

ARPG – THE AGILE RELEASE PLANNING GAME

Facilitator's guide – version 1.1 April 2010

A game where participants run an agile project.

Contents

The game..... 3

- The story so far 3
- What needs to be done..... 3
- Equipment at hand 3

Getting started 4

- Break the group into teams..... 4
- Explain the story so far..... 4
- Set the board up 4
 - Place the deadline cards on the board..... 4
- Explain velocity and story points 4
- Prepare an initial release plan 4

Running the first iteration..... 5

- Confirm the plan for the iteration 5
- Determine the actual velocity 5
- Deal with existing defects 5
- Allocate the remaining velocity 5
- Identify new defects 5
- Make scientific discoveries..... 5
- Update the status reporting sheet 5

Running remaining iterations..... 6

Ending the game..... 6

Debrief 6

Variations and optional rules 6

- Everyone has the same luck 6
- A more focussed outcome 6
- Improved status reporting 6
- Iteration story wall..... 6
- Hang over stories..... 7
- Dependencies 7

The game

The game is designed to help participants understand the concepts involved in release and iteration planning.

Participants must plan their work across a series of iterations and then adapt their plans as the project evolves.

In order to be successful, participants must meet critical deadlines, minimise defects, monitor their velocity and replan as they gain further insight into the evolving project.

Variants of the game allow participants to explore further concepts through:

- Managing within iterations by using story walls; and
- Using velocity charts and other tools to track progress.

The story so far

The team are the advanced party for the first mission to Mars.

The team got to Mars without any issues and a robot team landed to establish a base. Unfortunately the team's own landing did not go so well.

For some reason, the automatic pilot fired the retro-rockets far too late. Instead of making a gentle descent, the team plunged into the Mars base, virtually destroying it.

All communication has been lost and the team need to set priorities without any guidance.

Worse still, the team will run out of crucial resources in only a short time.

What needs to be done

Just before shutting down, the mission computer prepared an analysis of the team's chances of survival.

If the team repair the landing module then they can access a vast cache of supplies and even go home. Survival is thus assured.

If not then they will need to be rescued.

Fortunately another exploration vessel is already on route. Unfortunately this will take time, so the team will need to survive for 8 complete iterations before being rescued.

It would also be good if the team could make use of their time on Mars to complete the original objective of their mission – scientific research.

If the team live long enough, they can rebuild the original science labs and fulfil their original mission.

Equipment at hand

The mission computer produced a series of story cards for the team to use. Each card contains the name of a feature the team can build (or repair) and the number of "story points" required to do so.

The amount of story points the team can complete each iteration (or turn) is known as the team's velocity.

The computer estimates that the team's velocity is approximately 15 story points per iteration. This number is expected to change over time as mission evolves.

The computer has also produced some other tools for the team to use:

1. A mission glossary, explaining the features that can be built, as well as the meaning of other terms;
2. A status report for the team to use in tracking their progress; and
3. An iteration planning board for the team to use in planning iterations.

Getting started

There are a number of variations of the game available and these are explained a little later. But most teams start with the basic game.

The game can be introduced by first presenting an explanation of the concepts of release planning or it can be run prior to the explanation of the theory.

Break the group into teams

To introduce the game, break the group into teams. Each team will need:

1. Team members. The best team size is between 2 and 8 people.
2. The game board, defect counters, deadline cards and story cards.
3. A pair of dice.
4. A status report sheet and a pen. You are welcome to photocopy the status report for use in the game.
5. A story glossary describing the stories they can work on.

Explain the story so far

Explain the background story to the group.

Set the board up

Have each team lay the board out in front of them.

Place the deadline cards on the board

The purpose of the mission is to make new scientific discoveries. But to survive, the team will need to either repair the Mars Lander or create the right features to last 8 iterations.

But the base is collapsing around the team's ears. In order to merely survive, the team will need to meet a set of critical deadlines.

Deadlines are marked on the board. They can be met by building certain features in time, or by deferring the deadline through implementing smaller stories.

Place the deadline cards on the board where indicated. This way, if the deadline is met the card can be deferred to a later iteration or removed from the board to show it has been permanently addressed.

Explain velocity and story points

The team will survive by "completing stories" to meet deadlines or build features.

Each story is represented by a story card. Each card has a heading, a story number and a story point rating.

- The story number can be used to find the story in the project glossary.
- The story point rating is the cost to complete the story.
- The velocity is the number of story points the team can expend to complete stories each iteration.

Prepare an initial release plan

The final step in the initiation phase of the project is to prepare a release plan.

Give each team 10 minutes to:

- Review the project glossary to understand the benefit of the different features; and
- Come up with a release plan.

Note the team will decide how many iterations they plan for, but they must at least allocate stories to the first iteration.

Players can allocate more stories than they expect to complete, since they will be able to confirm which order stories are done during each iteration.

Running the first iteration

Confirm the plan for the iteration

The team confirm the stories to be attempted before the start of the iteration.

In future iterations, the team need to decide whether to reduce outstanding defects. Fixing a defect requires 1 story point per defect.

However, in the first iteration there will be no defects to fix. So the team should place the "zero" defect counter on the board.

Determine the actual velocity

We predict the velocity for the first iteration will be 15. This is a good indicator of how much work the team will get done, but it is not a perfect prediction.

So we roll dice to represent the impact of unpredicted occurrences, as well as the ongoing improvements made by the team.

Have each team roll two dice and consult the velocity change table on the game board.

Add (or subtract) the amount shown in the table to calculate the actual velocity for this iteration. This number will also be the expected velocity for the following iteration.

Deal with existing defects

Where defects exist, the team need to make running repairs, use workarounds and perform other work that will distract the team from both creating new features and even repairing the defects themselves. We refer to this as technical debt.

Consult the "Drain due to technical debt" table on the game board. The number shown represents the amount of effort the team must spend coping with existing defects in production.

This number is subtracted from the velocity each iteration before the velocity can be spent fixing defects or building stories.

Allocate the remaining velocity

If the team had committed to fixing defects during the iteration then these would need to be completed before work begins on stories..

The team can then choose which stories to complete with the remaining points. Incomplete stories must be deferred until a later iteration and any points that cannot be allocated to complete a story will be wasted.

Identify new defects

The team is likely to find defects in the work once it has been completed.

To represent this, the team roll two dice and consult the new defects table on the game board. This represents the number of defects that will carry forward into the next iteration.

Make scientific discoveries

In addition to merely surviving, the team has the opportunity to become one of the greatest scientific expeditions of their age.

If the team has built any science labs, then for each one, the team roll two dice and consult the "New discoveries" table.

Note that the team can add 1 to each roll for each Mars Rover they have built.

In addition, the first time a 10 or more is rolled for the life research project, the team discover life. And they gain 10 kudos points instead of 2.

Update the status reporting sheet

The team update the status report to reflect what has occurred during the iteration.

Running remaining iterations

Assuming the team survive they now move onto the next iteration.

The process is essentially the same as running the first iteration, except that defects may now exist.

Ending the game

For a single team, the game ends when either:

- The team do not survive the iteration;
- The team are rescued at the end of the 8th iteration; or
- The team have completed the repairs to the Mars Lander.

If more than one team is playing then other teams can continue, or the game can end for the whole group if preferred.

Debrief

Once the teams have completed the game, perform a debrief as a group.

One way to do this is to hold a retrospective:

- Have each team answer 3 questions:
 - What worked well?
 - What caused problems?
 - What can we learn from this?
- If there are multiple teams, then have the teams discuss their findings as a group.

Variations and optional rules

Everyone has the same luck

Allowing each team to roll the dice for themselves allows the facilitator to move freely between teams during the game. But it also means that different teams may have different outcomes based on luck rather than planning.

An alternative approach is for the facilitator to roll the dice each iteration and announce the results.

A more focussed outcome

The basic game requires people to set their own priorities.

Instead it is possible to have a more defined outcome for the teams to race towards.

In this case the game ends when one team achieves the objective.

Two typical objectives are:

- **We will not be rescued.** The teams will therefore need to build the Mars Lander. The first team to do so wins.
- **It's all about the science.** This is the same as the basic game, except that the winning team is the one with the highest scientific kudos at the end of 8 iterations.

Improved status reporting

The status sheet allows the teams to keep track of what is going on. But better tools are available to Agile development teams.

The game works well with participants completing the same reporting tools that they would utilise in a real project, for example:

- Velocity chart (or burn up chart);
- Defect tracking chart; or
- Iteration contents chart.

Using these tools will require longer turns and in return it will help to explain and imbed the use of these tools.

Iteration story wall

There is an extension game that tracks the same stories as they are built within the iteration.

This helps to explain the use of the story wall in a project.

However using this extension throughout the whole game will lengthen the playing time.

An alternative is to use the story wall only after iteration 4, or to use it as a standalone tool to explain story walls as a separate exercise.

Hang over stories

If the team is happy to play a more complex version of the game then they can introduce "hang over stories".

These are stories that are almost completed in an iteration but "hang over" into the next iteration before being completed.

However the team will experience a level of inefficiency when completing stories over multiple iterations. So they must "waste" a story point.

Furthermore, if the story is not completed in the next iteration then the points allocated to the story are lost and the story is no longer a hangover one.

For example:

- The team then has 2 points left to allocate to stories at the end of the iteration and decides to allocate them to a 5 point story that will be hung over to the next iteration.
- The team can now complete the story in the next iteration, but they have wasted one point, so they still need to use a further 4 points to complete the story.

Dependencies

Dependencies add great complexity to planning projects in the real world.

To duplicate this in the game, make the following stories dependent:

Story	Is dependent on
19c Mars Lander fittings	19g Mars Lander hull structure
19d Mars Lander communications	19e Mars Lander CPU
19f Mars Lander controls	19e Mars Lander CPU
19i Mars Lander navigation unit	19e Mars Lander CPU
19k Mars Lander atmosphere unit	19j Mars Lander power plant
20 Lander final tests	All other Lander stories (stories 19a through 19k)
22 Robot development engineer	26 Robot factory
23 Robot test engineer	26 Robot factory
24 Robot TV crew	26 Robot factory
25 Robot chef	26 Robot factory
34 Water recycling	33 Water production plant

Where a story is dependent on another one, it can be completed in the same iteration as the other story, but cannot be completed in a prior one.