

BACKGROUND

The Dot Game is a one-hour simulation of software development that is both fun and instructive. It can be played with as few as six people and a facilitator and as many as you like. You should have one facilitator for every 60 people or 9-10 teams.

It is good to get as many people playing the game as possible, but it is very important to have several observers since they notice things that the players don't notice. In many room setups, getting everyone involved can be a challenge. In these situations, getting half the people playing is definitely sufficient.

The game is played in three rounds, each lasting about 20 minutes: Five minutes for prep, five minutes for running the game, and ten minutes for group discussion. The group discussion is where the real learning takes place!

GOALS

Understanding the principles of Lean-flow is important whether you are currently doing or planning to do Scrum, Kanban or any form of Agile. A mantra of Lean is to reduce delay in workflow and feedback. Sometimes, this requires you to select what you work on differently than you might otherwise. Effecting long-term change requires an understanding of why these delays are bad by experiencing it, not just learning it intellectually. The Dot Game is a quick way to get a simulated experience of this.

LEARNING OBJECTIVES

Here is what participants will learn by playing this game.

- Why individual team members getting ahead of others causes problems, and what to do when that happens
- Why breaking work into small batches is important
- Why having too much work in process adds to the waste of the system and makes it harder to detect and fix problems (and why managing work in process is important)
- Why the system people are in often has more to do with how well they get their job done than their own abilities

- Why when individual team members are hyper-focused on their own work and their performance, they often start working against the overall success of the team

SUPPLIES REQUIRED

Here are the supplies required to host the Dot Game.

- 3x3 yellow Post-it™ notes (enough packs for 300 stickies)
- 3/4 inch round assorted color dots: (3 Packs)
- Letter or A4 size printer paper (½ to 1 ream)



Figure 2: Each "done" Post-it note

INTRODUCTION TO THE DOT GAME (INSTRUCTOR NOTES)

If you are planning to lead a group of participants through the Dot Game, there are some important things to understand that will help you make the game an effective learning experience.

PRIMARY OBJECTIVE

The primary objective of the Dot Game is to teach the underlying principles of Lean and Agile. This essentially revolves around finding better ways of working that remove waste of all kinds, such as delays, rework, quality problems, and not producing the right things.

ADDITIONAL OBJECTIVES

Here are additional objectives for the game.

- Learn the principles in a common way, regardless of their domain, so that all people who have played it have a similar experience;
- To experience a set of problems and then learn about and experience the Lean-Agile solution to the problems;
- Layer the experiences and learning through successive rounds so as not to inundate people with too many problems and solutions at the same time.

The result is an understanding that lasts for a very long time, coupled with experiences that provide easy recall of the lessons learned.

ELEMENTS OF THE GAME

Here is the important elements of the game.

- There are three rounds. Each round starts with five minutes of instructions, followed by five minutes of work, followed by then minutes of discussing observations and improvements.
- Rounds 2 and 3 incorporate improvements from the previous rounds so that people experience the improvements first-hand, which is a powerful way to learn.
- People work in teams at their table; and,
- The instructor acts as the ever-important client.

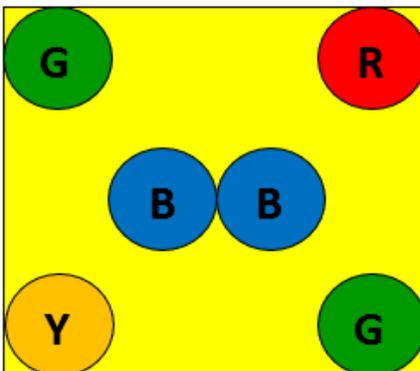


Figure 2: Each "done" Post-it note

BEING THE INSTRUCTOR

As the instructor, it is important to role-play. You are essentially setting people up for failure in each round, but that's not because you're being mean, it's because you're engineering an experience. So make it **fun!** The first round sets the stage, and each successive round will be better because they will have learned something and will incorporate improvements, but they will nonetheless fail at something. That's an important aspect of engineering the experience and the learning.

If it is your first time to lead the game, be prepared that you will make some mistakes as well, and you may even forget something. So, the first time you do this on your own, you may want to let the guinea pig teams know that it's your first time *too* and that *you* will *also* be learning with them.

ROUND ONE

THE WORK PRODUCT

The purpose of the game is to complete "software features", which are defined to be one yellow Post-it note that has six colored stickers as shown in Figure 2.

Important: Do not to show them the example above until **after** Round One. Instead, draw it imperfectly (see Figure 3 for an example). Then, when they do the game, they certainly won't get it right. You can then discuss acceptance criteria during the discussion. However, people typically don't get it right even if you give them an exact example.

Have all the materials on each table, and have each of the "roles", other than the customer, arranged around the table in a U order (see Figure 4).

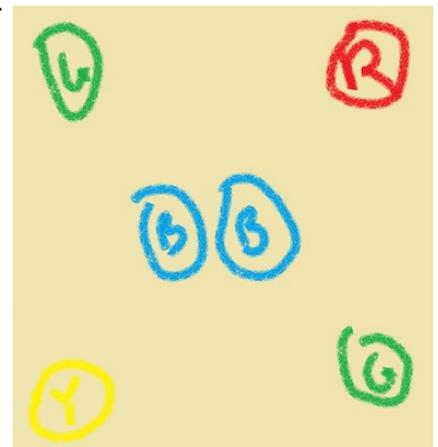


Figure 3: An imperfect hand-drawn representation of the requirement.

OVERVIEW OF ROUND ONE (NOTES FOR THE INSTRUCTOR)

In Round One, you introduce six very specific problems.

- **Localized optimization**, which is a term for optimizing a small part of the whole system without regard for the overall flow of the whole system. Management is traditionally focused on individual utilization and performance and this usually results in every person having to be busy and fully utilized or else the perception is that we're wasting money. So, for Round One, you are telling everyone that they must continuously produce parts and push it to the next station... after all, their performance reviews depend on this.
- **Batch production**, which is a term for when parts are produced in batches instead of single pieces and is often associated with obtaining bulk-purchasing discounts but can also be associated with a desire to minimize the cost of task-switching. There are pros and cons to this, which we'll cover later, but for now, this is an important part of the game's learning experience. So in Round One, each person is going to complete their work on six parts at a time and then push all six at a time to the next station.
- **Push**, which is a manufacturing or production term for when one work station (call it Station 1) makes a part and then pushes the part to the next station (call this Station 2) whether Station 2 is ready for it or not. This means that, unless Station 2 is consuming the parts at a rate equal to or faster than Station 1 is producing them, parts will start piling up at Station 2. This will happen in Round One because of the prior two problems of localized optimization and batch production.
- **Too Much Work-in-Process (WIP)**, which is a term that describes partially finished goods waiting for completion, including goods currently being worked on and goods waiting for further processing in a queue. Resulting from localized optimization, batch production, and push, there will be lots of WIP in Round One and very little finished product.
- **Low Quality**, which, in our terms, means not meeting customer

expectations. In the game the instructor is the acting as the customer, and unbeknownst to the teams, the "customer" will reject all the work items that make it to the end. Why? Because for the first round, you **are not going to show them precisely what you want** but you are going to describe the layout and produce a rough drawing of what you want. If all goes as it almost always does then everyone will assume they understand what you're expecting, and since you are not going to be precise in your drawing, they will be wrong.

- **Low Value Realization**, which is the result of all the above, in that the customer has rejected all the work items that made it to the end because they didn't meet the customer's expectations, and almost all the work in process will be wrong as well. Even though you, as the instructor, will state at the start of Round One that you will be happy to validate the final results, you'll find that it is rare for anyone in to take you up on that offer. Even if they do, that's not a big deal because everything else will prevent them from producing a meaningful amount that meets your expectations, and if you have multiple teams the likelihood of all of them taking you up on this offer is almost zero.

Now this may seem like a perfect storm, but our experience has shown that the vast majority of teams work under most of these conditions, if not all. Even if your teams are only experiencing one of these

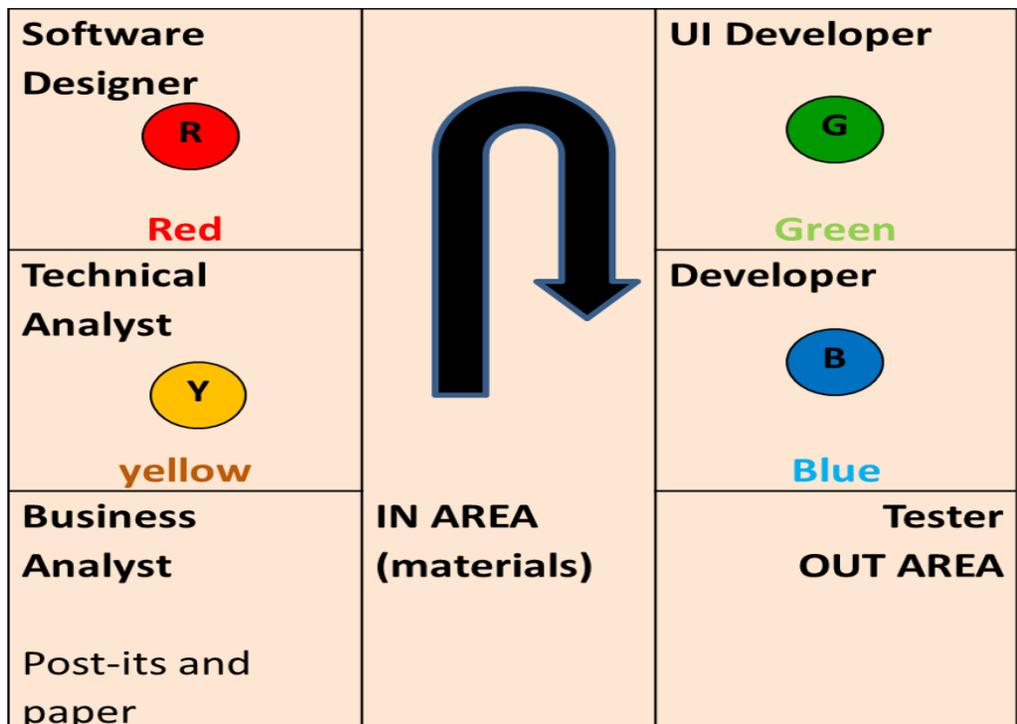


Figure 4: Roles around the table

conditions, the lesson will serve to underscore the importance of avoiding these, but more so, will give people the vocabulary with which to describe why they are undesirable.

And finally, for each round we're going to measure how many items were still in process, how many completed, and how many accepted. These will be our key performance indicators or KPIs for the game.

- # of Post-it notes **completed**
- # of Post-it notes **in process** (WIP)
- # of Post-it notes in **total** (completed + WIP)
- # of Post-it notes **accepted**

In the PowerPoint file, you will find an example of how to capture and display the information, as seen in Figure 5.

SETUP

Here is how to set up the game.

1. Make sure you have a **hand-drawn version** on a flip chart of what they are supposed to build.
2. Present the slide that shows figure 4.
3. Have the supplies on the tables.
4. Make sure there exactly 6 people per table. If you have less than 6 people left over, assign them to observe the teams.
5. After people are seated it is good to have the business analyst raise their hand and say the technical analyst is the person to their left, the software designer is to the technical analyst's left and so on.

INSTRUCTIONS TO GIVE PARTICIPANTS

Here are typical instructions to give verbally to participants.

"Those not playing are observers. From this point on, until we do the report out, you cannot talk or make funny facial expressions or hand movements!"

"This game simulates a software development process of building what's on this flip chart

<<point to your hand-drawn requirement>>. There are only two rules of the simulation. First, each role can touch only their colored dots. Second, for each Post-it note, dots have to be put down in a certain order – yellow, then red, then green, then blue.

"I will play the role of the customer. When you are working I will be happy to validate the final results.

"Studies have shown that hand-offs are expensive and time consuming so the business analyst is supposed to put six Post-it notes down per sheet and pass the sheets to their left. This will save us the hand-off time. Also, to maximize throughput, each person should work as fast as they can on their task. It simply makes sense that if we do each step as fast as it can be done with as high quality as it can be done the overall result will be optimal.

"In Round One, the work will proceed as follows:

1. The BA picks six Post-it notes, puts them on a sheet of white paper, and passes them to the tech analyst
2. The tech analyst puts on the yellow dot on all six Post-it notes, and passes them to the designer
3. The designer puts on the red dot on all six Post-it notes, and passes them to the UI developer
4. The UI developer puts on the green dots on all six Post-it notes, and passes them to the developer
5. The developer puts on the blue dots on all six Post-it notes, and passes them to the tester
6. The tester verifies the end result against Figure 3 (they won't have seen Figure 2 yet!), separating the defective ones from the correct ones

"Each table has an "in" area and an "out" area. Each person can only do work defined by their role, they cannot do the work of another role.

"You have five minutes to produce as many as you can!"

Team	Round	Completed	WIP	Total Count	Accepted
Team A	Round 1 Round 2 Round 3	34	26	60	0
Team B	Round 1 Round 2 Round 3	42	48	90	0
Team B	Round 1 Round 2 Round 3	24	47	71	0
Team D	Round 1 Round 2 Round 3	18	54	72	0
Team E	Round 1 Round 2 Round 3	30	57	87	0

Figure 5: Example layout for capturing the Dot Game's Key Progress Indicators (KPIs)

DEBRIEFING AT THE END OF ROUND ONE

After five minutes, stop everyone at the same time, ensuring nobody is still working. Some people will be so engrossed in their tasks that they'll ignore you, so be sure to stress that everyone has to stop immediately.

Step 1: Ask each tester how many were accepted by the customer.

The answer will be zero. You can go from team to team to "review" them, but the answer will still be zero. As explained in the instructors overview, for the first round you didn't show them precisely what you wanted, and instead described the layout and produced a rough drawing of what you wanted. People will ask why you're rejecting everything. Your answer is that none of them meet your expectations. Don't engage in debate over this, just go to Step 2

Step 2: Present the slide that has a picture of a pipe (the first one, without words), and ask the question, "**Is this a pipe?**"

People will say "yes", which should be countered with the question of whether they could roll up whatever that picture of the pipe is being shown on, light it, and smoke it. This is very literal, but the point is that it's a *representation* of a pipe, not a pipe itself, and as such, requirements and our assumptions based on the requirements are merely representations and interpretations of the customer's expectations; neither may accurately reflect the customer's expectations.

Click to the next slide to reveals the original words in the painting, "This is not a pipe", written in French.

You will get some protests here, but we usually joke about it at this point, saying something like, "Oh yeah, I forgot that you guys always get complete and accurate requirements from your customers. Sorry!" This usually gets

out



several
laughs
of the

Figure 6: René Magritte. "This is not a pipe",

participants as they realize they were protesting reality.

Step 3: Click to the next slide to show the more accurate representation of the requirement.

You may want to point out that they blue dots must be centered both horizontally and vertically, with no gap between them, and that the other four dots must perfectly line up in the corners with no gaps between the dots and the edge of the Post-it note.

Step 4: Click to the next slide to show the example of how to capture and display the KPIs (Figure 5) and explain to the participants that, per team, you are going to capture:

- # of Post-it notes **completed**
- # of Post-it notes **in process** (WIP)
- # of Post-it notes in **total** (completed + WIP)
- # of Post-it notes **accepted**

Step 5: Have each team tally their numbers and read them out to you. If you would like, the example slide is followed by one without the example that you can edit directly in PowerPoint as people read out the results.

GROUP DISCUSSION AFTER ROUND ONE

Let those participating share first. Then ask observers to share. Then interject.

Typical insights will include

- The set-up on people doing their best caused lack of collaboration.
- Very often when the customer gives feedback only the tester listens. And then when he/she tries to tell the team the team barely slows down while maybe listening some. They are so into "getting it done" they don't worry what done means. After all, they have the requirement.
- The mood is dramatic and not fun. Little collaboration.
- "testing" could have been done each step after a piece of work was done – no need to wait for test – discuss the role of tester. Should it be at the end? The beginning? Or throughout the workflow.
- That the delay in testing means that after the other steps are done testers will have a need to continue and that the others will start another project.

Have some discussion before revisiting and doing Round Two. Talk about proper WIP management, pull, and

smooth flow, better testing and how it helps productivity and quality at the same time. Emphasize that in Round Two we are going to do the right thing at the right time even if it means to “slow down to speed up”!

Solutions for Round Two offers a detailed discussion.

ROUND TWO

INTRODUCING SOLUTIONS TO SIX PROBLEMS

In Round Two of the Dot Game, you introduce solutions to the six problems introduced in Round One, as explained in the section, **Overview of Round One to instructors**. Some solutions will be of an interim nature, and for those, Round Two will help people see that there are even better solutions that address the root causes of the problems rather than just the symptoms. This specific concept of finding and addressing the root cause is critical to sustainably solving problems.

Using the “Five Times Why” technique, work backwards from problem six in order to establish the solutions. Number six was “Low Value Realization.” You should lead folks through this exercise so that they see the problems, the causes, and the solutions. It also introduces them to this extremely powerful root-cause analysis technique.

However, instead of going five levels deep with our “whys”, we’re only going two levels deep after Round One because we want to take small steps in the dot game in order to layer the learning.

So, consider these problems.

Why did we have low Value Realization?

Because (1) we didn’t produce enough finished products that met the customer’s expectations, (2) we created a lot of wasted inventory, and (3) everyone was focusing on their own work without helping others finishing products.

Why did we not produce enough finished products that met the customer’s expectations?

Because we made assumptions about the customer’s expectations and we didn’t validate our assumptions with the customer. This also resulted in lots of wrong/defective in-process inventory.

Solution 1: Validate the requirements and your assumptions with the customer before starting production.

Why did we create a lot of waste?

Because we produced inventory at a higher rate than we were completing finished products, and at the start, everyone after the business analyst was waiting for the previous persons to finish their batches of six. This also resulted in lots of inventory that couldn’t be turned into finished products before the five-minute deadline.

Solution 2: Starting with the business analyst, each person works only on one product at a time and sends it on as soon as it’s ready.

Why did everyone focus on their own work without helping others?

Because the focus was on individual performance and utilization, and not on overall throughput of value realization.

Solution 3: Focus instead on finishing products, crossing boundaries where necessary. One way that teams can do this is by reorganizing who’s doing what. Encourage them to experiment because they’ll quickly learn what works and what doesn’t.

INSTRUCTIONS TO GIVE PARTICIPANTS

Here are typical instructions to give participants.

1. Get an agreement from everyone to adopt the three solutions outlined in the previous section.
2. Give them another five minutes to discuss any additional improvements they’d like to experiment with in this round.
3. Repeat the game for another five minutes.
4. Capture the KPIs for Round Two and compare to Round One.

DISCUSSION AFTER ROUND TWO

By the end of Round Two, teams will have much more completed and much less WIP compared to Round One and they will likely have some accepted as well.

Teams will likely report that there was a lot less waiting, while at the same time things felt less rushed.

However, things will still not have been “awesome”:

- They will still have experienced areas where inventory was piling up; less than before, but still resulting in excess WIP.

- They will still have discovered defective products at the end that will either have been rejected or pushed back to be fixed. Pushing defects from the test station back upstream to be fixed delays other items currently in process and, because it takes longer to fix these items will result in fewer accepted items than would've been accepted had these items not been defective in the first place.

ROUND THREE

INTRODUCING SOLUTIONS TO ISSUES IN ROUND TWO

It's time for the "Why's" again.

About excess inventory and WIP

Why did we still produce inventory at a rate higher than we could complete items?

Because even though we were producing only one at a time, there are bottlenecks in the system, specifically the two development activities (two green dots and two blue dots), and the folks before them were producing more parts than they could process.

Why did the upstream folks produce more parts than the development stations could handle?

Because (1) they were still focused on individual productivity, and (2) they weren't working in a system that signaled them when they needed to produce more parts, nor did they have the permission to stop producing parts if the next stations were "full".

Solution 1: Institute a system that will signal each station when they need to produce more people and grant them permission to not produce more parts when it will result in excess inventory. This is called a Kanban system with Pull. To make this clear, you'll need to demonstrate to everyone how a Kanban system with Pull works. See the next section, **Teach the essence of Kanban and Pull**, for instructions. and then have them adopt this solution for Round Three.

About rejects and defects

Why did we still discover defective products at the end (the test station) that were either rejected or pushed back to be fixed?

Because defects weren't detected and fixed mid-stream.

Why were defects not stopped and fixed mid-stream?

Because everyone was focused on producing their "parts" and weren't validating the "parts" they received.

Note to instructors: It's possible that some people did this in Round Two, but it doesn't happen that frequently in Round Two. Nonetheless, even if some people did, tell them that it wasn't an official policy before and now it will become an official policy.

Solution 2: Everyone is responsible for quality and should not let defects flow past them. This means (1) they're validating their own parts as well as the parts they've received, and (2) stopping production to fix it right there and then.

TEACH THE ESSENCE OF KANBAN AND PULL

Position yourself and one other person as shown in the Figure 7 with your chair positions at 90 degrees to each other so you can see the table but not the work area.

There are two parts to this exercise. The first is describing what a Kanban system is.

Part 1: A Kanban system. Put a Post-it note on the table between yourself and the volunteer and write an x on it. Then tell the person to put a Post-it note on top of the Post-it note with the x whenever they see the x. When they put a Post-it note down, you pick it up, take a few seconds to put a dot on it, during which they put another Post-it note down (because they see an x again since you picked it up) and then wait for you. Point out how while you're working, neither of you can see what the other is doing, meaning that you can do this in a virtual environment. Note that this is a simple Kanban system.

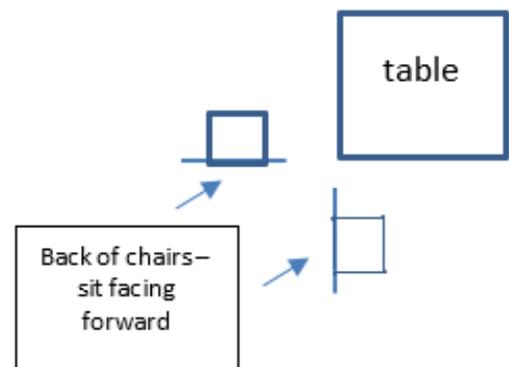


Figure 7. Setup to demonstrate Kanban and Pull

Part 2: Ask how much faster folks think the person is. Put down that many Post-it notes, putting an x on each (typically 3-4). The rule is that they now put Post-it notes down as fast as they can whenever they see an x, but when all are covered, they should walk away as if they're doing something else and come back when you've removed the last covered Post-it note.

This often takes a little bit of patient correction. This illustrates how one can manage a bottleneck. This can also illustrate how you can be wrong and adjust. For example, what if you had put six Post-it notes and it proved to be too many, or two Post-it notes and proved to be too few? You adjust. Kanban is a system that allows for experimentation and improvement.

INSTRUCTIONS TO GIVE PARTICIPANTS

Here are typical instructions to give participants.

1. Get an agreement from everyone to adopt the two solutions outlined in the previous section.
2. Give them another five minutes to discuss any additional improvements they'd like to experiment with in this round. However, the roles and responsibilities can't change yet.
3. Repeat the game for another five minutes.
4. Capture the KPIs for Round Two and compare to Round One.

DISCUSSION AFTER ROUND THREE

By the end of Round Three, teams will have significantly lowered their WIP and their rejected items. If they haven't increased their accepted items, that's perfectly fine because they've been spending their time on adding value, not creating waste. However, we almost always see a significant increase in accepted items in Round Three.

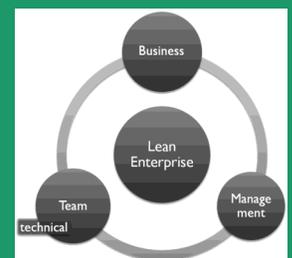
We often spend 10-15 minutes after Round Three in discussion,

Ask people these questions:

- How do they feel about the results?
- Did they feel more or less pressure?
- Did it feel like things were running more smoothly?
- Even though the game is over, do they see any additional improvements they could make? Someone will usually point out that the bottlenecks can now be seen clearly, and perhaps they can do some load-balancing. They may have known or felt this before, but now they have proof.
- How did the roles, challenges, and solutions compare to their world?

NET OBJECTIVES

We are committed to delivering the principles, practices, and perspectives that businesses must know in order to maximize their return on their technology solution and software development efforts. We combine our experience and a time proven approach based on lean thinking to continuously extend the capability of what is possible in creating effective technology delivery organizations (IT or product). We provide these learned methods to our clients to assist them in achieving their goals and in assisting them in making their organizations more successful.



Full course descriptions may be found at
www.NetObjectives.com/training

FLEX Enterprise Transformation
Lean • Agile • Team Agility Patterns • TDD • ATDD
Assessments • Consulting Training • Coaching