[reative Commons License](http://creativecommons.org/licenses/by-sa/4.0/)  
High Performance Teams Game by [Mark Levison - Agile Pain Relief Consulting](http://agilepainrelief.com/)

is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](http://creativecommons.org/licenses/by-sa/4.0/).

Your team is working on the World’s Smallest Online Bookstore, a site that provides the best results (just a few) for every search, not every result on earth. We’re a vulture capital funded company, so if we don’t deliver, our funding will be cut.

My goal is to help you see the effects of choices/tradeoffs on productivity and team cohesion. While some of the benefits of Agile happen at the individual level, there are many things that affect the relationships between team members, and therefore the overall cohesion and productivity of the team.

Will your team win the award of finishing the most User Stories?

Will it win the award for being most improved (i.e. best final capacity)?

## Ground Rules

* *Rounds represent 6 weeks or three 2 week sprints*
* *Our goal is to play 6 rounds == 36 weeks*
* *Each table will have 5 to 9 team members*
* *In each round you have budget for the amount of work/stuff you can do based on your team’s capacity. Some of that budget must be spent on delivering features, otherwise the business will threaten to let you go. Some of it should be spent on growing the team and their engineering skills, otherwise you don’t get more budget capacity.*
* *There is a risk of failure, and not all teams make it.*
* *There is a sense of risk and randomness. Every round, you roll two dice to see what gremlins will affect you. Human interactions are imperfect and the gremlins will simulate that.*
* *In each subsequent round, all prior actions are still available.*

Drag effects:

* **Moral/Communication Drag** - If nothing has been done to improve the team on the communication level, their capacity will be reduced by 1 Story Points (-1) at the of second round and again in each subsequent round.
* **Engineering/Tech Debt Drag** If nothing has been done to improve the team engineering practices, their capacity will be reduced by 1 Story Points (-1) at the of second round and again in each subsequent round.

*This isn’t an action in of itself – rather the default that will happen for any uncommitted capacity.*

Commit to User Stories- **Cost**: 1 per Story Effect: Roll one die per User Story committed. 3 or higher = commitment met, User Story is completed

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Title** | **Action Description** | **Cost** | **Explanation** | **Effect Next and Future Rounds** |
| **Round 1** | | | | | |
|  |  |  |  |  |
| Same Floor | Get all Team members on the same floor (can’t be used with Team Room) | 3 | Even if you can’t create a team room, just getting people on the same floor reduces the cost of communication. | +1 Capacity, per round, for 5 rounds, accounting for face to face time. |
| Team Room | Set up a team room with walls | 5 | Teams form faster in a Team room because we can use eye contact to ask for permission to chat and most conversations happen in the real world | +2 Capacity per round, for 5 rounds |
| Protected from Outside Distraction | ScrumMaster protects the team from outside distraction | 1 |  | +10% on User Stories |
| Build Server | Setup build server and Continuous Integration | 2 |  | None – only long term effect is enabling Unit Testing, TDD |
| Working Agreements | Create Team Working Agreements | 1 |  | +1 Capacity |
| Clarify Product Vision | PO and Development Team collaborate on understanding Product Vision using an exercise like Product Box | 2 | Teams that aren’t involved in the creation of their product vision are doomed to build Product that neither PO, Customers nor Stakeholders want | User Story Success Percentage +15 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Title** | **Action** | **Cost** | **Explanation** | **Next & Future** |
| **Round 2** | | | | | |
| Create Remote Avatars | Distributed teams benefit when team members better understand each other. |  |  | +1 Capacity |
| Eliminate Long Lived Feature Branches | All Work is done on Main or Trunk | 2 | When teams use Feature Branches – then they’re not really using Continuous integration.  Feature branching optimizes for the individual while harming the Team | + 1 Capacity |
| Backlog Refinement | Introduce regular Refinement sessions |  | The Development Team and the Product Owner sit down every Sprint. They take the time to better understand upcoming features. They might estimate or split stories, create acceptance criteria. They also create new stories and delete one's that no longer make sense. All of this improves their understanding the product they're building. | +15% on User Story percentage |
| Social Time | Promote some social time, e.g. common coffee breaks, team lunch, etc. | 1 | Building Trust requires relationships beyond the work you do. Taking the time to socialize has helped your team. | +1 Capacity |
| Problem Solving Bonus | Offer a firefighter award to any team member who solves …. | 1 | Promoting a firefighter culture promotes individual behavior and, surprisingly, the starting of fires. | +1 to start but after two rounds -1 |
| Unit Testing | Introduce Unit Testing (can only be used if you set up a Build Server/CI system) | 2 |  | +1 Capacity |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Title** | **Action** | **Cost** | **Explanation** | **Next & Future** | |
| Move Water Cooler | Move the water cooler/coffee near the team space. | 1 | Social conversation happens around the water cooler/coffee and tea. If team members have to leave the team space then the social benefit is diffused | +1 Capacity  *If you already have a team room this is not applicable.* | |
| **Round 3** | | | | | |
| Observe People + Relationships | ScrumMaster spends time observing people, how they interact, and the quality of their relationship. | 1 | Watching the Team tells you where to put your coaching energy. | +1 Capacity | |
| One-on-Ones | ScrumMaster meets with all team members for a regular one-on-one. | 1 \* | Once ‘Gremlins’ start to popup, this action mitigates the worst of the effects, because you already have a deeper understanding of team needs  Keep this card | None | |
| Pair Programming | Two team members – one computer | 2 | Team Members working in pairs have a lower defect rate, simpler code and learn from each other. | +1 Capacity | |
| Test Driven Development | Writing Unit level Tests before writing the code  (requires a build server) | 2 | By writing the tests before the code – the Developer is forced to consider the simplest solution to their problem.  Result: Less code; simpler design and fewer defects | +2 Capacity | |
| Story Mapping or other Strategic technique |  |  |  | +10 User Story rolls | |
| **Round 4** | | | | | |
| Personal Productivity Bonus | The company will offer anyone who exceeds their performance goals an extra $5,000. | 3 | Personal Productivity Bonus increases individual focus on achieving the goal. Unfortunately teamwork suffers. |  | User Story Success 1 time goes up by 50%.  Long term reduced by 10%.  -2 Capacity |
| New Specialist | Hire a new specialist for the team. This person will fill in missing pieces. | 4 | Hiring a new person and bringing them up to speed slows the team down and damages existing Team member relationships | -2 Capacity | Three rounds after doing this, roll one die – if you score 4 or higher this person is a boon to your team, your original capacity is restored, and goes up +1.  If not, oh well *☺* |
| Informal Cross-Training | Informal cross-training for existing team members in an area the team is weak. (Testing anyone?) | 1 | Counts for the social cohesion of the team as well as improving the engineering practice. | None | +1 Capacity |
| Overtime | Overtime for the whole team. | 2 | You get more work done this round but are tired and make more mistakes/ messes in the code base | + 2 User Stories Completed this time | -20% on User Story success rolls |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Formal Cross-Training | Formal cross-training for existing team members in an area the team is weak. (Testing anyone?) | 3 | Will boost the technical skill of a team member – improving capacity – but doesn’t help the social cohesion. | None | +2 Capacity |
| Collaborate on Understanding Vision | Spend two to three hours as a team collaborating on what the goal means and what is important. | 1 | Turned a meaningless goal into something that matters for the team. You have a more focused Sprint.  Teams see through artificial pointless goals that they’re not involved in setting. If you didn’t discuss the goal with the team, they’re demotivated and not buying it and actually don’t try as hard | Complete all User Stories to which you committed (one time effect)  *Want more often? Buy more often* | None |
| **Round 4** | | | | | |
| Pre-Allocate Capacity for fires | Set aside some of your team’s capacity to deal with the fires as they come up. | 2 |  |  |  |
| Sacrifice One Team Member | One team member deals with all the fires as they come up, allowing the others to stay focused on delivering new value. | 2 \*\* |  |  |  |
| Behavior Driven Development | We write our Acceptance Criteria as examples – with at least one of each: Business Analyst, Developer and QA participating.  The Acceptance Criteria are created before the code is started | 2 | By writing the Acceptance Criteria first – many misunderstandings are avoided and defects are eliminated.  If these are automated well, they provide “living” documentation and a regression suite |  | +3 Capacity |
| (\*\* when used with the “Pre-allocate Capacity”, combined cost is 3) | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Title** | **Action** | **Cost** | **Explanation** | **This round** | **Next & Future** |
| Ignore Fires | Refuse to deal with fires until the next sprint. | 2 |  |  | *None. You get yelled at by Management. You still have to deal with the fires.* |
| **Round 5** | | | | | |
| Accept new Stories mid-sprint | **Either:** Accept New Stories Mid-Sprint | 0 |  |  | *Teams that accept new User Stories lose the 3 Stories this round. Accepting change mid-sprint is very expensive. The team was focused on meeting the goal. In addition….* |
| Refuse new stories mid-sprint | **Or:** ScrumMaster and Product Owner work together to refuse new stories mid-sprint | 2 |  |  | *This costs you political capital, but helps the business see that making arbitrary changes mid-sprint is expensive.* |

Action Ideas:

Backlog Refinement – increase chance to succeed at User Stories

Work on Retrospectives

BA, Development Testing Collaboration – increases productivity, improves likelihood of completing a User Story. *Could necessary to do BDD?*

Improve DOD - …?

Skip Retrospective get 2 additional story rolls?

We could code actions so that their effect is negative in the round after and then slowly increasing positive. *Example Eliminate Feature Branches capacity effect “-1” one round in the future, +1 after two rounds.*

Missing Gremlins that affect the product itself – market changes

Two – three rounds after go live a customer makes a big request that requires a big change in strategy. *Prior Vision Work or Strategy work would have mitigated this*

How to incorporate lessons around the definition of done

At the start, we’re a brand new team who have never worked together before. We have no established relationships or history. We were hired by the company because our resumes say we’re the best in the field.

Round I – Sprint 1:

“Team, welcome to the World’s Smallest Online Bookstore. We hired you because you’re the best individuals in your respective areas. Please remember that we’re Vulture Capital funded and we have only a few months runway, so you must deliver. This first Sprint, the company really needs you to prove that you can deliver a working ….”.

Initial User Story Point Capacity: 10

*Facilitator’s notes*

* *Force the team into storming.*
* *Ensure that few, if any, teams actually got their work completed.*
* *If your ScrumMaster protected the team from distractions, rolls of 2 or higher for Stories succeed.*

## Round II – Sprint 4

“None of you met our expectations that you would complete 10 User Stories in the last round. Our vulture capitalists are becoming concerned and ask if you can really deliver.”

*Facilitator’s notes*

* *Teams that don’t do something to improve their communication or teamwork (in this or the previous round) have their capacity reduced by 2. (Specifically Team Room, Working Agreements, Social Time)*
* *Teams that don’t set up a build server by the end of this round reduce their capacity by 2.*
* *Teams that have set up working agreements gain an additional budget point the round after they were established.*
* *If your ScrumMaster has consistently protected the team from distraction for the previous two Sprints, grant them now a gain of 1 additional Capacity point forever, as many people outside the team are starting to appreciate the cost of actions.*

## Round III – Sprint 7

“We must go live with an early version of the product this round, for CES. Due to your limited productivity in past rounds, management are prepared to offer some options to help you out. We will pay an extra ‘4’ points for anything that helps. Another team member? Overtime?”

*Facilitator’s notes*

* *Announce now that changes like team rooms take a long time to pay off – often as long as 12 weeks. Teams that took on the team room in round 1 now gain 2 additional budget points per round hereafter.*
* *In teams that made no changes to improve Engineering Practices, Technical Debt is starting to take hold. Reduce capacity by 1, and continue reducing by 1 every other round hereafter.*

## Round V – Sprint 13

“We’re live, congratulations. In addition, if we’re unlucky some teams will have to deal with a fire.”

*Facilitator’s notes*

* *Teams that do nothing to protect themselves from interruption due to fires automatically lose half their capacity this round.*
  + *Teams that pre-allocated time to deal with fires roll a die: 1-3 no additional losses; 4-5 lose one additional story; 6 lose two additional stories*
  + *Teams that dedicated one sacrificial team member roll a die: 1-3 no additional stories lost – same outcomes pre-allocating time to deal with the fires*
  + *Teams that pre-allocated capacity and sacrificed a team member, congratulations, you took care of all of the fires that came up.*
* *Teams that invested in Unit Testing two rounds ago gain 1 capacity.*
* *Check to see if anyone gains benefits from Team Room; Social Time; Moving Desks closer together; Working Agreements;*

## Round VII – Sprint 13

“Wahooo, we’re making money. We’re starting to get great customer feedback with all sorts of new ideas. Even though it’s mid-sprint and you’ve already committed, the company needs some changes.”

## Gremlins or Chance Events

These can happen in any round. At the end of every round, roll 2 dice.

|  |  |  |  |
| --- | --- | --- | --- |
| **Event** | **Effect** | **Die Roll** | **Mitigating Factors** *known only to the facilitator* |
| Team Member/Management Relationship | The team perceives their peer is being treated differently than they are and feel this is unfair.  Capacity -2 until something changes. If nothing is done in the next round, reduce capacity by 1 more point. | 2 | Teams that have established social interaction reduce this effect to -1 Capacity.  Establishing one-on-ones reduces the effect by 1 point. |
| Management yells at a team member in public for not pulling their weight | The team member who was yelled at feels their personal status was reduced. The whole team fears they will be next.  Capacity -2 | 3 | Teams whose ScrumMaster has been protecting them from distractions may ignore this. |
| We’ve had an emergency on another team, we need your best tester for a while. | Capacity - 3 for three rounds | 4 | Teams whose ScrumMaster has been protecting them from distractions reduce the number of rounds by 1.  Informal or Formal cross-training also reduces the impact by 1 Capacity Story Point.  These are cumulative effects. |
| One of the people on your team isn’t pulling their weight - not even close. However, nothing seems to get done about it. | The rest of the team is pulling together and taking this person’s work on, but it’s harming morale and productivity. Capacity - 2 | 5 | ScrumMaster who conducts one-on-one’s spots this issue early and reduces the impact to 1.  Informal or Formal cross-training also reduces the impact to 1. (Note: this is not cumulative). |
| One team member is consistently late or misses Daily Scrum. In addition, they do most of their work on their own. | Other team members are annoyed that this person feels that they’re more important than the team. Capacity -1 until something changes. | 8 | ScrumMaster who is already conducting one-on-one’s action avoids this problem  Same with, Working agreements also work because they empower team members to raise the issue. |

[reative Commons License](http://creativecommons.org/licenses/by/4.0/)  
Team Games by Mark Levison - [Agile Pain Relief Consulting](http://agilepainrelief.com)is licensed under a [Creative Commons Attribution 4.0 International License](http://creativecommons.org/licenses/by/4.0/).

--

Mike’s notes on the game:

 ​Mechanics and playability?

* Mechanics: Feels like the game could be extended for a few more rounds, to provide more time to feel the compounding effects of the modifiers.
* Mechanics: How will this be played?  Single/multi player? Local/online?
* Mechanics: There is a focus on capacity/velocity, backlog items done. Should product and sprint goals be a factor (demonstrating that sometimes it's awesome to not complete stories, but still hit your goal).
* Mechanics: Would be neat for the player's team to have certain attributes, which would impact the cost and benefit of improvements.  The player would decide on which improvements to take based on the team's strengths/weaknesses.
* Mechanics: Items seem to be pure positive/negative impact.  Double-edged swords might be interesting
* Mechanics: Idea - Bypass DOD to get more stories "done", but pay for it with increased tech debt drag (or production fire).
* Mechanics: Idea - Skip retro to get a few more points in the sprint, but reduce/disable improvements in the next sprint.
* Mechanics: Nothing to do in sprint 5/6?
* Mechanics: Confucious said: "Best time to plant a build server was last round, the second best time is now".  I can't implement unit testing because I didn't build a build server in round 1... why can't I build one in round 2?

 Is it obvious how the game progresses?

* Visualize product backlog progress
* Show sprint velocity history as a graph
* Visualize customer satisfaction with product (neat to show this diverging from velocity)

 Is it obvious the effect actions are having?

* Some options are obviously negative (e.g., firefighter award), but it is unclear why I would want to pick some beneficial items over the others.
  + One exception that becomes clear later, is the Build Server, as it unlocks unit testing.
* Visually represent active effects
* What's the impact of the PO or stakeholder telling the player to prioritize feature work?
* Maybe stakeholder happiness as well.

 Is the rule text clear enough?

* The significance of the relationship between rounds -> sprints -> weeks is unclear.
* Would it be simpler to just say:
  + Game runs for 6 rounds.  Each round is one sprint

 Anything else around the basics of the playability?

* Assuming the UI is in a very early stage, but more visuals/graphics
* Selecting from tooltips is not great.  Can you show a description on the main page?
* Add more visuals/graphics for the UI - Perhaps bars/graphics to represent the capacity breakdown section.