

-1



## Scope creep

Some zombie requirements crept in your user stories while you weren't looking

-1

-2



## Motivation Drop

Motivation is overrated, bark at your developers a bit more to increase productivity

-2



## Definition of undone

So, is this done or what? There is no "well, it depends"

-2



## Murphys-Law

Sometimes the demo gods seem not to be on your side

-2



## Threat Talk

Going all staff sergeant-style on knowledge workers works... sometimes...

-2



## Decaf'

Intern Josh went to buy some coffee and bought some decaf'....

-2



## Ran out of coffee

You ran out of coffee. It's not as bad as replacing the coffee with decaf' (intern Josh did that once), but bad enough

-2



## Unclear stories

At least Socrates and Jon Snow knew that they knew nothing...

-2



## Electrical Powercuts

No electricity, no code.

-1



## AC is broken

Machines and developers both generate a lot of heat. What if the AC broke down in the summer?

-1



## Windows Update

Windows updates always come when you don't expect them to.

-1



## Tab vs Spaces

Another day in paradise. Someone chose to re-ignite the tab vs spaces war.

-1



## Merge chaos

You remained on a branch for too long, merging is a bitch!

-1



## Stress

There's good stress and bad stress. Most often the latter lately.

-1



## Silent refactoring

Who the f\*\*k rewrote my module?

-1



## Ghost PO

When your PO is nowhere to be seen, you do what you can

-2



## Server burned down

What do you mean you have only a backup from last week?

-1



## Fire drill

Everybody out!

-1



## Proxy PO

When your PO is a man-in-the-middle, things tend to take more time

-1



## Technical Debt

You shouldn't have ignored this bad design for so long.

-2



## Patent war

Yep, they patented that idea also

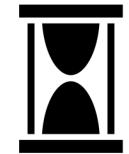
-1



## Horizontal slicing

Let's make that data access layer a team on its own!

-1



## Waiting for delivery

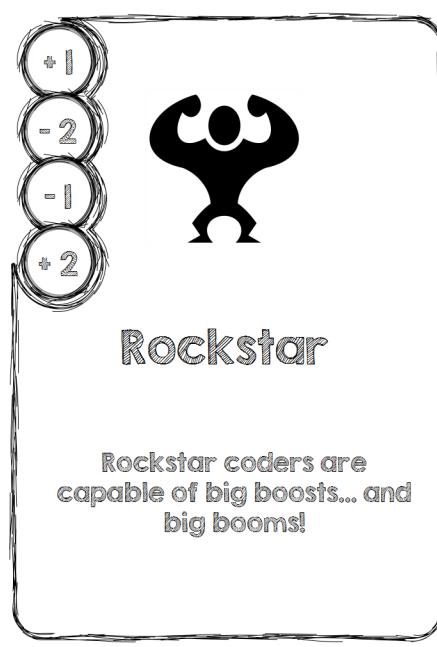
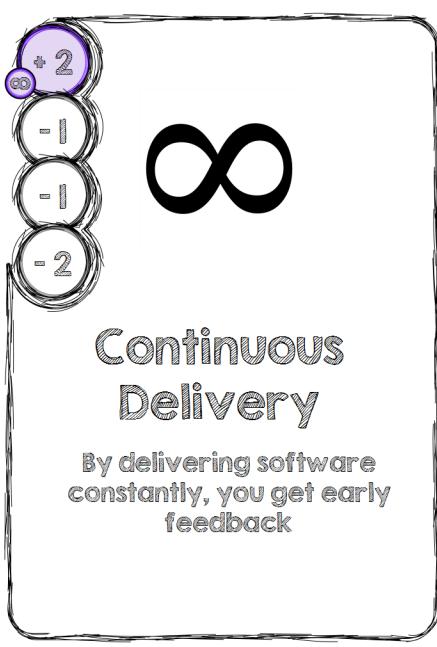
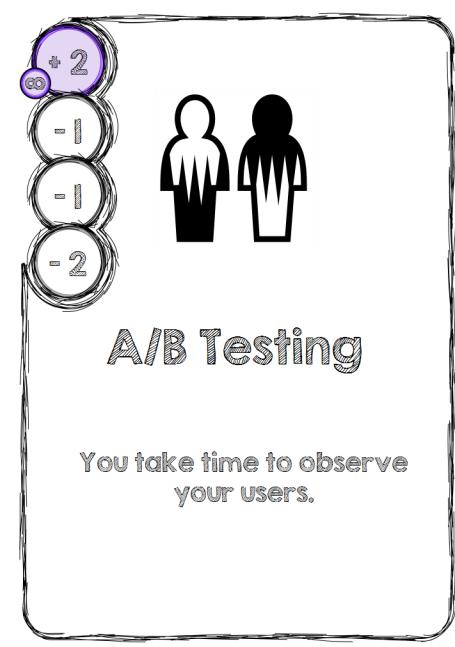
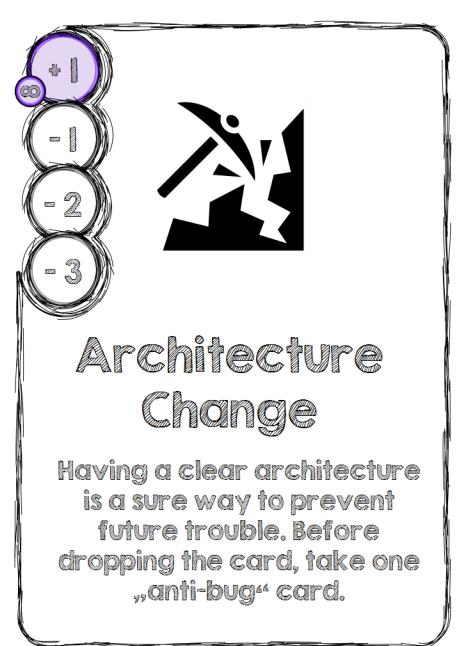
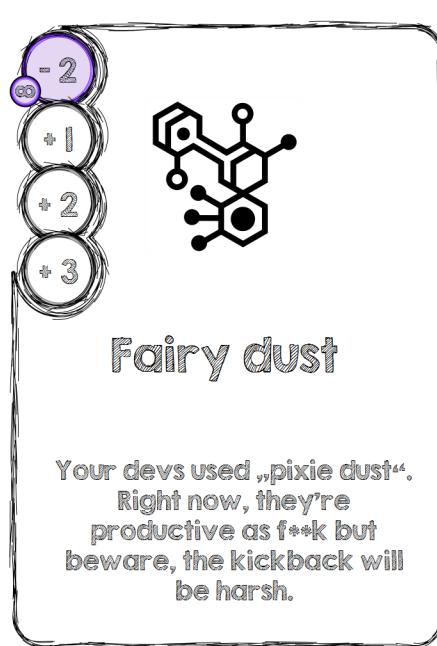
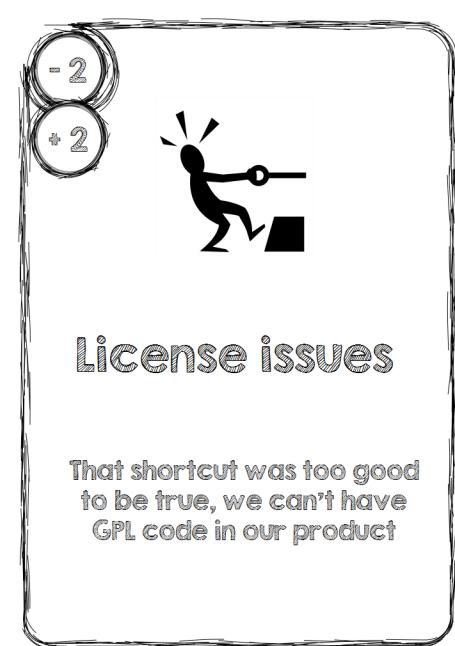
Your code is ready, but team-X isn't.

-1



## Flu

Coming to work sick to show everyone how hardcore you are, seemed like a good idea



-1  
-1



## Premature sales

Someone sold a feature without asking engineering first

+2  
co



## Priorizing

Your product owner is a priority-ninja. You create value faster than Silicon Valley venture capitalists.

+2  
co



## Refinement Meeting

It helps to know where you're going

+2  
co



## Cross functional teams

The team has all the needed skills to create the software, no need to wait for handoffs

+2  
co



## Vertical slices

The team can modify every layer of the software, no need to wait for handoffs

+1  
co



## Feature flags

The team integrates faster and releases to production early and often

+2  
co



## Daily Standup

What did I do yesterday? What will I do today? What is blocking me?

+1  
co



## Client on site

You are getting first hand feedback, the client is pleased

co  
n  
+ 1



## Hands-on

Sometimes, you don't know what you need until you've seen and used it

co  
n



## Scrum of Scrums

You get 1 velocity point for every player - other than yourself - who played an SoS card

co  
n



## Scrum of Scrums

You get 1 velocity point for every player - other than yourself - who played an SoS card

co  
n



## Scrum of Scrums

You get 1 velocity point for every player - other than yourself - who played an SoS card

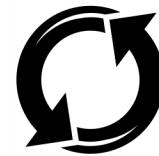
co  
n



## Scrum of Scrums

You get 1 velocity point for every player - other than yourself - who played an SoS card

co  
n  
+ 1



## Continuous Integration

You get instant feedback. Bugs are solved in minutes, not days.

co  
n  
+ 1



## Collocated Teams

Collocation makes communication way easier

co  
n  
+ 1



## Management Champion

You have a champion high-up in the organization. Changes are smoother with pinch of hierarchical help.

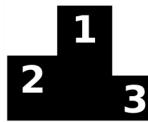
+ 1  
- 1  
co



## Interested management

Recognition from high up is rarely counterproductive

- 1  
co



## Everything is prio 1

You have a beautiful database, a login screen and some printing features. You are missing a product...

- 1  
co



## Virtual board

A virtual Kanban board is great... unless it prevents you from talking to each other

- 2  
co



## Offshoring

Outsourcing for financial reasons has never been a good idea

- 2  
co



## Reporting StoryPoints

Let's use those Story Points to estimate the project's progress and compare teams

- 1  
co



## Virtual PCs

Let's virtualize all the developer's PC and save bazillions on hardware!

- 1  
co



## Micromanager

Who better than the chef knows what's to be done and how? That's why he's the chef in the first place, right?

+ 1  
- 1  
co



## Global code review

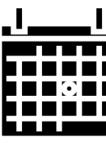
UnexpectedWinException seemed like a good exception name



+1  
0  
-1  
-2

## Mandatory Staffing

Add two members to the team. Tuckman says it's going to be „forming“, „storming“ and „norming“ before „performing“



-2  
0  
+2  
+2

## Work on Saturdays

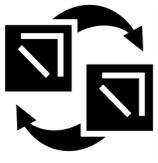
Two weekend days are overrated. Let's work 6 or 7 days and see where that leads us!



-2  
+1  
+2  
+2

## Death March

You now work day and night. But beware, the positive results of the beginning are not to be counted upon in the long term.



+2  
-1  
-2  
-3

## Git-Introduction

Git is like a very sharp saw. It will make wonder in the long run... but only if you keep all your fingers long enough



-2  
+1  
+1  
+2

## Bonus based on Velocity

Let's use those Story Points to give out bonuses, nothing could go wrong



+2  
-1  
0  
+1

## 3 Amigos

Each story is designed by the joint effort of a developer, a tester and the product owner



+1  
-1  
-1  
-1

## Conference Season

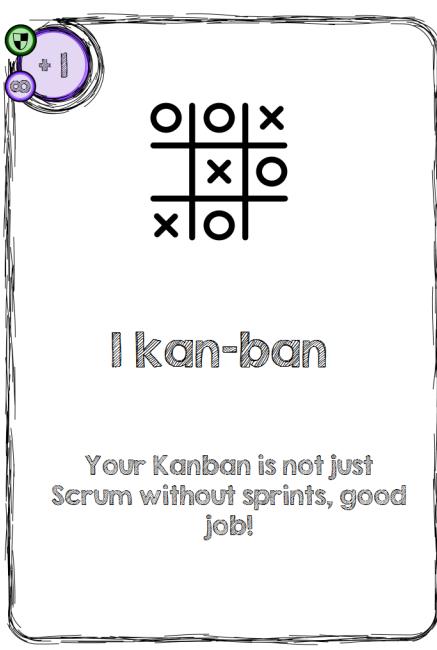
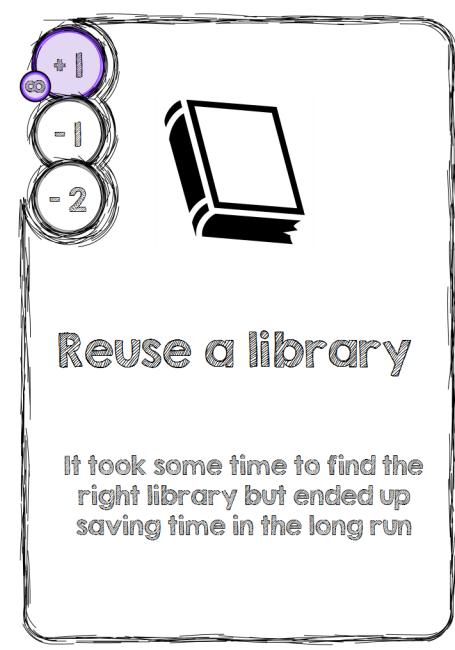
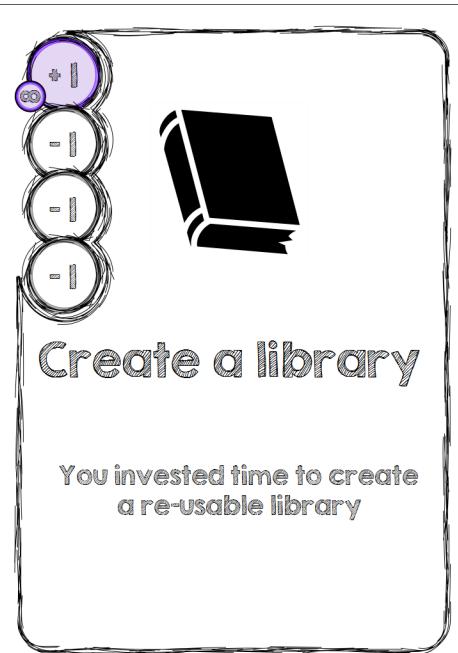
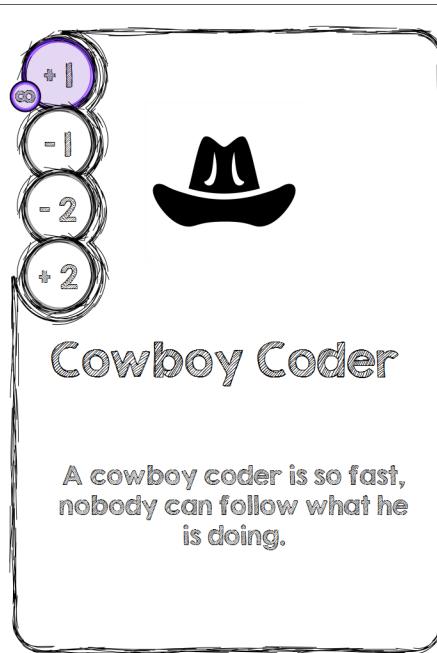
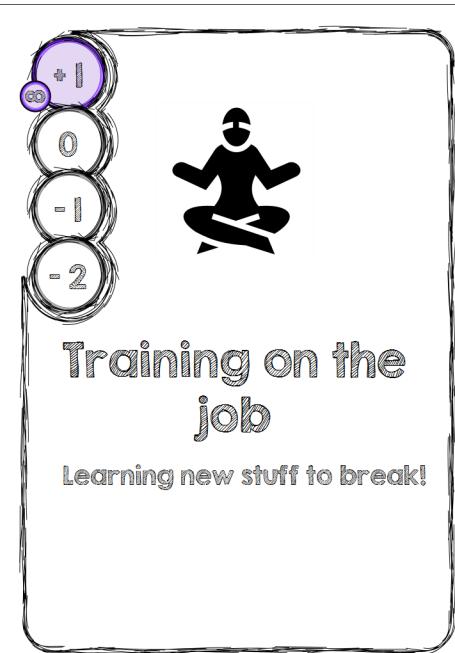
It's conference season. Your team members are who knows where but in front of their keyboards.

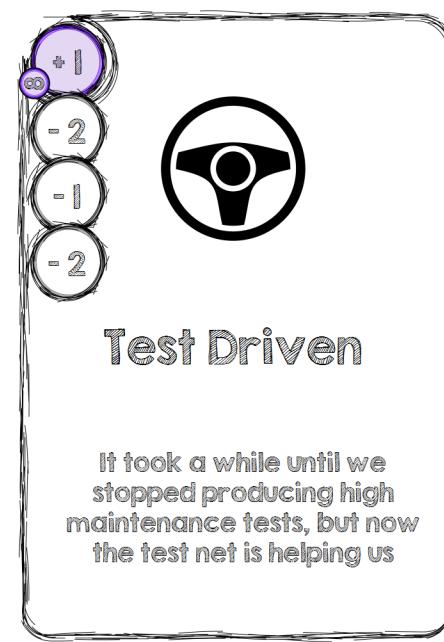
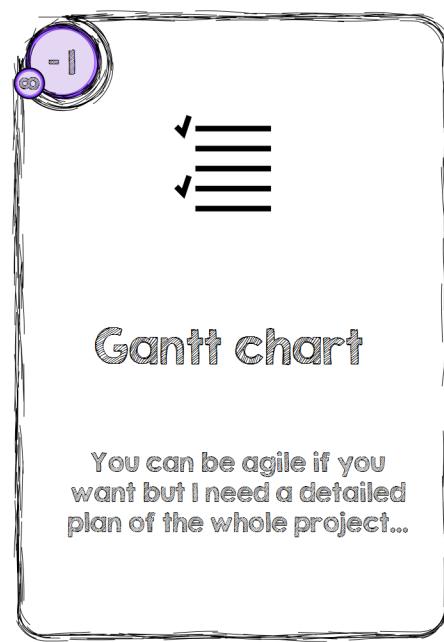
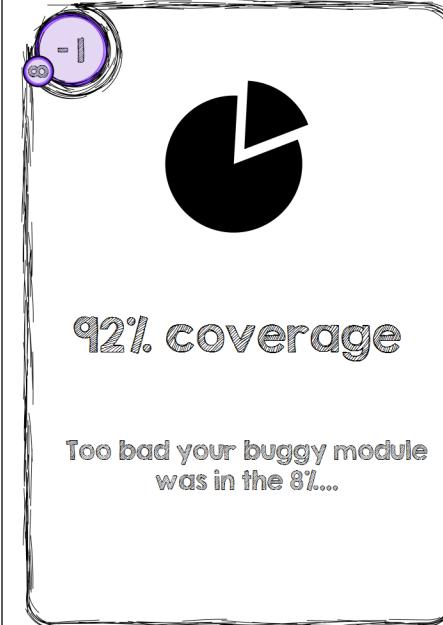
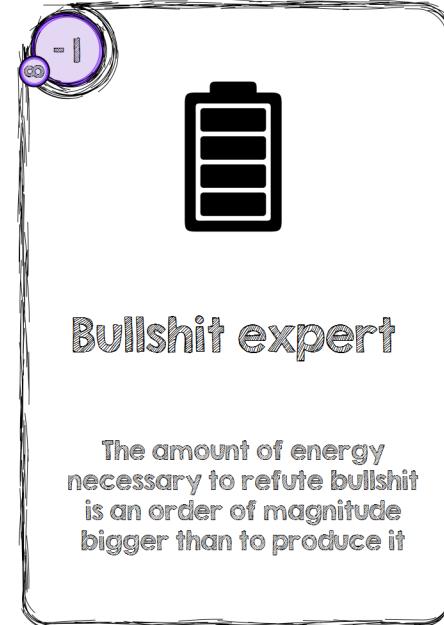
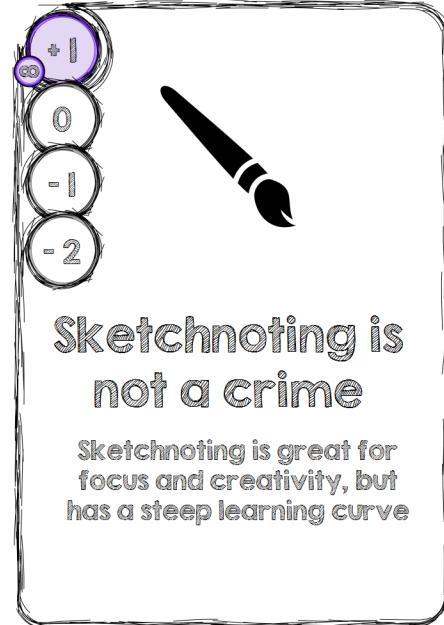
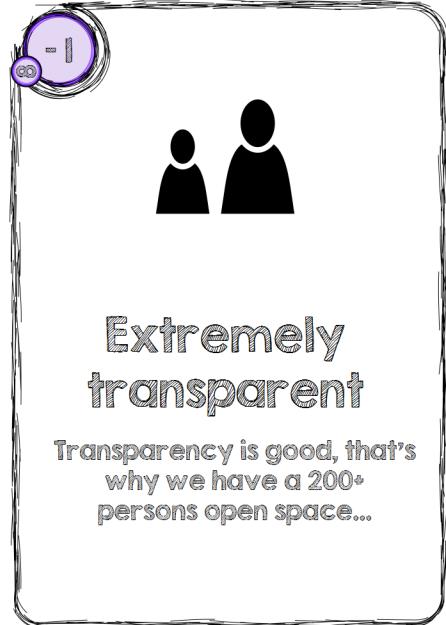


+2  
-1  
-2  
-2

## 20% Innovation Time

20% of your time goes to innovation. It is costly but it will bring a lot to the company in the long term.







## Venn Diagram overuse

It sure is a lot of fun to trace Venn Diagrams to explain everything, but it does not help productivity



## No documentation

Having no documentation is a sure way to have no "comprehensive documentation"



## Agile-cadabra

Renaming 'managers' to 'Scrum Masters' was a brilliant idea!



## Agility assessment

As if agility could be assessed via a spreadsheet...



## Performance Review

It's everyone for himself now



## Shu-Ha-Ri

Your team reached the 'Ha' state, they are now improving on their own!