Force of nature **Discard any 2 cards** **Explanation** The team go out for a long lunch and have many philosophical discussions that lead to surprising changes. Discard any two cards from your hand now, or just throw this one back.	**Force of nature** **Discard any 2 cards** **Explanation** A senior executive takes an interest in the team's progress. Discard any two cards from your hand now, or just throw this one back.	**Force of nature** **Discard any 2 cards** **Explanation** Key team members head off to a conference and return with new ideas. Discard any two cards from your hand now, or just throw this one back.	**Force of nature** **Discard any 2 cards** **Explanation** A new project manager joins the team and makes some changes. Discard any two cards from your hand now, or just throw this one back.
Restructure **Swap 2 cards with any other player** **Explanation** The teams face a restructure. Give any 2 of your cards to a player of your choice and take 2 random cards from them in return. This card must be played immediately	**Restructure** **Swap 2 cards with any other player** **Explanation** The teams face a restructure. Give any 2 of your cards to a player of your choice and take 2 random cards from them in return. This card must be played immediately	**Restructure** **Swap 1 card with any other player** **Explanation** The teams face a restructure. Give any 1 of your cards to a player of your choice and take1 random card from them in return. This card must be played immediately	**Restructure** **Swap 1 card with any other player** **Explanation** The teams face a restructure. Give any 1 of your cards to a player of your choice and take 1 random card from them in return. This card must be played immediately

Agile coach **+3 Points** **Explanation** The team gain access to an experienced agile veteran. Having the coach there really helps the team understand the impact of the decisions they make while experimenting with new approaches.	**Retrospectives** **+3 Points** **Explanation** The team make mistakes and also find better ways of doing things as they go along. They regularly discuss and implement both trivial and significant changes.	**Different strengths** **+3 Points** **Explanation** Rather than just adopt the technical practices the team look at their own strengths, weaknesses and communication styles. They learn to build on each individuals strengths and also to deal with all the other annoying people in the team.	**Visible architecture** **+3 Points** **Explanation** The team continuously review and question the architecture of the system as it evolves. Seeing the impact of their decisions, combined with detailed analysis where it is needed leads to a better understood and better built system.
Agile zealot **-3 Points** **Explanation** The team thought they were getting a new coach, but instead they got a zealot. The zealot insists on everything being done in accordance with the One True Path of Agile, significantly distracting the team from learning and adding value.	**Code optimiser** **-3 Points** **Explanation** What began as a good idea has been corrupted for evil. Rather than solving customer problems the team are using customers to speed up the writing of beautiful code.	**Non-non-functional** **-3 Points** **Explanation** In their race to embrace stories and other exciting practices, the team completely forget about non-functional requirements. They are building worse products a lot more efficiently now.	**Random architecture** **-3 Points** **Explanation** The team adopt the mantra of “no upfront architecture”. Fortunately the system evolves its own architecture automatically as it evolves. Unfortunately the architecture that evolves without guidance is not pretty.

Decision rights **+2 Points** **Explanation** The business representatives know what decisions they should make and when to check with others This and the fact that the team know where they can innovate and where they need to stick to the plan is really speeding up development.	**Stand ups** **+2 Points** **Explanation** The team are having a quick meeting every day to see who needs help and who is travelling well. The meetings cost the team 15 minutes per day but save the team an hour a day in avoiding confusion.	**Automated testing** **+2 Points** **Explanation** The team are sick of all the manual testing they have to do. So the team work on a joint test strategy and find opportunities to automate a lot of repetitive testing.	**Story walls** **+2 Points** **Explanation** The team have a visual understanding of how work is progressing and where the bottlenecks are. Team members are now able to focus more of their time on work that is on the critical path.
Accommodations **-2 Points** **Explanation** Even though the team are replacing old practices with new ones, they need to keep performing a lot of the old tasks. But many of the old tasks only there to accommodate the old way of working. A lot of time is being lost.	**1001 status updates** **-2 Points** **Explanation** The project office want the team to keep reporting along the lines of the "standard" projects in the company. This means that the duplicating all their agile reports and having to spend even more time translating them into "Waterfallian".	**Robots turn evil** **-2 Points** **Explanation** The team thought automated testing would help and started using machines to do their testing. But the automated tests are bug ridden and maintaining them is turning out to be harder than manual testing would have been,	**Too many meetings** **-2 Points** **Explanation** The team were glad to see the end of all the bureaucratic overhead of their old approaches. But that was before they became stuck in 4 hours of meetings every day. Now they can't get any work done.

Test Driven Development **+2 Points** **Explanation** The developers adopt Test Driven Development. They are writing automated acceptance tests as they develop their code and the team is producing better quality software.	**Given When Then** **+2 Points** **Explanation** Rather than just provide stories or requirements to the developers, the team are define acceptance tests for each story. The tests are written in the form: GIVEN (an existing state) WHEN (something happens) THEN (we expect this outcome).	**Continuous Integration** **+2 Points** **Explanation** In the old days the team would wait for weeks before integrating their code. Now they integrate their code as they go and quality is improving all the time.	**Automated migration** **+2 Points** **Explanation** The team create an automated build to migrate code between environments. As a result they reduce the time and errors involved in migrations.
Absentee sponsor **-2 Points** **Explanation** The sponsor is too busy to spend time understanding the project so the team are making many decisions on the sponsor's behalf.	**Forgotten infrastructure** **-2 Points** **Explanation** The team assumed that because they are agile, they will be able to order infrastructure in the same iteration they need it. Unfortunately there is a 6 week lead time and they are very disappointed.	**Forgotten handover** **-2 Points** **Explanation** The team were so busy building new things they forgot about how to support what they are building. Now all they can hand to the production support team is a pile of story cards and a "get well soon" card.	**Forgotten knowledge** **-2 Points** **Explanation** The team have dispensed with all unnecessary documentation. Unfortunately some of the documentation contained information that is now lost.

Agile training **+2 Points** **Explanation** The team attend agile training courses that give them a more holistic understanding of agile.	**Testing with the customer** **+2 Points** **Explanation** Frequent interaction with customers is allowing bugs to be uncovered far earlier. A surprising benefit is that the team are also learning which bells and whistles can be left out of the solution.	**Just the basics** **+2 Points** **Explanation** The increased scrutiny of work as it progresses is encouraging the team to avoid complexity. The team are really focusing on the core components of work rather than the waste that used to occur.	**Obviously that is not what I meant** **+2 Points** **Explanation** The business are not getting what they asked for anymore, they are getting what they want. Seeing what the team is doing is allowing the customer to identify their assumptions and clarify their real needs..
Cowboy fever **-2 Points** **Explanation** The team see agile as a way to escape all the chains and shackles of the companies procedures so they can do anything they want. Unfortunately it seems the one thing they don't want to do is to build robust code that meets real customer needs.	**Not my problem** **-2 Points** **Explanation** People mistook shared accountability for "no-accountability". Ambiguity around roles and responsibilities means that nobody is making any of the hard decisions.	**Scope randomiser** **-2 Points** **Explanation** The business have been liberated from the need to think and plan. Stakeholders can turn up whenever they want and ask for anything. But the team is so busy juggling they are not delivering anything of value.	**All trees, no forrest** **-2 Points** **Explanation** The team are flat out every week delivering new features. But nobody has the time or opportunity to see the big picture. So the team is travelling in a random direction.

All else being equal **+1 point** **Explanation** The team are able to actively measure their velocity. While the team's measurement is not precise, understanding how much work gets done each week is really helping the team predict how much will get done in the coming weeks.	**Early failure** **+1 point** **Explanation** The team realise their whole project was a mistake. But rather than being doomed to see it through they abort early. The team can now focus on adding value.	**Bad estimates** **+1 point** **Explanation** The team's estimates are way out every time they interact with the new "Invergolator". Looking into the errors reveals some hidden issues that the team needs to deal with. The team deal with the issues and find considerable improvements.	**A problem shared** **+1 point** **Explanation** The team adopt pair programming, which means two people deal with each problem together. The team start doing far better work now that they are always bouncing their ideas off others.
No boundaries **-1 point** **Explanation** The team have been empowered to be agile, but have no scope and no direction. It seems like "agile" really means "having no idea what is expected but needing to do it faster".	**Micro manager** **-1 point** **Explanation** The agile practices turned out to be no protection from micromanagement. The boss is demanding daily and hourly updates on everything, without sharing what others are up to.	**Money, what money?** **-1 point** **Explanation** The team are inviting the sponsor to a lot of meetings. But when the sponsor asks for an update on the budget she is told money is not part of the agile vocabulary. Now she is looking a little nervous.	**Access minefield** **-1 point** **Explanation** In their haste and excitement to deliver business value the team don't have time to build robust databases. But every release seems to cause several existing access databases and macros to explode.

Tomorrow is better than yesterday

+1 point

Explanation

The team are discovering existing defects and technical debt (unnecessary complexity) in the existing system.

Stopping to fix things as they find them is slowing the team down but also making life easier over time.

It's a risky business

+1 point

Explanation

The team are talking about risk every week, making it easier to monitor and deal with risks as they emerge or recede.

The team is too big

+1 point

Explanation

The overhead of communicating on an agile project is causing issues … and the business doesn't have the time to dedicate to so much work.

This is forcing the team to really focus on the must haves and to break projects down more.

Process integration

+1 point

Explanation

Delivering completed features every iteration is allowing the customer to test and understand the process impact.

This has led to a lot of small changes in both their processes and the system design.

Suspicious pace

-1 point

Explanation

The team estimated on what they thought was a sustainable pace.

But the deadline got squashed and the team started to get a bit behind.

So everyone is working some extra hours until the team catches up. But when will things level out?

No time to think

-1 point

Explanation

The story wall, the stand-ups and the iteration planning meetings are giving great visibility of what to do next.

Unfortunately everyone is so busy as a result that nobody has time to stick their head up to see where the team is heading.

Ego factoring

-1 point

Explanation

The frequent reviews by both developers and business customers is allowing a lot of feedback, which is leading to rework.

But rather than adding value, the changes are really just satisfying the egos of some key players.

Panic waterfall

-1 point

Explanation

The developers feel that they still need complete specifications to build from,

But now the business analysts have only one week to produce every document. They are starting to look frayed and distressed.

Old stones resurface **+1 point** **Explanation** Adopting new approaches has uncovered some existing issues and conflicts . This forces the team to reflect on the issues, resulting in gradual improvements in spite of the frustration.	**Developer testers** **+1 point** **Explanation** New practices enable the developers to better understand what their code needs to accomplish. In response they are able to better able to test their own code, leading to much better quality.	**People talk** **+1 point** **Explanation** Agile has resulted in people spending more time talking to each other. The team discover their work was more interconnected than they realised and start to stumble on better ways of working.	**Agile conference** **+1 point** **Explanation** Some of the team attend an agile conference. The team discuss some new ideas in their next retrospective and find some simple improvements.
Unmentionables **-1 point** **Explanation** The adoption of new practices is exposing existing tensions and issues. Fortunately the team can sweep them under the carpet. Unfortunately doing so is holding them back.	**Back to the future** **-1 point** **Explanation** Agile is being used as an opportunity for others to roll out their old initiatives again. The team are distracted by having to adopt tools and processes that have very little to do with their own needs.	**New toys** **-1 point** **Explanation** People are buying lots of new tools to make the team more agile. Unfortunately, the new tools are getting in the way of the team doing their work.	**Not enough testers** **-1 point** **Explanation** The team are writing lots of good code, but there doesn't seem to be enough time to test it all. They advertise for more agile testers but can't seem to find any.

Users are strange beasts **+1 point** **Explanation** Showing the evolving solution to the users every iteration is leading to a lot of surprises. The users just don't seem to behave the way they were expected to. So the system is evolving to work for the users the way they will really use it.	**The team is learning** **+1 point** **Explanation** Everyone in the team is the team is learning more about each others' expectations, style and roles. But surprisingly they also seem to be learning more about how to do their own role through the process.	**The mathematics of traffic lights** **+1 point** **Explanation** Work flows through the team like traffic flows through a city. Some is going fast, some is stuck in traffic and some is completely lost. So the team's increased focus on traffic flows is making everything flow much better.	**The customer is the constraint** **+1 point** **Explanation** The team would be able to go faster if the paying customer had more time. In other words the team is now going exactly as fast as the customer can manage.
Inevitable complexity **-1 point** **Explanation** As the project continues, the team keeps adding new features. So it seems that work will naturally slow down as the system becomes more complex. But if you think about it, that means tomorrow will be worse than today.	**Empires in danger** **-1 point** **Explanation** New practices are bringing changes to the organisation and as a result some empires are starting to shrink. So managers are racing to lock down knowledge and to impose additional checks to stop their empires shrinking.	**Single point standoff** **-1 point** **Explanation** Going agile has highlighted the fact that some key people are the only holders of critical knowledge. But they are feeling threatened by the idea of handing over their knowledge and so they are keeping a tight rein on things to prove how valuable they are.	**No time for testing** **-1 point** **Explanation** Having to deliver every two weeks means having to retest every two weeks. There simply isn't time to do that in the real world so the team are going faster by not testing. But bad karma is brewing in the evolving system.

Motivating astronauts **+1 point** **Explanation** A lot of companies spend a lot of time worrying about how to motivate people. But when people see the impact of their contribution to an emerging solution they seem to become motivated automatically.	**Constraints suck** **+1 point** **Explanation** Some really annoying constraints are stopping the team from being more agile… Which means the team is doing the best job possible within those constraints. Now the team can communicate or even deal with the constraints	**No more agile** **+1 point** **Explanation** The team have gotten using the agile techniques. Strangely though they have stopped talking about how to be more agile. They are now just focusing on how to do things better.	**Enough is enough** **+1 point** **Explanation** An agile project only delivered around 40% of what it set out to achieve before it got cancelled. The customer is ecstatic because he got the real thing he wanted and is now spending money on his next adventure.
Damn the torpedoes **-1 point** **Explanation** The way the team is working has highlighted a number of issues. But the project is so important that the team decide to ignore the risks and issues and simply steam ahead.	**Velocity by any measure** **-1 point** **Explanation** The team have confused the measure for the goal. Estimates are corrected and work is adjusted to maintain the target velocity.	**Not suitable for children** **-1 point** **Explanation** Agile has been implemented in a way that removes the safety net that used to protect junior team members from making mistakes. The team are now working on such critical and urgent work that only the senior team can get the job done.	**Business too busy** **-1 point** **Explanation** The business stakeholders want to participate but they are too busy at the moment. Fortunately the team can make decisions for the business. Unfortunately the decisions are made in ignorance.