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.

Longer term change

* **Product Drag** – if nothing is done to improve Product Ownership i.e. Vision, Product Backlog Refinement etc. then the team’s success chance is reduced.

Much longer term – add Morale, Defects found and Cycle time as measures? See if Google Project Aristotle can help create new actions.

*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

In an ideal world I would balance the game choices so that each round has one improvement from each of: SM, PO, Eng, Collaboration and one item for remote teams

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Title** | **Action Description** | **Cost** | **Explanation** | **Effect Next and Future Rounds** |
| **Round 1** | | | | | |
| 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 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Title** | **Action** | **Cost** | **Explanation** | **Next & Future** |
| **Round 2** | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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** | | | | | |
| **Round 4** | | | | | |
| 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 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Round 7** | | | | | |
| 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 in two rounds time accounting for the fact this is hard to learn |
| (\*\* 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 8** | | | | | |
| 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.* |

Round 1:

Protected from Outside Distraction

Clarify Product Vision

Working Agreements

Build Server

Team Members On Same Floor

Round 2 – Failed to Deliver

Remote Team Avatars

All Work is done on Main or Trunk

Unit Testing

Social Time

Problem Solving Award *Needs a rename*

Backlog Refinement

Round 3: Work Harder

Observe People + Relationships

One on One

Pair Programming

Story Mapping moved

Refactoring

Round 4 –TeamBottlenecked

Informal Cross Skilling

Send Team Members on a testing course *Rename for clarity*

Personal Productivity Bonus

New Person on the Team

TDD

Limit WIP slows team down for now and speeds them up later

How to work Psychological Safety into the game

Round 5

Work with PO to ensure PB is prioritized, up to date and remains a manageable size

Include stakeholders in updating/creating product vision and strategy

Establish Sprint Goal

Create a public impediments or slowdown list

BDD – Consider moving to round 6

Round 6 Go Live Soon -

Overtime

Improve Forecasting – move to an earlier round since its impact is mainly preventative *Reduces stakeholder interruption gremlin*

Sprint Backlog or Kanban board – help the team feel they own it. The intention being the more attuned it is to their needs the more they will want to use for themselves.

Work to make Daily Scrum more effective https://agilepainrelief.com/blog/daily-scrum-pain.html

Learning Time/Coding Dojo – can ref <https://agilepainrelief.com/blog/scrummaster-tales-the-team-learn-how-to-learn.html> Story can be used for descriptive text sample as well

Mechanics: Idea - Skip retro to get a few more points in the sprint, but reduce/disable improvements in the next sprint

Round 7 – Go live (currently in round 4)

Work to eliminate long ignored technical problems – requires public impediment list

Continuous Delivery

Work on Retrospectives *Will have to make this more concrete*

Round 8 – We’re making money but our customers are demanding – Unplanned work in the Sprint – use: <https://agilepainrelief.com/blog/scrum-by-example-scrum-anti-patterns-unplanned-work-disrupting-the-sprint.html> - *the default built into the Sprint is that team accepted the work – notes as the table above*

consider – affects capacity that sprint – also user story success reduced by 20% that round. Mitigation – Strategy or Backlog refinement?

Improve DOD - …? DevOps – popular but not enough meaning

Mob Programming

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.

Once we hit the final round – the game calcs another 2-3 rounds to calc final effects

Round N introduce handling of Dependancies

Gremlins as they currently exist:

'GREMLIN\_MANAGEMENT\_YELLS'

  | 'GREMLIN\_EMERGENCY\_ON\_OTHER\_TEAM'

  | 'GREMLIN\_NOT\_PULLING\_THEIR\_WEIGHT'

  | 'GREMLIN\_NOT\_AT\_DAILY\_SCRUM';

Interruptions From stakeholders – mitigate with including them in strategic work, also forecasting

Messy Code Found *Avoided by TDD*

Product Backlog a Mess – *Limit Backlog size avoids this*

Dissatisfied Users

PBI Not clear – *Backlog Refinement avoids*

Work not from the Sprint Backlog *– Sprint Goal mitigates to a point*

Quality Issues

Integration Problems with another team

Computer SDD Failure or Other technical problem

*Waiting for another team*

*Search what goes wrong in a Scrum team*

Dependency – another team injected work in your product backlog that harmed you

Poor Communication

Unnecessary Meetings occupying up 25% of the Sprint

Skip a Retro

*Should any of these be builtin as round events?*

Action Ideas:

* All action code should be checked to see what types of action they affect

Put Agile Engineering practices picture on the site. Link to from game. Give its own link URL. Promise a clickable one will come later

Celebrate successes within the team

Collective Code Ownership

10 minute build and test

Firefighter award? Under a name that intrigues and catches attention?

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

Indecisive PO

Gremlins that could be avoided by Vision/Strategy and PB Refinement – one example unhappy customers or stakeholders.

Gremlin for Team Member who wants to be hero or work alone.

Gremlin for Team Member as a Bottleneck.

Skim SBE stories for more Gremlin ideas

Pair Programing would help both Emergency on other team and Team member leaves org.

Notes for game rules:

* More emphasis on resilience

**Distributed Teams**

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.

Mechanics notes:

* Missing the per round and end of game action notes
* Each action and action gremlin should have a field for people to click to learn more
* Could Gremlins document in the text avoidance of problems?

Missing – round descriptions for a company that has a live product.

## 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.