Glossary

Lead time - the time is takes for a work item to be completed, once work has started
Work item - a unit of work in your process. Also called PBI, user story.
Work Item Completion Cycle Time - how often a work items is completed (done done)
Cycle time - for this exercise cycle time will mean Work Item

Facilitator’s Guide

Your job is to guide the team, reveal useful information and to encourage discussion. There are 5 rounds. In each round the players will have questions to answer; some will be from the CEO and some will be questions designed to foster discussion. Each round will contain facilitator hints that allow you to understand the story of delivery. These hints should help you guide the players towards answering questions.

First please pick a name for your team. Ask the team to write the team name on all their sheets.

Record any interesting discussions or thoughts that the team have. If there is time at the end you will have a chance to tell everyone else about these.

Instructions for you, the facilitator are in black. Instructions for the players are in courier, blue. Privileged facilitator hints are in Cambria green.

Let’s get started, we need to set the scene. First, read this to the team:

> You work for a company call Birds Ltd. You are a team of advisors who work directly under the CEO. After some market research a feature has been requested on one of your products called BirdSpotter. The feature is called BlackBird and will allow users to automatically identify pictures of blackbirds. You have a team of people who are going to implement the feature. Your job, as a team, is to help the CEO to direct the resources of Birds Ltd and to keep the CEO informed of the finish date. It's currently late in August and work is about to begin. Being agile development, the goals of the product are clear, however the delivery hasn’t been broken down into work items yet.
>
> The CEO needs this feature to be ready for Christmas. The financial success of the company for the next 6 months is riding on this feature. At the beginning of each month the CEO comes to you and asks when the feature will be done.
>
> On average (mean) it takes 3 months to deliver a feature at Birds Ltd. Although there is a lot of variance around the mean, lots of features take 1 month and lots take 6 months.
>
> The BlackBird feature looks similar to the robin feature. The robin feature took 2 months to develop, with a team that was

familiar with the BirdSpotting application code base. The current team is made up of 7 people, 4 developers, 2 manual QAs and 1 automation QA.

They are:
Betty - Developer
Tim - Developer
Gary - Developer and Team Lead
Robert - Developer
Larry - Automation QA
Shilpa - Manual QA
Daman - Manual QA
Alastair - Product Manager

Only Daman, Betty and Alastair have worked on the BirdSpotting application before. Everyone else is new to the product. Normally it takes 3 months to get someone up to speed if they are working through documentation. It takes a lot less time when they pair with someone. Birds Ltd tend to protect a team once they are working on a feature and they will be dedicated to just that feature.

Facilitator Hints

The nature of the product and feature are unimportant. If the players ask questions about that, repeat that the feature allows automatic identification of blackbirds, but that this is not important for the game. More useful would be for the players to try to find out things like what the composition of the team is, whether they have worked on the code base, how long features take to develop, whether they will do any other work.
Cycle time is based on calendar days. This means that some work items span over weekends. The variation that comes from work items spanning over a weekend evens itself out.

A key discovery that the team needs to make is that you can multiply the cycle time by the number of remaining work items to derive how many more days are left in delivery. For example, if the cycle time is 5 days and there are 10 work items then you can expect delivery to take 50 calendar days.

ROUND 1

1st October: End of Month 1

Goal: Players understand they need to get some way into the feature until they start to generate data to produce an estimate of the delivery date

Hand the players sheet set 2: End of Month 1

Read this to the players

> The team has been working hard for just over a month. Take a look at the data on what they have delivered and try to answer the questions.

Facilitator hints:

At this stage the only data that is available is that the second work item is taking a long time. Talking to the team is how the players should find out what's happening. If the players choose to talk to the team, reveal that they have been posse programming. This has meant that the second work item took longer than if the dev who knows the code had written it on her own. However, the developers are learning lots about the code base and the quality of the learning is very high. The team have also done t-shirt size estimation and they think the current work-item is the biggest. They have also arranged the stories so that the bigger stories with unknowns will be executed first.

If the players have trouble getting a discussion going, ask them these questions:
Q: Given that all 4 developers have been posse programming, what do you expect to happen to the cycle time once they start working in pairs?
Q: Do you think that all the work has been discovered. How many work items would you expect to be added over the coming months?

If the team are stuck read them this:

> The team come to you to tell you what's happening on the feature. They tell you that although the second work item is taking a long time to complete they are learning a lot. They tell you, based on T-shirt size estimation, that the biggest work items with the most unknowns are at the front of the queue of work. The team reports that the work is proceeding in a very orderly fashion. Rather than working on multiple work items at once they are focussing on fewer work items in order to amplify learning. Once this work item is completed they expect to pair or work on their own. The team reports that they think the work items in the backlog are cover all of the work that they need to

```
do to deliver the feature.
```

ROUND 2

1st November: End Of Month 2

Goal: To understand that the cycle time can be used to estimate delivery date
Goal: To understand the different averages on the graph

Hand the team sheet set 2: 1st November End of month 2

Read the team the following

> The team has been working hard and the pace of work item completion has increased. No new work has been found this month. The team reports that once they completed the second work item a lot of the subsequent items were repeating the same work in different areas. Take a look at the user stories completed and the graphs and answer the questions. The team reports that two work items completed on the same day.

Facilitator hints:

The team reports that after work item 2 they have been doing work that is well understood and is to some extent a repetition of the work done already. They report that all of the shorter work items were in the same area of the code, but they have identified a work item that is in a new area of the code and expect that work item to take longer than the current work items. This is mainly because they have to learn a new area of the code and a new area of the product. After that they think they have de-risked the feature and the stories will be repetitions of work already done and well understood. The developers report that they are now pairing on each work-item, and the QAs report the same. This is driving learning and accelerating the pace of delivery.

There are two averages. The average over the whole feature delivery and the rolling average. For predicting how much work is left and therefore the delivery date, the rolling average can sometimes be better. To make this call you have to decide whether the rest of the delivery has the same characteristics as the recent history or the long term history. If recent history, choose the rolling average; if long term history, choose the overall average. When making predictions you can even use both and produce a range of dates.

It should be clear by now that cycle time can be used to predict delivery. If the players haven't figured this out, read them the following text:

> Work Item Completion Cycle time is a measure of the pace of delivery. As such it can be used to derive when the feature can

be delivered. Based on the average, the date of delivery can be predicted to be 100 days. This is calculated by multiplying the average by the remaining number of work items (16*6.22).

Now ask them, based on the rolling average, how many days are left until delivery?

If the team get stuck answering the discussion questions read them the following:

Work item completion cycle time is simply the difference between the completion dates of two work items. While the second work item (id 45143) was in testing, the third work item (id 45371) was started by the developers. There was a lot learning and testing for the QAs for second work item. The third work item was tested with automation which meant a couple of hours work for the QAs to test it. This meant that the team completed two work items on the same day.

If the players have trouble getting a discussion going, ask them these questions:

Questions for the players

Q: How long did the second user story take?
Q: Do you think the second user story is typical or atypical. Is it something that will repeat or is it a special case?
Q: How many work items were added in the last month?
Q: How many more will be added, do you think?
Q: Rather than a single date of feature completion, can you produce multiple dates and multiple confidences?

ROUND 4

1st December: End Of Month 3

Goal: the players observe and act on the trends seen in the graph to give a better prediction of when the delivery will be ready

Read this to the team:

> The team reports that this month they hit a work item that was hard to complete and took longer than they expected. It required them to work in a new area of the code and product. Once they completed this work item they found that work items started to complete quickly again. The team have looked at the work left and some of the work items will only a very small amount of testing effort. They expect the pace of work item completion to increase.

Facilitator hints:

There is a clear trend in the graph that the players should notice, it's present in the last few work item completion cycle times. The trend can be used to predict that the cycle time of work items is going to decrease all the way through the end of feature delivery. If players spot this, they should make predictions based on a lower cycle time than the one they currently see in the data.

If the players have trouble getting a discussion going, ask them these questions:
Q: Can you see any work items with a cycle time of 0?
Q: Are the raw and rolling average staying the same or changing? What might that give a hint to?
Q: Contrast the way the average has moved in the last work-item with the way the raw and rolling average changed.

ROUND 4
13th September: END

Read the following to the players:

> The team has finished their work in good time for christmas. The CEO is very pleased with your efforts. The team noticed that lots of the stories at the end could be completed very quickly. Now, reflect on your ability to predict and how your predictions changed over the course of delivery. Please answer the questions on your sheet.

Facilitator hints:

The graph contains two distinct phases, first half and second half. You can spot the two phases by looking at the raw cycle-time and 3 work item rolling average. Halfway through the project both of these change from being centred around 6 to being around 2. There are a couple of reasons for this. Most of the risk and learning was front loaded on the project. The team chose to do the integration work first. Also, as time went on the team learnt how to move more work to the QA function which was experiencing a lot of slack. Since many members of the team were new to the BirdSpotting application they took time to come up to speed. The team chose to do this by having everyone in a function work together on the same subtask of the work item. This allowed a very deep knowledge transfer from the developer who knew the code and product. That approach slowed things down a lot at the beginning but the team betted on the fact they would get a big payback within the delivery of the BlackBird feature. Much of the work in the final month(actually 2 weeks) was a rinse and repeat of previous work. The team also refactored the interfaces around a core component early. This allowed them to swap the core component out for a replacement easily and meant that the predominant effort in the final work items was testing. Even with this testing effort they realised that they didn't need to re-test the functionality within the new core component, they only had to test that the component had been connected to the existing code properly.

If the players have trouble getting a discussion going, ask them these questions:
Q: The team started out with little experience. How does the graph show that?
Q: What would a graph look like if there was a big integration phase at the end?