yellow Story Cards onto the board *in numbered order* and they *may not* look through the backlog to see what's coming up. Once stories have been played onto the board, they can be reprioritized freely.

Ready to Play!

Check in with each team and make sure they are set up more or less like this:

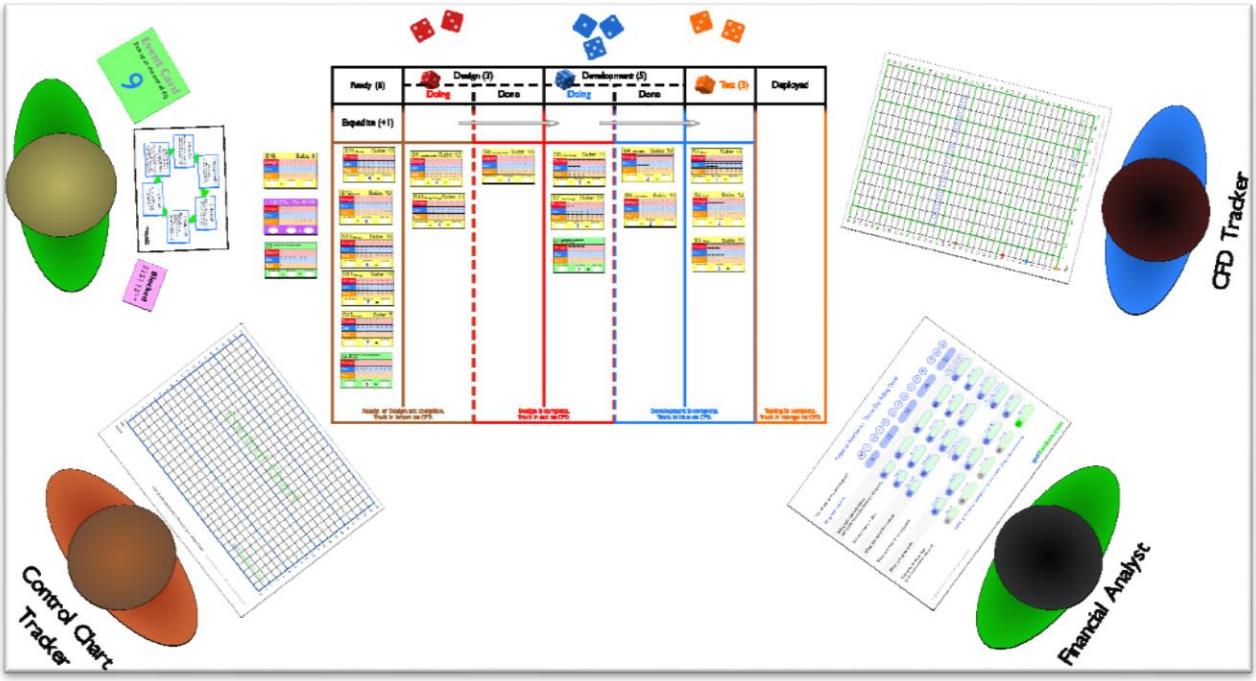


Figure 7: All set up and ready to play!

You're ready to go!

Begin Play

You can pull out the following 4 pages of this Guide to use on their own.

Day 9

You will now guide the teams through Day 9. All decisions for the day have been made, and dice rolls will be controlled, so teams can focus on learning the game process together. Ask teams to follow along on their Daily Steps chart. You can use this script, or adapt it as you wish:

"Teams, we are going to play Day 9 all together, so we can all learn the process before I set you free to play on your own. Then you're going to play Days 10, 11, and 12 independently, and at the end of Day 12 everyone is going to stop for some discussion. We'll talk about how the game is going, and then you'll go back to playing for the rest of the session. Got it? Let's get started!"

1. Standup Meeting

> "It looks like every specialization has plenty of work to do. There are no apparent bottlenecks or problems with flow. I am happy with the order the stories are in. We might be able to deploy our first story today."

Figure 8: The Version 2.0 script is a bit misleading—there are problems on the starting board—but go with it for now

"Just for today, our starting board is already prioritized, and we're going to play each worker in their column of specialization. Tomorrow, during your Standup, you'll be able to reprioritize your board and allocate your dice as you choose."

2. Play Board

"On Day 9, you're not really going to roll dice. I'm going to tell you what your dice rolls will be, and you'll all roll exactly the same. Tomorrow you'll roll your own dice. Today I roll for you. Ready?"

Tester #1 rolls a 4

"We're going to start by playing *one* of our Tester dice, and today this Tester is going to **roll a 4**. Please take one Tester die and make it a 4."

Yes, some of them may try to actually roll the die. Go fix them and make their Tester be a 4.

For best results, give light guidance and have them calculate the dots and figure out for themselves where to mark the effort, e.g.:

"So, your Tester rolled a 4, and it's playing in Test today. So how much effort is that? <*wait*> Right, 8. Find your top priority story in Test—that's S1—and mark 8 dots to represent completed effort."

*Confirm that they have all marked **8** test dots on story **S1**.*

Developer #1 rolls a 3

"Next, we'll play one of our Developer dice. Today for all of you, this developer **rolls a 3**."

*Guide and confirm that they mark **6** development dots from story **S6** in Dev Doing.*

Designer #1 rolls a 5

"You're seeing how this works, right? OK, let's take one of the Designer dice. All of you, your Designer **rolls a 5**. Productive day! Maybe they got out of some meetings!"

*They have **10** dots to spend; they will find that the top story in Design Doing, **S9**, only has 6 dots remaining.*

"All the design work on story S9 is finished! That means you can **move it to Design Done**. You must keep it in priority order, so take it from the top of Design Doing to the bottom of Design Done. You can reprioritize it tomorrow if you would like, but today it has to stay at the bottom."

*Confirm that they marked 6 design dots and moved **S9** to the bottom of Design Done.*

*Guide them to mark the remaining **4** design dots from **S10**, the next story in Design Doing.*

Tester #2 rolls a 3

"Now let's go back to our remaining Tester. Our Tester **rolls a 3**."

*They have **6** dots to spend; they will find that story **S1** has only 3 dots remaining.*

"Story S1 is now DONE! Woohoo! Mark those dots and **move story S1 into Deployed**."

*Don't forget to have them also mark the remaining **3** dots on story **S2**.*

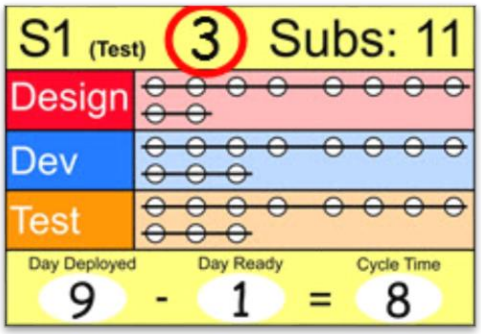


Figure 9: Finished story S1 looks like this (with earned subscribers written in)

"So, now, a whole bunch of things need to happen. Let's start by calculating the **cycle time** and **earned subscribers** on that story we just deployed."

*You might want to project the Story Card slide to help out with this step. Double-check everyone's results: cycle time is **8** and earned subscribers is **3** (see figure).*

"**Control Chart Trackers**, remember I said you should update your chart every time a story is deployed? That's now. Please update your chart."

*Double-check and help with the Control Charts. They should mark one line for Day 9 with a cycle time of **8**.*

"**CFD Trackers** and **Financial Analysts**, we're not to the end of the day, so you *don't* need to update yet."

Guide them to pull stories across the board to fill capacity, e.g.:

"What do you see in the Test column now that we've deployed story S1? <*wait*> That's right: your WIP limit in Test is 3, and right now you only have 2 stories there. So you have room to **pull a story** into Test! You still have to honor priority, so you must take the *top* story from Dev Done—that's S4—and move it to the *bottom* of Test."

"And that makes room to **pull a story** from Design Done into Dev Doing. Do you see how it's kind of a chain reaction? Deploying a story creates **capacity**, and you pull everything forward as your team has capacity. That's the essence of Kanban. Those WIP limits create a system where you pull work instead of pushing it."

They might notice that because you can pull stories immediately when you have capacity, it makes sense to roll the dice from right to left—from Tester to Dev to Design.

"So now you can pull from Ready to Design Doing, and finally, you pull from the backlog into the bottom of the Ready column. Tomorrow, you'll be able to choose whether to pull a yellow or purple or green story, but today, you all **pull story S16** from the top of the backlog. Remember, you *must* take the yellow stories off the top in numerical order—you can't dig through the backlog to play different ones."

If someone happens to ask whether they are required to pull a new story into Ready, just tell them no, they don't have *to. Act casual. There's a big reveal about the Ready column, but save it for later if you can. It will be more impactful if you let them do it "wrong" first.*

"One more thing we need to do. See how all the starting stories already had their 'Day Ready' filled in? From now on we need to fill in the 'Day Ready' spot ourselves whenever we move a story onto the board. Today is Day 9, and we just played story S16, so please **fill in a 9** on the S16 Story Card."

Developer #2 rolls a 2

"Our second Developer **rolls a 2**."

*Guide and confirm that they mark **2** development dots from story **S6** in Dev Doing, **move it** to the bottom of Dev Done, and spend the remaining **2** dots on story **S7**.*

Designer #2 rolls a 1

"We have one Designer remaining. Our Designer **rolls a 1**. Maybe they stayed out a little too late last night?"

*Guide and confirm that they mark **2** design dots on story **S10**.*

Developer #3 rolls a 4

"We're not done yet! We still have one Developer left. Our Developer **rolls a 4**."

*Guide and confirm that they mark **8** development dots on story **S7**.*

"That's it! We're finished applying our team's work for the day."

3. Sanity Check

"Before we fill out our remaining charts, we take a quick look at the board to make sure we've pulled stories to fill our capacity, and filled out cycle time, earned subscribers, and day ready values on all the Story Cards. At the end of Day 9, everyone's board should look like this:"

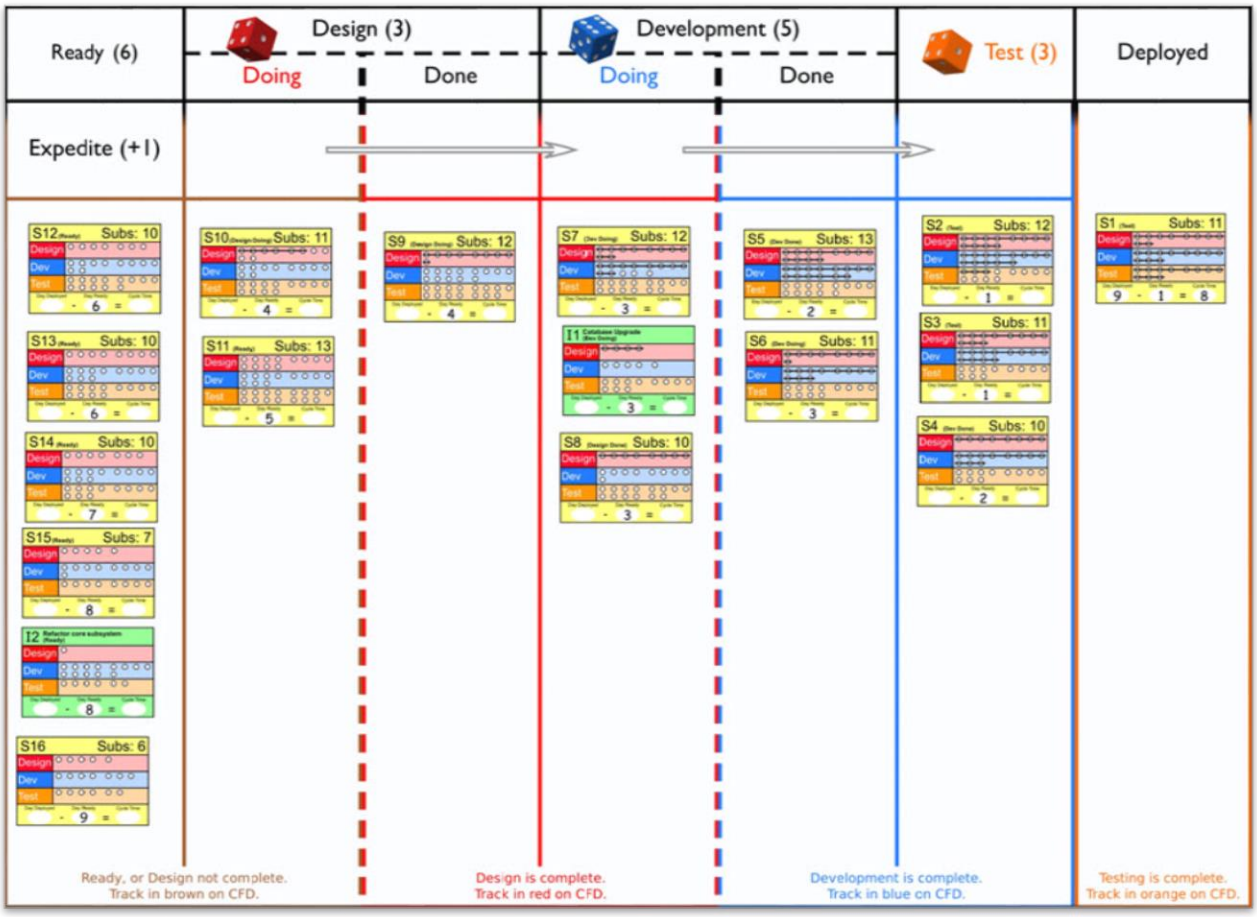


Figure 10: The board at the end of Day 9 should look like this

Check each team's board to make sure their Story Cards are in the right columns in the right order, and they've marked the starting day on story S16.

4. Track Charts

"**CFD Trackers**, I told you that you would update your charts at the end of each game day. That's now. Let's update the chart together."

Step through filling out the chart. Instructions are on the back of the CFD sheet. When done, it should look like the illustration (which is also on the back of the CFD sheet).

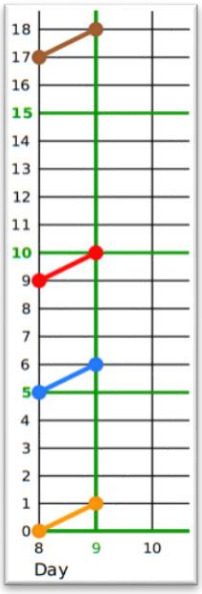


Figure 11: CFD

5. Day Complete / Financial Summary

"**Financial Analysts**, you fill out your chart at the end of every third day, and it happens to be Day 9. Let's update your finances."

*It really is self-explanatory. Just step them through the directions on the front. Their total should be **$300**. Mark $300 for all teams on the scorekeeper's flip chart.*

6. Event Card

"At the end of each game day, you will turn over the Event Card—don't do it yet, wait a moment—you'll read the back, and you'll do what it says. Some days are nothing special, but other days there will be important instructions. If you want to get your third Tester die, and you want to get your Expedited story, the Event Cards will tell you when you can ask for those. The Event Cards will also tell you when and what your Intangible stories reward will be."

"And remember, we're all going to pause after Day 12, let everyone catch up, and then we'll get together for a group discussion. So when you get to the end of Day 12, go ahead and fill out your Financial Summary, and then stop and wait for the rest of us."

"Finally, you can alter your WIP limits at any time. If you've been paying attention to Lean principles, you'll know that this means you should look for opportunities to *lower* WIP, not raise it, but it is your decision."

"Now let's all turn over the Day 9 Event Card and read the back and do what it says."

Good luck!

Teams Play Independently

At the conclusion of Day 9, you'll release teams to play getKanban on their own. Take a break, get a coffee; you deserve it! But you should also wander among the teams, listening, coaching, and learning from them as they play.

To Carlos, or Not to Carlos?

When we first started teaching with getKanban, we found that the Day 13 "Carlos" rule—*only* Testers may test, and Testers may *only* test—distracted teams away from our desired learning objectives. Instead of figuring out how to address their testing bottleneck by limiting WIP, teams tended to blame "Carlos" for their problems, to the point where they actually stopped trying to solve them!

Plus, one time we actually had a player named Carlos. Awkward.

Some of us have stopped using Carlos altogether, but others of us have kept him in. The focused Day 12 coaching around queues and WIP is critical to keep teams improving instead of following Carlos down the rat-hole.

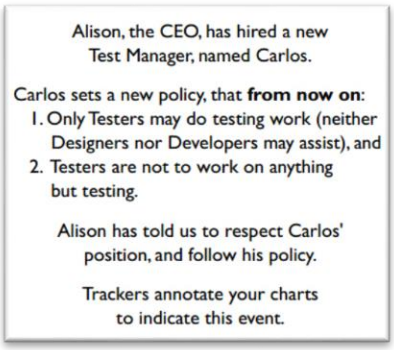


Figure 12: Enter Carlos (or not)

Why Carlos?

> "The game uses resource allocation exclusively to address bottlenecks. This is to keep the game simple yet maintain tension in the process."

Teams will almost universally immediately struggle with a bottleneck of stories in Test, and many teams will try to solve it through resource allocation instead of limiting WIP: by piling all their dice into Test, day after day. It's an unrealistic and unsustainable scenario, and players know it.

The "Carlos" rule in Version 2.0 catches teams over-relying on resource allocation, and stops them doing it. However, as noted above, they most often *do not* make the leap to limiting WIP on their own. You must **pair Carlos with fairly aggressive coaching on queues and WIP** to get them to see the correct solution. Subtle (and not-at-all-subtle) "tips" and "hints" while they play have never worked for us.

Carlos provides an *opportunity* to illustrate the importance of team self-organization. Once a queue has formed, extraordinary measures are required to remove it. Carlos' policy prevents this, which is especially frustrating for teams once they understand what they want to do and aren't allowed do it.

Why not Carlos?

After post-Day 12 coaching on queues and WIP, you want your teams to aggressively attack their queues to promote flow. However, Test is a particularly bad queue, and Carlos prevents the team from taking the extraordinary measures needed to bring it down. They can't swarm on Test, so only 2 dice are available to work the Test queue. Worse, the Carlos rule forces all 5 other dice to work in Design and Development, so they're compelled to pile *even more* work into Test—unless they go on strike!

Conclusion

Carlos, or no Carlos: it's up to you. Just make sure your choice supports your learning goals!

Coach through Gameplay

Listen closely and note which game-day each team is playing. Make sure they haven't missed an Event Card instruction, and point out any learning opportunities as they arise.

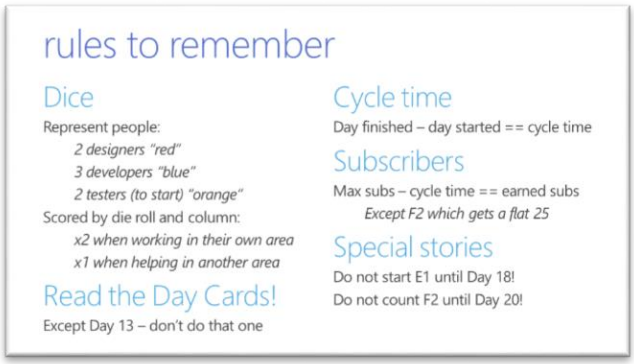


Figure 13: Project this slide to help teams remember rules

Day 10: Think about cycle time

Teams will usually deploy a story on Day 10. Make sure they pull work correctly all the way back through the Ready column.

If they happen to ask whether they are required to pull work into Ready, quietly tell them they don't have *to, but save the big discussion and reveal for later.*

As teams are debating which story to pull onto the board, suggest to them that they consider cycle time, e.g., for story F1. Ask them to tell you what day F1 is due (Day 15), and what day it is today (Day 10). Then coach them to ask their **Control Chart Tracker** what their current cycle time is (it'll be 8-9). Make sure they do the math and make sure they start to look dismayed by what the numbers are telling them.

Day 11: Blocked sticky

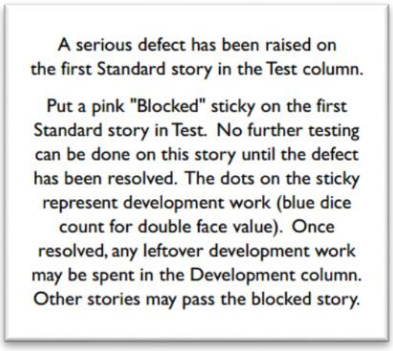


Figure 14: Complicated blockage

The end of Day 10 is the first time teams get a substantive instruction via their Event Card. Make sure they haven't missed it (some teams do!) and then when they are inevitably confused by it, walk them through how it works:

- Only the first story in Test is blocked. Dice assigned to Test may work on the *second* story in Test while waiting for the block to clear.
- The blocked sticky's dots are *development* dots. Dice assigned to *Development* do the work on the blocked sticky, and if they roll more than they need, their extra dots go back to the *Dev* column.

Day 12: Finances and break

Update Financial Summary. Check their math, and post earnings on the scorekeeper's flip chart.

Pause for discussion (see next section). If other teams are still playing, let early finishers take a coffee break.

Day 12 Discussion

This discussion time takes the material you've previously taught on Lean principles (see *Prerequisites*) and the emerging lessons from gameplay and brings them together in real time to create some hopefully strong associations and realizations among your students. Taking the time to call out these patterns, and explicitly show students where Lean theory can be applied in the game, has yielded great results when we've done it—vastly more effective than just commenting during gameplay, or following up after the game has ended.

Pain point

Ask teams what's frustrating them right now, where their top pain point is. If they say something high-level like "deploying" or "making money", dig deeper to find out where *they* think the problem is coming from.

By far the most common answer, when we ask this, is "bottleneck in Test". Remember that you also coached them, on Day 10, to be concerned about getting the F1 audit story deployed by Day 15; you might ask them how that's going, or ask about subscribers.

We present the next three topics/slides in this order, but any order that makes sense to you is fine.

Prioritization

The key decisions teams make on every game-day are outlined on the slide:

- Where to allocate their dice
- How to set priority within each column
- Which story to pull onto the board when they have new capacity

Figure 15: Get players to reflect on how they are identifying and managing in-game priorities

Each one of these is a form of prioritization. Ask teams how they're making those decisions and how they feel about the process. Are some agonizing over every choice? (Those will be your slow-play teams.) Are others just taking a guess and getting on with it? Are the more careful decisions associated with better performance? (In our history of running the game, we've found no correlation, or maybe even a negative correlation, between slow play and revenue.)

What metrics matter in this game? **Revenue is the only one**. How aware are they of this? Are they using revenue as their guide, or are they focusing on other things? As they're making prioritization decisions, are they able to predict which choices will result in more revenue? Why or why not?

The game makes prioritization difficult on purpose. For example, the different types of Story Cards are nearly impossible to compare. Consider the Intangible (green) stories. What effect might they have on revenue? Your players don't know, because the game is withholding that information from them! Ask them if they encounter this kind of difficulty in real life—because they surely do.

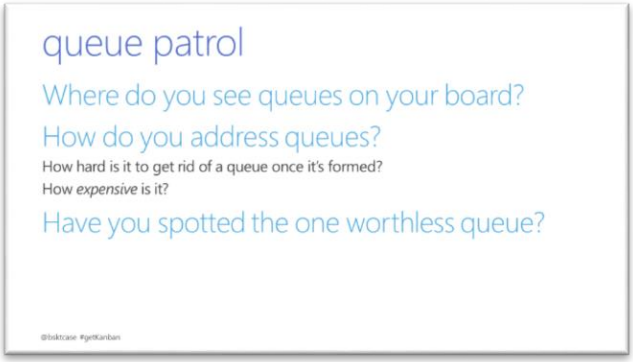


Figure 16: Lead players to identify in-game queues and recognize their impact

Queues

Another way of thinking about queues is touch time vs. wait time. In our Lean principles course, we teach that wait time is *by far* the most costly metric and the primary cause of long cycle times. You're working hard, everyone's "busy", but it still seems to take a long time to deliver anything. Queues are to blame.

If your students take away only one learning from the entire getKanban play session, we (well, Cheryl, anyway) **would argue that the ability to recognize queues and the desire to eliminate queues is the single most valuable thing Lean beginners can do**. Eliminating queues can *lead* a team to voluntarily adopt other good practices, such as limiting WIP and swarming, but in our experience the reverse creates more resistance and is less likely to be pursued wholeheartedly.

Motivation around queues is pretty much the reason we changed these rules and wrote this guide.

Ask teams to examine their board. Which columns represent *working* states, where actual effort happens, and which ones are *waiting* states, i.e., queues? Teams will usually do well at identifying the two Done columns as queues. The "one worthless queue" on the slide refers to a queue they sometimes *don't* notice on their own, the **Ready column**. Per the game rules, cycle time is calculated—the "clock starts ticking"—on a story as soon as it enters Ready state. Did they consider this? Some teams might have spotted it already during gameplay. If so, ask them to share their observations with the group.

Ask each team to pick up the first Story Card in their Ready column, whatever it may be, and calculate the cycle time on it *as if they were deploying it today*. In other words, from the current game day (Day 13, not 12), subtract whatever day it entered the Ready column, and report the result. It varies, but for us the vast majority of teams will report a cycle time of 5-6 days (!). Now ask each Control Chart Tracker to report the team's current average total cycle time (a rough estimate is fine). This will almost always be 9-10 days. Make sure the impact of this sinks in: *half*, or possibly *more*, of their total cycle time is happening in the Ready column, *for no reason!* What might happen to their cycle time if they could reduce or eliminate the time spent in Ready?

Finally, what about any hidden queues? Is there work sitting in a working column but not yet being worked on? This is especially interesting if anyone has raised their WIP limit in Test, as teams sometimes do. How does it help to raise the limit, if your dice can only work on the top story at a time? Does raising the WIP limit in Test cause more stories to be *worked on*, or only more stories to be *waiting?*

Cycle time

Is the Ready queue anything like real life? Ask them to consider when the "clock starts ticking" on the work they do. When does it start from their own perspective? When does it start from their *stakeholders'* perspective? (This is sometimes called cycle time vs. lead time.) Which one do their stakeholders care about? Which one does their leadership care about? Which one(s) do they have control over? Some teams and roles don't control lead time, but they have a lot more control over their own cycle time than they even realize. Both lead time and cycle time are impacted by queues.

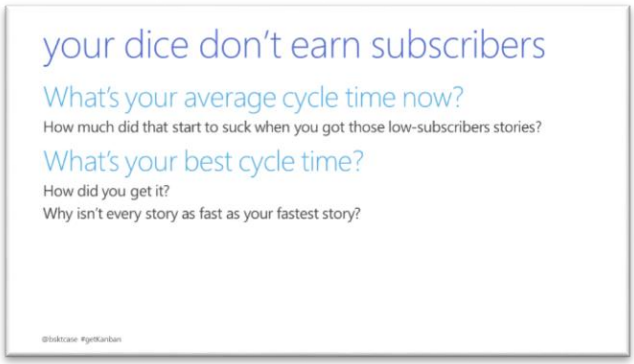


Figure 17: Coach players to stop managing utilization and start focusing on flow

Ask teams to reiterate: what single thing in the game appears to have the greatest impact on their cycle time? Since, above, you just calculated that the wait time in the Ready column likely accounted for half or more of the total cycle time, that must be it. What did they think it was before? Probably the dice. Where have they been spending their decision-making energy so far? Many teams will have recognized the Test column as a bottleneck and will have swarmed their dice onto it to try to knock it down. How difficult was that, compared to how easy it is to change their process in the Ready column?

Resume Gameplay

Day 13: Carlos!

If you are playing with the Carlos rule (see discussion above), now's the time!

If you are not playing with the Carlos rule, do still prompt teams to think about and talk briefly about what they think the impact of the rule would have been in their game.

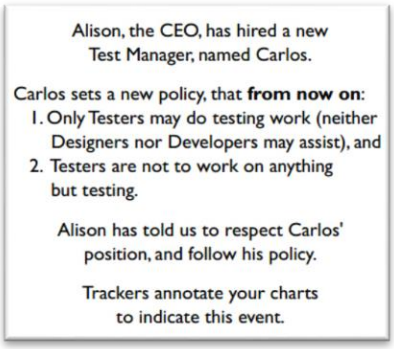


Figure 18: Enter Carlos (or not)

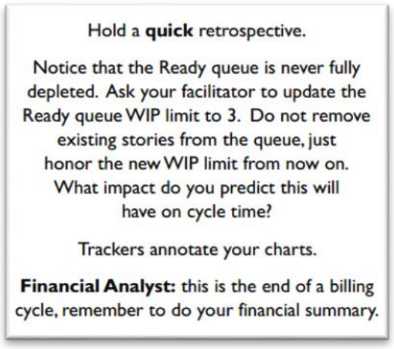


Figure 19: The Ready "queue" instruction may be moot by now

Day 15: Finances and the Ready column

Update Financial Summary and report earnings.

They have hopefully already reduced or eliminated their Ready column by this point; if so, the Day 15 instruction to reduce the WIP limit in their Ready column will be moot. (Point out that the Event Card even *calls* it a "queue"!)

If they've reduced Ready instead of completely eliminating it, now might be a good time for you to ask some leading questions about that. What good does it do to have the Ready column *at all?*

Day 16: Deming (no action)

This is the W. Edwards Deming card. They will be confused by it. There's no action—it's just there to taunt them and make them mad, especially if you are playing the Carlos rule.

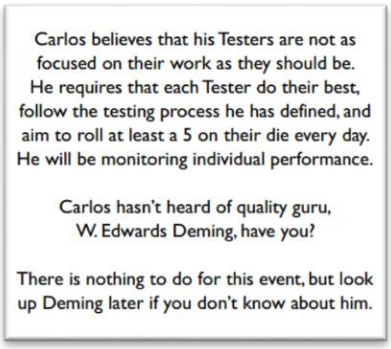


Figure 20: This card has no action; it just adds insult to the Carlos injury

Day 17: Collaboration wins!

Alison the CEO comes to her senses, and Carlos is vanquished! The team can now self-organize!

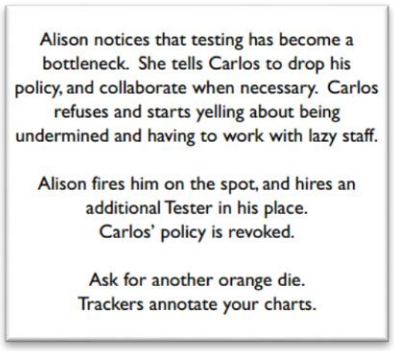


Figure 21: No more Carlos!

Whether you've used the Carlos rule or not, teams should eagerly ask you for their new Tester die, the one you set aside or put in your pocket during game setup.

If you've been playing with Carlos, now's a good time to ask teams about their progress against their queues. Were they able to take extraordinary measures to help clear the board? Did lowering WIP in other columns "help"? (Consider whether it may have helped by limiting the damage—if they hadn't lowered WIP, would the bottleneck in Test be *even worse*?) Carlos also provides a very good illustration of the **folly of utilization**. Did keeping Designers and Developers "busy" in Design and Development help the team deliver?

Day 18: Finances and Ernie

Update Financial Summary, check math, and post earnings on the scorekeeper's flip chart.

Teams should ask you for story E1, also known as the "Ernie story". That's the white one you set aside or put in your pocket during game setup.

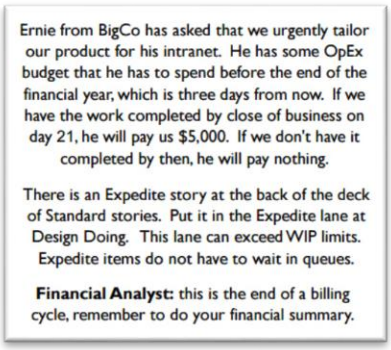


Figure 22: Deep-pockets Ernie

If you are playing with the 11x17"/A3 resized game board, then the Expedite lane is much less visible than the Version 2.0 board (it was reduced to make the board fit on the paper). Either way, the Expedite behavior usually requires some explanation: Expedited stories violate WIP, are always at the top of any column they're in, and skip over any Done columns. Teams usually realize that this means they should play their dice from left-to-right (Designer-Dev-Tester) instead of the right-to-left they may have been doing before. This is allowed and is a good observation.

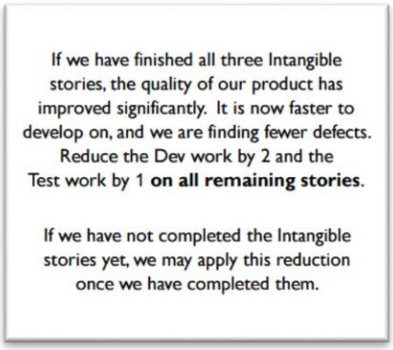


Figure 23: Intangibles

Day 19: Intangibles

For some reason, about half the time our teams miss this Event Card when it comes up. If they have completed all three green Intangible stories (I1, I2, I3), make sure they know to apply the bonus to *all* stories on the board as well as the future backlog. If they have not yet completed all three, make sure they know that they may apply the bonus as soon as they finish them!

Day 21: Finances and endgame

Update Financial Summary, check math, and post final earnings on the scorekeeper's flip chart.

If you have time, encourage early-finishing teams to continue playing until the other teams are done. Just use the standard rules with no special Event Cards, and calculate finances every third day.

An irritating cheat

Some teams inevitably figure out that they can avoid negative-subscribers penalties by simply *failing to deploy* stories with a low "Subs" value and a high cycle time—they prioritize them to the bottom of a column every day, avoid working on them, and wait for the game to end.

The best way to punish them for this is to require all teams to keep playing until all Story Cards are done, but we've never had enough time to enforce that. We just sternly admonish them. You might consider requiring teams to also calculate subscribers for all *in-progress* Story Cards when the game ends. That'd show 'em.

Wrap-up

When all teams have finished, review the scorekeeper's flip chart and teams' revenue results. Was there a clear "winner" (highest gross revenue)?

- Did the winning team lead throughout the session, or did the lead change (it often does)?
- Did the lowest cycle time correlate to the highest revenue (it doesn't always)?

- What decisions did high-revenue teams make compared to decisions by low-revenue teams?
- Did any teams pursue goals *other than* revenue? Why? What was their result?

Though we want to acknowledge the strong performers—and especially reward those who paid attention to the repeated instruction to maximize revenue—we don't usually emphasize "winning" in an overly competitive sense and we don't give a prize. No, not everybody is a winner, but we want all of them to appreciate their experience and learn from it, even/especially if their team didn't make the most money.

For all teams, discuss:

- How did they alter their WIP limits?
- How low did they get their cycle times?
- How did it all *feel?* What did they observe?
- Most importantly, what can they apply when they get back to work?

Finish

We hope you'll enjoy, and get a lot of value out of, including getKanban Version 2.0 in your Lean training and/or learning processes. If you have any questions about this guide or suggestions to make gameplay better, we'd love to hear them.

Good luck!