

XP Game

Coach

Thank you for being a coach at the XP Game. Your task is to explain the game to the team, and to assist them during the game. You measure the time played and write down the scores on the Scoring Sheet.

You answer questions about the game. You act as the customer when the players are developers; you act as the developer when the players are customers.

Terminology

- **Business Value:** the "value" of the completion of a story to the customer.
- **Story points or ideal seconds:** measures how difficult it is to implement a story.
- **Real time seconds:** the stuff you can measure with a stopwatch (but we won't).

Purpose

- Maximize the "Business Value"
- Business Value is expressed in *values* between 100 and 500.
- Team that ends with the most Business Value wins

An XP Game Iteration

The game begins

Each team has a bag with game props. Empty the bag on the table. The team can have a look at the props. Pick the pack with story cards that is marked "iteration 1".

Customer writes stories

You are the *customer*. The team members act as *developers*. Turn around the Customer/Developer Responsibility Sheet.

You explain to the developers that you've written some stories. You want them to have a look at the story cards. You hand them the first pack of pre-written story cards. The team may *briefly* study the cards.

Developers estimate stories

The team members still act as *developers*. You are the customer, and answer questions about the descriptions of the stories. Encourage them to ask questions and to look at the props. For every card, the team estimates the time needed to perform the task: 10-60 (ideal) seconds, or impossible.

If possible, the entire team is allowed to help for the implementation of a story. Remember that it shouldn't be too hard to do the acceptance tests. For example, for the 'Monkey dice' story, it's not very practical if everyone is throwing the dice at the same time.

Maybe they will have some questions about stories that look exactly the same to them, but that have different business value. You should look surprised, and say you didn't notice that they look the same (even if they do).

If they put aside a Card as being impossible, try to convince them to estimate it anyway, because it has so much business value.

If all the story Cards are estimated, they should give the Cards to you. Thank the development team.

Customers choose and prioritize stories

Now, the (same) team members act as *customers*. Turn around the Customer/Developer Responsibility Sheet. You tell them that the developers have estimated the cards, and that they, the customers, can now choose which stories have to be implemented in the next iteration. You tell them what the velocity of the development team is, and write that down on the Score Sheet. In the first iteration, the velocity will be 180 seconds.

The team selects stories with a total estimated time of no more than 180 ideal seconds. Explain that they have to try to maximize the potentially earned business value.

If that's done, the customers have to sort the tasks according to importance. Explain them that the stories will be implemented in this order. They have to take into account the fact that the plan is only a plan. Maybe, halfway the iteration, the development team will notify them that the plan can't be finished entirely. In that case, it would be nice to have the possibility to drop the least important stories. That's why those stories should be at the end of the plan.

The customers return the sorted stack of story cards to you. Write the resulting plan on the score sheet, and thank the customers.

Developers implement stories

The team members act as *developers* again. Turn around the Customer/Developer Responsibility Sheet. You tell the developers that the customers have made a plan, and that the plan can be implemented now. During the implementation, you act as customer to answer questions and to perform acceptance tests.

The team takes each story in turn, in the order defined by the plan. First, they can briefly discuss their strategy. Team members sign up for the task. For most tasks, the entire team can participate in the implementation. The team has to come to an agreement with you about the way they will let you know that the implementation is finished.

When the team is ready to start the implementation, turn the hourglass upside down. One side of the hourglass is marked, so that it's not too difficult to remember which side is 'up'. Stop the hourglass when the team declares the story implemented.

When the hourglass is stopped, you can do the acceptance test. When the story is implemented to your satisfaction, you can add its business value to the score sheet. You should pay attention on the quality of the implementation. If you think it's not really finished, the team doesn't earn the business value. It is possible to start the hourglass again, and let them complete the story. Tell them to have a look at the time. Maybe it's better to warn the customers, and ask *them* whether this story has to be finished, or if it's better to drop it.

If a story takes too much time, you may allow the team to stop. In that case, the story is put aside and the business value is not earned.

Halfway through the iteration, the team can assess how they're doing relative to the plan. Half of the time has elapsed. Have they performed more or less than half of the stories? This will give the team a clear indication of their performance. If things don't go as planned they will be able to warn their customers.

The complete iteration lasts until the hourglass is empty. Only implementation time is measured. Time spent allocating tasks, preparing, performing acceptance tests, etc, ... is not measured.

If all stories have been implemented in less than 180 seconds, the team (acting as customers) may select another story to implement from the stack of estimated stories.

Have fun!