

Evidence For How To Make Great Games

Actions to take from the "Game Outcomes Project"

Malte Skarupke

July 19th 2017

Schedule:

6pm to 6:30pm: Pizza, beer and chat

← Pizza in the yellow room on the left

6:30pm to 7:30pm: Presentation

7:30pm to 8pm: Q&A and more chat

Overview


- What is the game outcomes project
- Part 1: Do Financial Incentives matter?
- Part 2: What doesn't matter
- Part 3: What maybe helps/hurts a little
- Part 4: What probably helps
- Part 5: What really helps
- Part 6: Top 10 things that help
- Part 7: Practical advice

Game Outcomes Project

Survey to find out what makes great teams so effective



Series of articles on Gamasutra

I think everyone in game development should read these

 Blogs

The Game Outcomes Project, Part 1: The Best and the Rest

by Paul Tozour on 12/16/14 03:05:00 pm Expert Blogger Featured Post

21 comments   Share  

The following blog post, unless otherwise noted, was written by a member of Gamasutra's community. The thoughts and opinions expressed are those of the writer and not Gamasutra or its parent company.

.....

This article is the first in a 5-part series.

- **Part 1: The Best and the Rest** is available here: ([Gamasutra](#)) ([BlogSpot](#)) ([in Chinese](#))
- **Part 2: Building Effective Teams** is available here: ([Gamasutra](#)) ([BlogSpot](#)) ([in Chinese](#))
- **Part 3: Game Development Factors** is available here: ([Gamasutra](#)) ([BlogSpot](#)) ([in Chinese](#))
- **Part 4: Crunch Makes Games Worse** is available here: ([Gamasutra](#)) ([BlogSpot](#)) ([in Chinese](#))
- **Part 5: What Great Teams Do** is available here: ([Gamasutra](#)) ([in Chinese](#))
- For extended notes on our survey methodology, see our [Methodology blog page](#).
- Our raw survey data (minus confidential info) is now available [here](#) if you'd like to verify our results or perform your own analysis.

The Game Outcomes Project team includes Paul Tozour, David Wegbreit, Lucien Parsons, Zhenghua "Z" Yang, NDark Teng, Eric Byron, Julianna Pillemer, Ben Weber, and Karen Buro.

[Editor's Note: The results of the Game Outcomes Project will be [addressed at length during GDC 2016 as part of Paul Tozour's talk](#) on "The Game Outcomes Project: How Teamwork, Leadership, and Culture Drive Results."]

The Game Outcomes Project, Part 1: The Best and the Rest

What makes the best teams so effective?

Veteran developers who have worked on many different teams often remark that they see vast cultural differences between them. Some teams seem to run like clockwork. and are able to craft



Methodology

- Anonymous survey asking game developer about their project
- Data from 272 completed projects
- Statements could be answered on a 7 point scale from “Agree Completely” to “Disagree Completely”
- Example statements:
 - The team composition didn’t change during the course of the project (other than growing when needed)
 - It was difficult to get or give honest feedback without feelings getting hurt
 - The project leads frequently called out when team members performed well

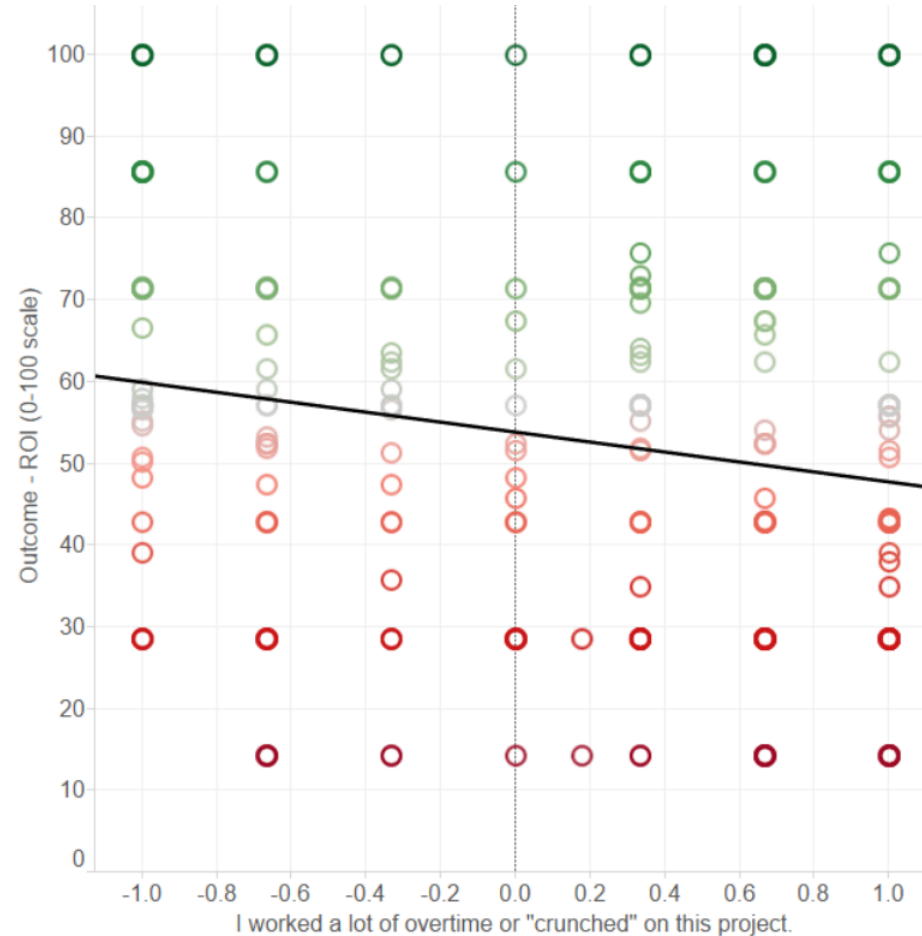
Methodology

- When taking the survey, people self rated their success in four categories
 - Was the project delayed
 - How much money did the game make
 - What was the Metacritic score of the game
 - Were the internal goals of the team met / was the team happy with the game?
- “Aggregate Score” averages the above four

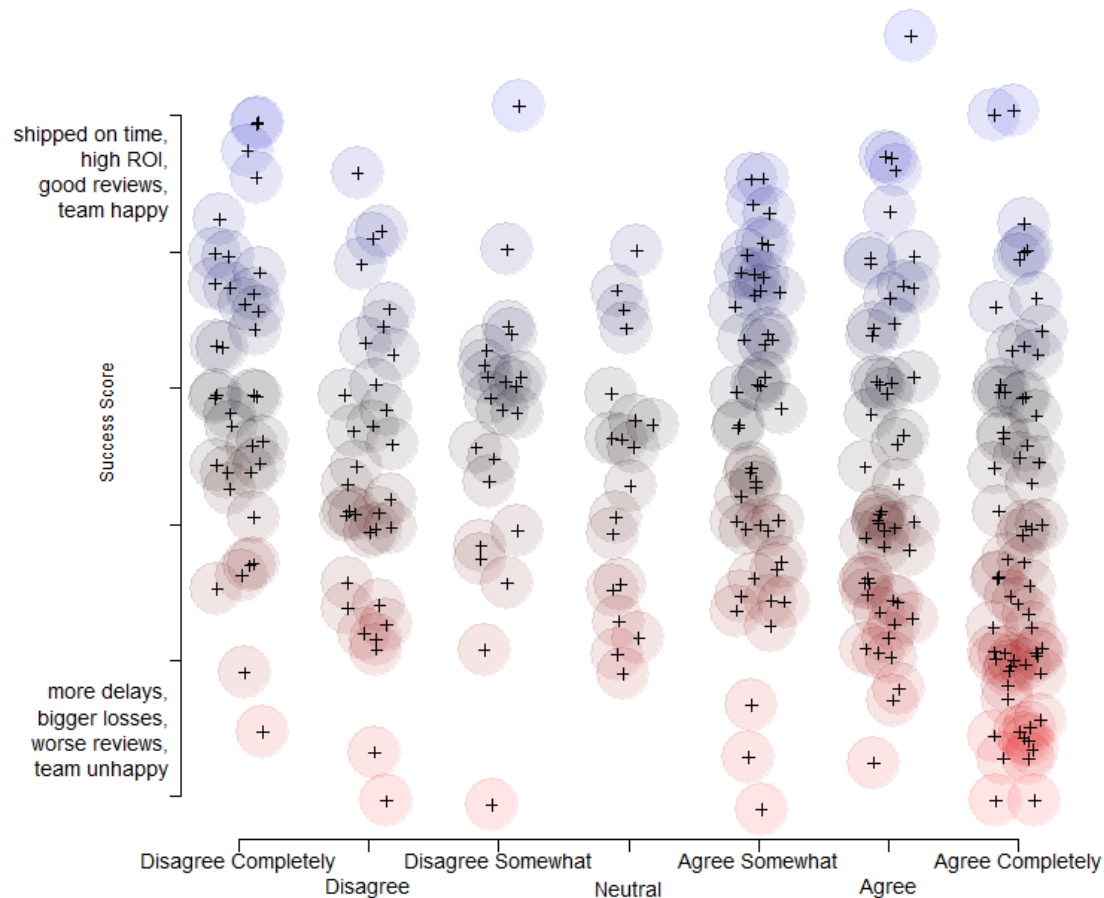
Data Display Methodology

	Correlation with:					Category Score
	Project Delays	\$ ROI	MetaCritic	Internal Goals	Aggregate Outcome	
It was safe to take a risk on this team, and stick your neck out to say something that needed to be said.	0.24	0.33	0.32	0.47	0.43	0.43
It was difficult to ask other members of the team for help.	-0.17	-0.15	Not S.S.	-0.23	-0.21	
No one on this team would deliberately act in a way that undermines my efforts.	0.17	0.18	Not S.S.	0.23	0.22	

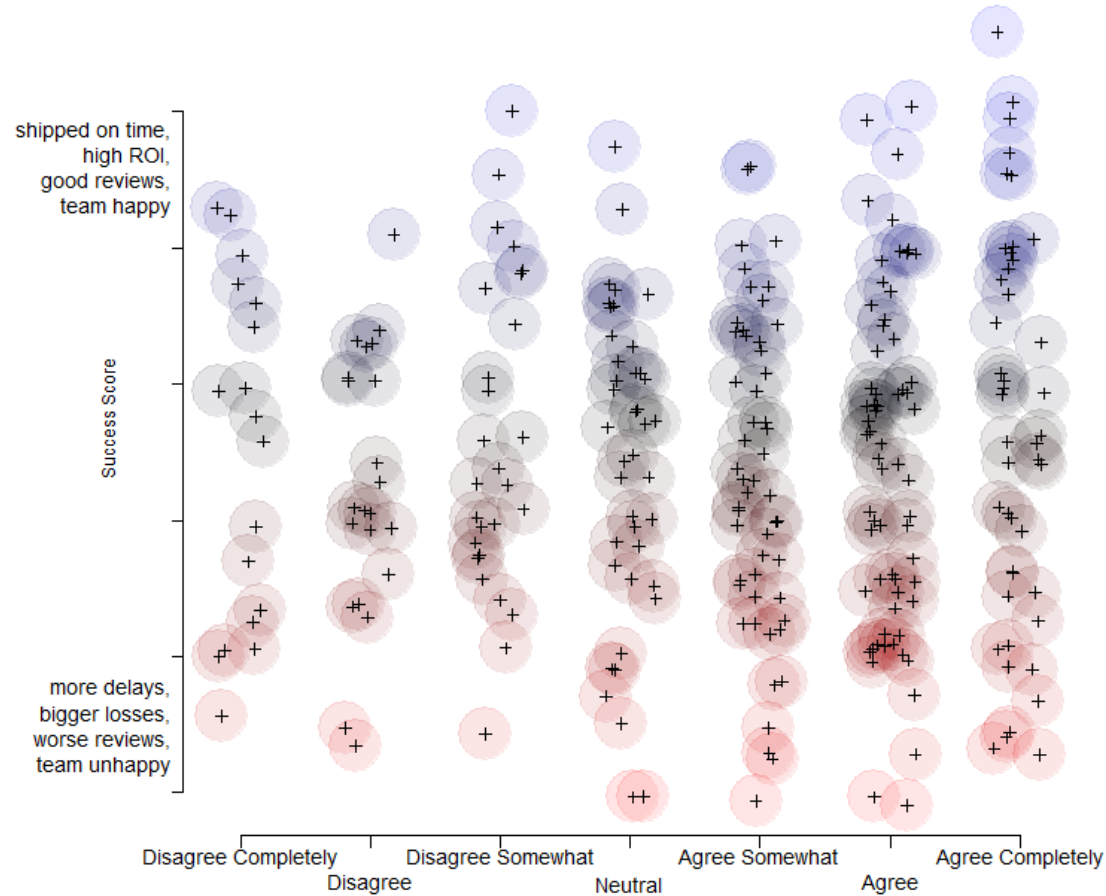
Data Display Methodology



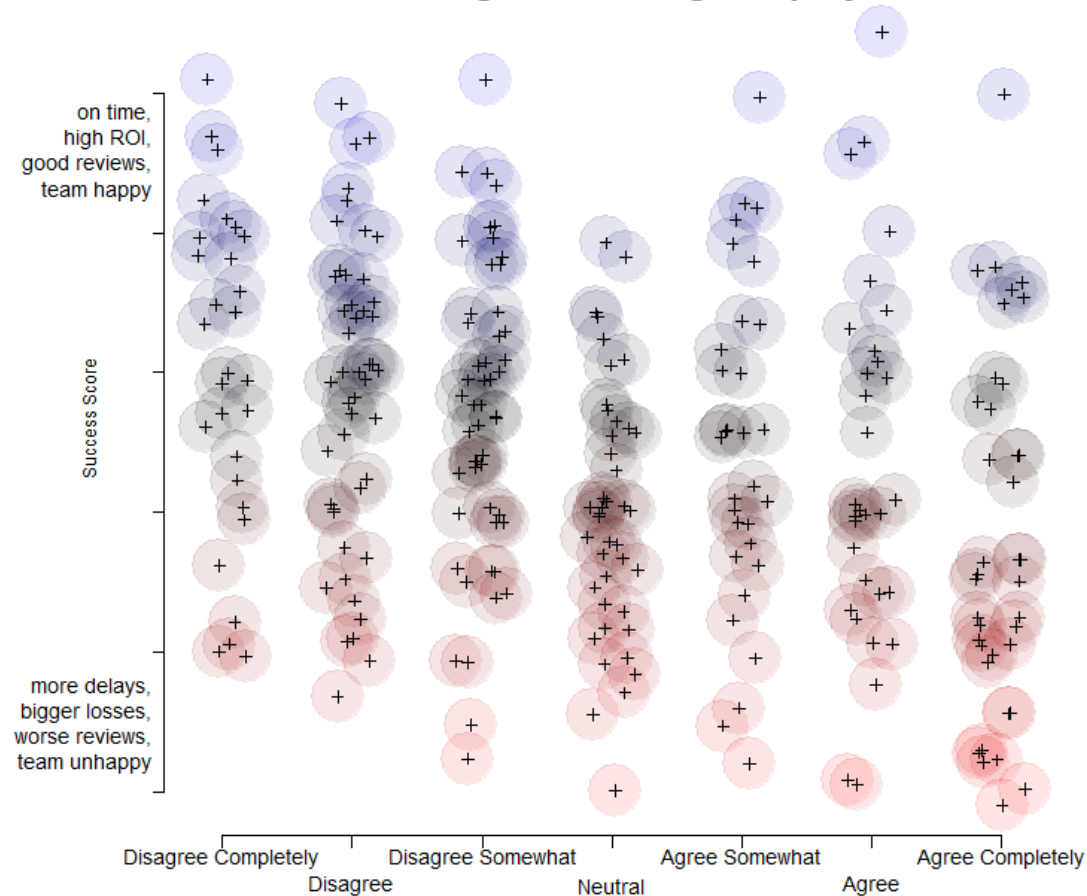
I worked a lot of overtime or "crunched" on this project.

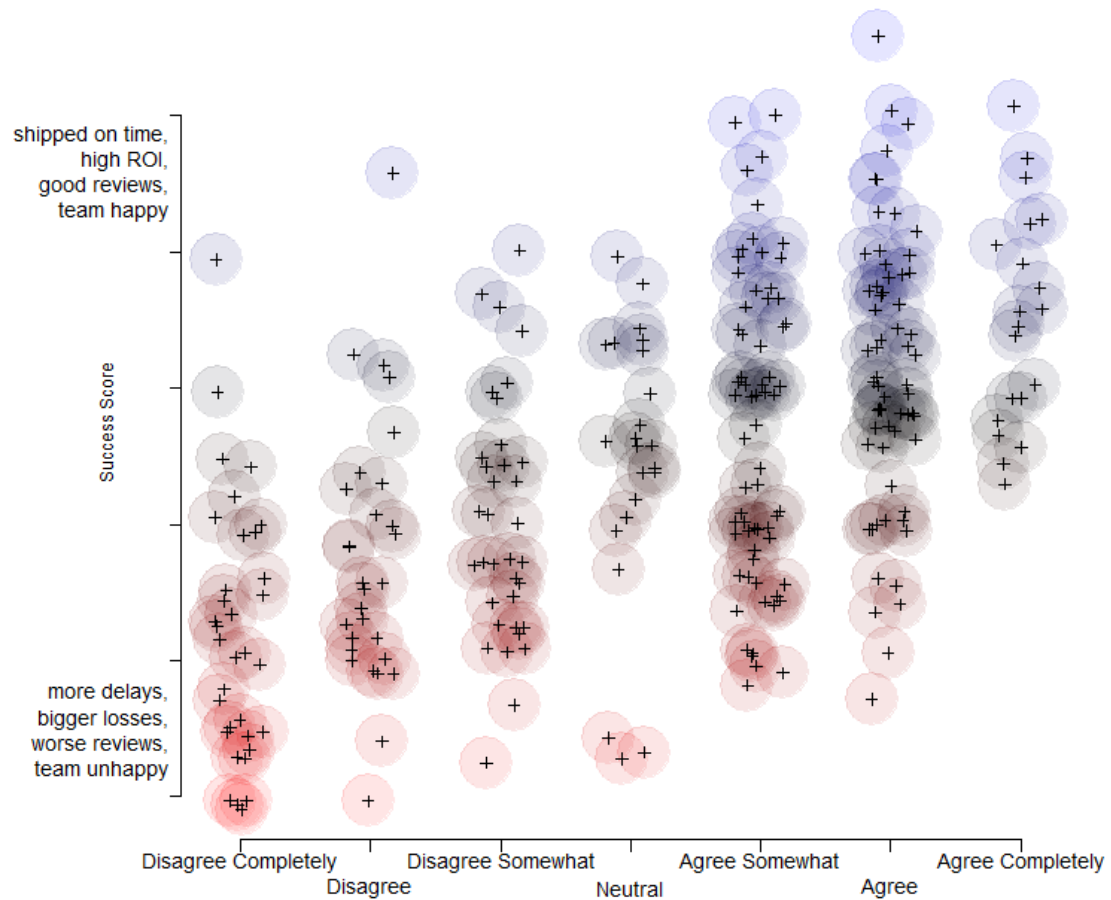


The company's culture preferred face-to-face communication over e-mail.



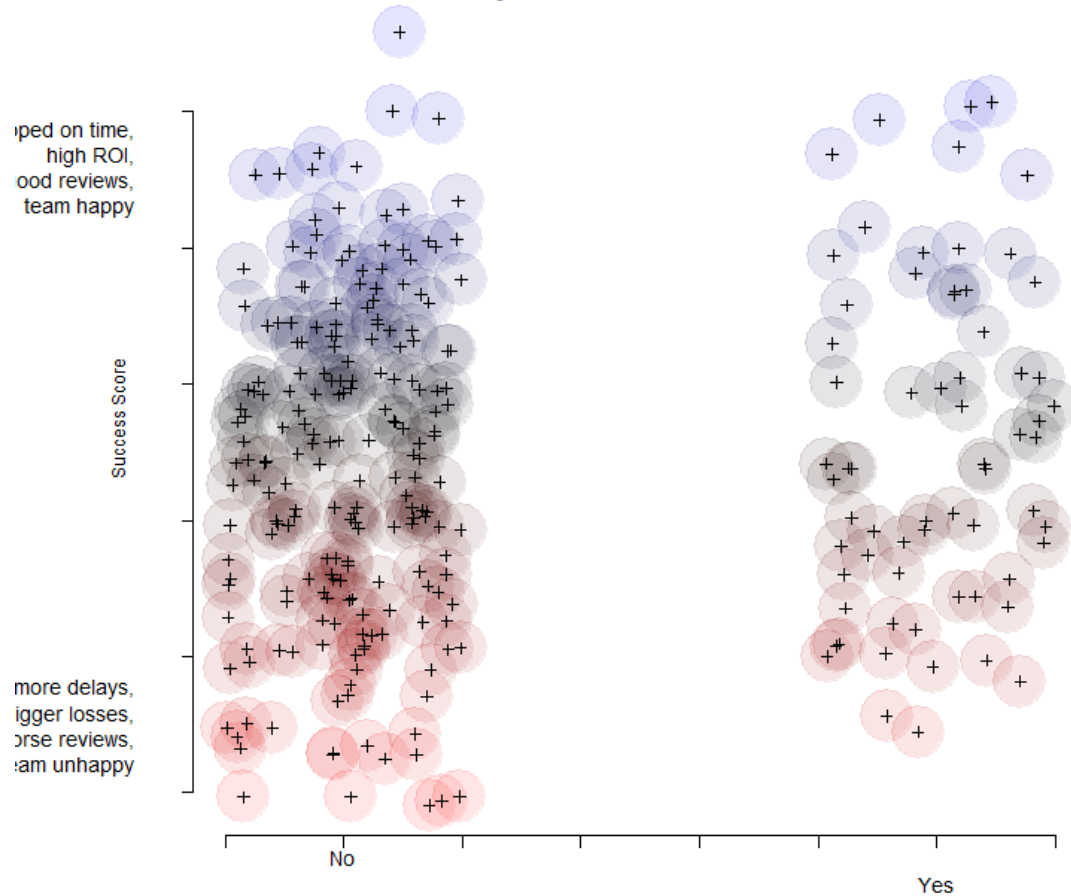
Some team members put their own careers above the collective goals of the game project.



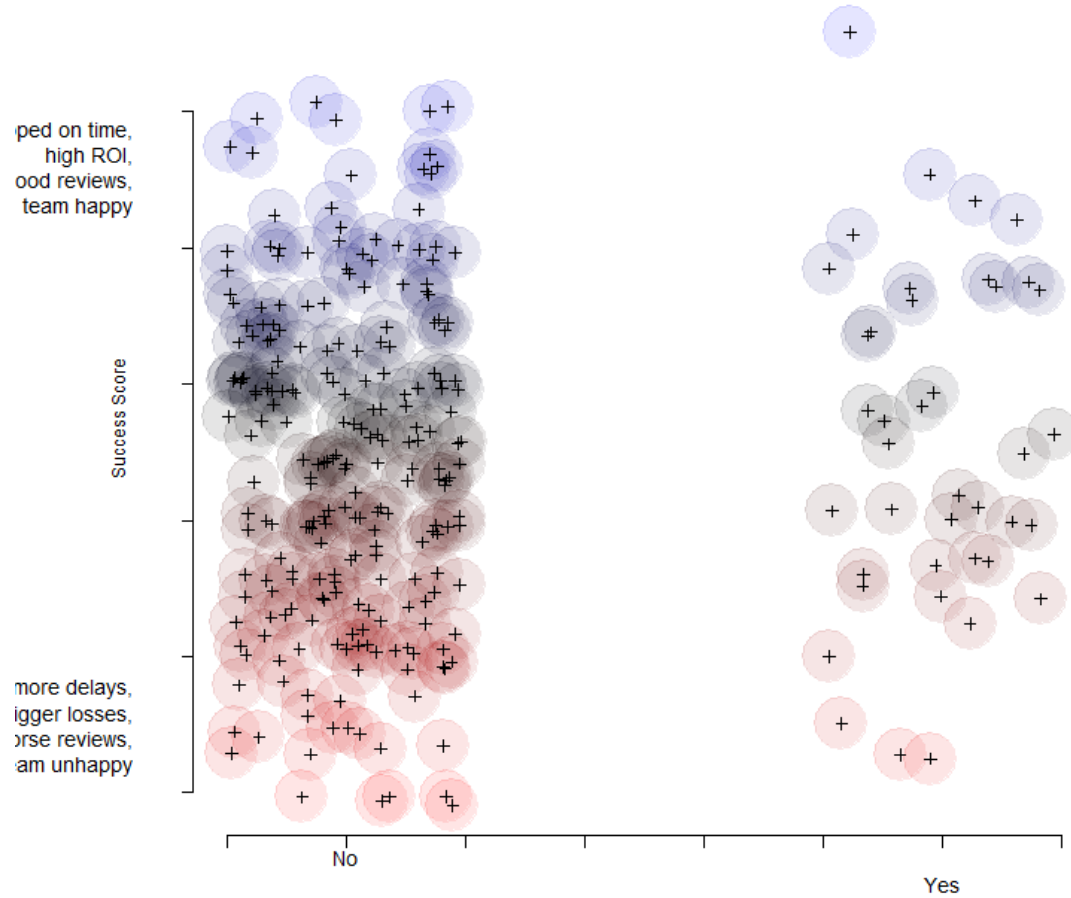


Part 1: Financial Incentives

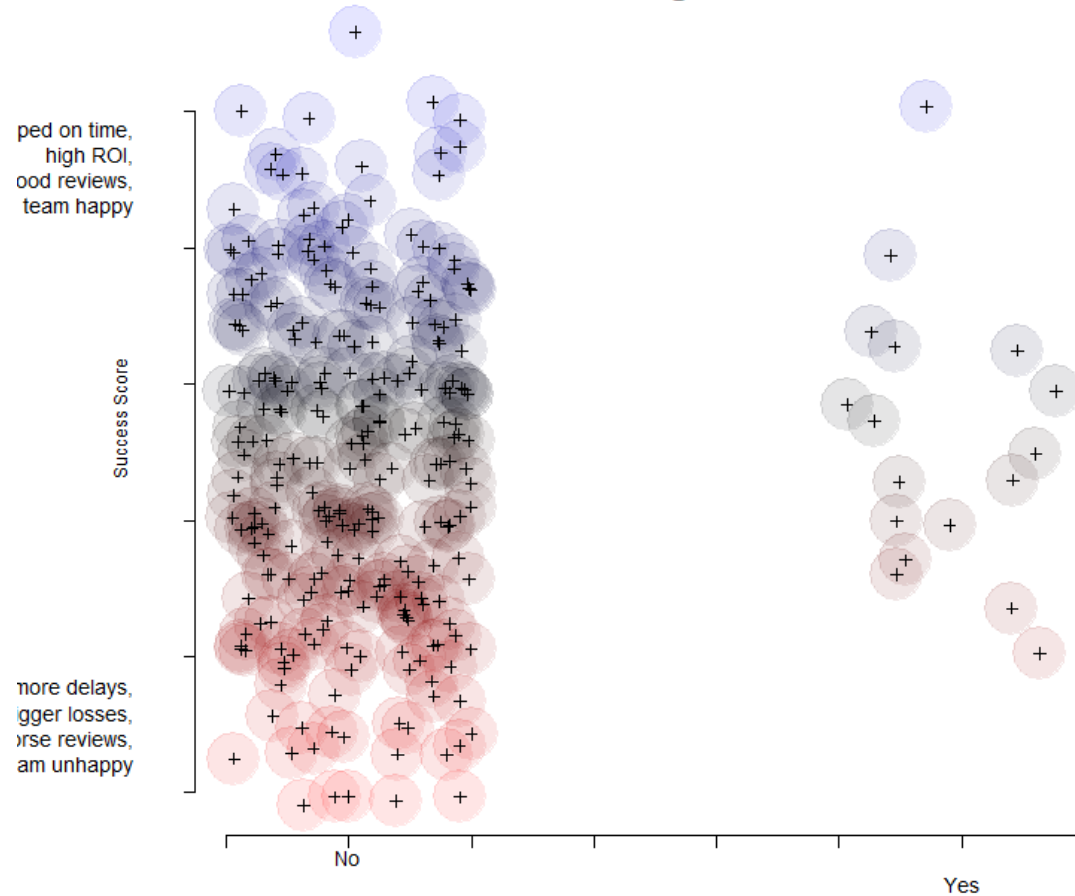
Royalties or revenue sharing tied to the product's financial performance or total sales



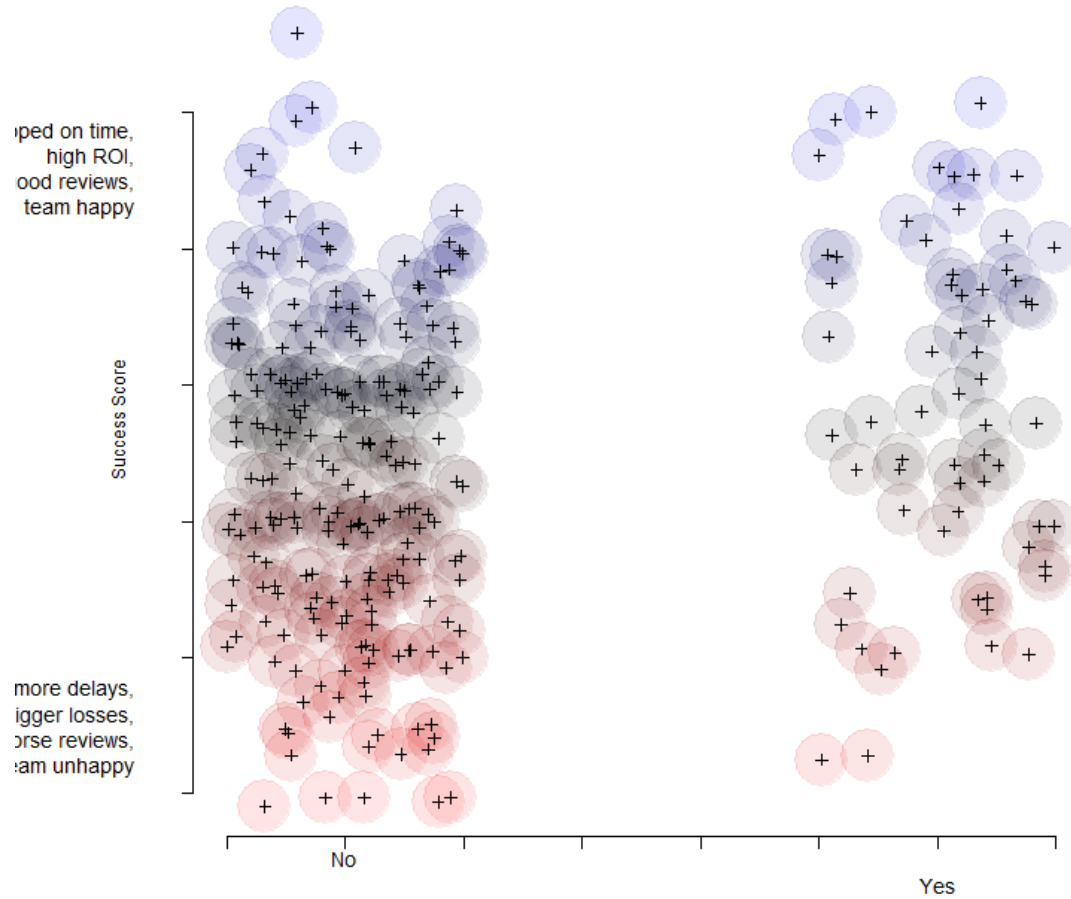
Bonuses tied to team performance, as evaluated internally



Bonuses tied to the game's critical reviews or MetaCritic rating



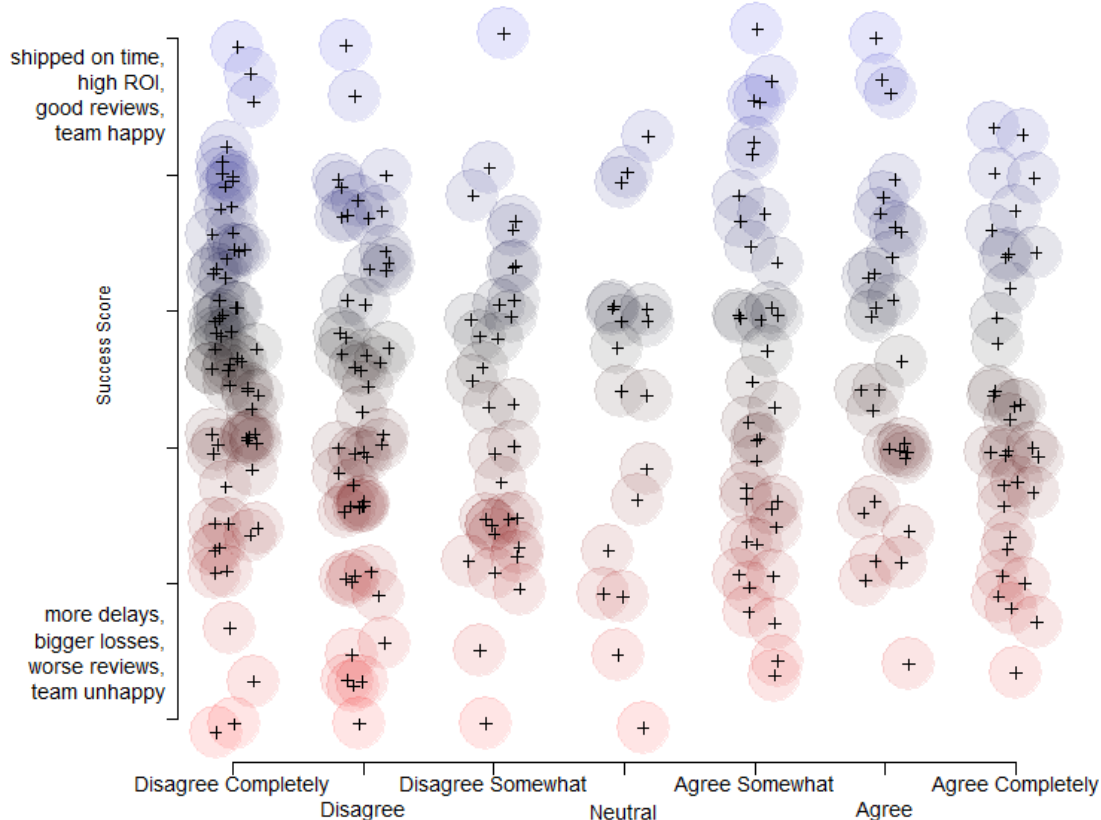
Bonuses tied to individual performance



Part 2: Things that don't Matter

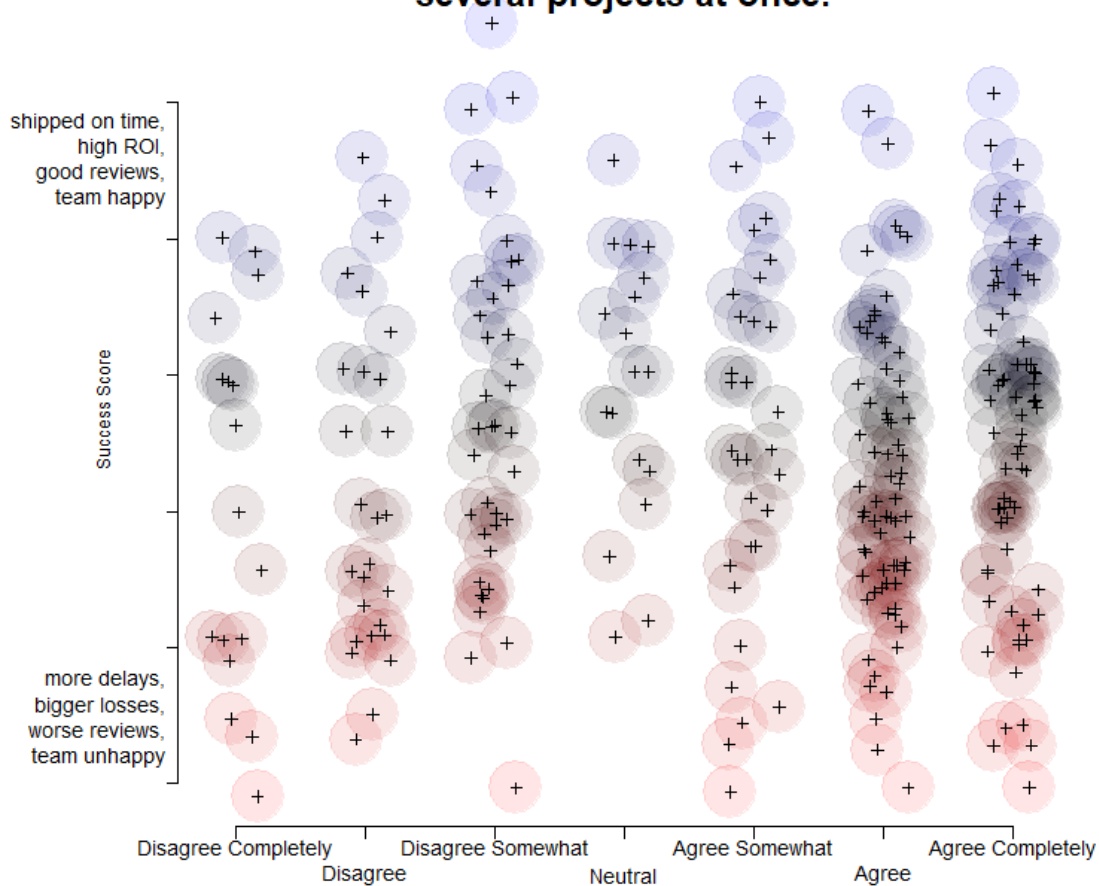
**Our team relied significantly on temporary workers,
contractors, or other team members who were not full-time
members of the studio.**

- Outside vs full-time work



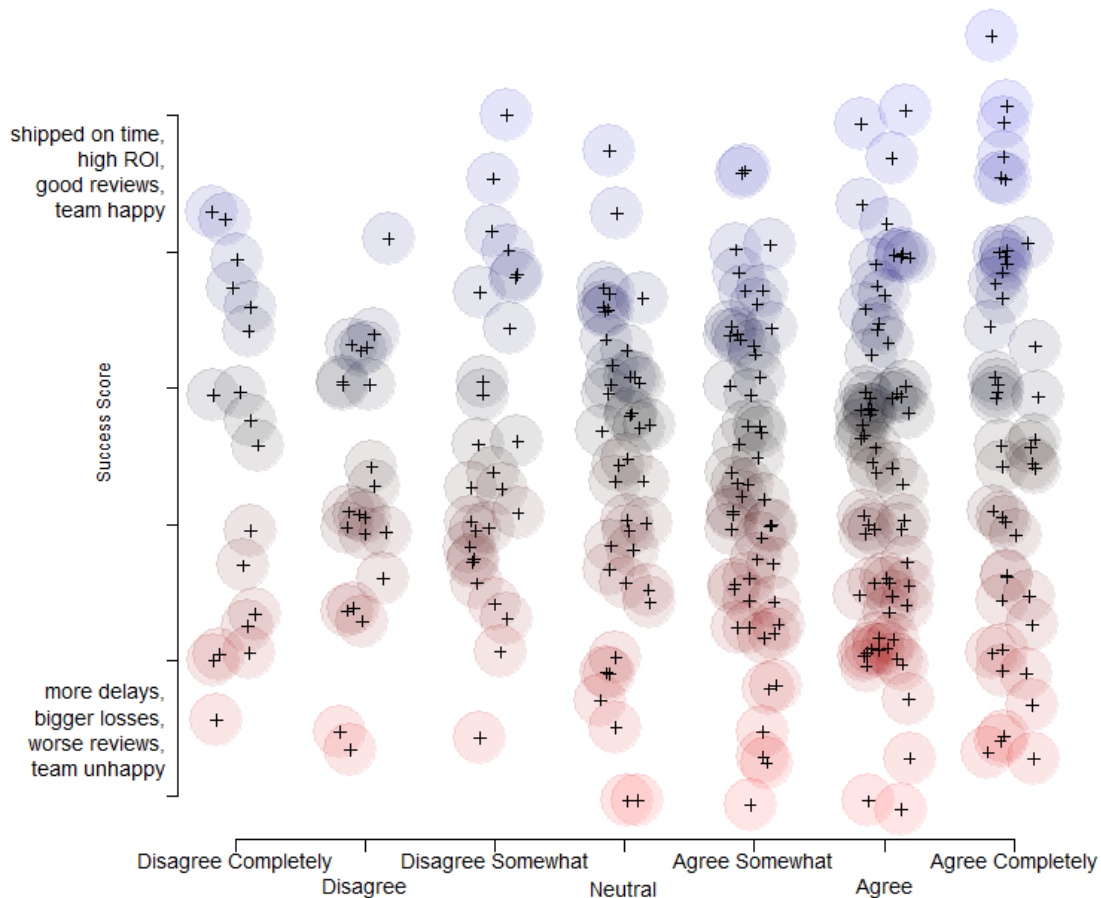
All team members were exclusively dedicated to this project throughout the development process, rather than working on several projects at once.

- Outside vs full-time work

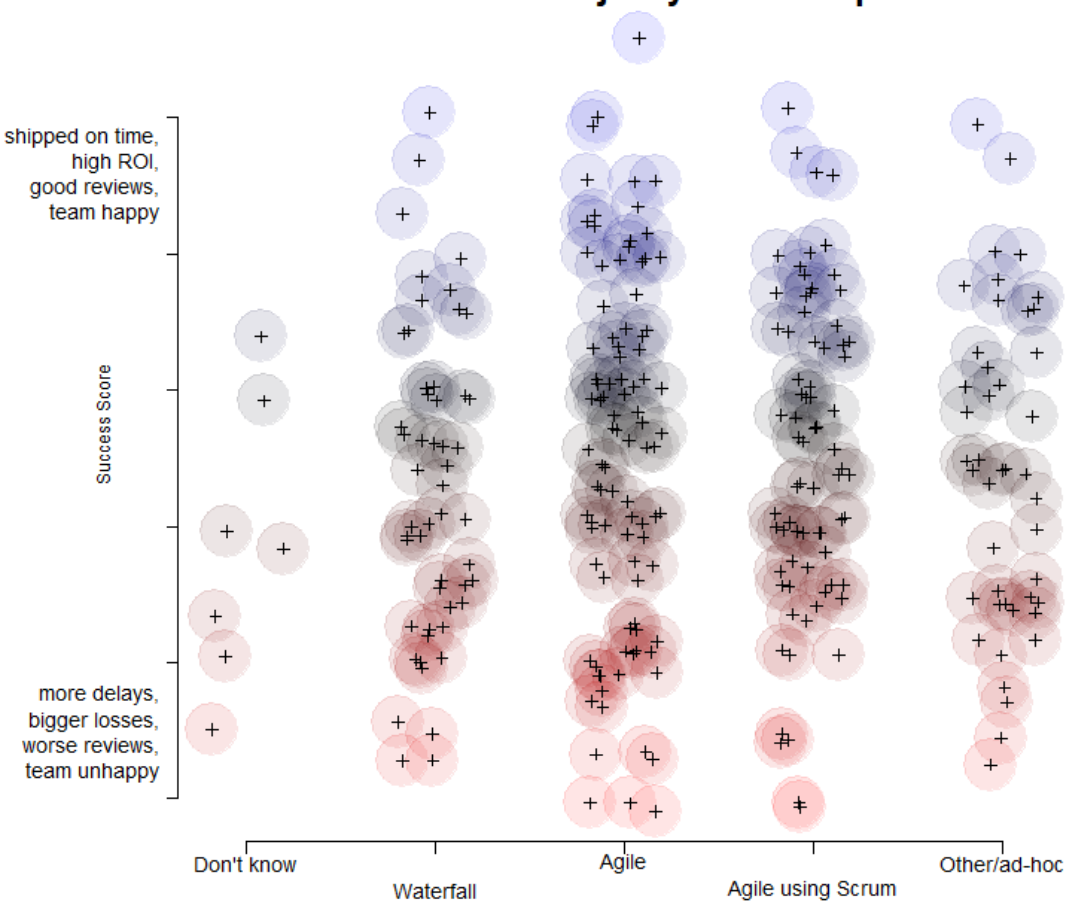


The company's culture preferred face-to-face communication over e-mail.

- Outside vs full-time work
- face-to-face vs email

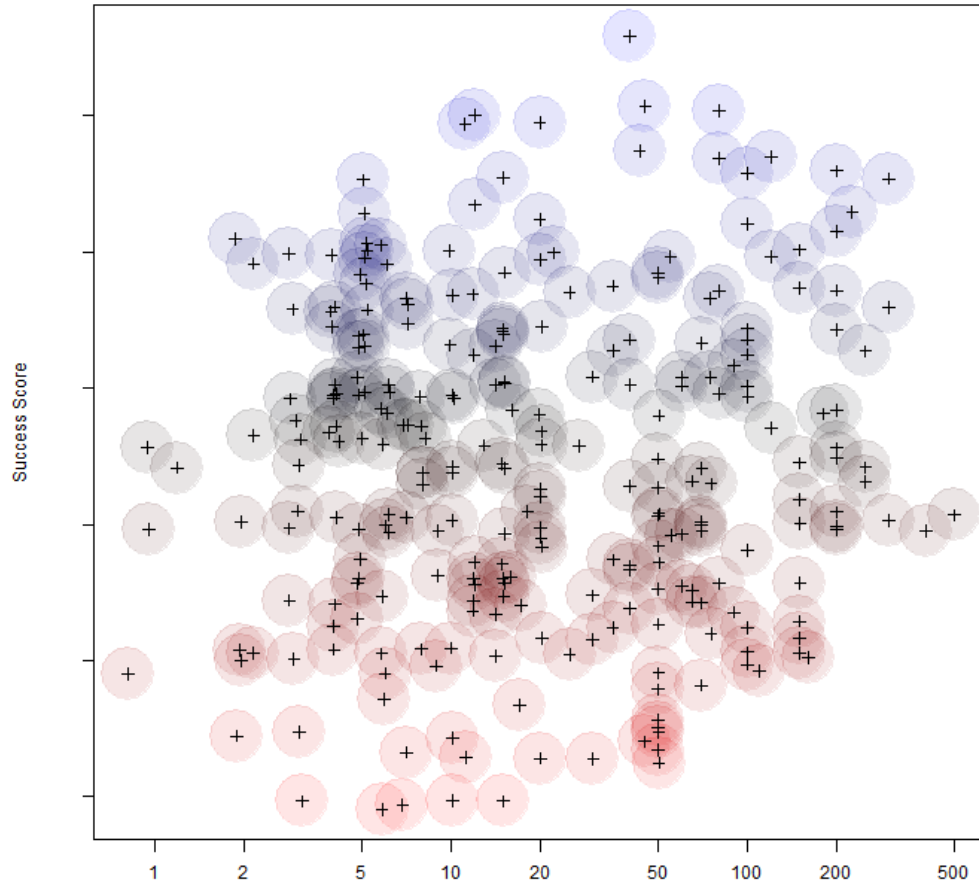


What software development methodology did the development team use for the majority of development?



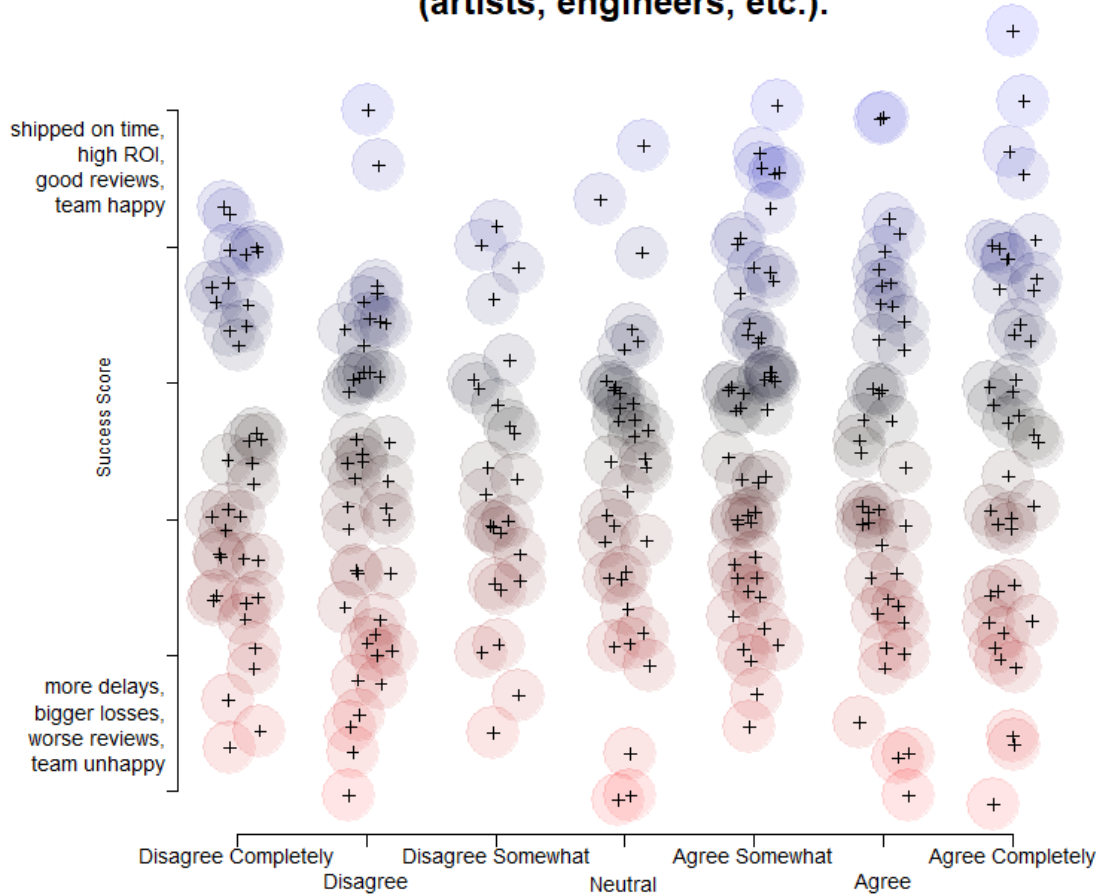
- Outside vs full-time work
- face-to-face vs email
- Development methodology

Average Team Size:



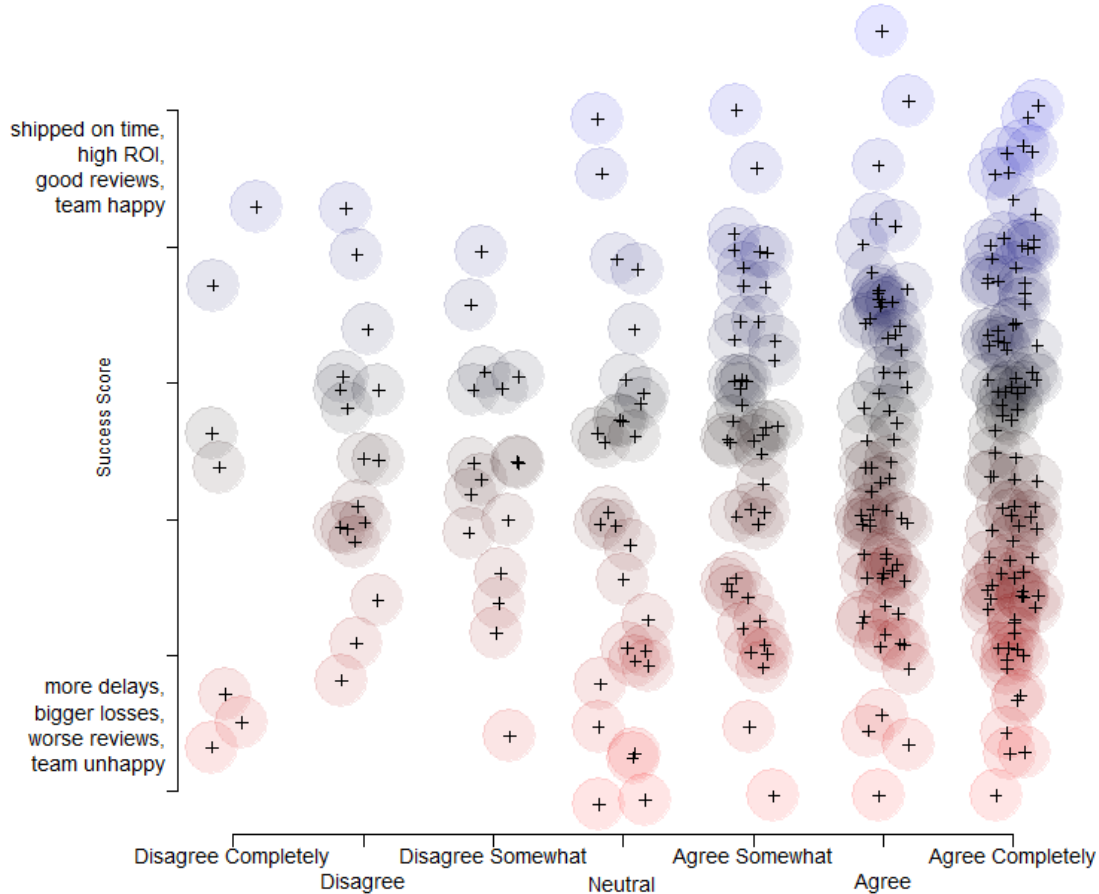
- Outside vs full-time work
- face-to-face vs email
- Development methodology
- Team size

Our developers were organized into cross-functional teams or teams that combined developers from several disciplines (artists, engineers, etc.).



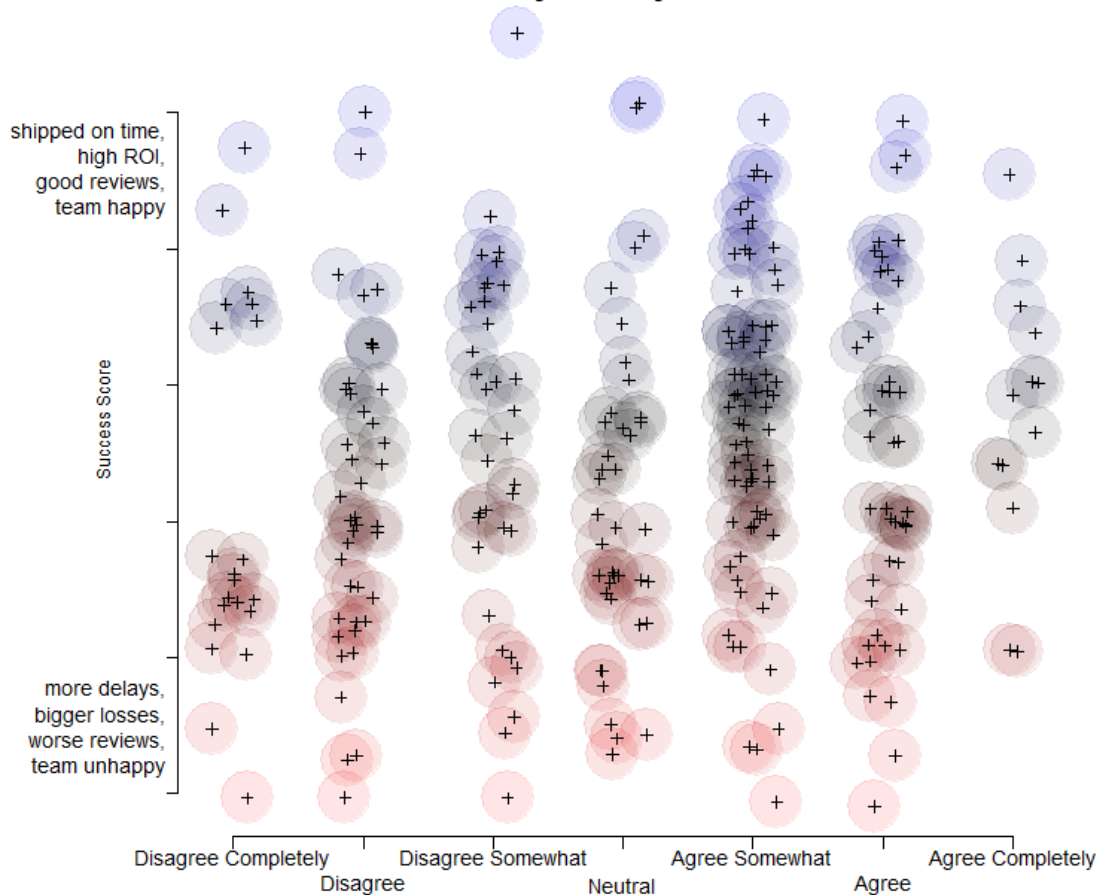
- Outside vs full-time work
- face-to-face vs email
- Development methodology
- Team size
- Cross-functional teams

There was someone on this team I considered a close friend.



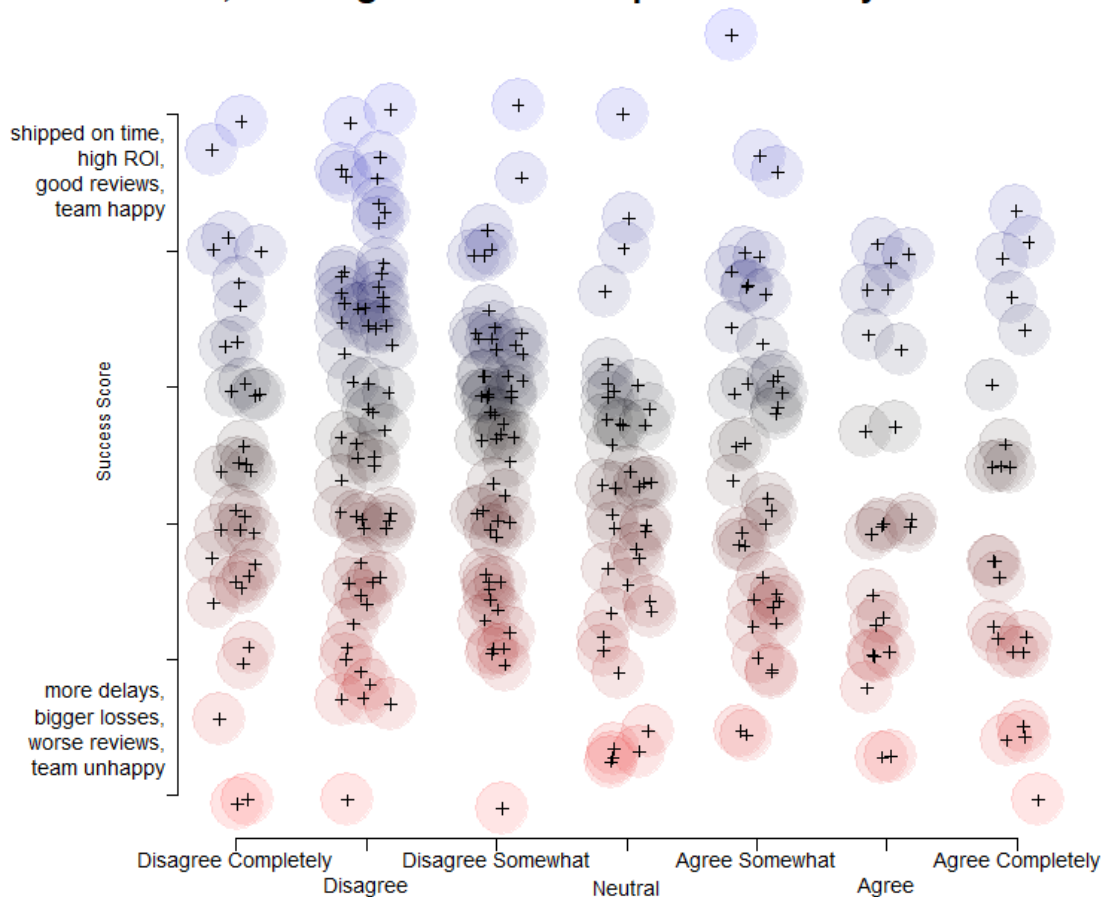
- Outside vs full-time work
- face-to-face vs email
- Development methodology
- Team size
- Cross-functional teams
- Close friend on the team

We re-estimated how long tasks would take frequently, or on a day-to-day basis.



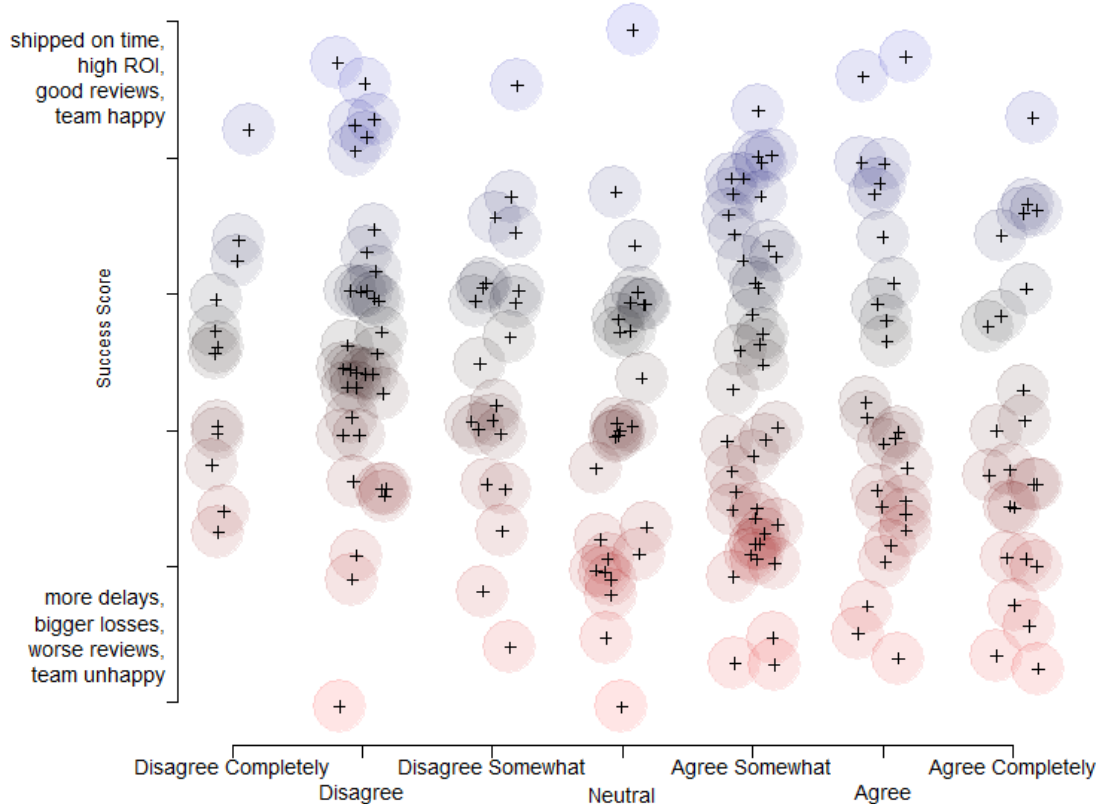
- Outside vs full-time work
- face-to-face vs email
- Development methodology
- Team size
- Cross-functional teams
- Close friend on the team
- Frequent/daily estimates

Deadlines in our studio were treated as matters of life and death, with significant consequences if they were not met.



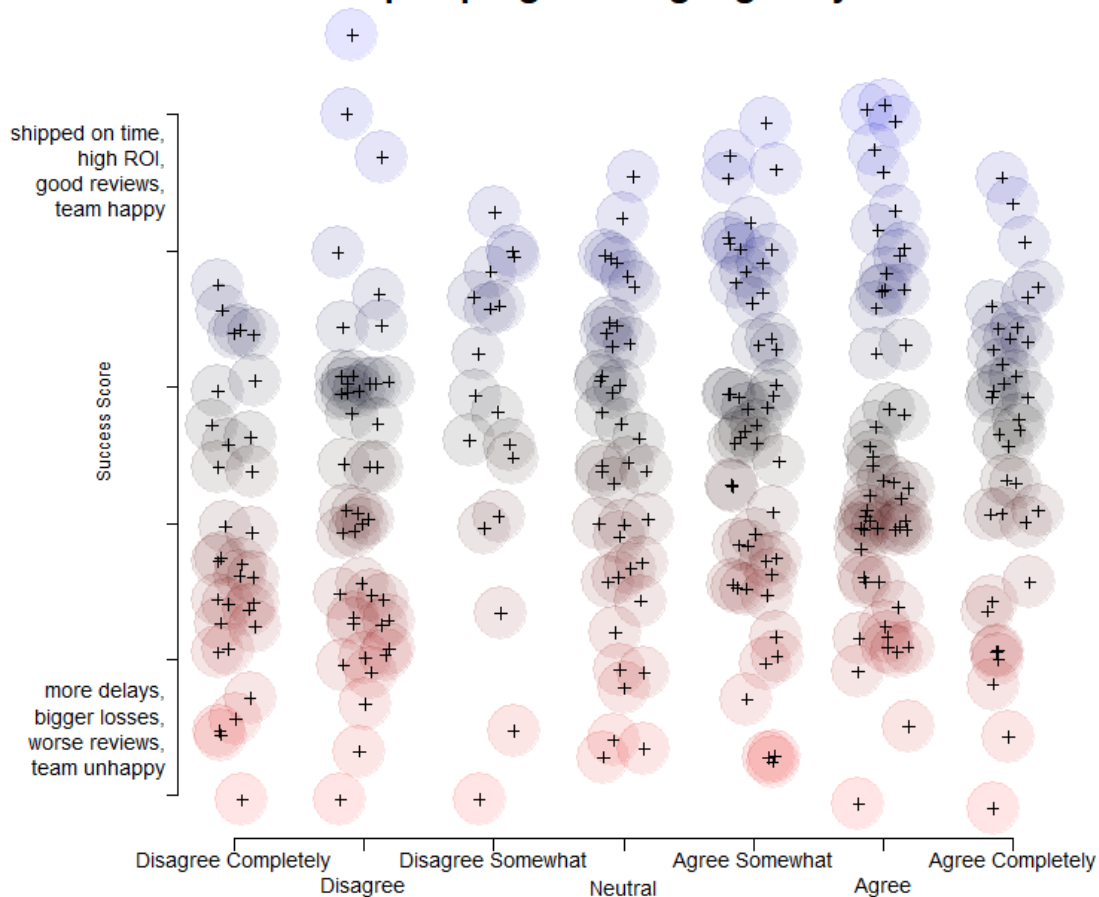
- Outside vs full-time work
- face-to-face vs email
- Development methodology
- Team size
- Cross-functional teams
- Close friend on the team
- Frequent/daily estimates
- Strict deadlines with consequences

We experienced difficulty communicating with outsourced or external developers working with our team (leave blank if no external developers).



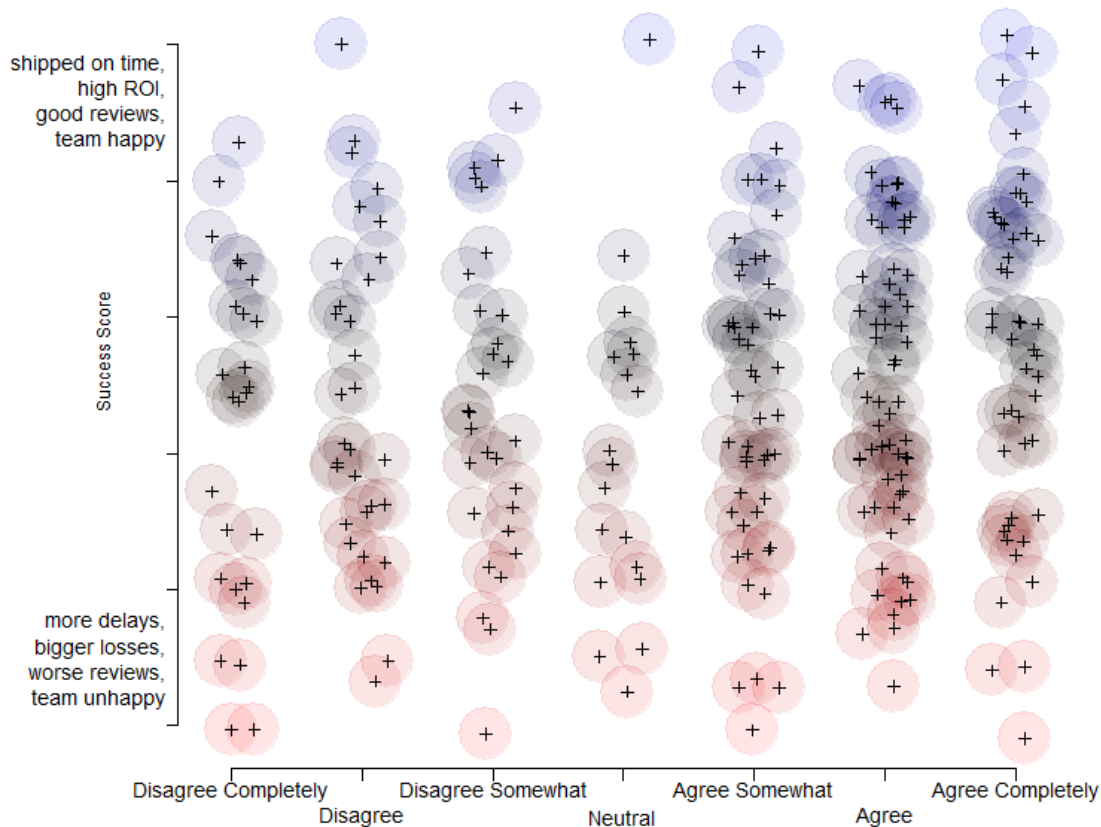
- Outside vs full-time work
- face-to-face vs email
- Development methodology
- Team size
- Cross-functional teams
- Close friend on the team
- Frequent/daily estimates
- Strict deadlines with consequences

The team's engineers practiced some form of code reviews or pair programming regularly.



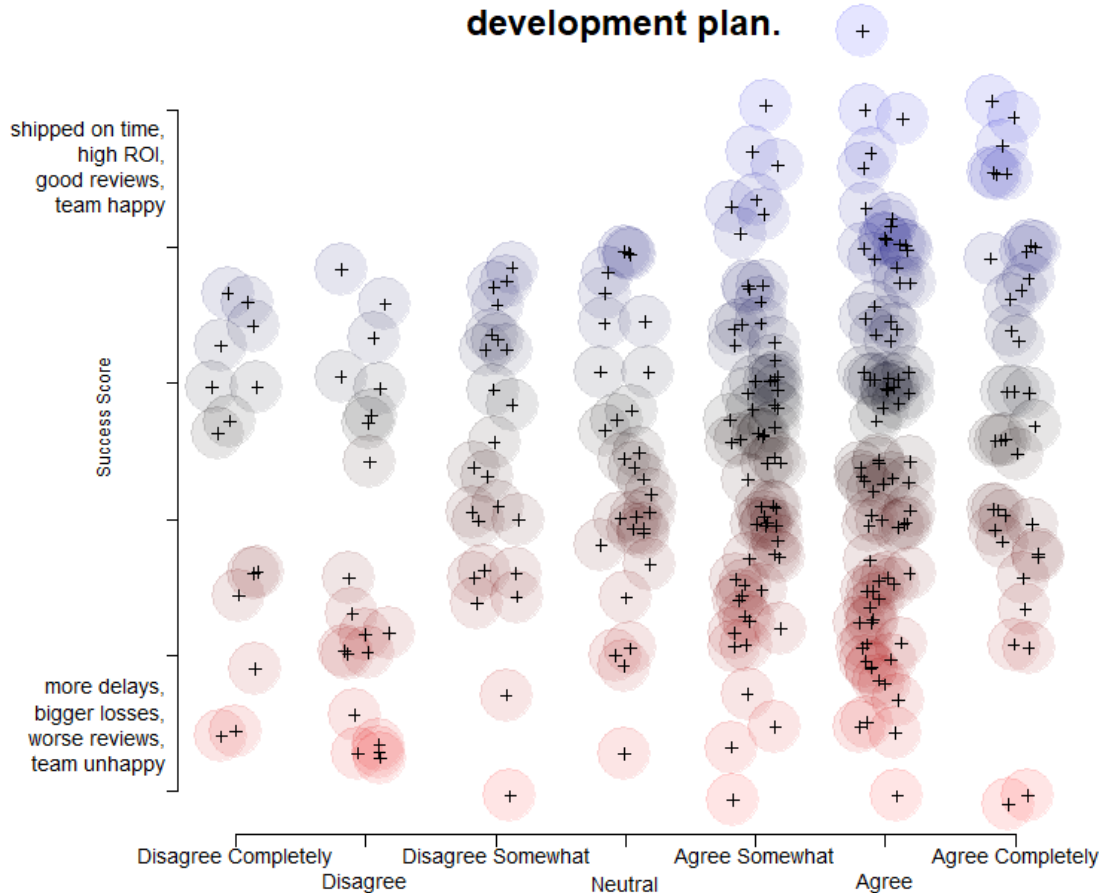
- Outside vs full-time work
- face-to-face vs email
- Development methodology
- Team size
- Cross-functional teams
- Close friend on the team
- Frequent/daily estimates
- Strict deadlines with consequences
- Code reviews/pair programming

We met daily to discuss what we were working on and identify production bottlenecks.



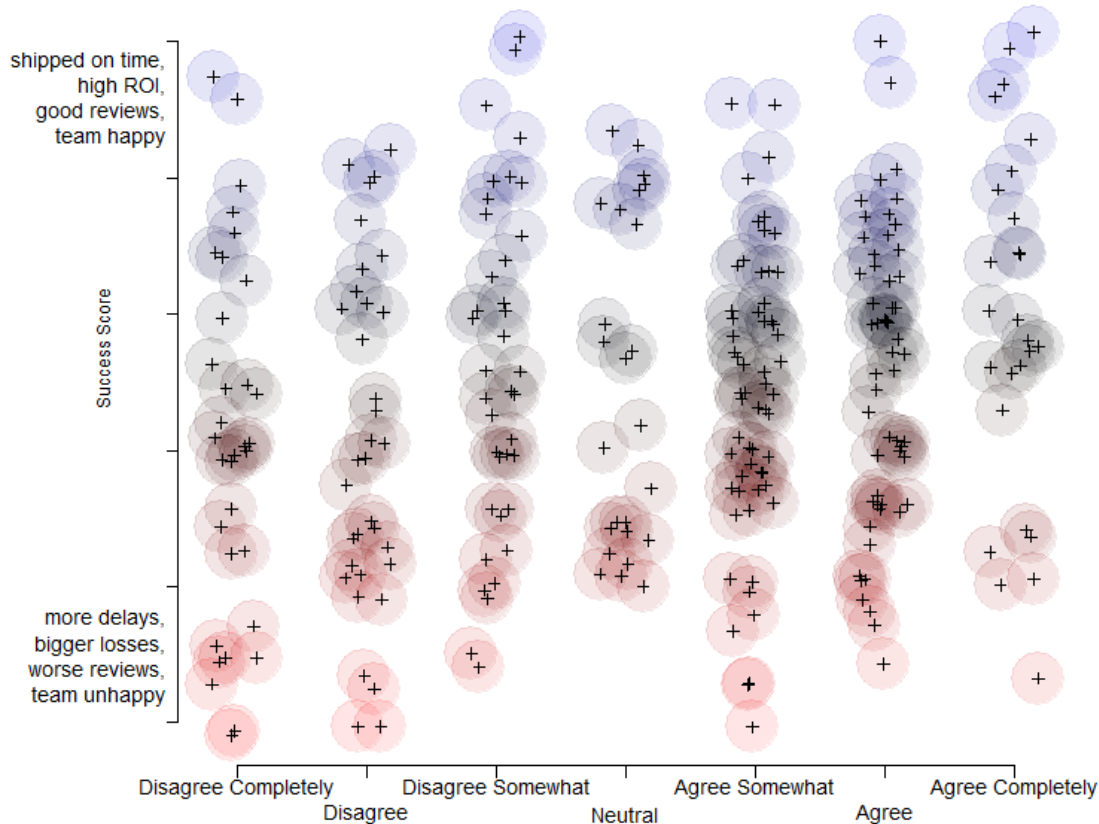
- Outside vs full-time work
- face-to-face vs email
- Development methodology
- Team size
- Cross-functional teams
- Close friend on the team
- Frequent/daily estimates
- Strict deadlines with consequences
- Code reviews/pair programming

The priorities for each new milestone or sprint were determined based on the current state of the project each time, rather than deriving from a predetermined overall development plan.



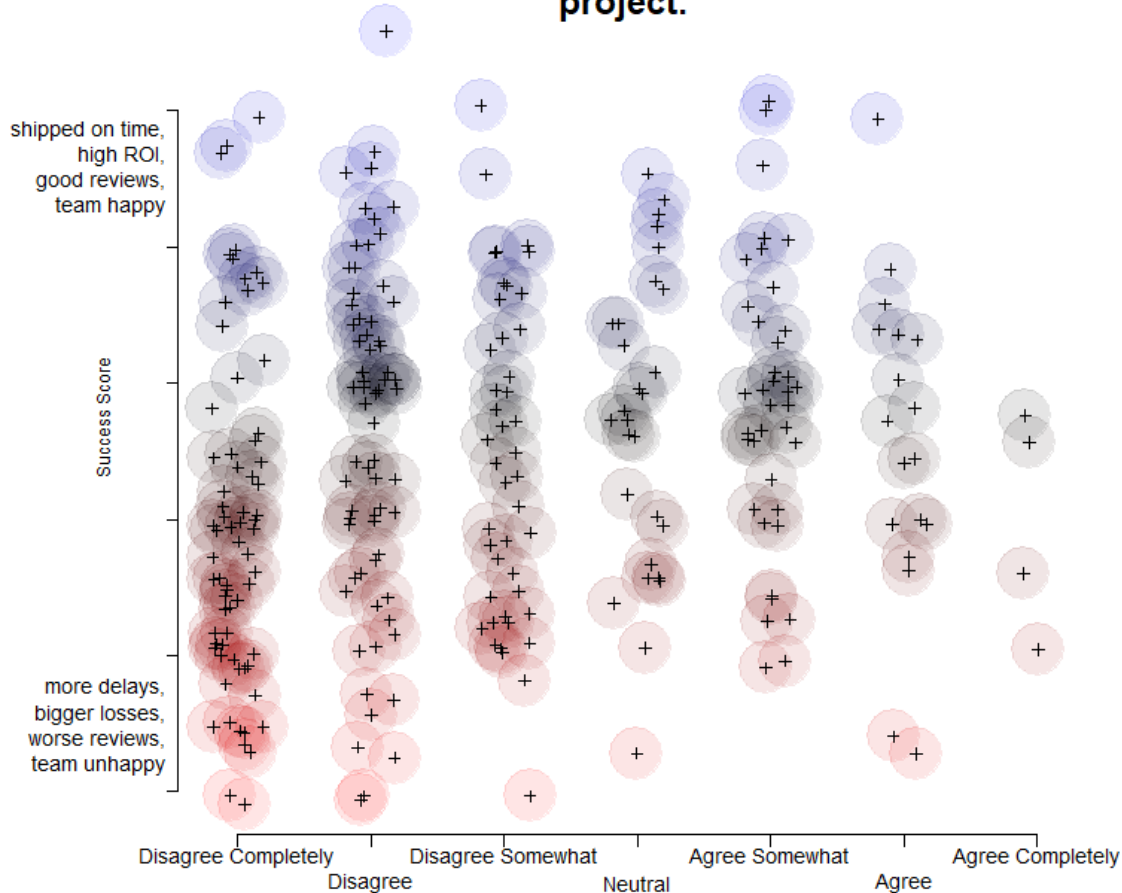
- Outside vs full-time work
- face-to-face vs email
- Development methodology
- Team size
- Cross-functional teams
- Close friend on the team
- Frequent/daily estimates
- Strict deadlines with consequences
- Code reviews/pair programming

There was some sort of design document available to the team near the beginning of development that clearly specified the game we were trying to make.



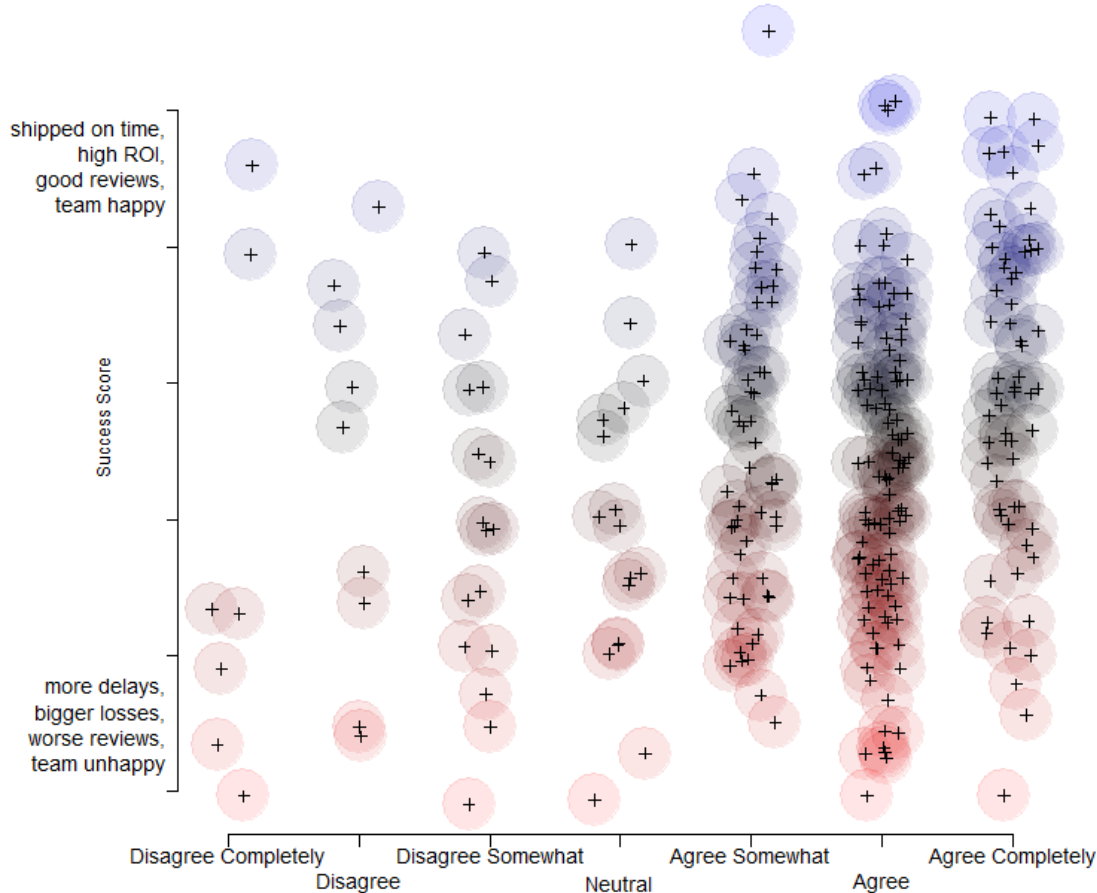
- Outside vs full-time work
- face-to-face vs email
- Development methodology
- Team size
- Cross-functional teams
- Close friend on the team
- Frequent/daily estimates
- Strict deadlines with consequences
- Code reviews/pair programming
- Early design document

Project leaders created detailed plans for our project when we started and followed them very closely throughout the project.



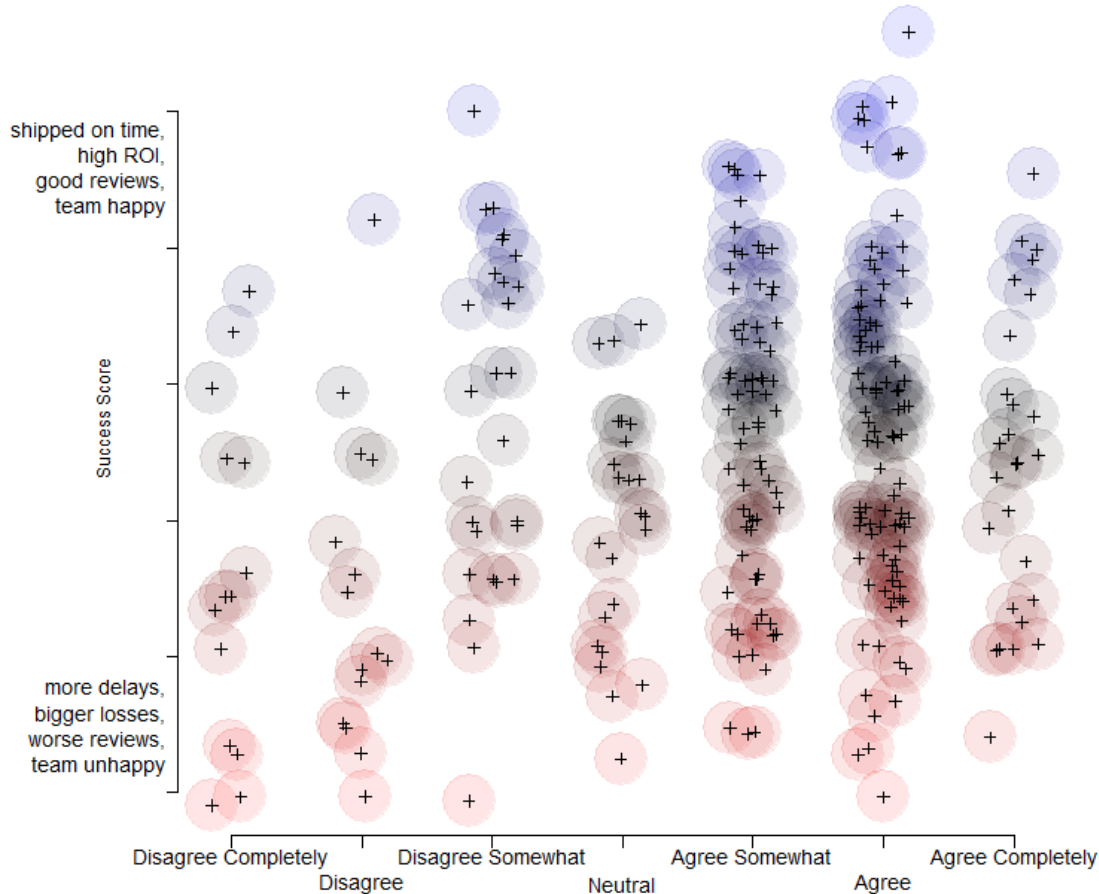
- Outside vs full-time work
- face-to-face vs email
- Development methodology
- Team size
- Cross-functional teams
- Close friend on the team
- Frequent/daily estimates
- Strict deadlines with consequences
- Code reviews/pair programming
- Early design document

The tasks I worked on were a good fit for my skill set.



- Outside vs full-time work
- face-to-face vs email
- Development methodology
- Team size
- Cross-functional teams
- Close friend on the team
- Frequent/daily estimates
- Strict deadlines with consequences
- Code reviews/pair programming
- Early design document

The person who was responsible for doing a task determined how much time would be allocated to it.



- Outside vs full-time work
- face-to-face vs email
- Development methodology
- Team size
- Cross-functional teams
- Close friend on the team
- Frequent/daily estimates
- Strict deadlines with consequences
- Code reviews/pair programming
- Early design document

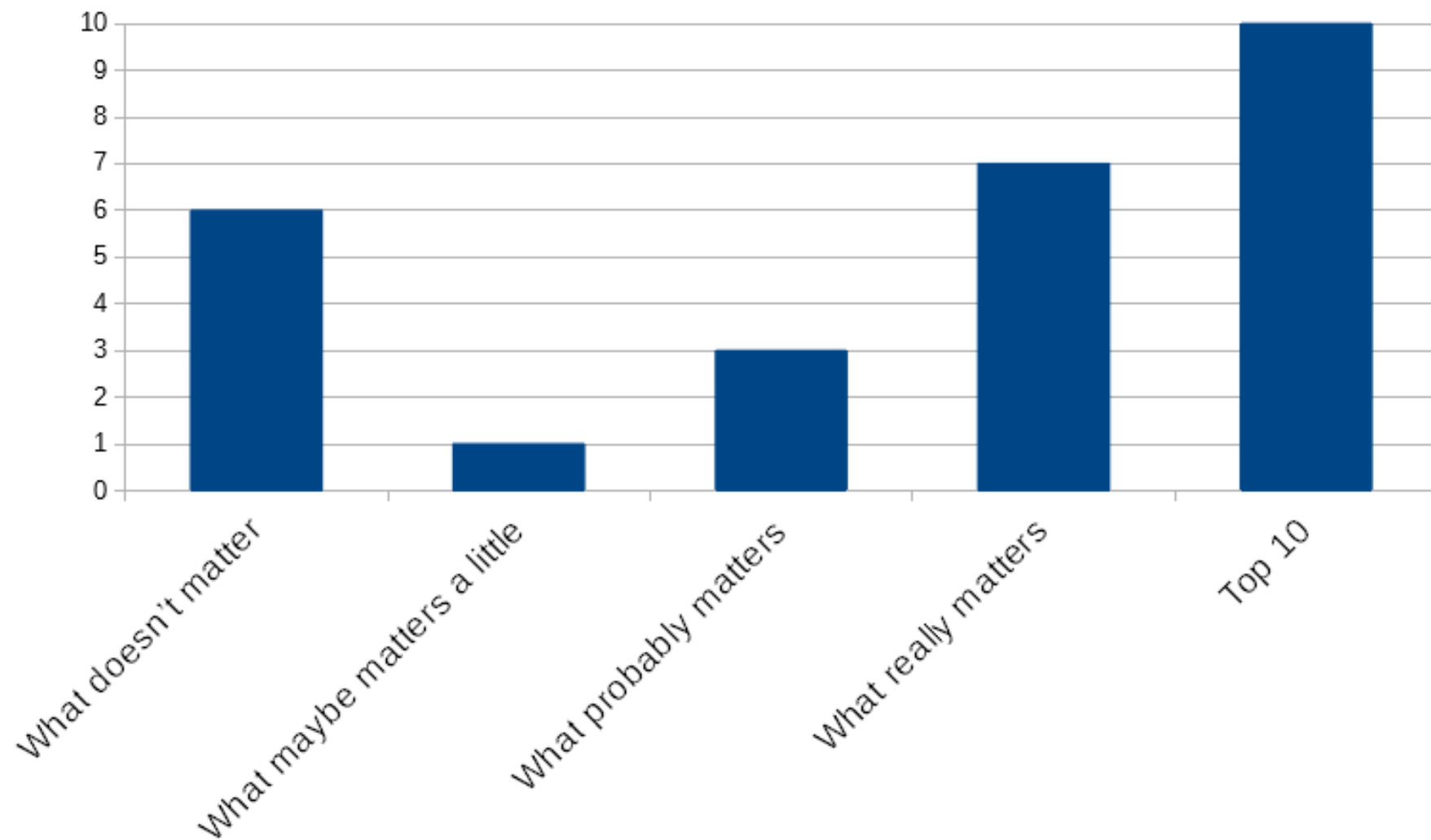
Part 2 Main Points

Whatever you choose for these won't impact whether you make a good game or a bad game:

- Production methodology
- Team size and location
- Cross-functional teams or not
- Outsourcing or not

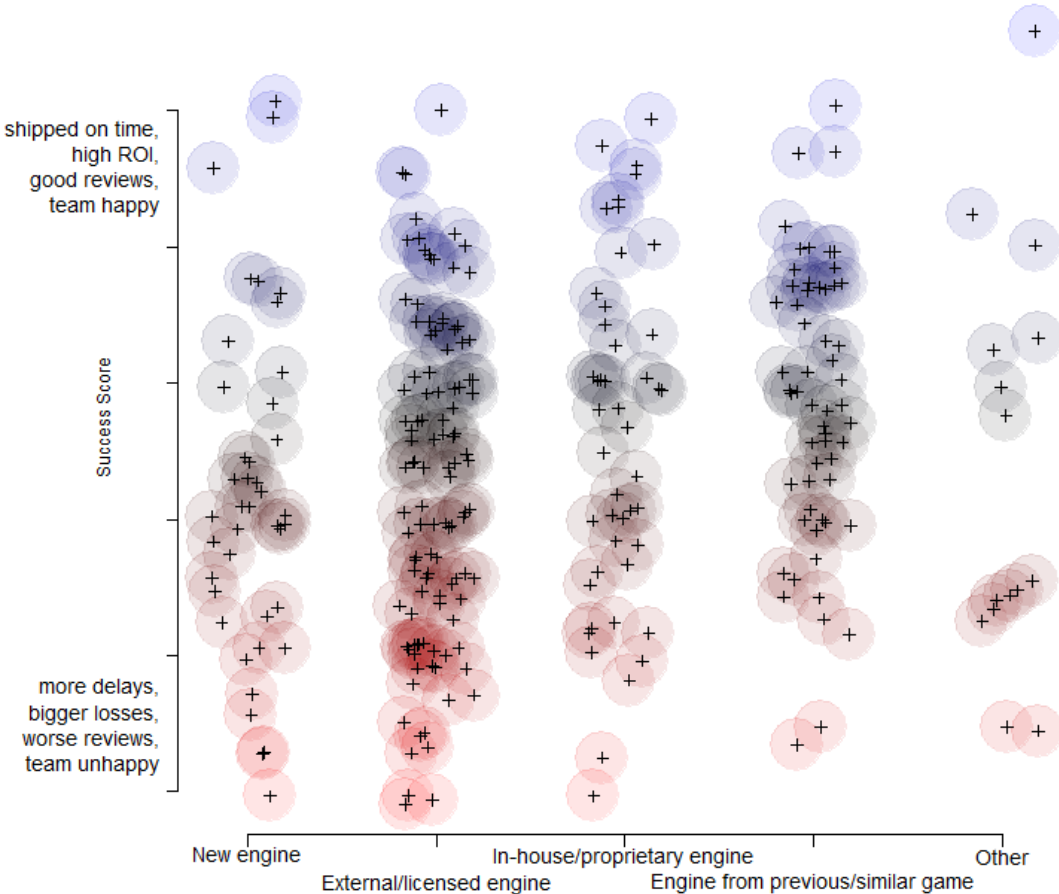
Intermission

How interesting each section is



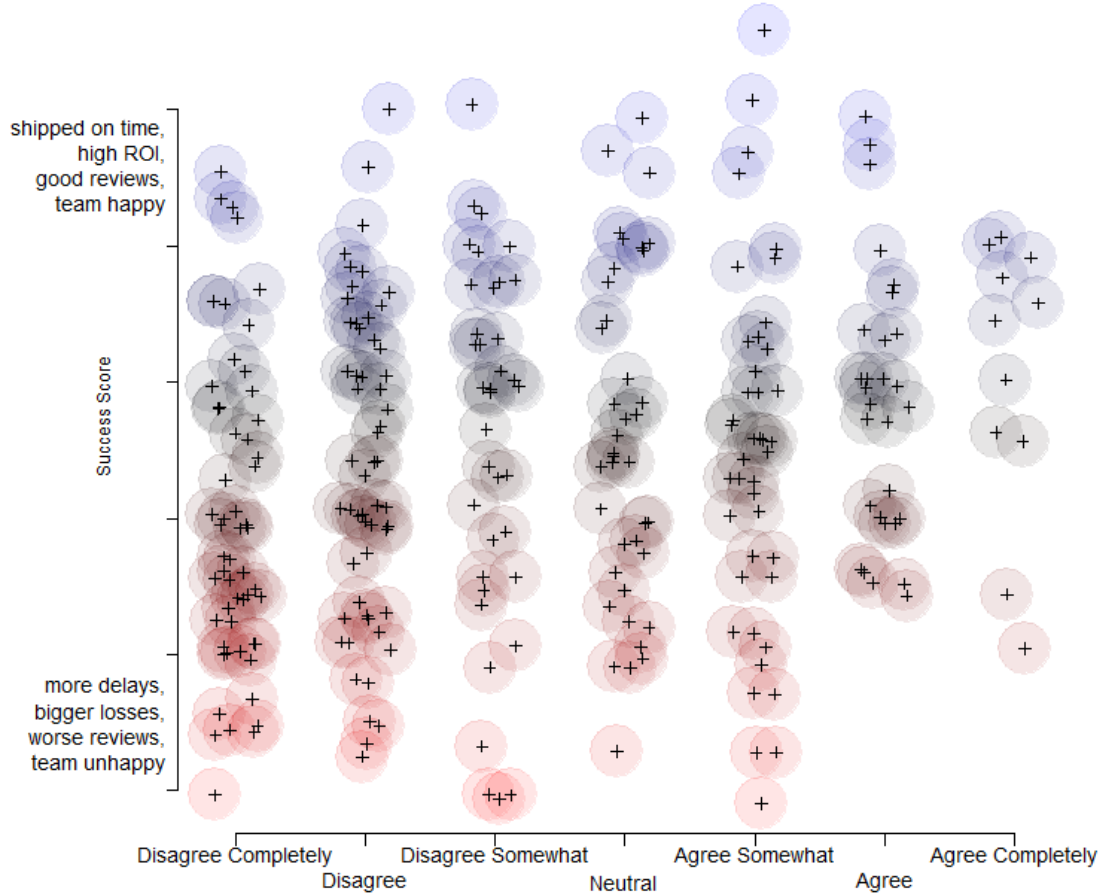
Part 3: Things that Maybe Help/Hurt a Little

What sort of technology solution did the team use to create the game?



- Maybe Bad:
- New engine

We received some form of coaching or guidance to enhance our effort or improve our effectiveness as team members.



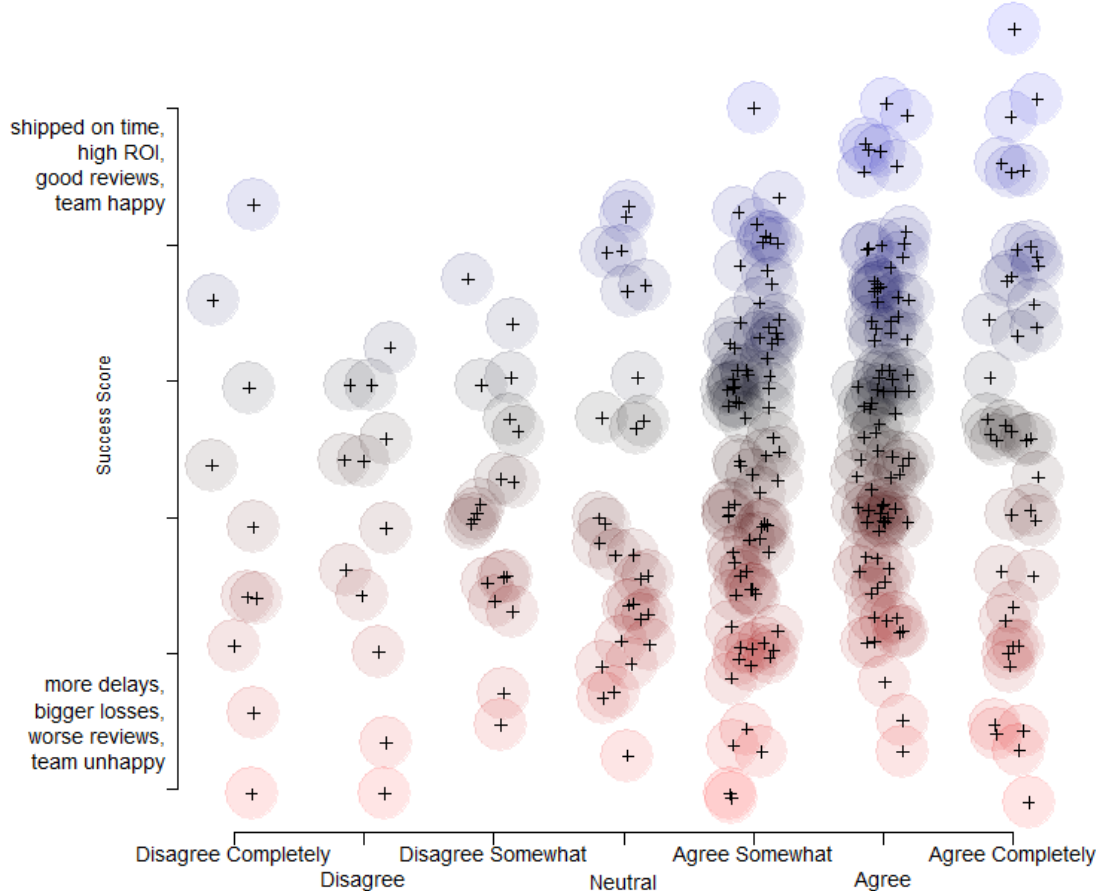
Maybe Good:

- Coaching

Maybe Bad:

- New engine

**When team members had difficulty performing a task, others
were quick to offer them support.**



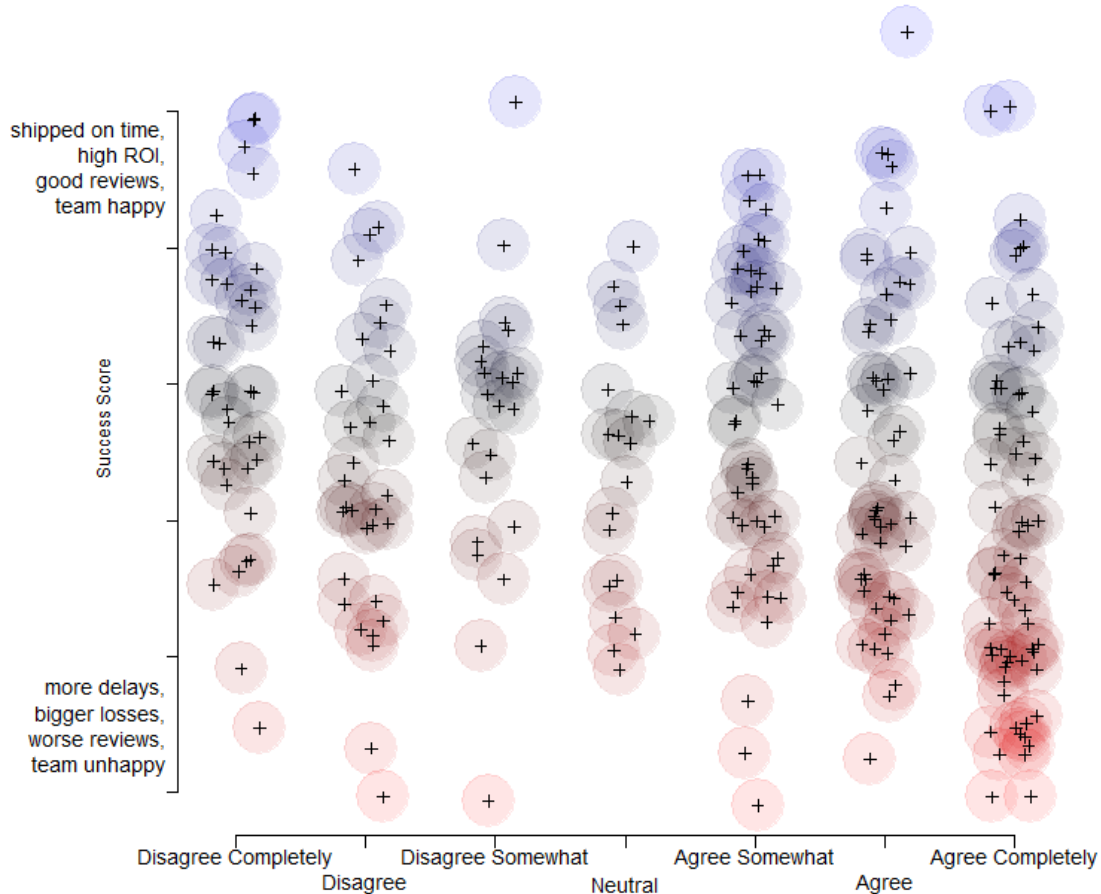
Maybe Good:

- Coaching
- Helpful team

Maybe Bad:

- New engine

I worked a lot of overtime or "crunched" on this project.



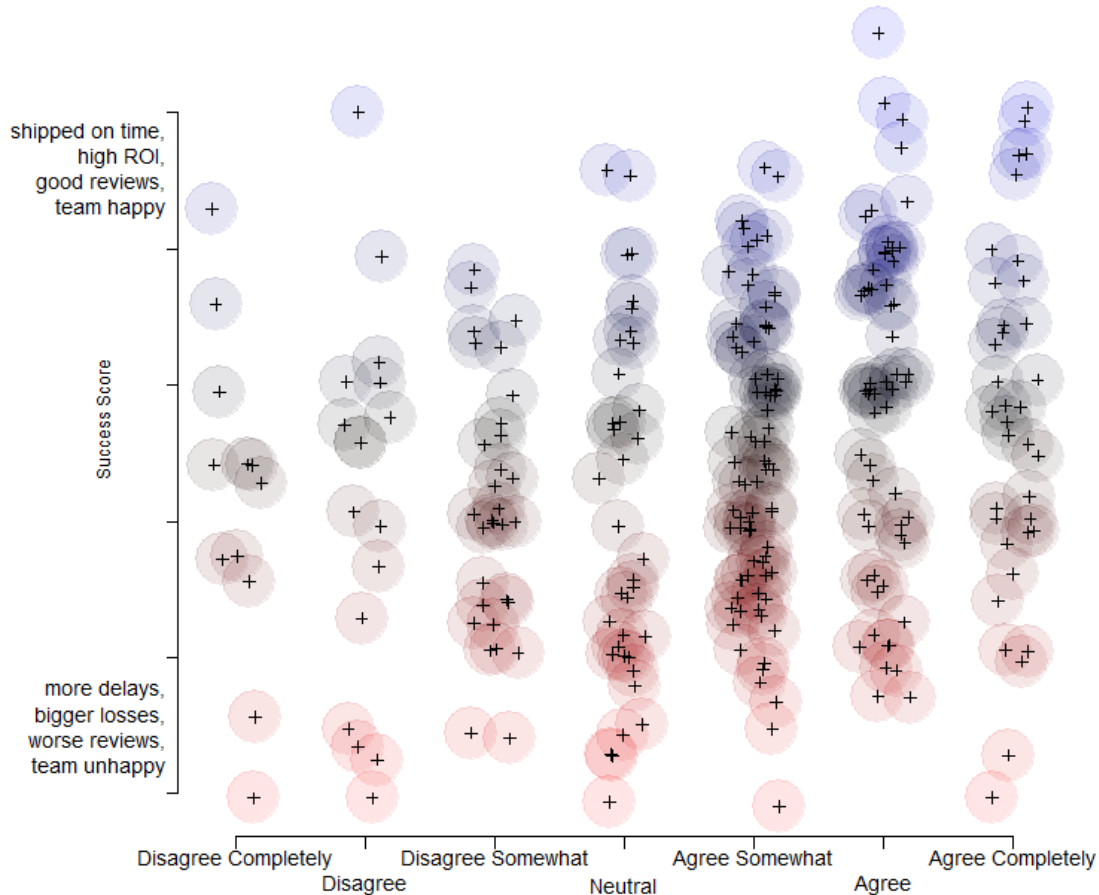
Maybe Good:

- Coaching
- Helpful team

Maybe Bad:

- New engine
- Crunch

The team often engaged in unfiltered and passionate debate of ideas.



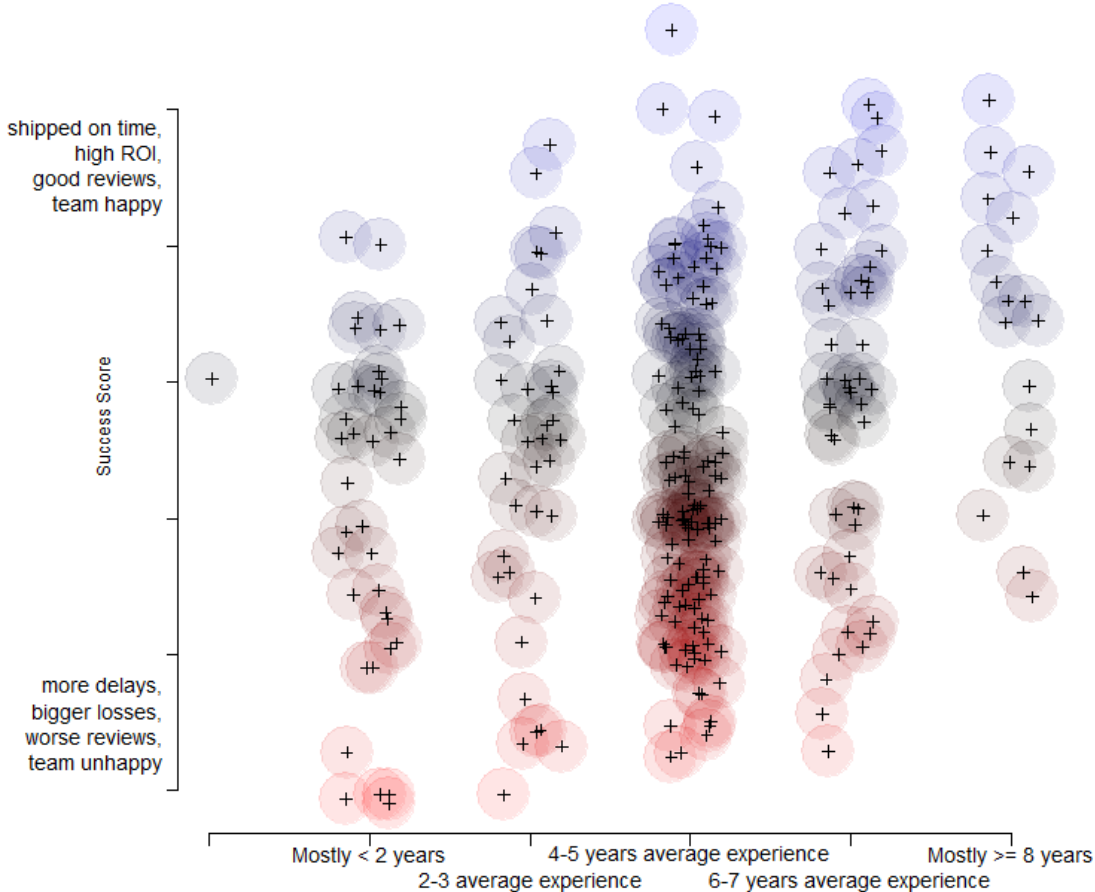
Maybe Good:

- Coaching
- Helpful team
- Unfiltered discussions

Maybe Bad:

- New engine
- Crunch

Please select the answer below that best describes the overall level of experience of the game development team.



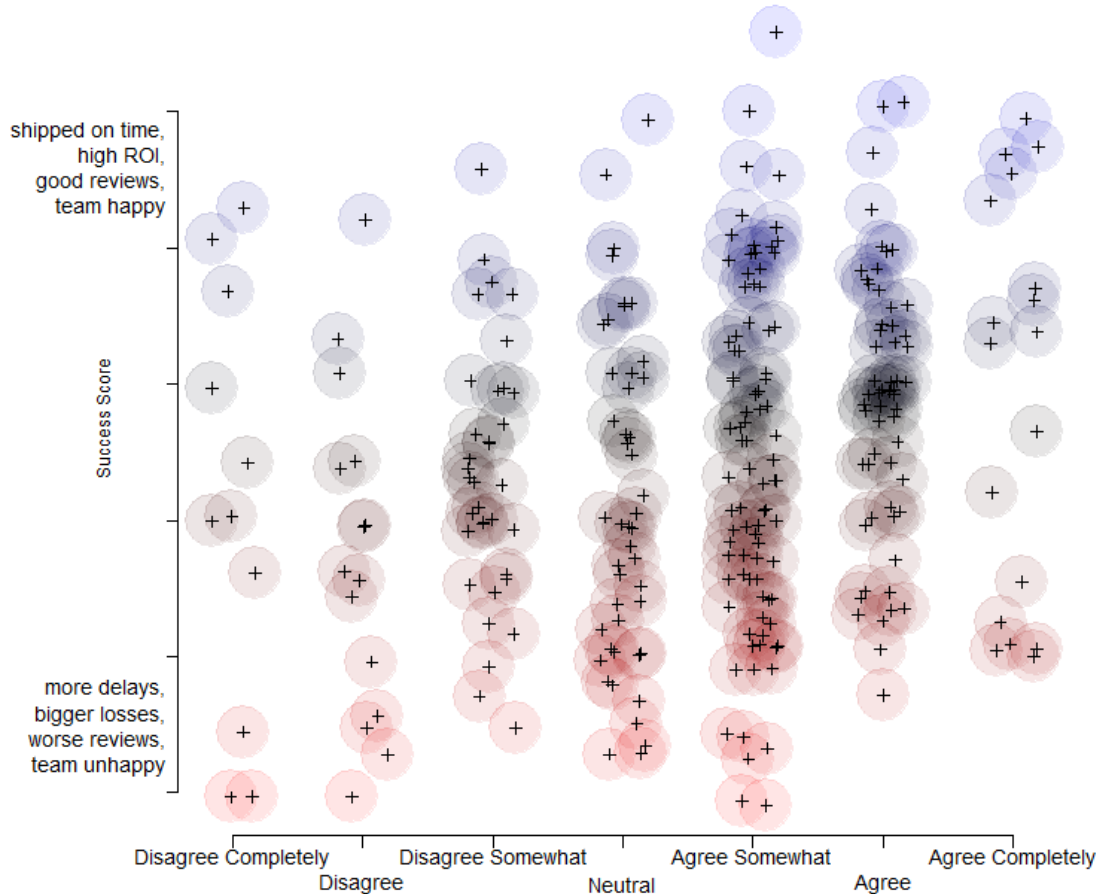
Maybe Good:

- Coaching
- Helpful team
- Unfiltered discussions
- Team experience

Maybe Bad:

- New engine
- Crunch

Team members held one another accountable for their actions.



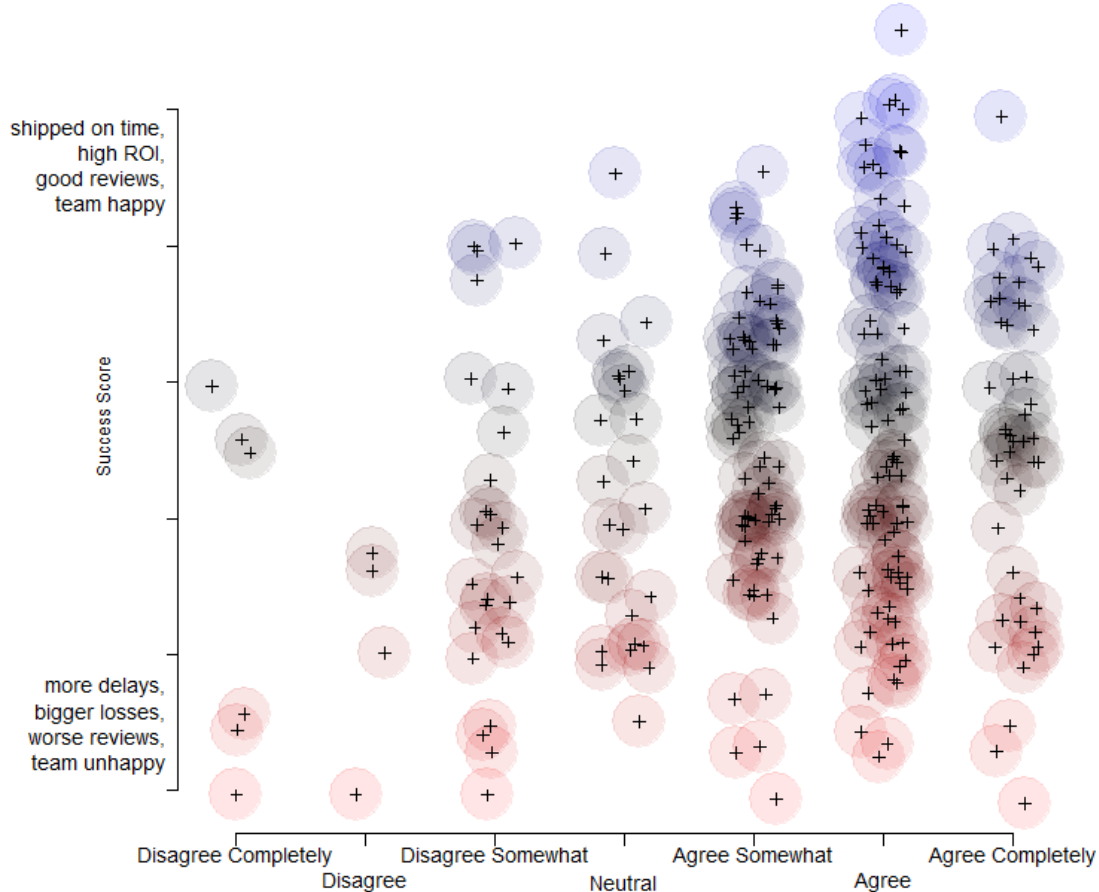
Maybe Good:

- Coaching
- Helpful team
- Unfiltered discussions
- Team experience
- Mutual accountability

Maybe Bad:

- New engine
- Crunch

Team members frequently discussed the issues and challenges they faced.



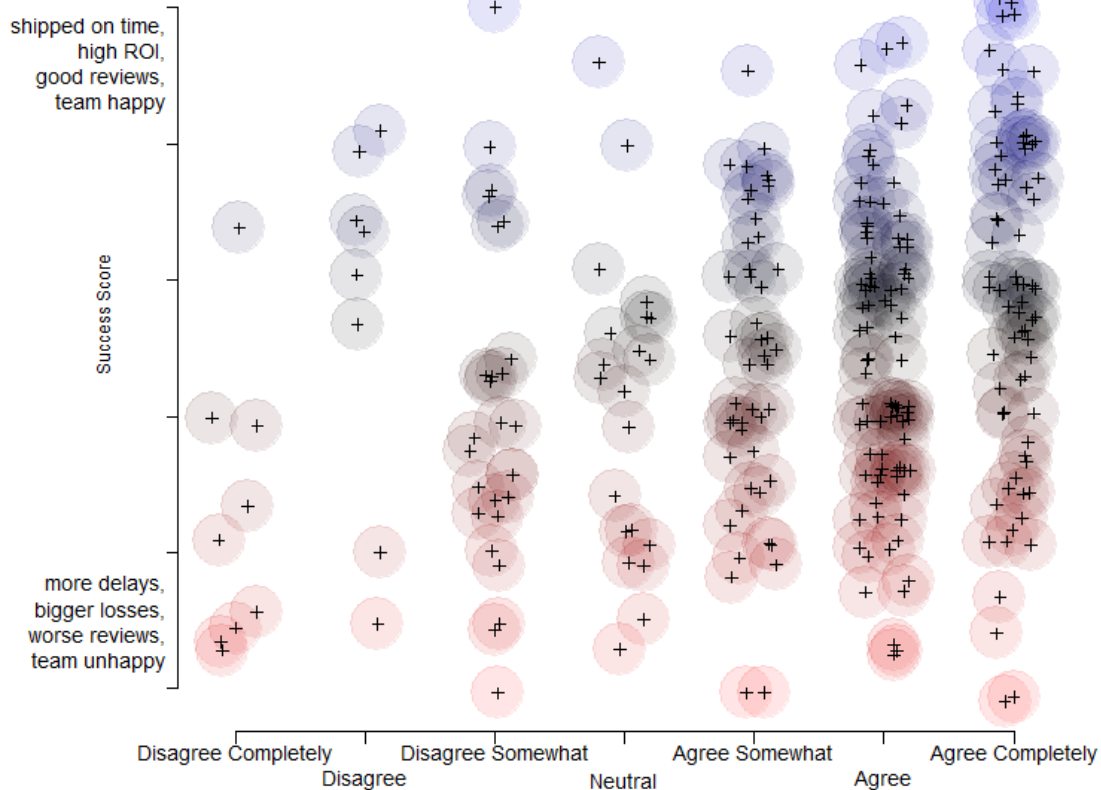
Maybe Good:

- Coaching
- Helpful team
- Unfiltered discussions
- Team experience
- Mutual accountability

Maybe Bad:

- New engine
- Crunch

I had all the software and hardware I needed to do my job effectively.



Maybe Good:

- Coaching
- Helpful team
- Unfiltered discussions
- Team experience
- Mutual accountability
- Good dev equipment

Maybe Bad:

- New engine
- Crunch

The project experienced a major technology revamp in development, such as a new game engine or major engine changes.



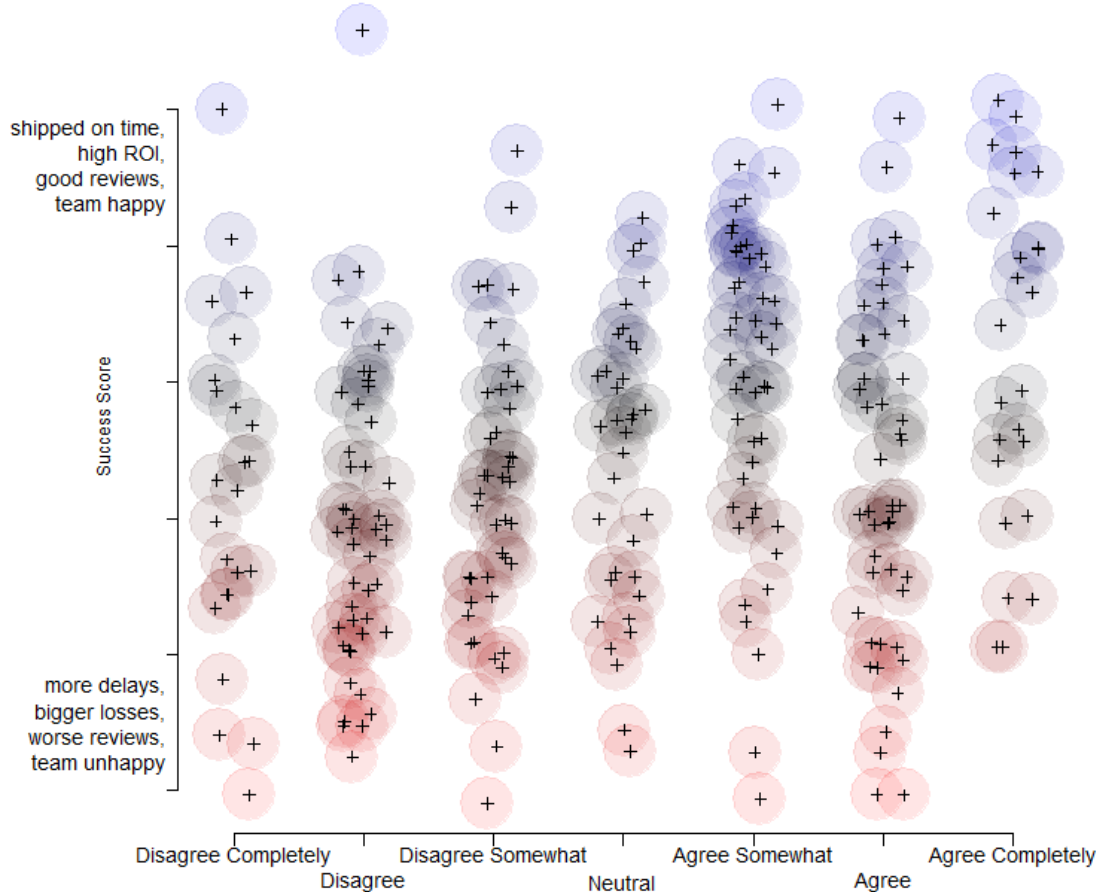
Maybe Good:

- Coaching
- Helpful team
- Unfiltered discussions
- Team experience
- Mutual accountability
- Good dev equipment

Maybe Bad:

- New engine
- Crunch
- Big tech changes

The entire team was involved in prioritizing the work to be done for each milestone or sprint.



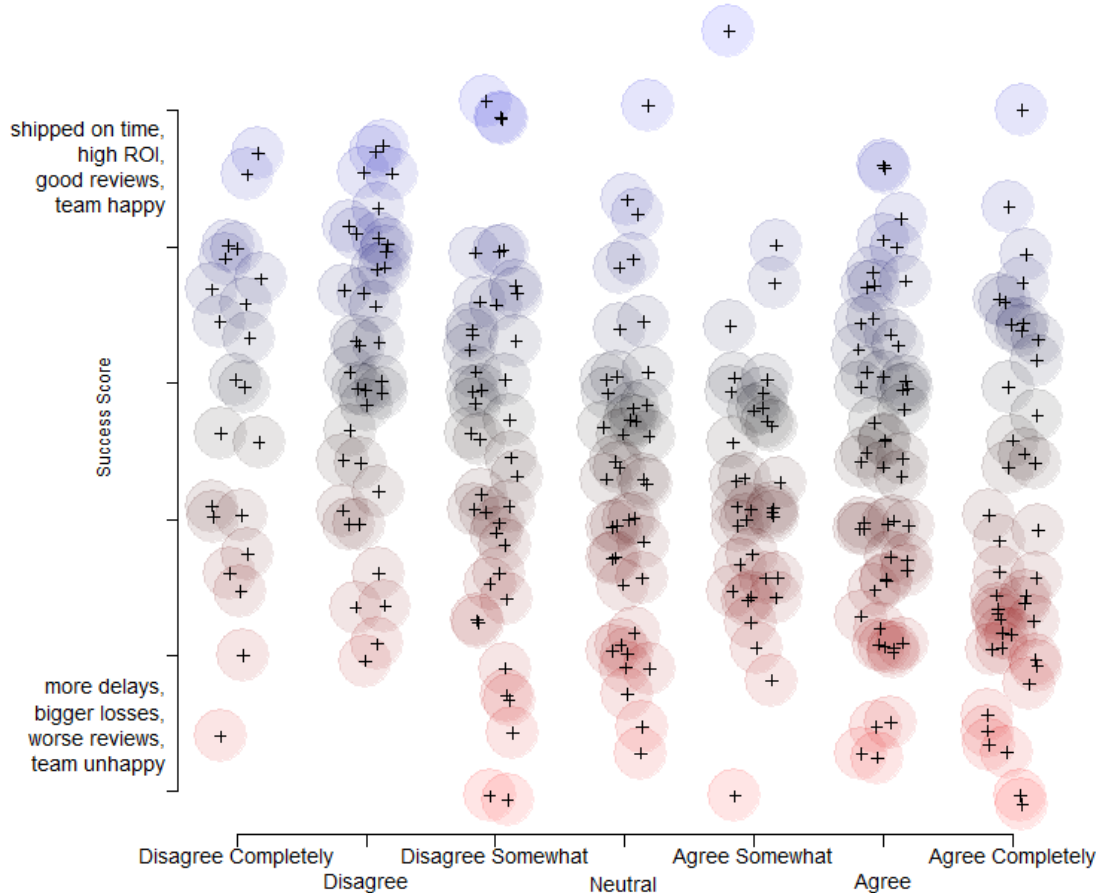
Maybe Good:

- Coaching
- Helpful team
- Unfiltered discussions
- Team experience
- Mutual accountability
- Good dev equipment
- Whole team involved in prioritization of work

Maybe Bad:

- New engine
- Crunch
- Big tech changes

If the team leadership didn't like an idea, it was quickly swept under the rug.



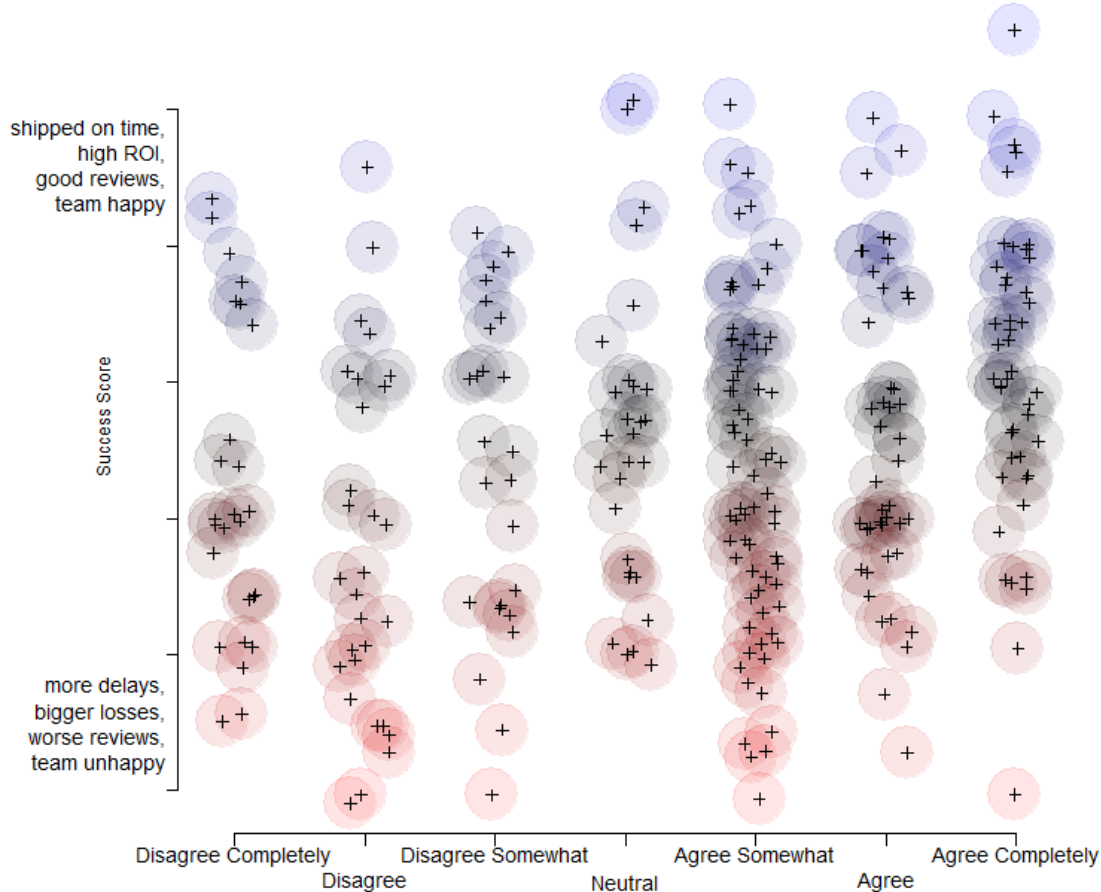
Maybe Good:

- Coaching
- Helpful team
- Unfiltered discussions
- Team experience
- Mutual accountability
- Good dev equipment
- Whole team involved in prioritization of work

Maybe Bad:

- New engine
- Crunch
- Big tech changes
- Leaders suppress “bad” ideas

There was someone at this organization who encouraged me to develop my skills further.



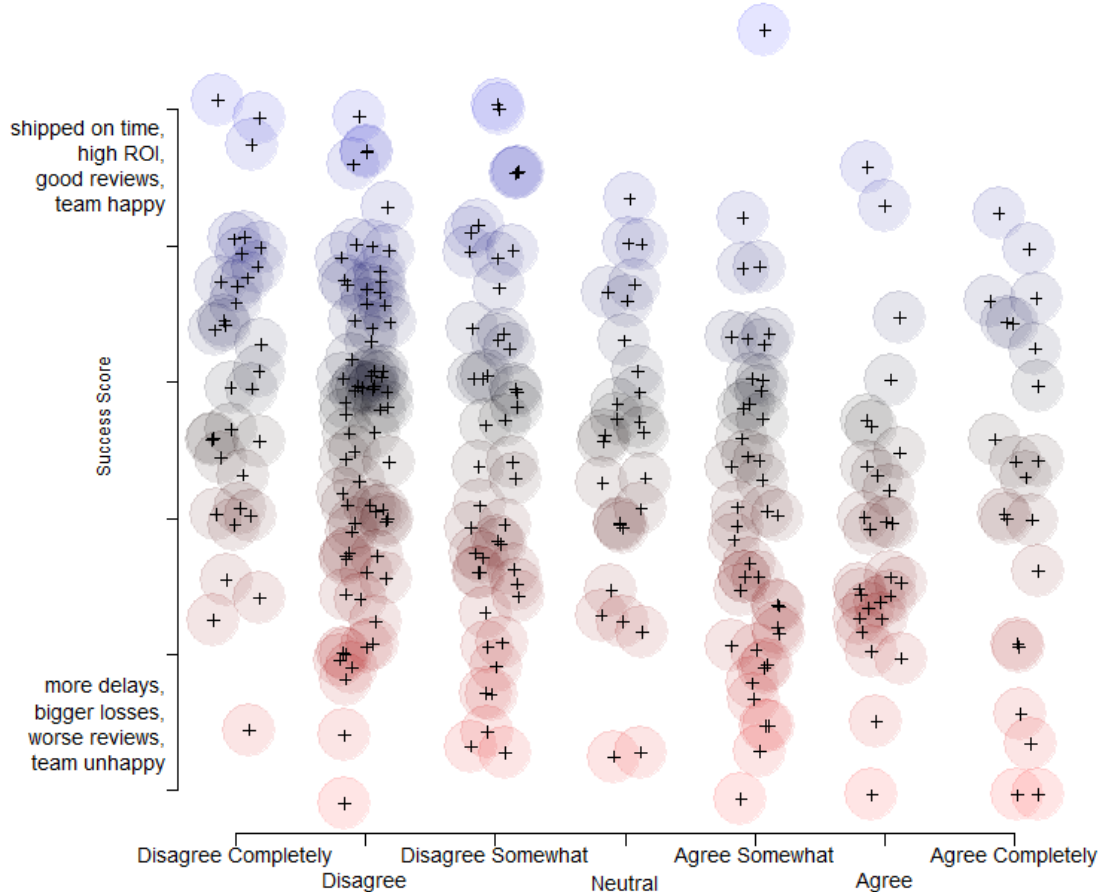
Maybe Good:

- Coaching
- Helpful team
- Unfiltered discussions
- Team experience
- Mutual accountability
- Good dev equipment
- Whole team involved in prioritization of work

Maybe Bad:

- New engine
- Crunch
- Big tech changes
- Leaders suppress “bad” ideas

It wasn't always clear who was supposed to be doing what on the project.



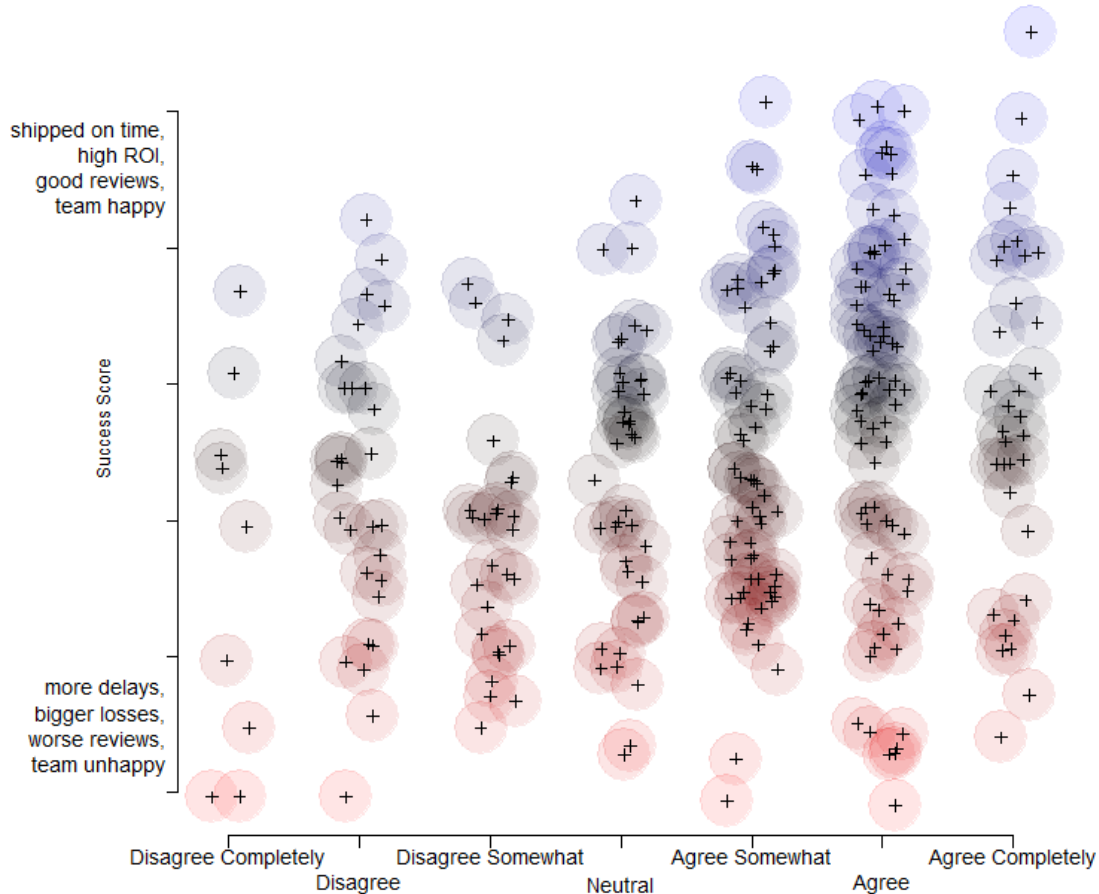
Maybe Good:

- Coaching
- Helpful team
- Unfiltered discussions
- Team experience
- Mutual accountability
- Good dev equipment
- Whole team involved in prioritization of work

Maybe Bad:

- New engine
- Crunch
- Big tech changes
- Leaders suppress “bad” ideas
- Lack of planning/coordination

Team members were held accountable for meeting their deadlines.



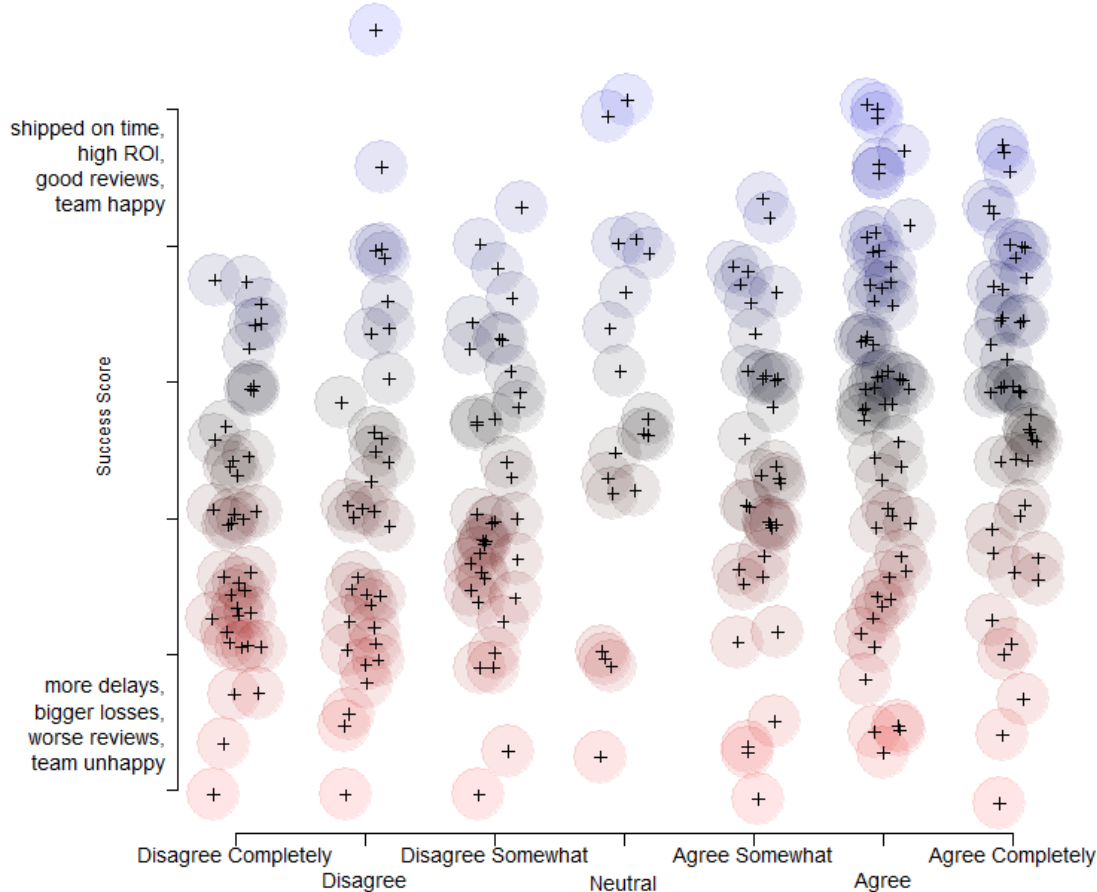
Maybe Good:

- Coaching
- Helpful team
- Unfiltered discussions
- Team experience
- Mutual accountability
- Good dev equipment
- Whole team involved in prioritization of work

Maybe Bad:

- New engine
- Crunch
- Big tech changes
- Leaders suppress “bad” ideas
- Lack of planning/coordination

The team composition didn't change during the course of the project (other than growing when needed).



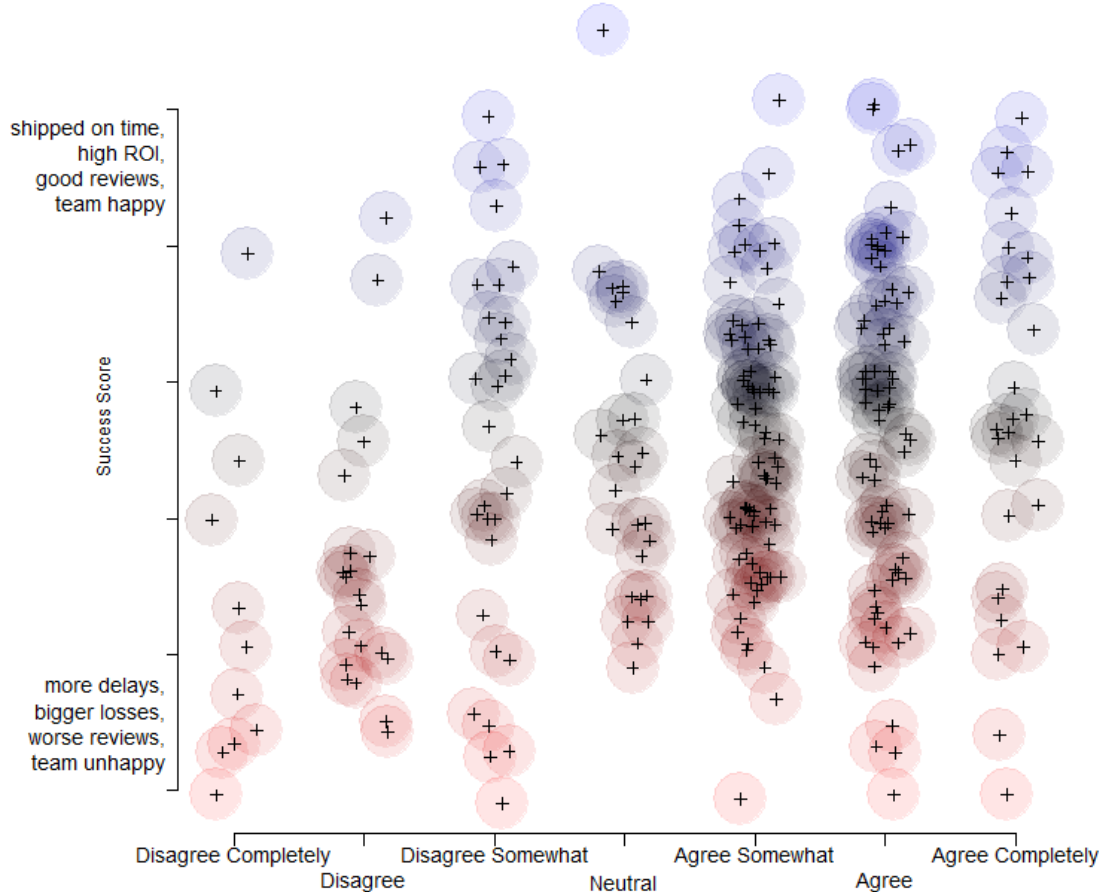
Maybe Good:

- Coaching
- Helpful team
- Unfiltered discussions
- Team experience
- Mutual accountability
- Good dev equipment
- Whole team involved in prioritization of work
- Stable team

Maybe Bad:

- New engine
- Crunch
- Big tech changes
- Leaders suppress “bad” ideas
- Lack of planning/coordination

Most team members were able to determine their own work processes and workflow.



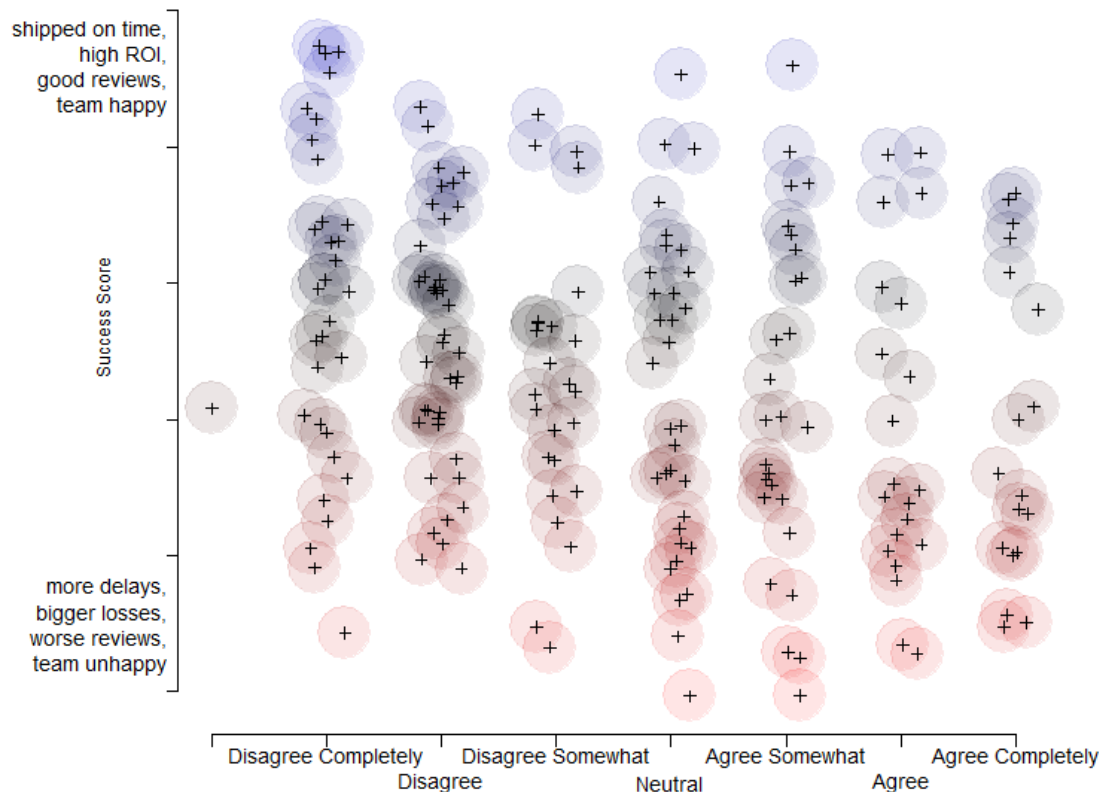
Maybe Good:

- Coaching
- Helpful team
- Unfiltered discussions
- Team experience
- Mutual accountability
- Good dev equipment
- Whole team involved in prioritization of work
- Stable team

Maybe Bad:

- New engine
- Crunch
- Big tech changes
- Leaders suppress “bad” ideas
- Lack of planning/coordination
- Rigid processes/workflow

The portions of the project that were outsourced to external developers were not worth the resources spent and yielded a generally unsatisfactory product (leave blank if no outsourcing).



Maybe Good:

- Coaching
- Helpful team
- Unfiltered discussions
- Team experience
- Mutual accountability
- Good dev equipment
- Whole team involved in prioritization of work
- Stable team

Maybe Bad:

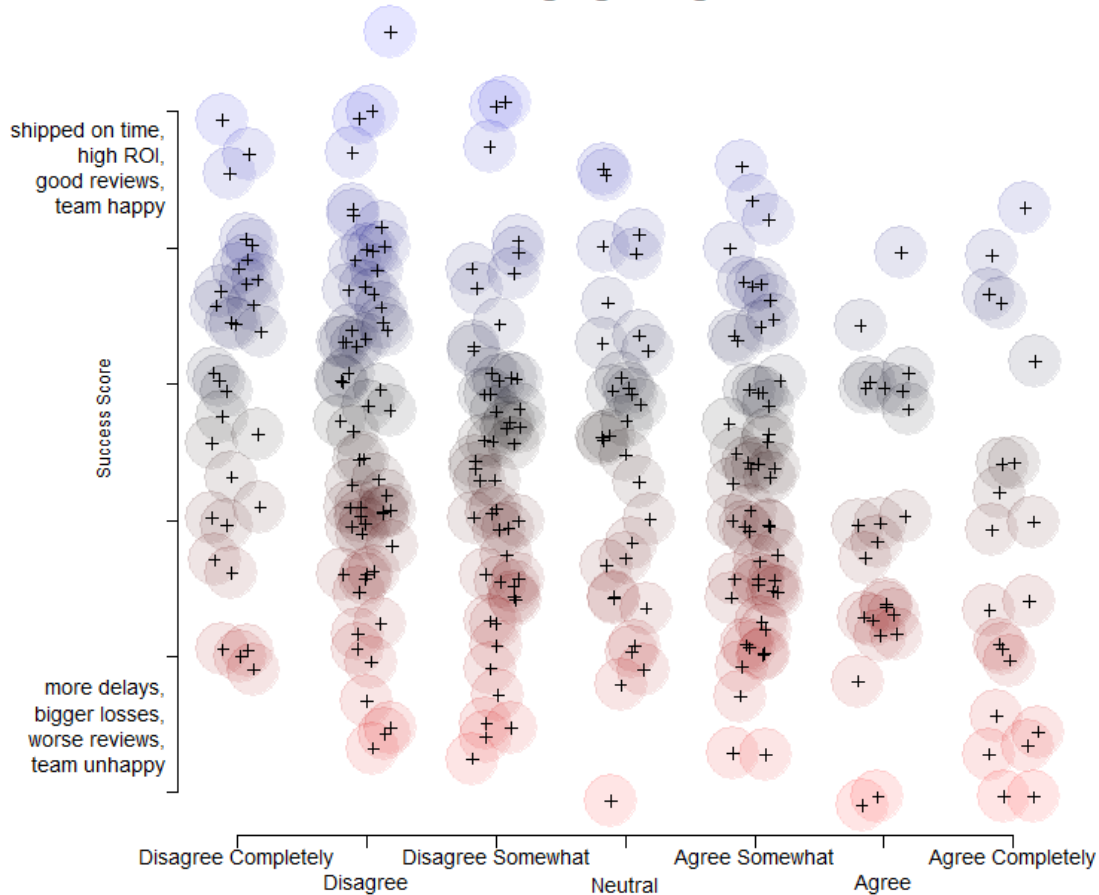
- New engine
- Crunch
- Big tech changes
- Leaders suppress “bad” ideas
- Lack of planning/coordination
- Rigid processes/workflow

Part 3 Main Points

- You can maybe get a leg up by doing these:
 - Re-use tech from previous game, avoid big tech changes
 - Frequent unfiltered/open discussion
 - Stable team with clear responsibilities
 - Skill development/coaching
 - Mutual accountability

Part 4: Things that probably help/hurt

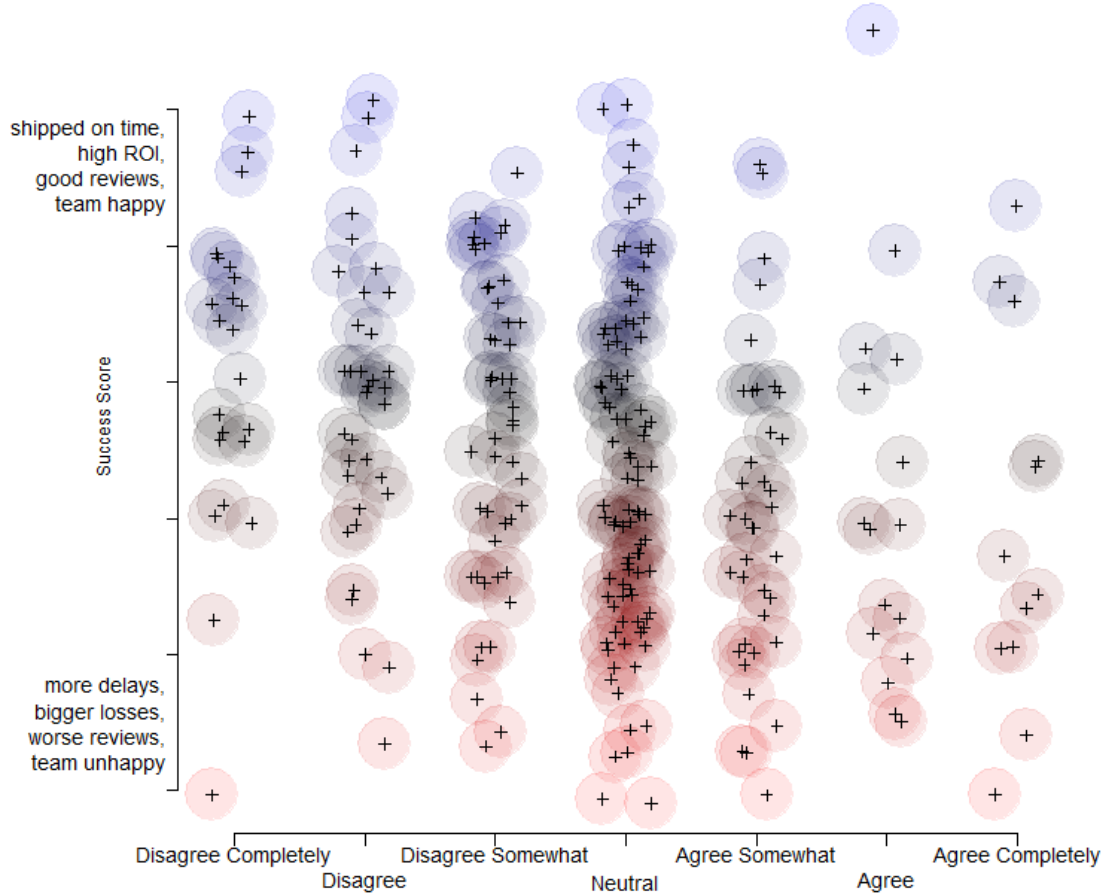
It was difficult to get or give honest feedback without feelings getting hurt.



Good:

- Being able to receive feedback

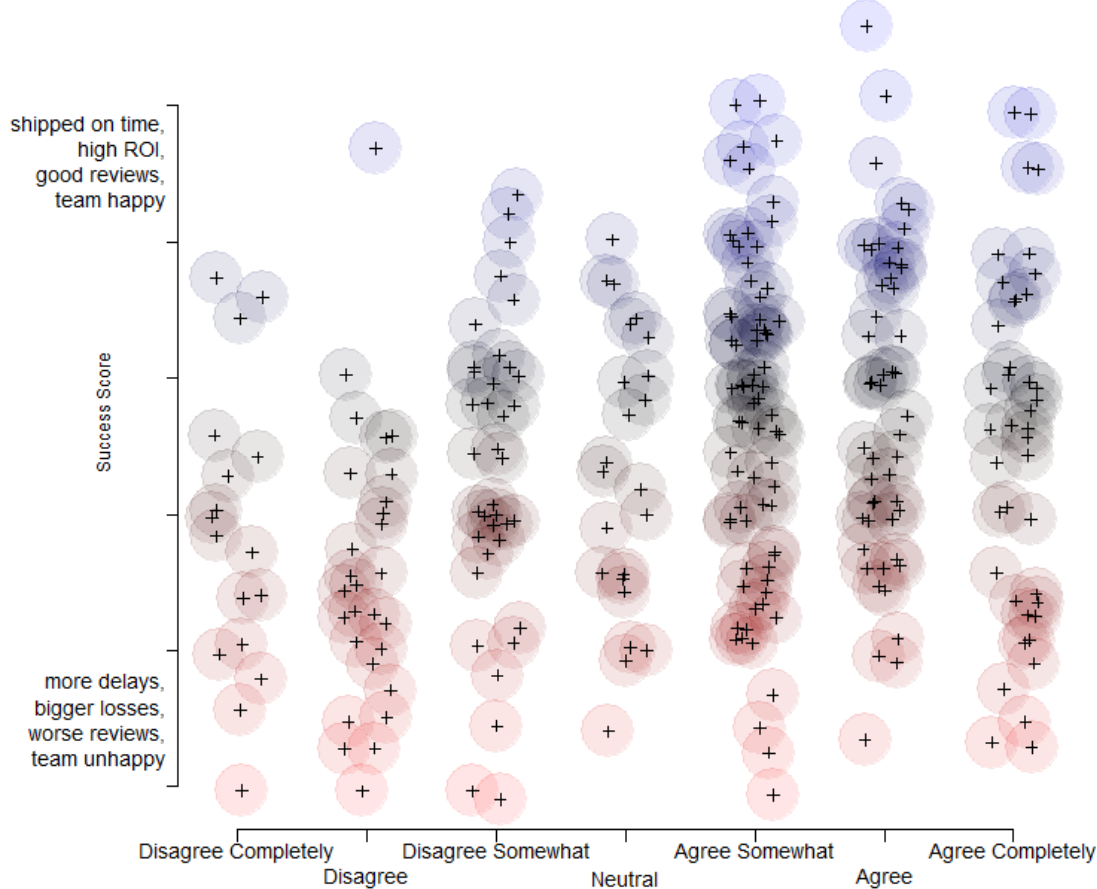
People tended to keep failed ideas to themselves in this organization.



Good:

- Being able to receive feedback
- Fearless communication

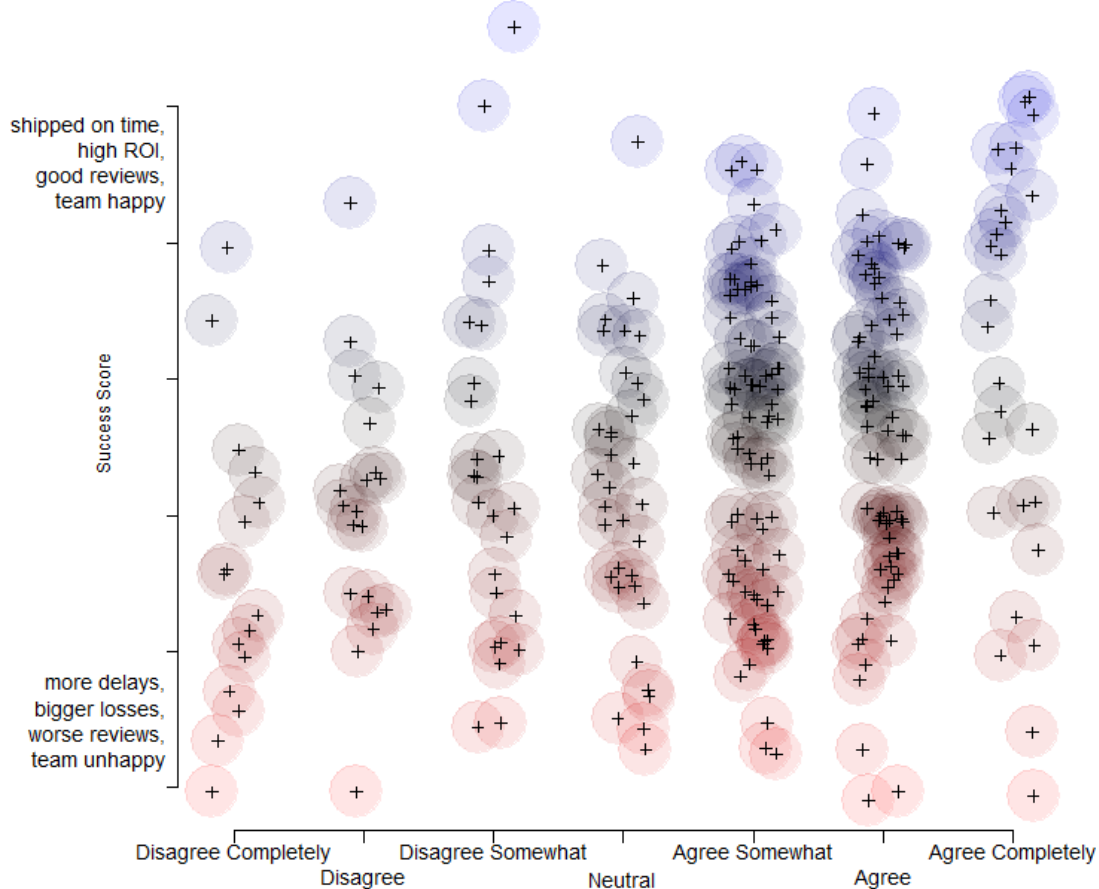
The entire team met frequently to openly discuss topics of interest and ask questions.



Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A

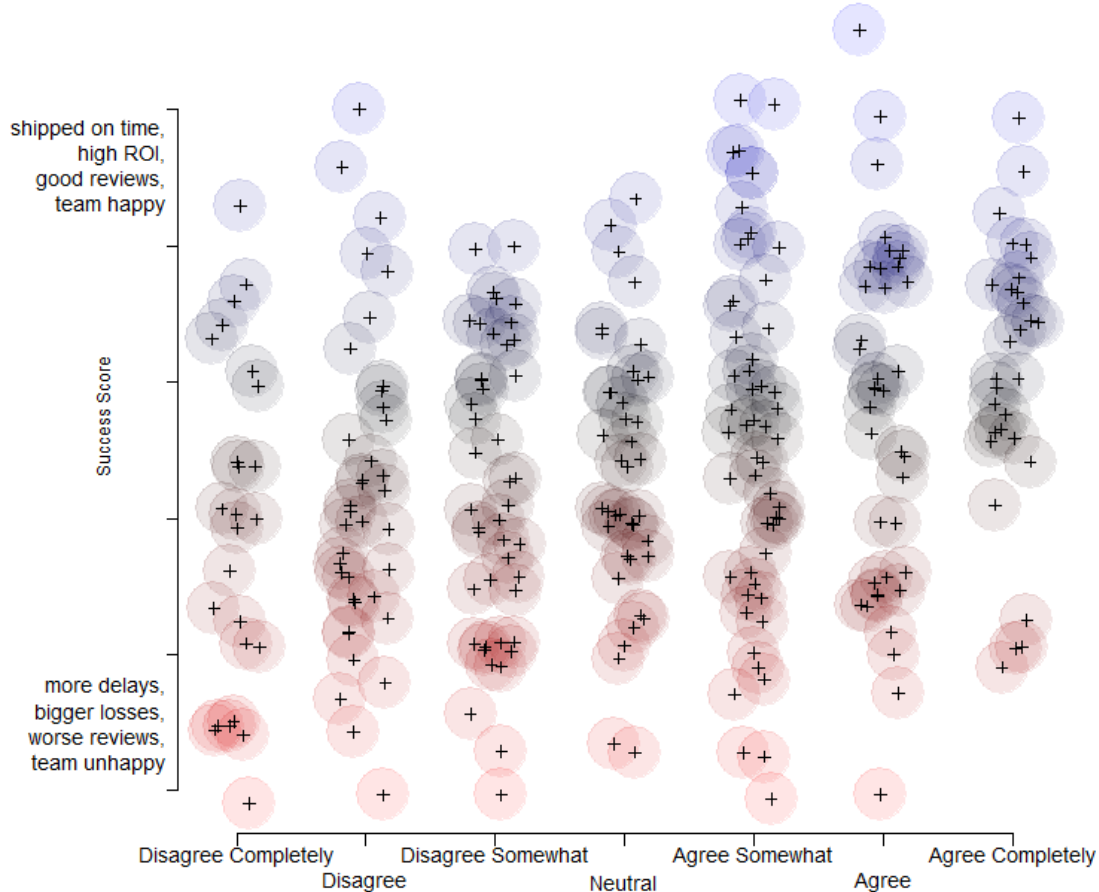
The tools we used generally worked well and allowed us to be productive.



Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well

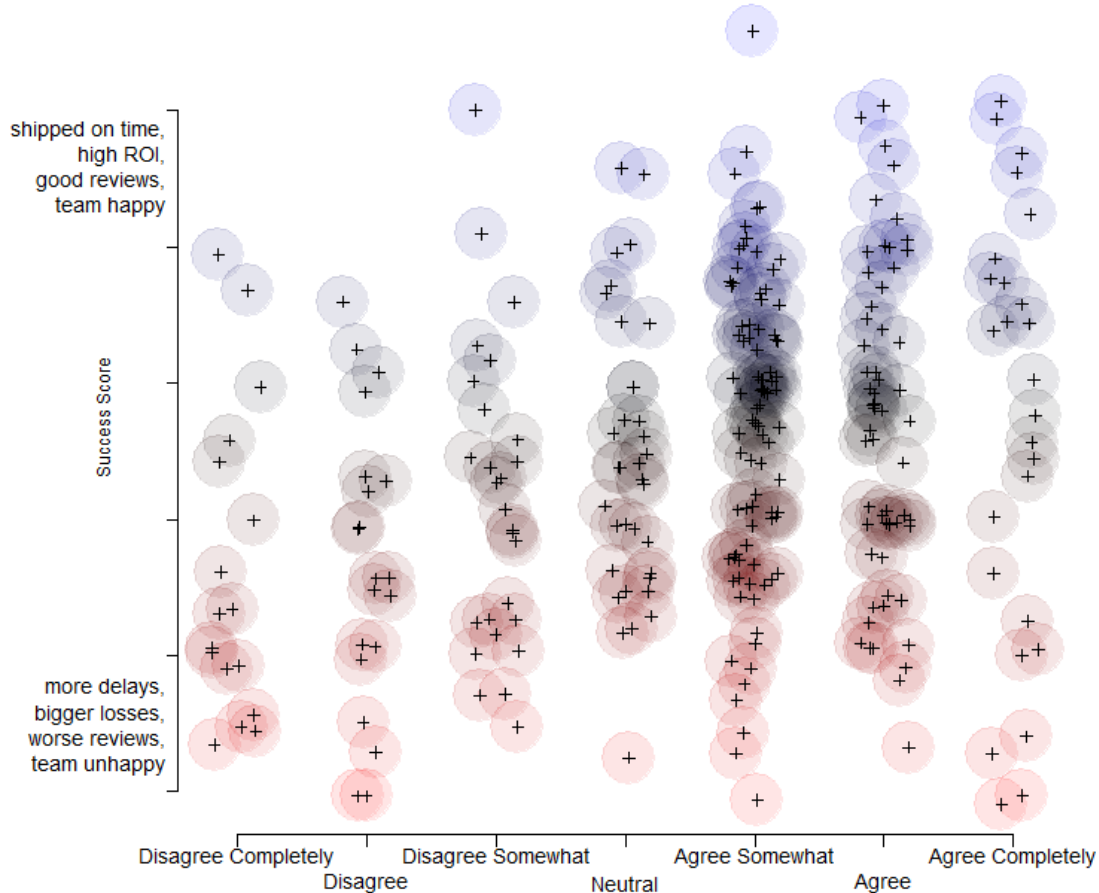
The project leads frequently called out when team members performed well.



Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public positive feedback

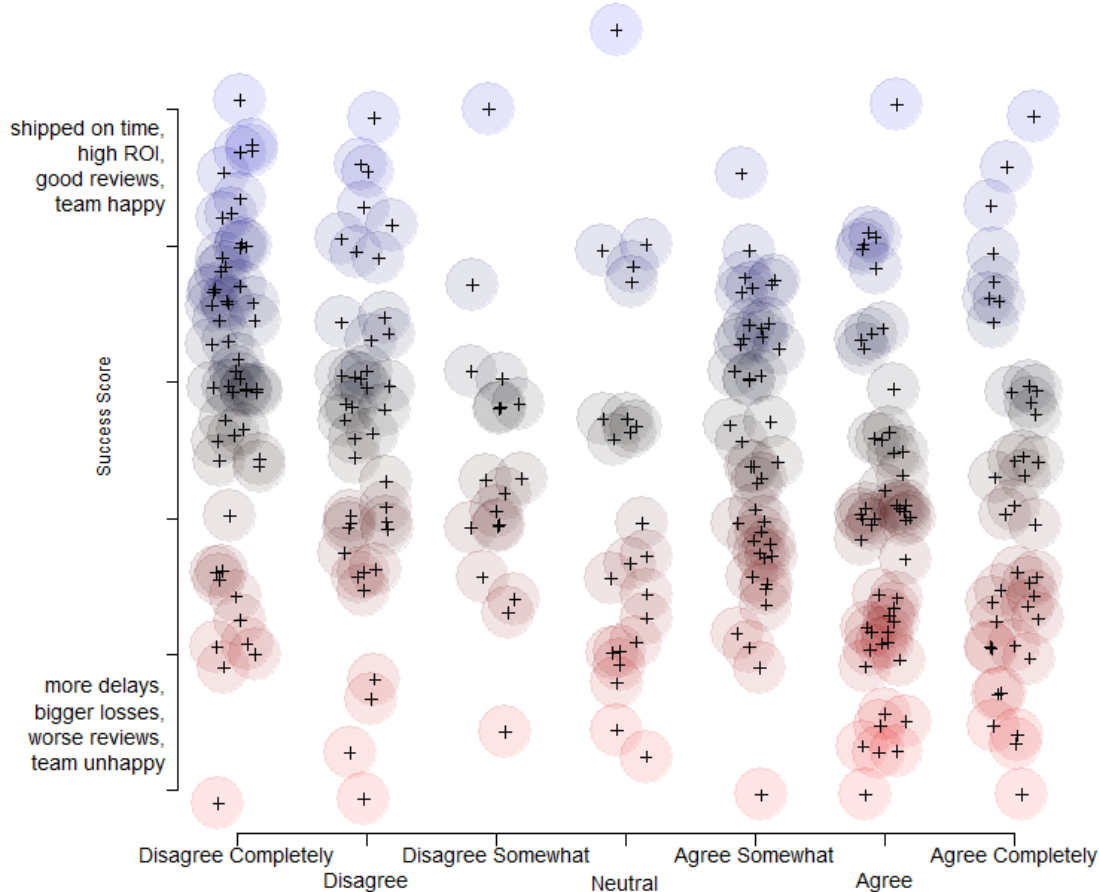
Team members' tasks were well-defined and clearly specified.



Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public positive feedback
- Well-defined tasks

The team experienced one or more reorganizations during development that changed its composition or its structure.



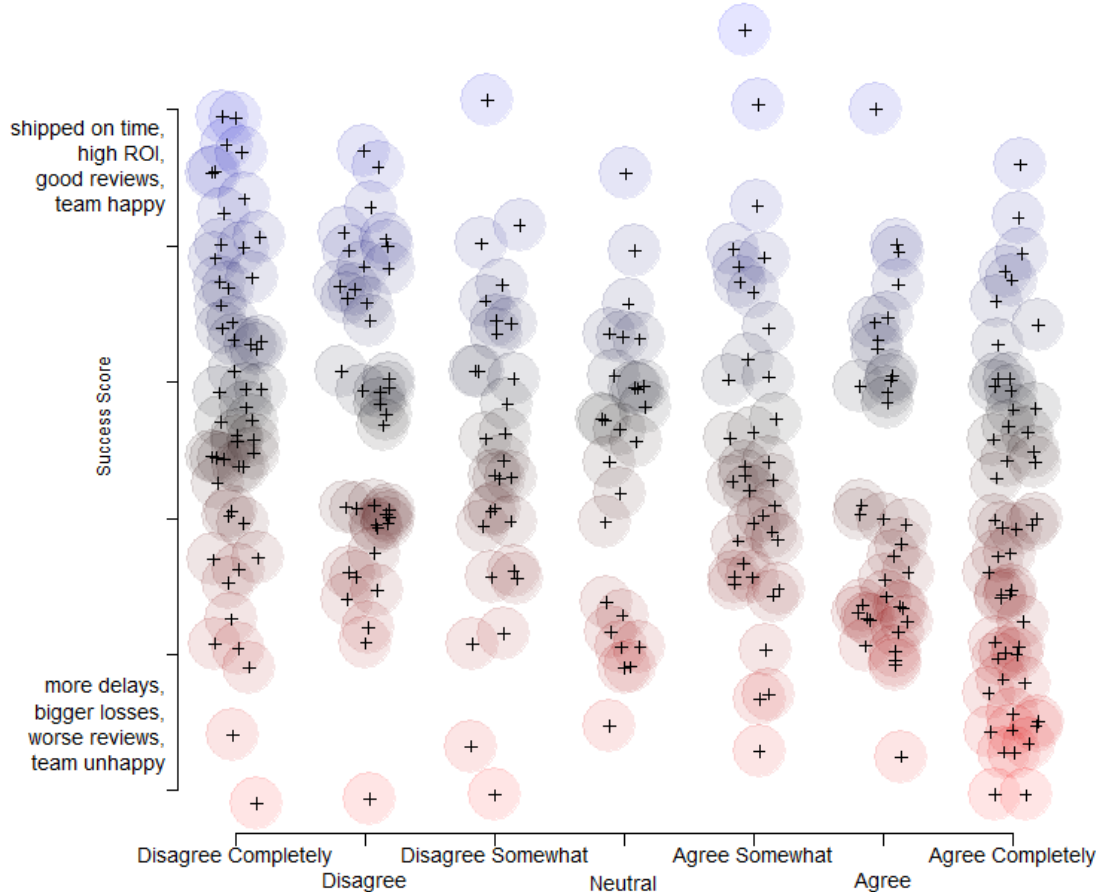
Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public positive feedback
- Well-defined tasks

Bad:

- Team reorganizations

I often worked overtime because I was required or felt pressured to.



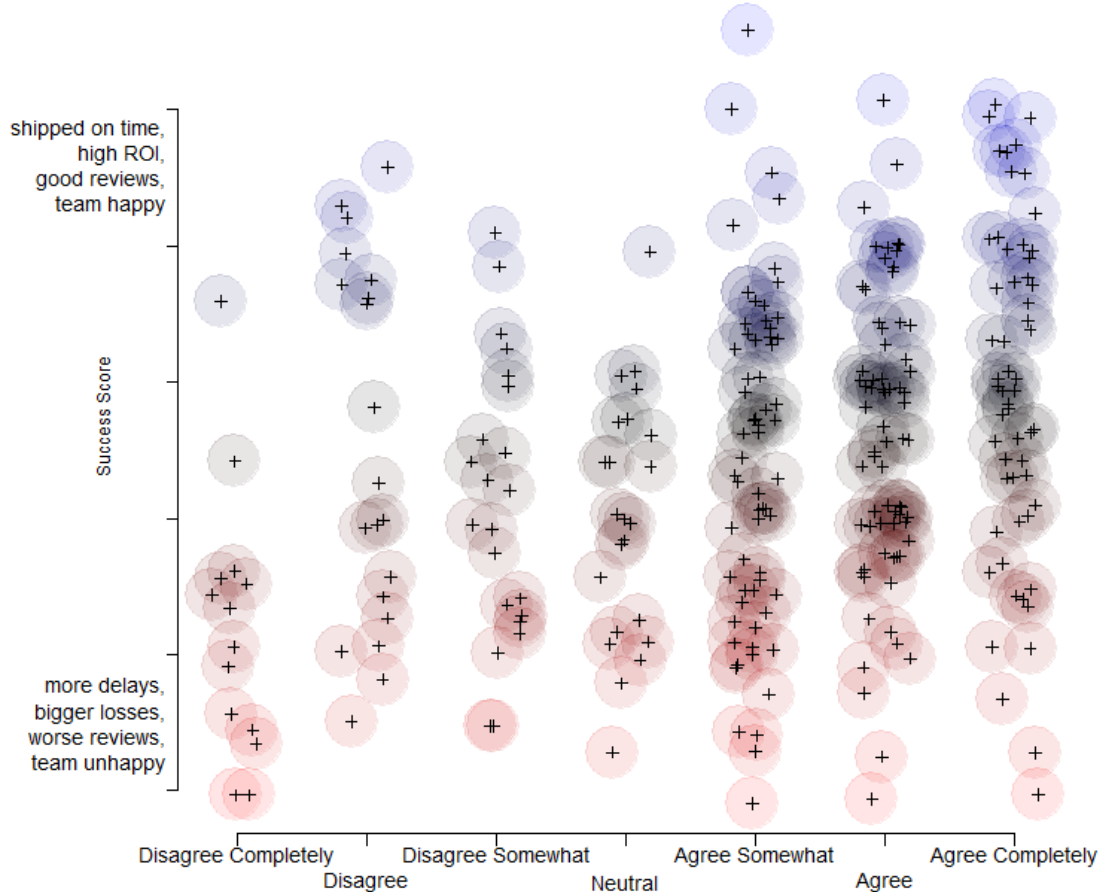
Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public positive feedback
- Well-defined tasks

Bad:

- Team reorganizations
- Required/pressured overtime

**The team gave me plenty of opportunities to learn, grow,
and improve my skill set.**



Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public positive feedback
- Well-defined tasks
- Opportunities for growth

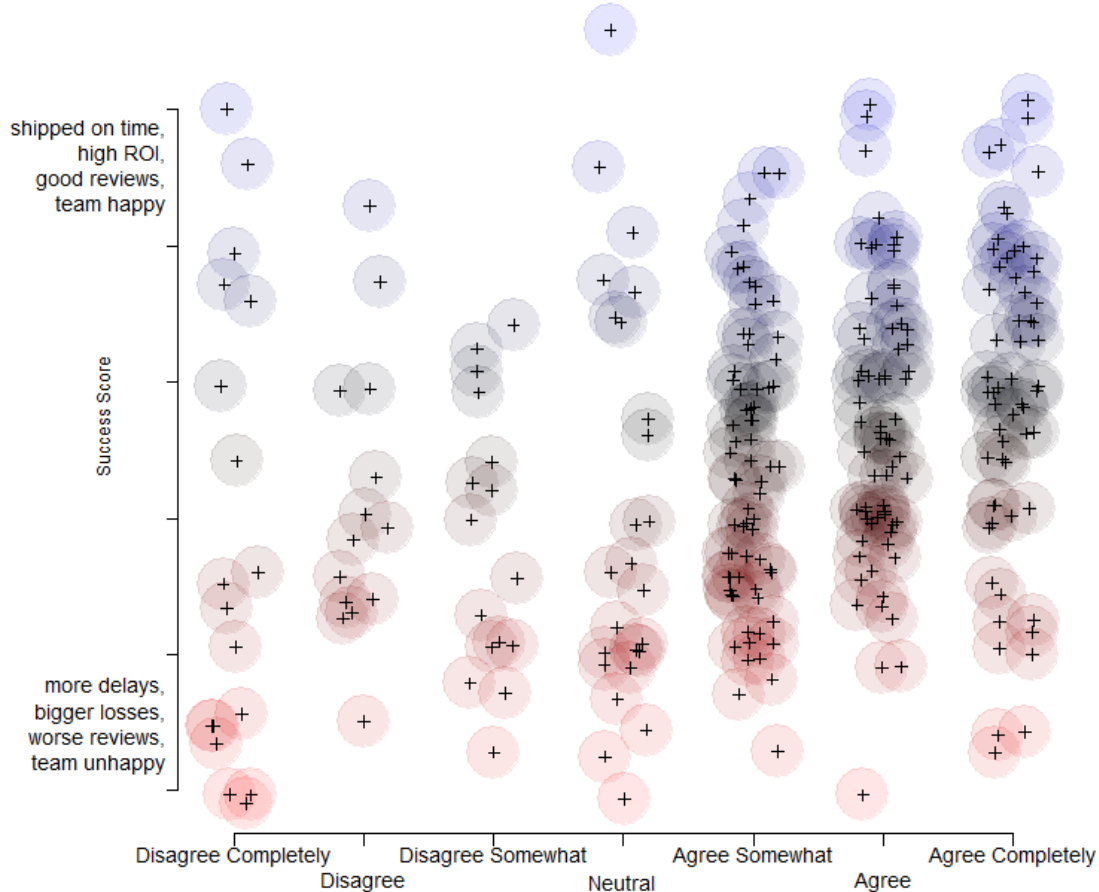
Bad:

- Team reorganizations
- Required/pressured overtime

Side Story: Growth and Training

- “The team gave me plenty of **opportunities** to learn, grow and improve my skill set” has a higher correlation than
- “There was someone at this organization who **encouraged** me to develop my skills further” which has a higher correlation than
- “We received some form of **coaching** or guidance to enhance our effort or improve our effectiveness as a team.”
- Would you have put these in this order?

My unique skills and talents were valued and utilized while working on this team.



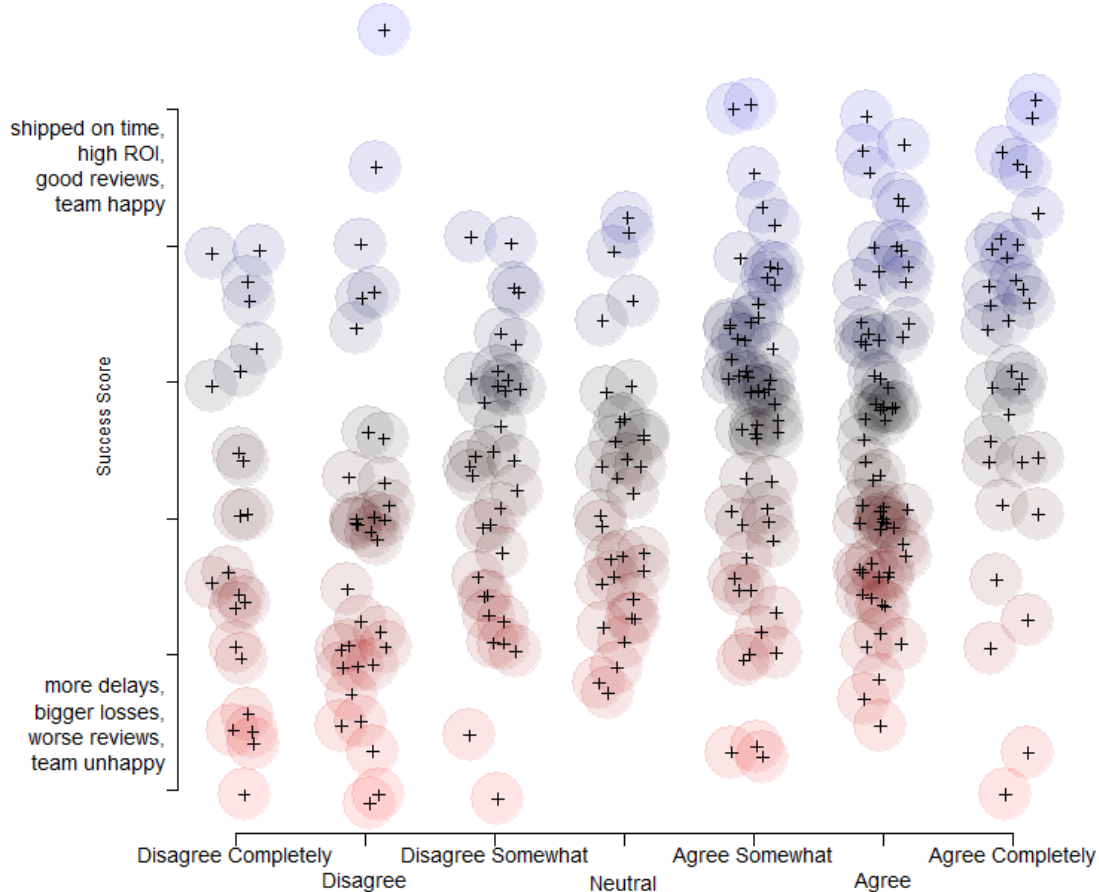
Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public positive feedback
- Well-defined tasks
- Opportunities for growth

Bad:

- Team reorganizations
- Required/pressured overtime

The organizational structure and membership of the team were clear from the outset of the project.



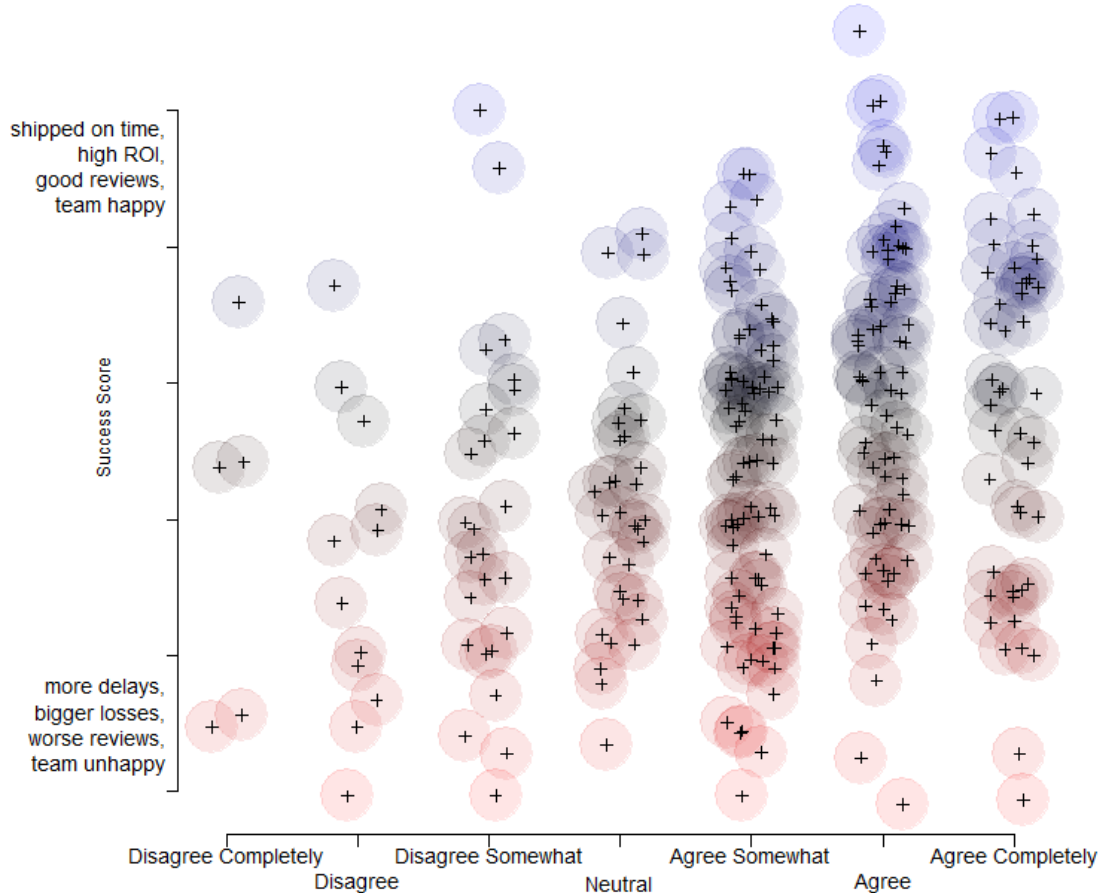
Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public positive feedback
- Well-defined tasks
- Opportunities for growth

Bad:

- Team reorganizations
- Required/pressured overtime

Team members often praised each other for a job well done.



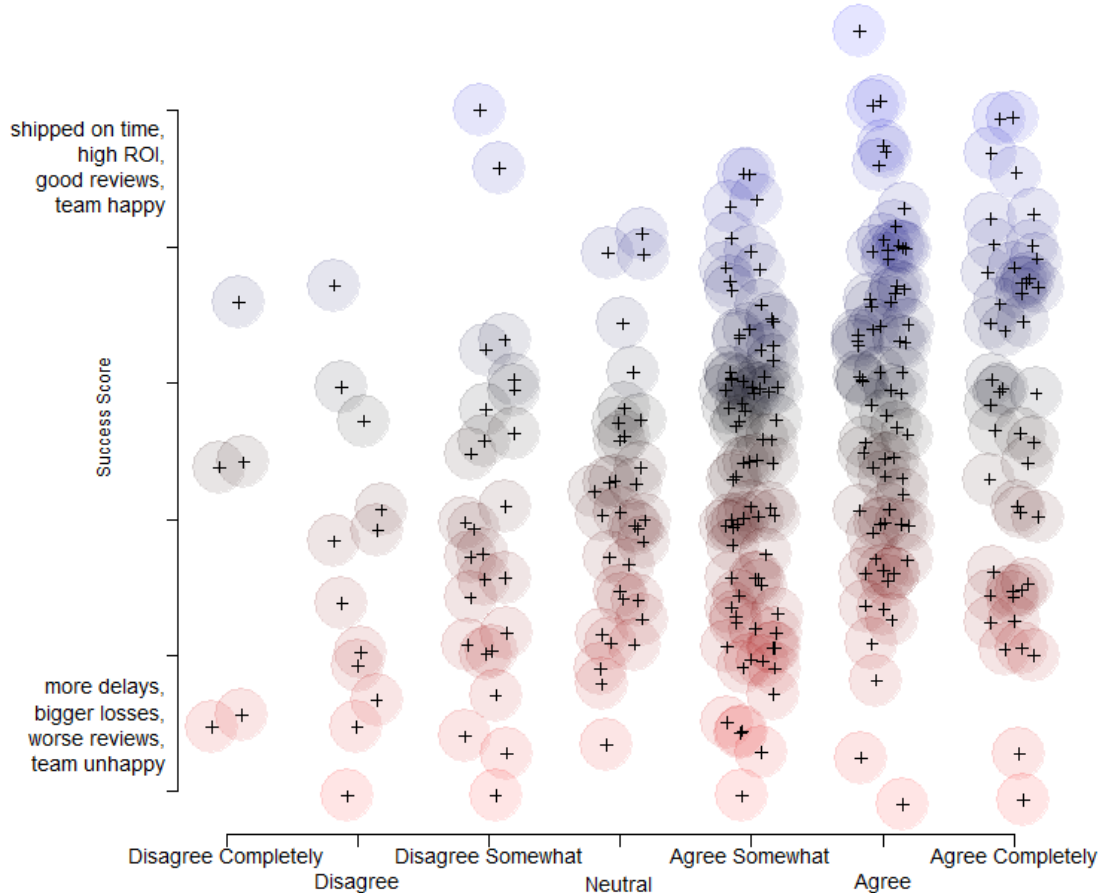
Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public positive feedback
- Well-defined tasks
- Opportunities for growth

Bad:

- Team reorganizations
- Required/pressured overtime

Team members often praised each other for a job well done.



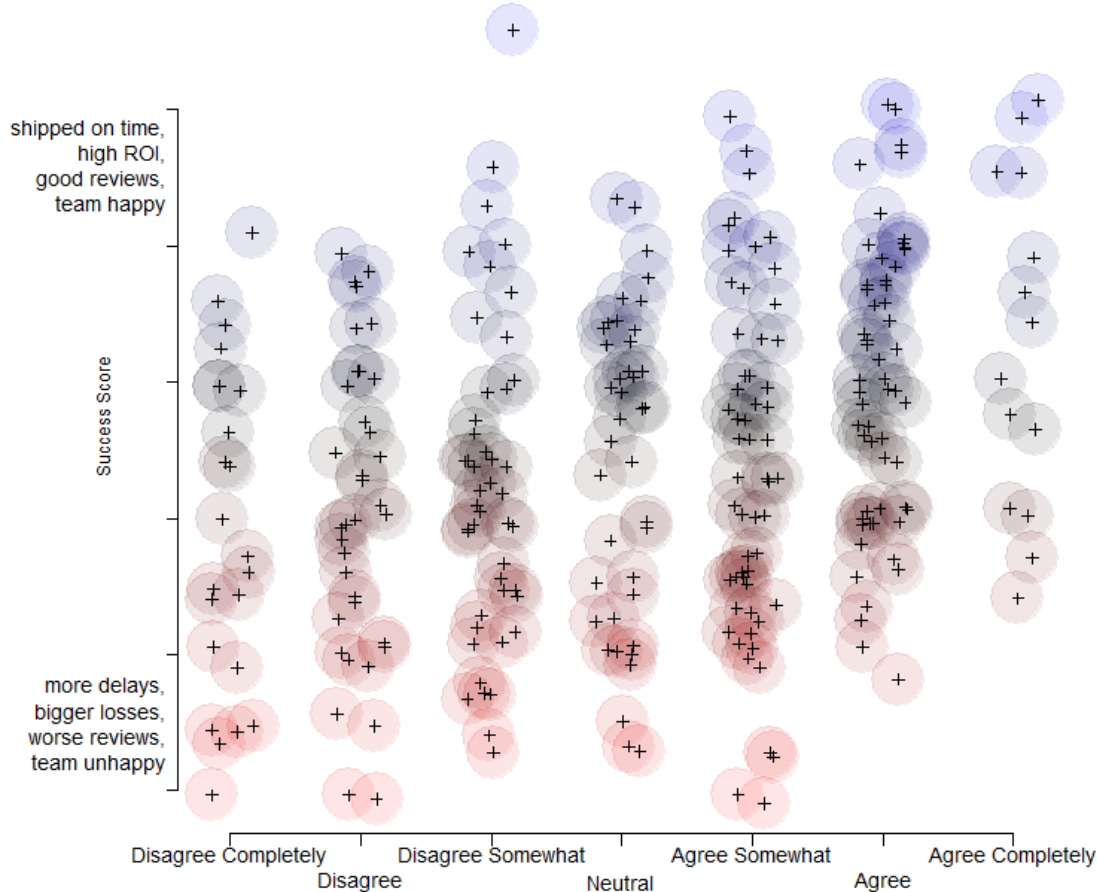
Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public/mutual positive feedback
- Well-defined tasks
- Opportunities for growth

Bad:

- Team reorganizations
- Required/pressured overtime

Most team members were well-trained in our studio's production methodologies.



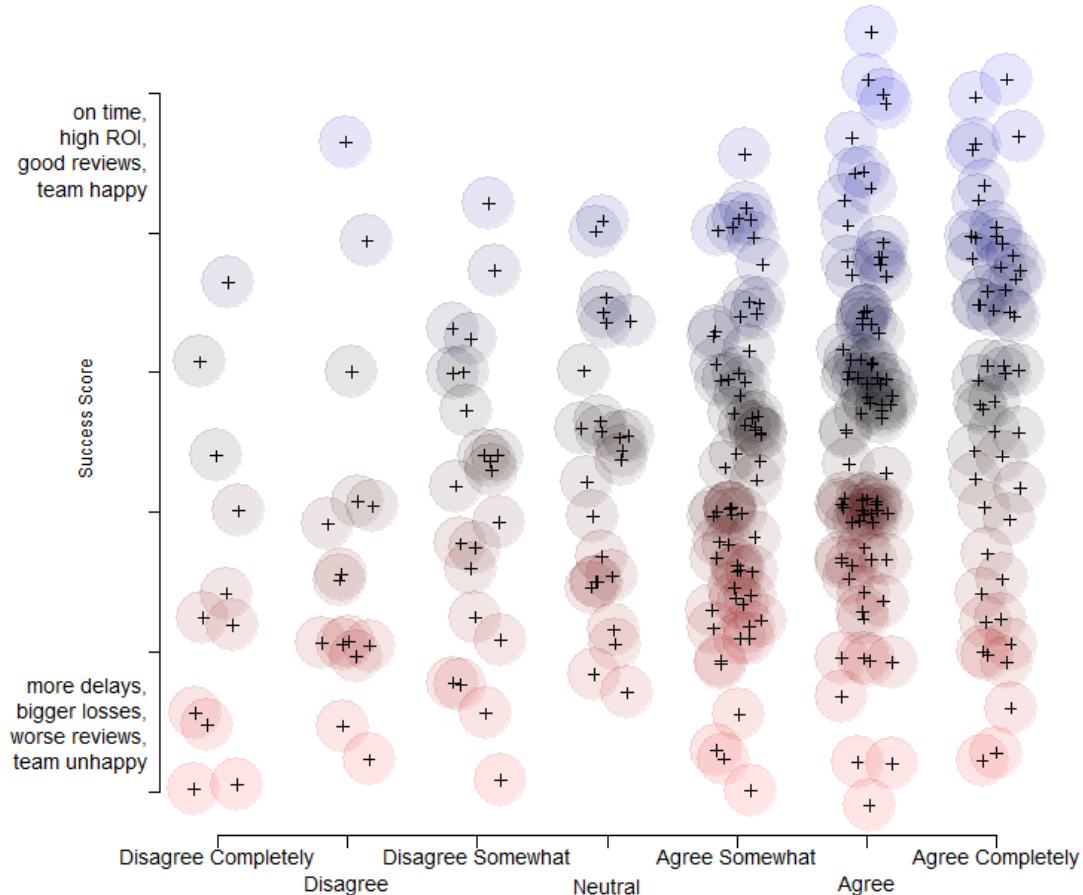
Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public/mutual positive feedback
- Well-defined tasks
- Opportunities for growth
- Explain how the studio works

Bad:

- Team reorganizations
- Required/pressured overtime

I understood clearly what was expected of me as a team member.



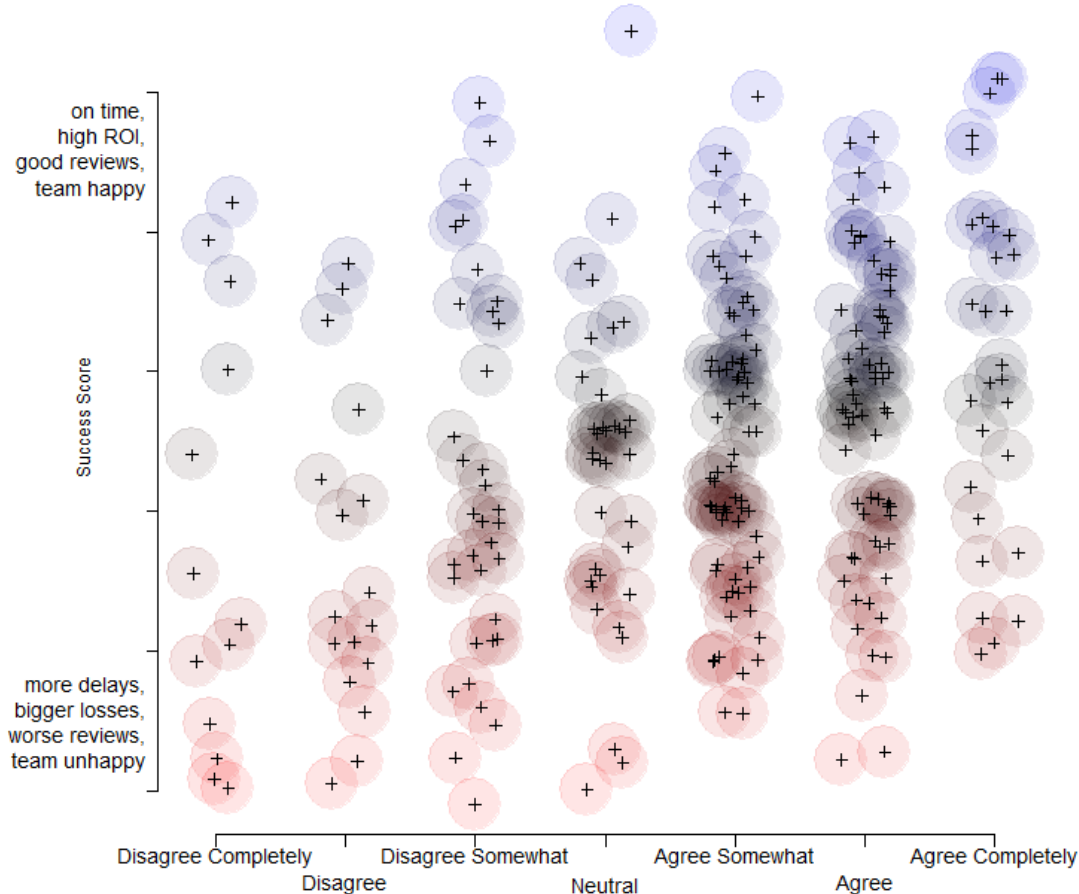
Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public/mutual positive feedback
- Well-defined tasks
- Opportunities for growth
- Explain how the studio works

Bad:

- Team reorganizations
- Required/pressured overtime

Team members' responsibilities and job roles were carefully matched with their particular skills and abilities.



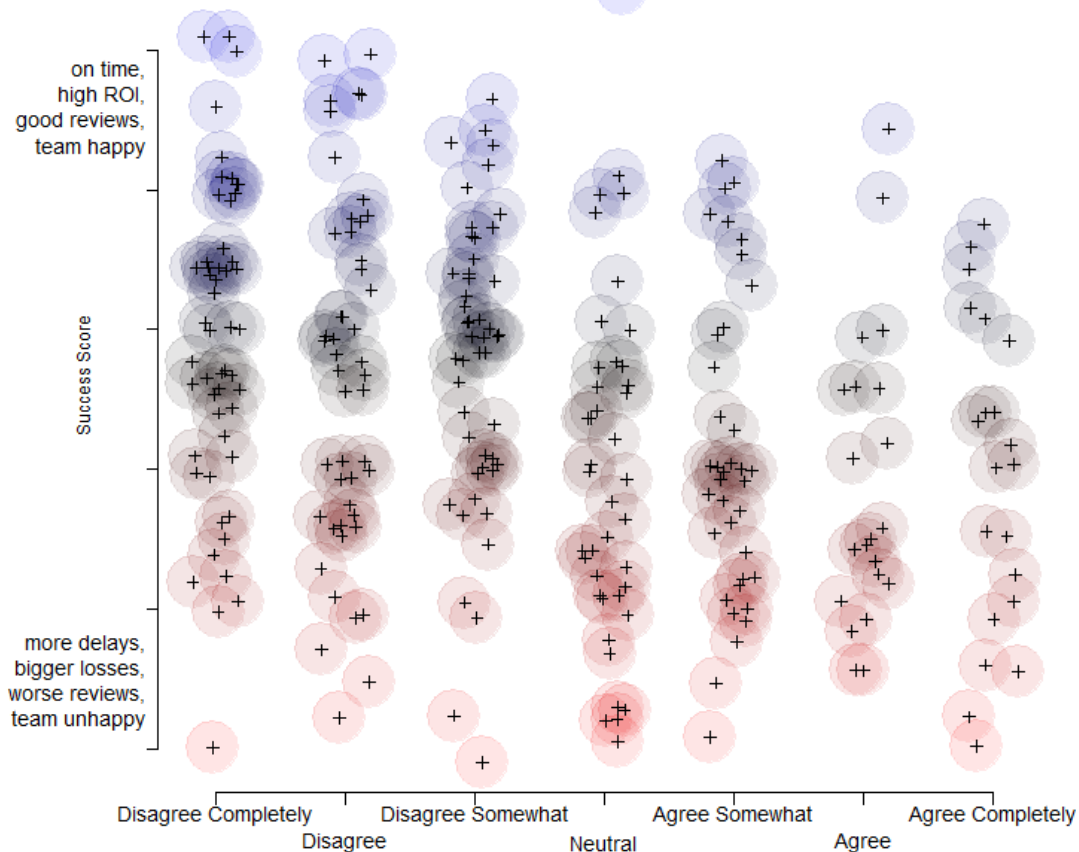
Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public/mutual positive feedback
- Well-defined tasks
- Opportunities for growth
- Explain how the studio works

Bad:

- Team reorganizations
- Required/pressured overtime

As a result of the separation of disciplines, smaller teams often fought for different priorities rather than collaborating.



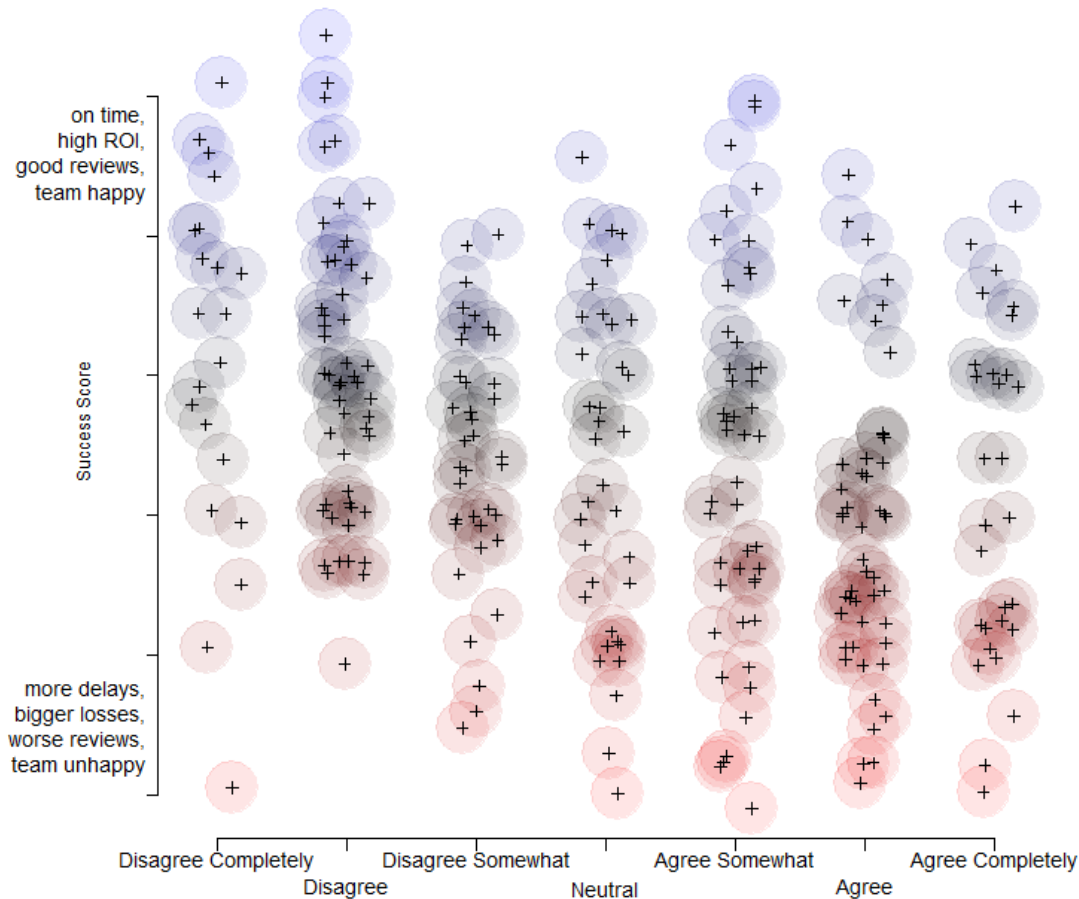
Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public/mutual positive feedback
- Well-defined tasks
- Opportunities for growth
- Explain how the studio works

Bad:

- Team reorganizations
- Required/pressured overtime
- Competing priorities between teams

Some important things went unsaid during development because the team feared conflict.



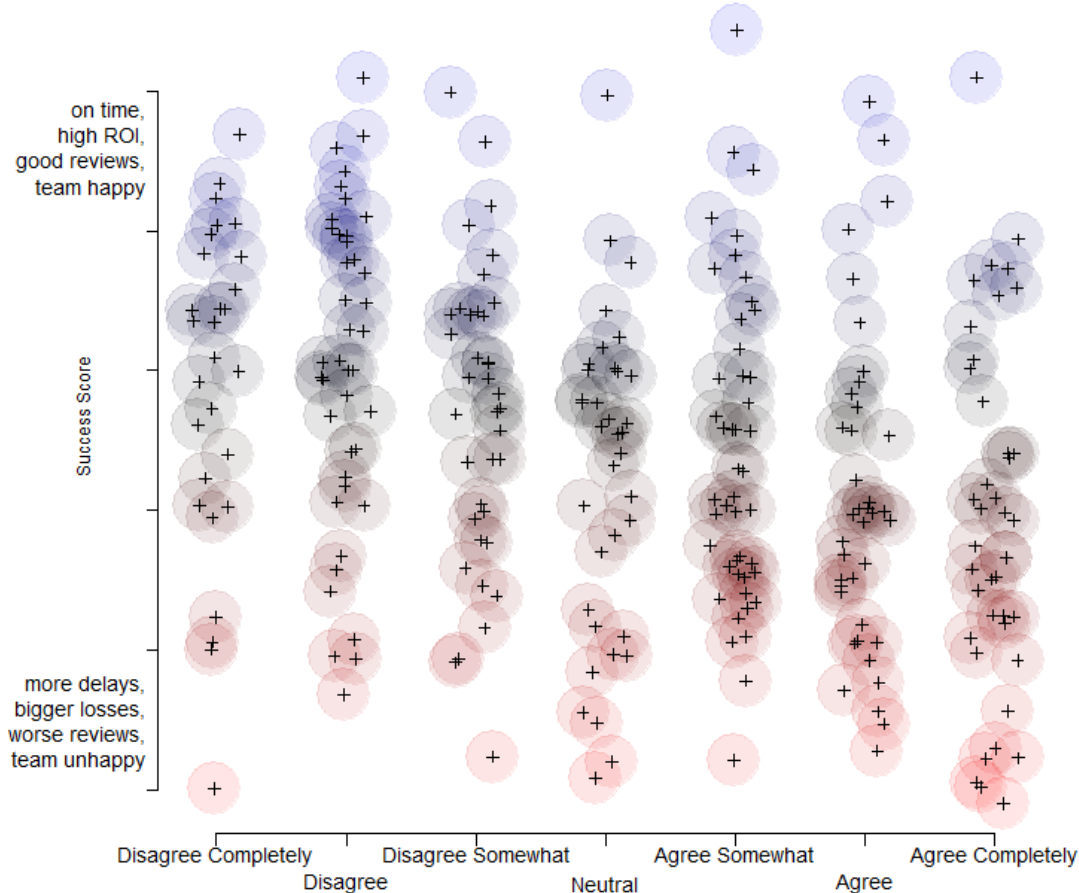
Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public/mutual positive feedback
- Well-defined tasks
- Opportunities for growth
- Explain how the studio works

Bad:

- Team reorganizations
- Required/pressured overtime
- Competing priorities between teams

Some team members put their ego or their own need for recognition above the collective goals of the game project.



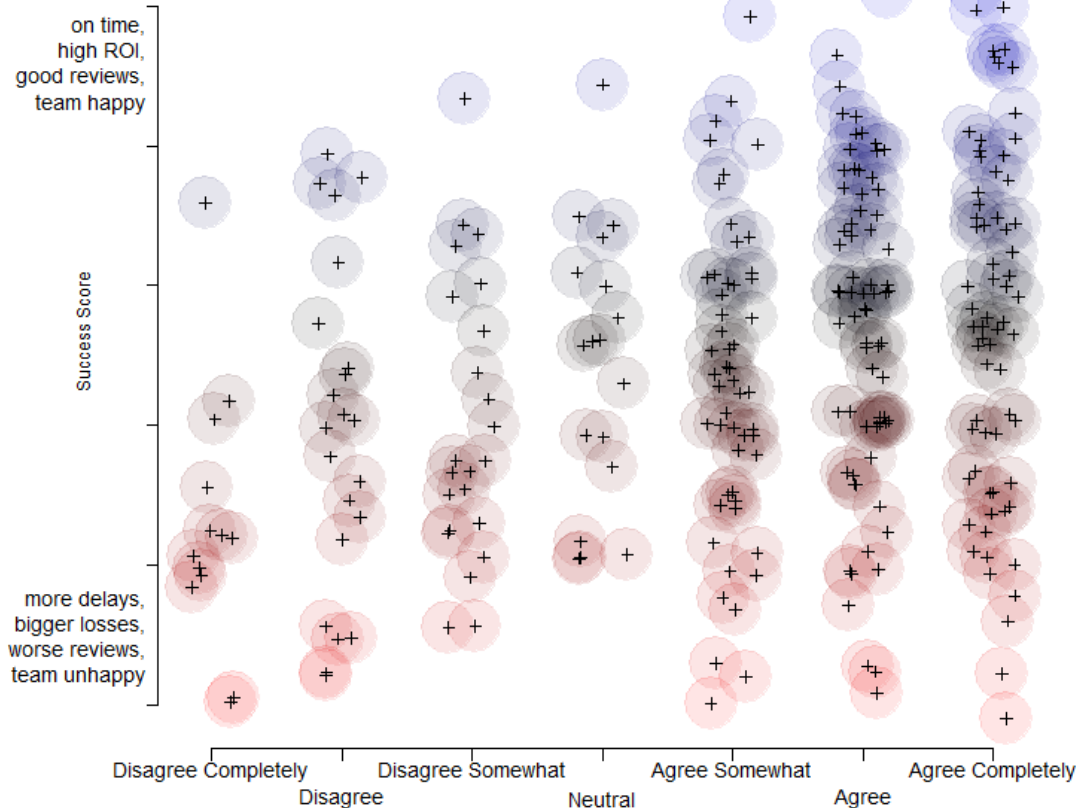
Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public/mutual positive feedback
- Well-defined tasks
- Opportunities for growth
- Explain how the studio works
- Collective goals

Bad:

- Team reorganizations
- Required/pressured overtime
- Competing priorities between teams

Members had easy access to communicate with senior leadership about the project, raise concerns, or discuss personal issues.



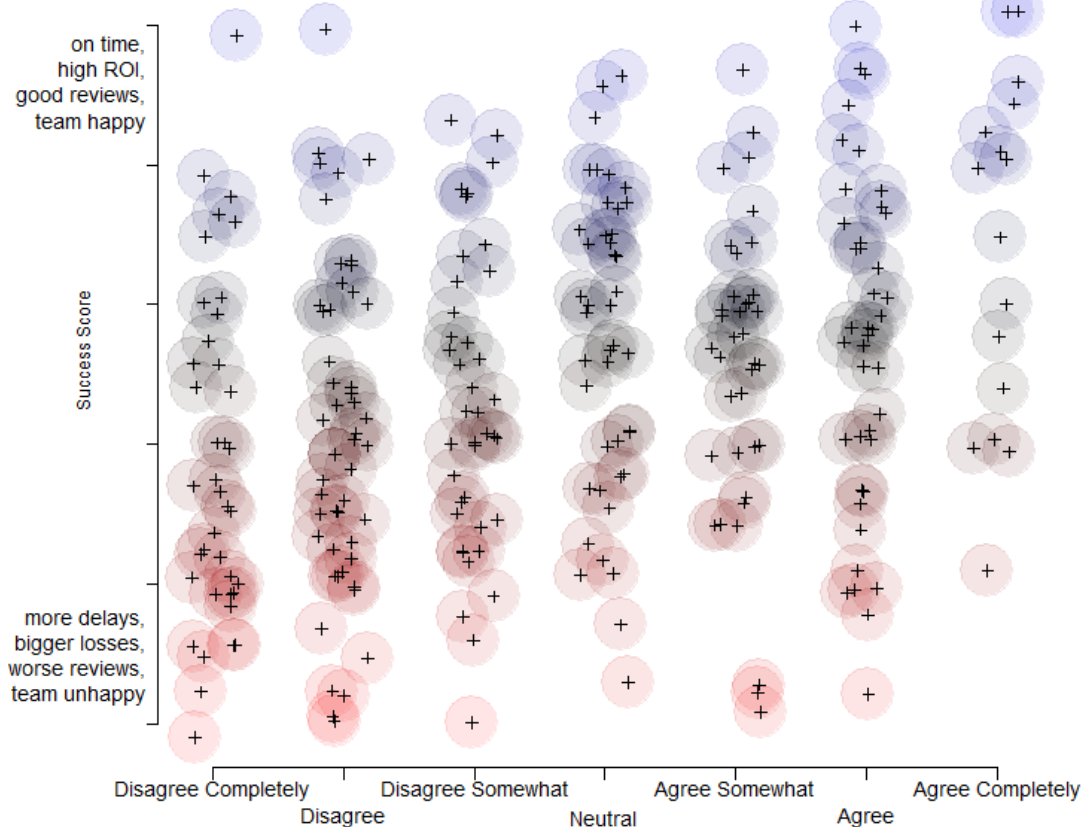
Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public/mutual positive feedback
- Well-defined tasks
- Opportunities for growth
- Explain how the studio works
- Collective goals
- Accessible leads/seniors

Bad:

- Team reorganizations
- Required/pressured overtime
- Competing priorities between teams

When significant changes to the core design or architecture of the game occurred, all stakeholders in the studio participated in the decision to make changes.



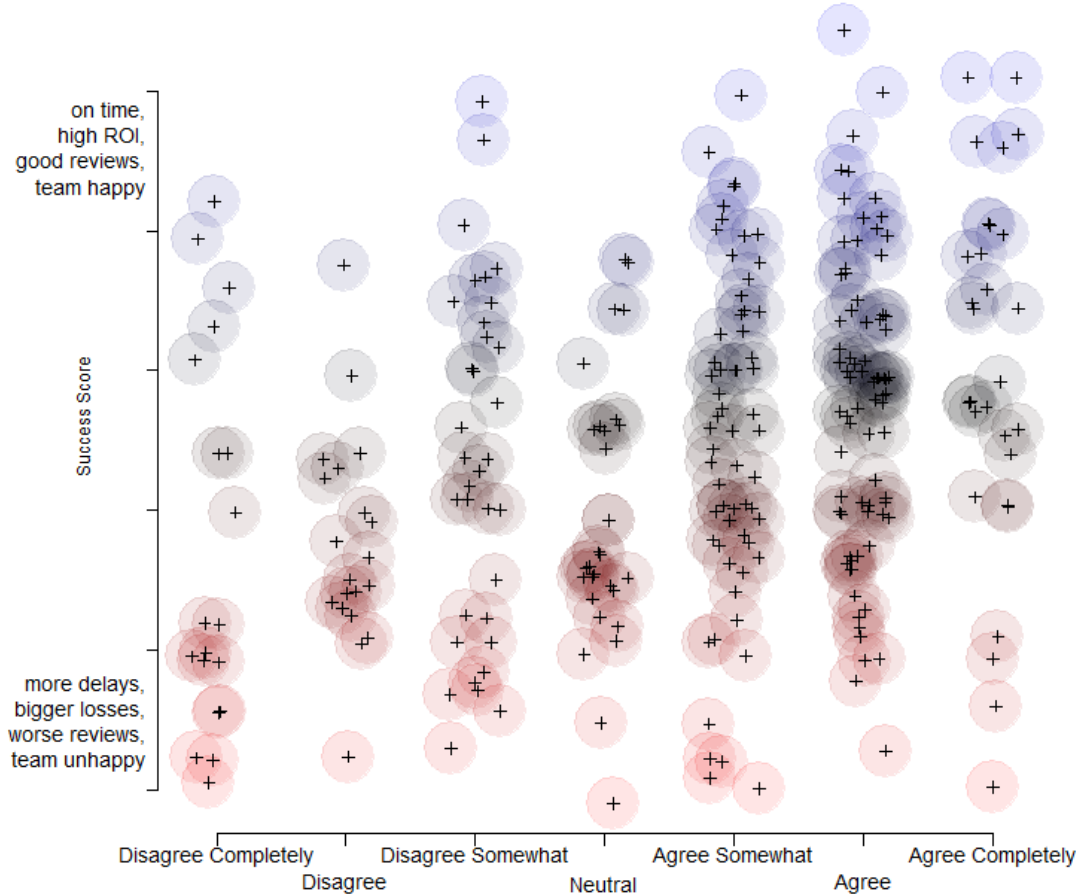
Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public/mutual positive feedback
- Well-defined tasks
- Opportunities for growth
- Explain how the studio works
- Collective goals
- Accessible leads/seniors

Bad:

- Team reorganizations
- Required/pressured overtime
- Competing priorities between teams

Mistakes were treated as learning opportunities or a chance to improve, not a nail in your coffin.



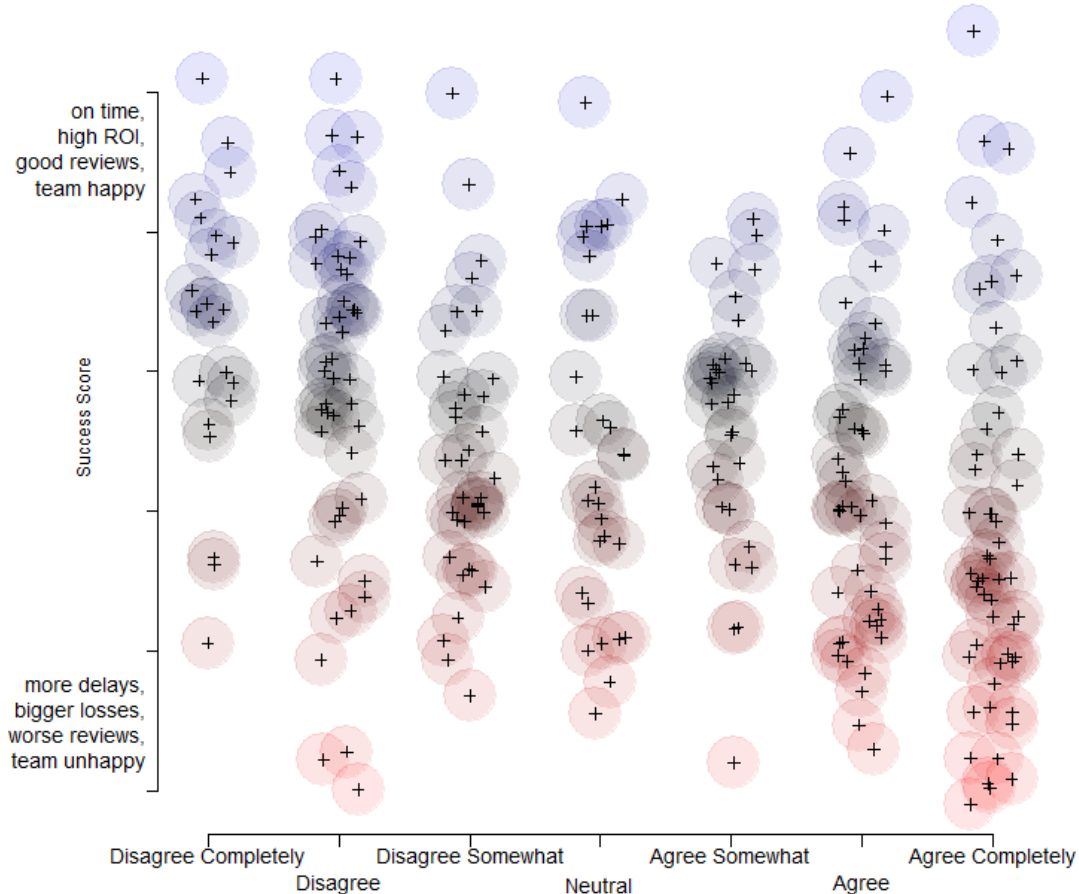
Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public/mutual positive feedback
- Well-defined tasks
- Opportunities for growth
- Explain how the studio works
- Collective goals
- Accessible leads/seniors

Bad:

- Team reorganizations
- Required/pressured overtime
- Competing priorities between teams

Internal politics played a big role in the dynamics of the team.



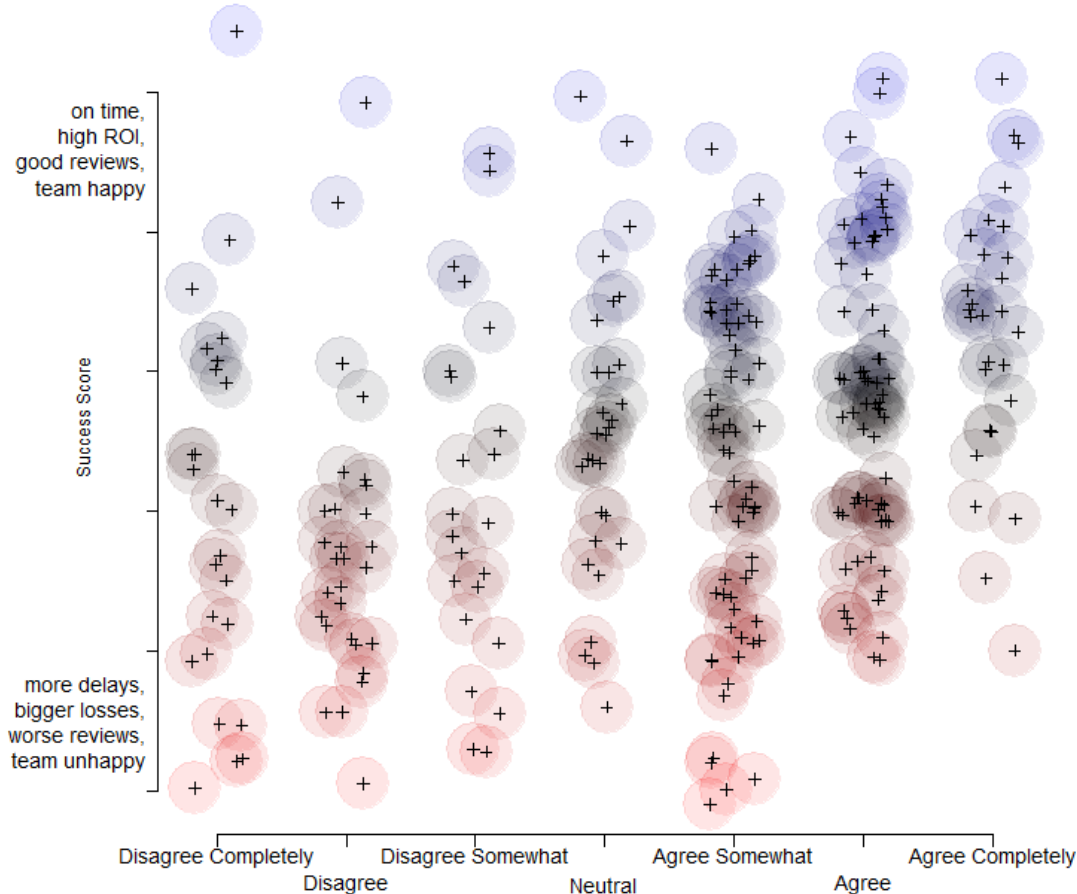
Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public/mutual positive feedback
- Well-defined tasks
- Opportunities for growth
- Explain how the studio works
- Collective goals
- Accessible leads/seniors

Bad:

- Team reorganizations
- Required/pressured overtime
- Competing priorities between teams

I felt confident that if I expressed an opinion, people would listen, and I could change their minds.



Good:

- Being able to receive feedback
- Fearless communication
- Whole team open discussion or Q&A
- Tools that work well
- Public/mutual positive feedback
- Well-defined tasks
- Opportunities for growth
- Explain how the studio works
- Collective goals
- Accessible leads/seniors

Bad:

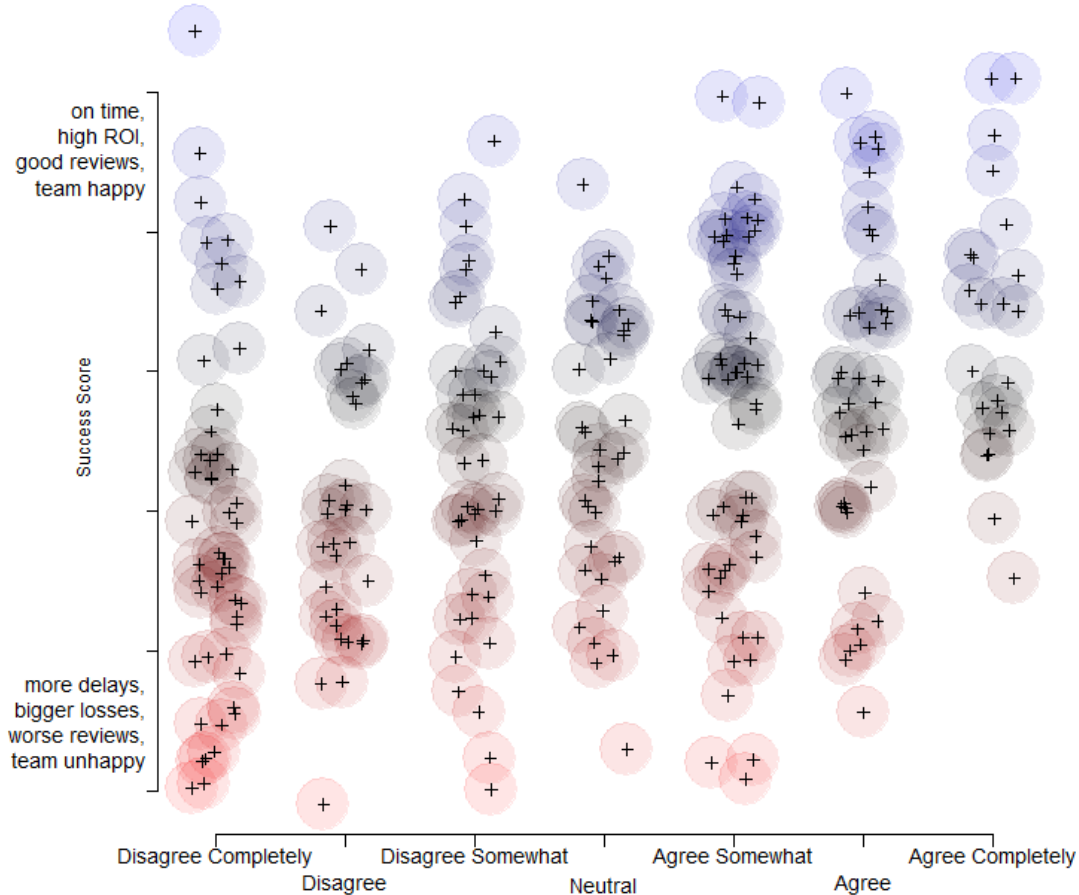
- Team reorganizations
- Required/pressured overtime
- Competing priorities between teams

Part 4 Main Points

- These things probably help
 - Being able to receive feedback
 - Fearless communication
 - Stable teams with clear roles/responsibilities
 - Collective goals instead of competing teams or individuals
 - Positive feedback

Part 5: What really helps/hurts

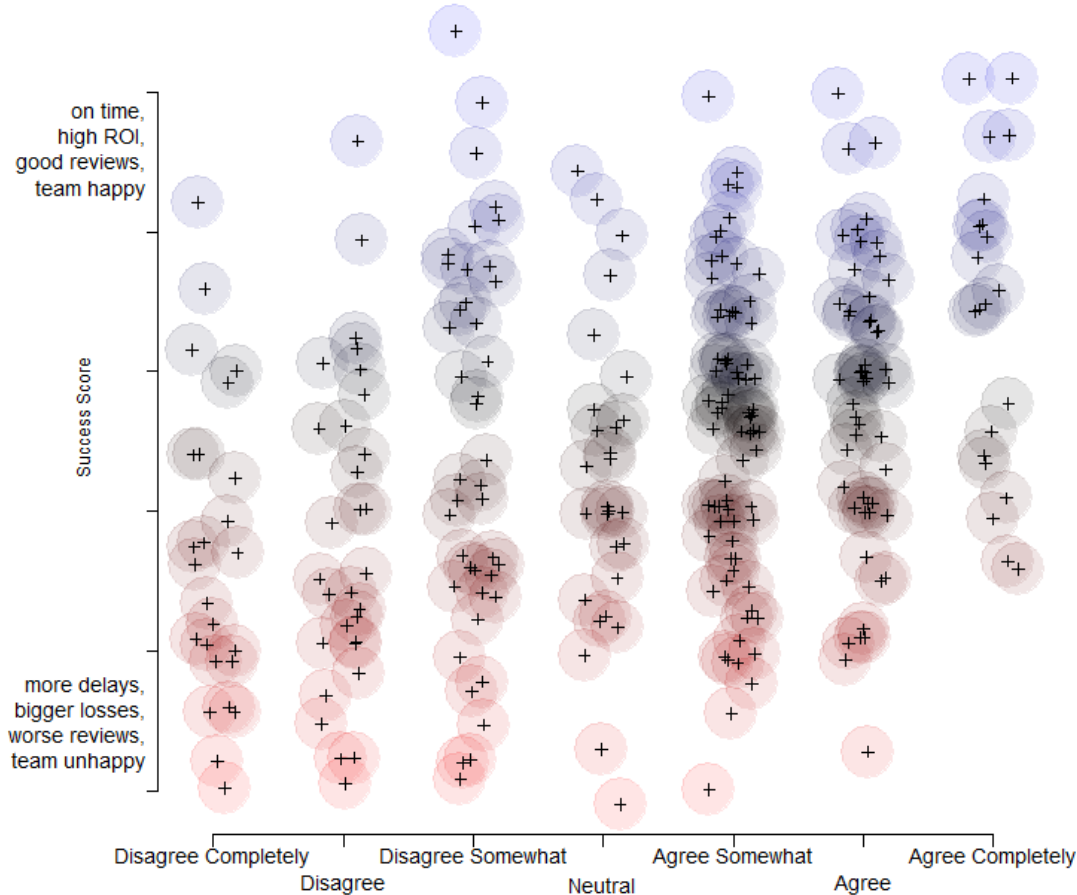
**This organization encouraged taking creative risks,
regardless of outcome.**



Good:

- Encouraging creative risks

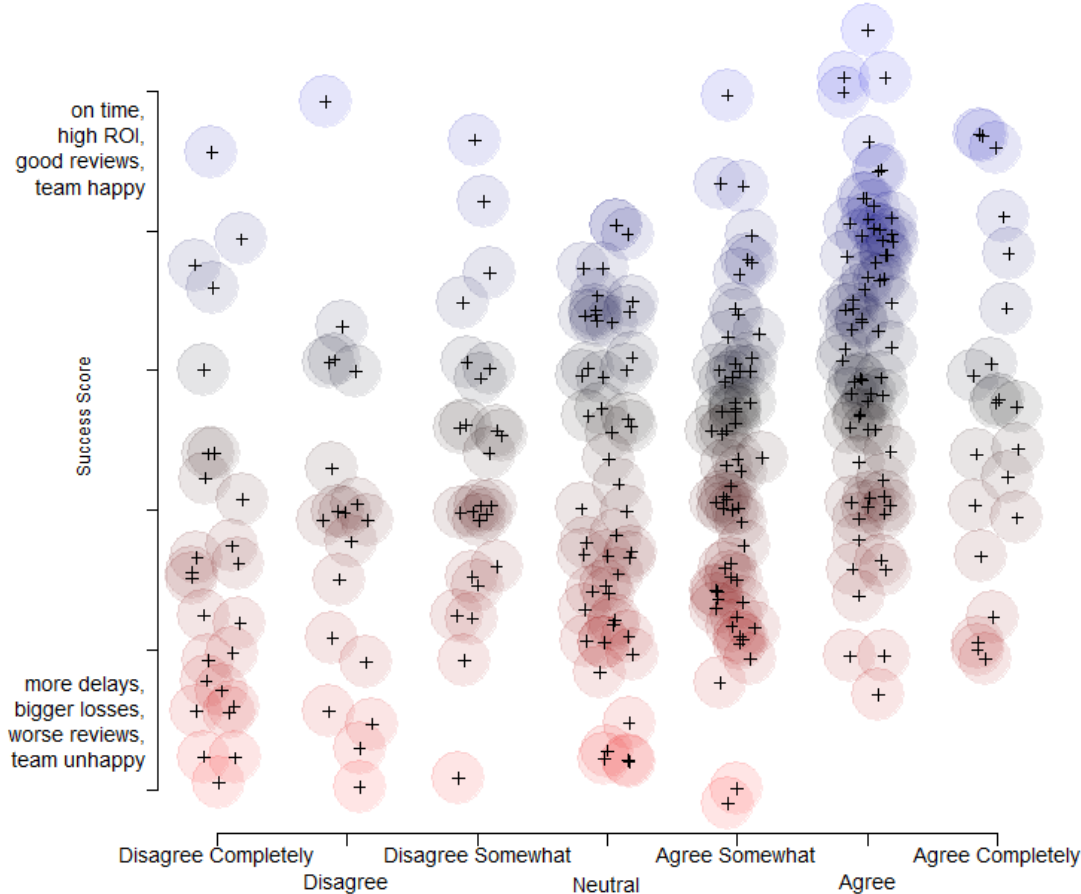
We discussed our failures openly in this organization.



Good:

- Encouraging creative risks
- Being open about failures

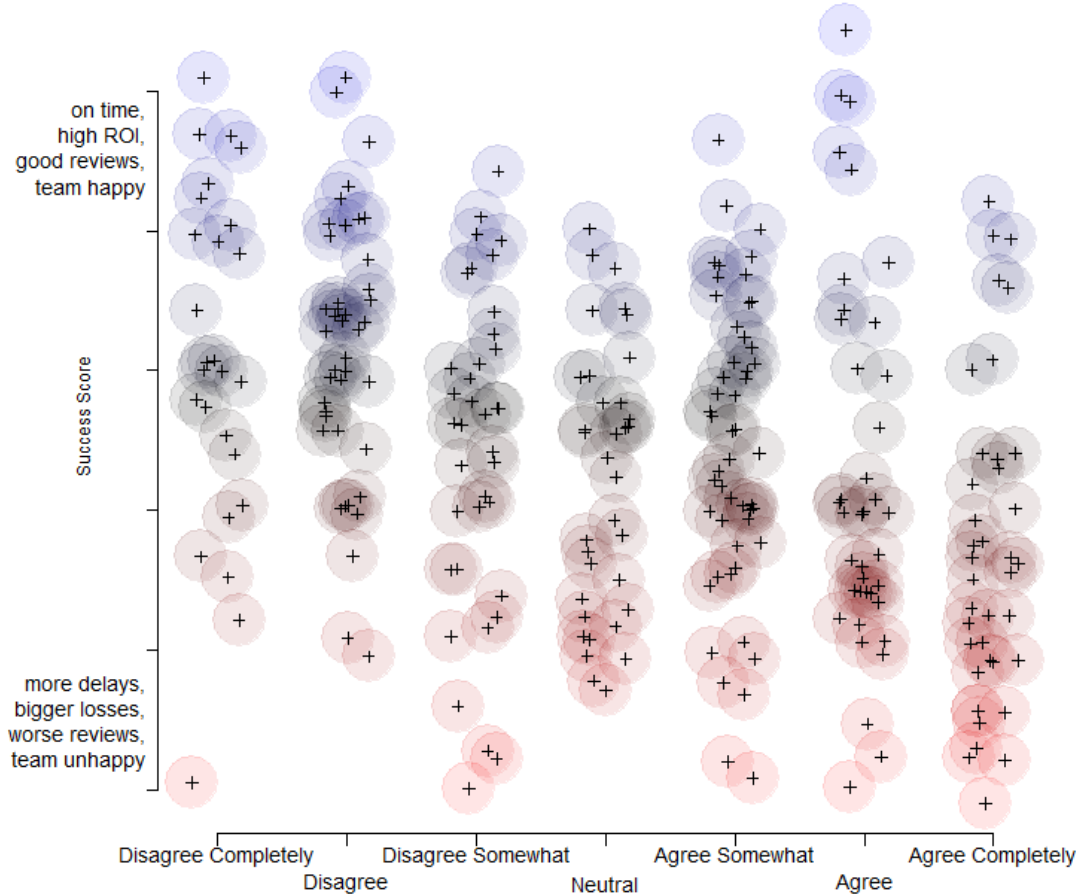
Personnel issues within the team (teamwork problems, HR issues) were dealt with professionally and appropriately.



Good:

- Encouraging creative risks
- Being open about failures
- Resolve team issues

Team members often expressed frustration that their opinions were ignored.



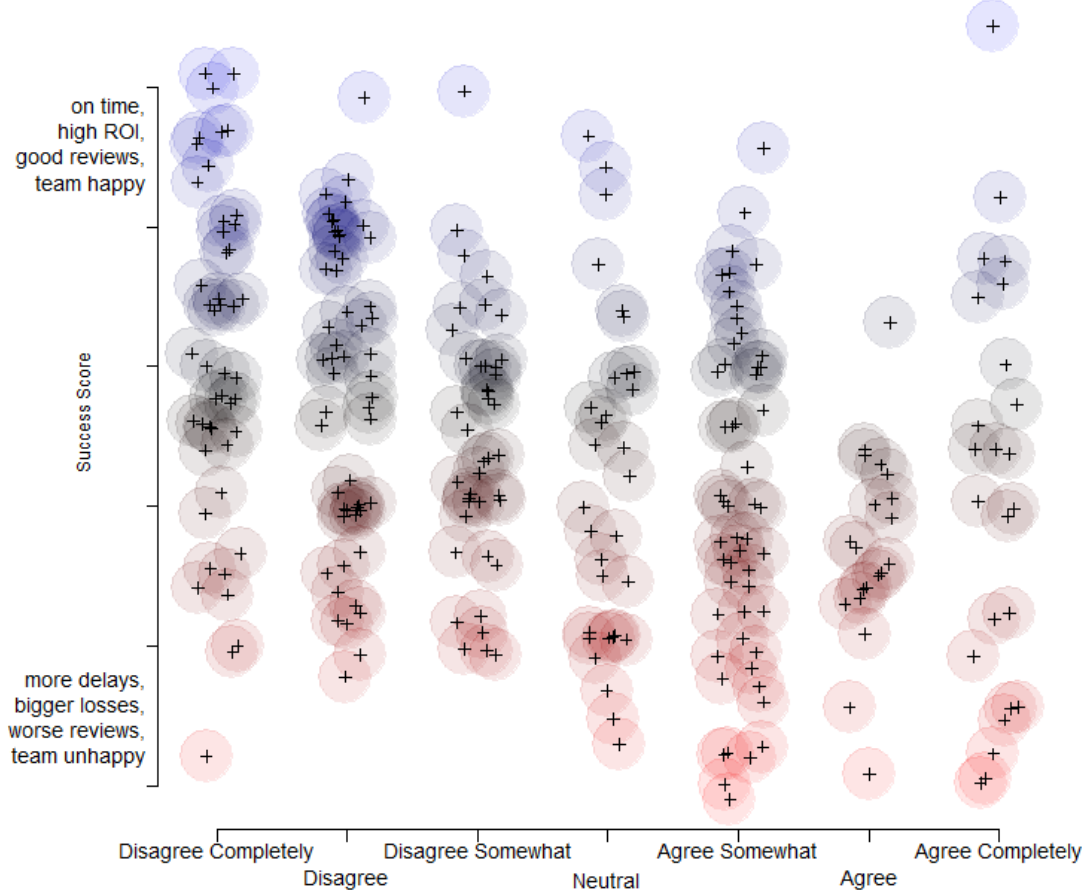
Good:

- Encouraging creative risks
- Being open about failures
- Resolve team issues

Bad:

- Ignoring feedback

I was hesitant to take creative risks in this organization.



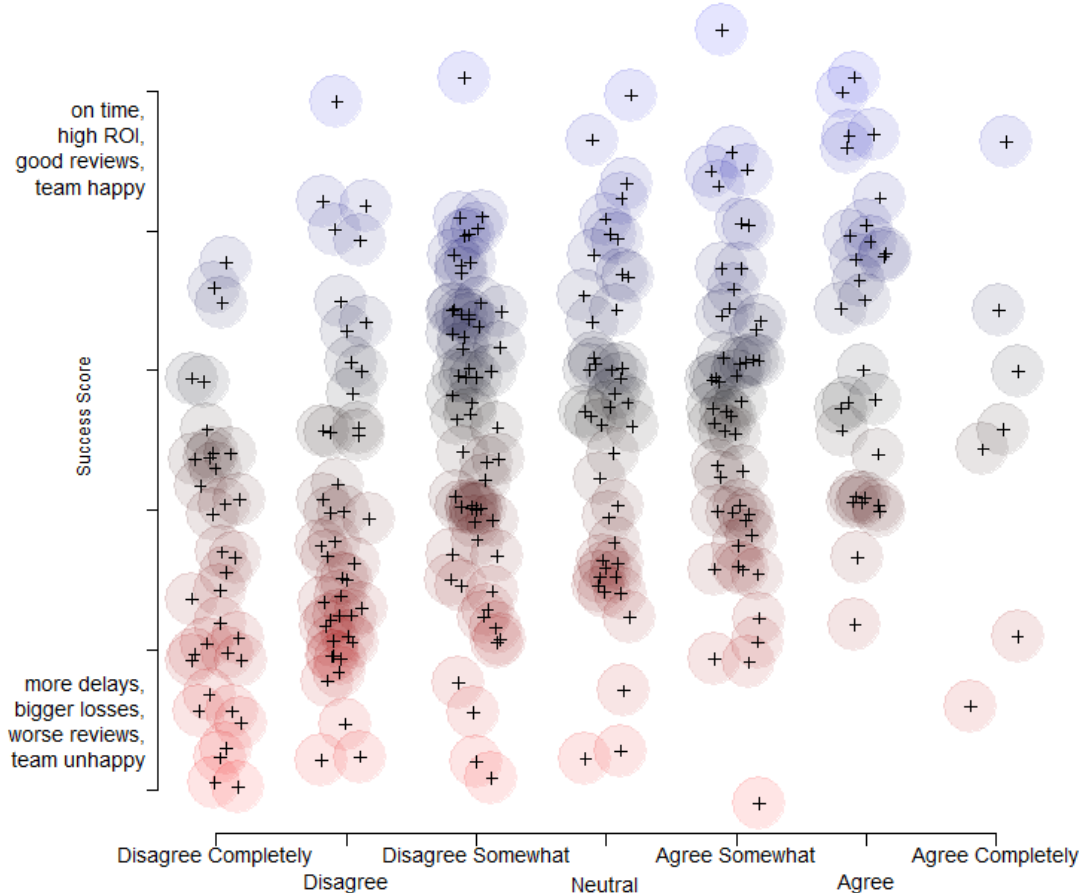
Good:

- Encouraging creative risks
- Being open about failures
- Resolve team issues

Bad:

- Ignoring feedback

Team members or studio leaders accurately estimated how long tasks would take.



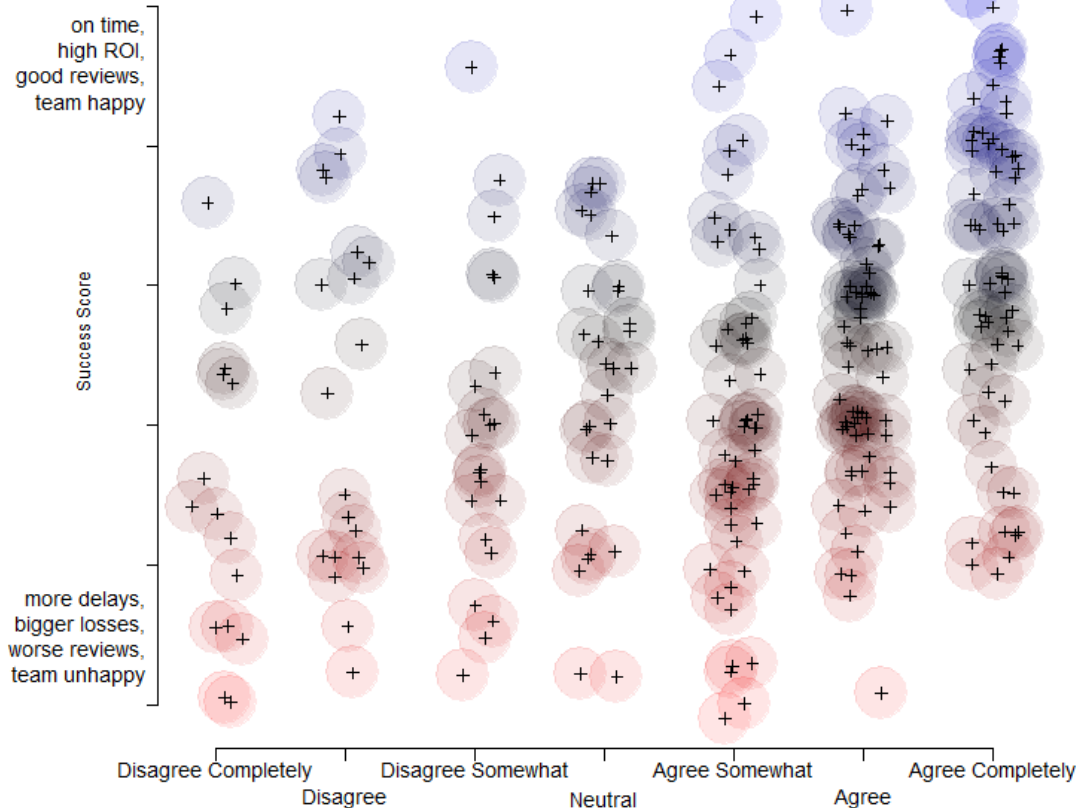
Good:

- Encouraging creative risks
- Being open about failures
- Resolve team issues
- Realistic schedule

Bad:

- Ignoring feedback

I felt comfortable sharing my creative attempts with my team leads.



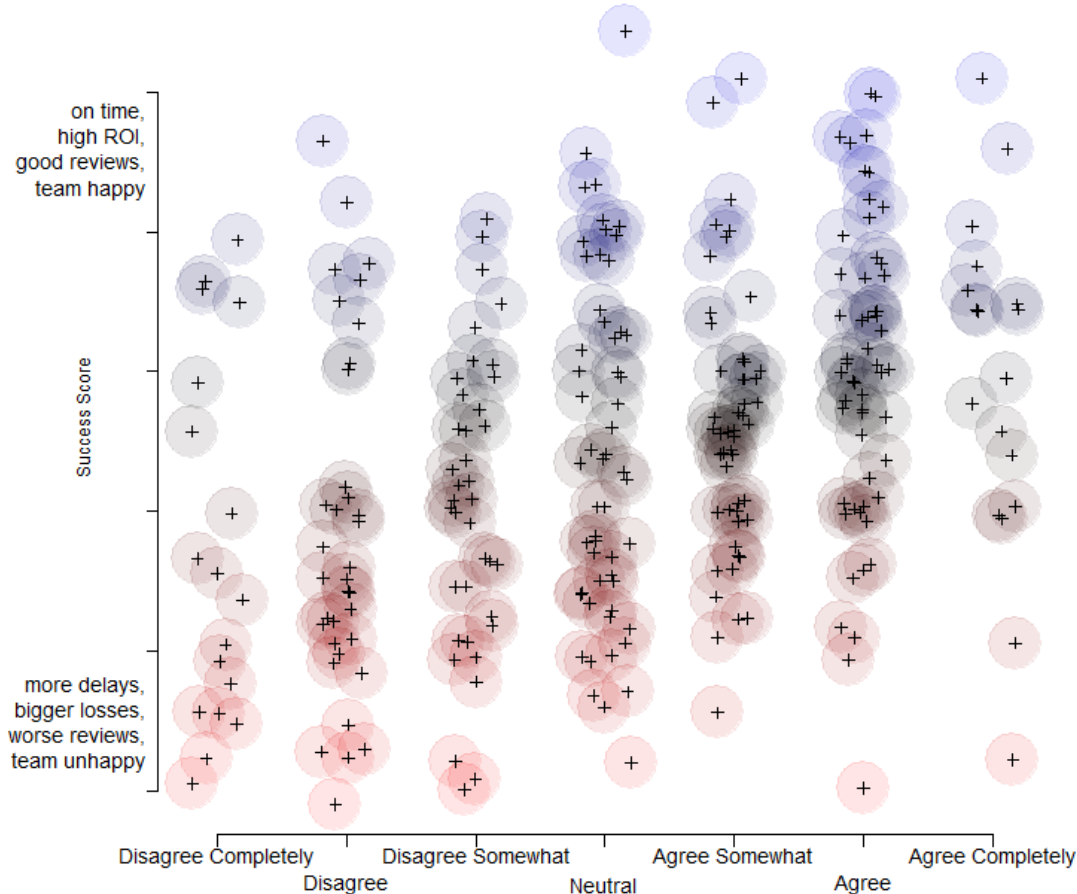
Good:

- Encouraging creative risks
- Being open about failures
- Resolve team issues
- Realistic schedule

Bad:

- Ignoring feedback

Decisions regarding technological changes were well communicated and understood by all team members.



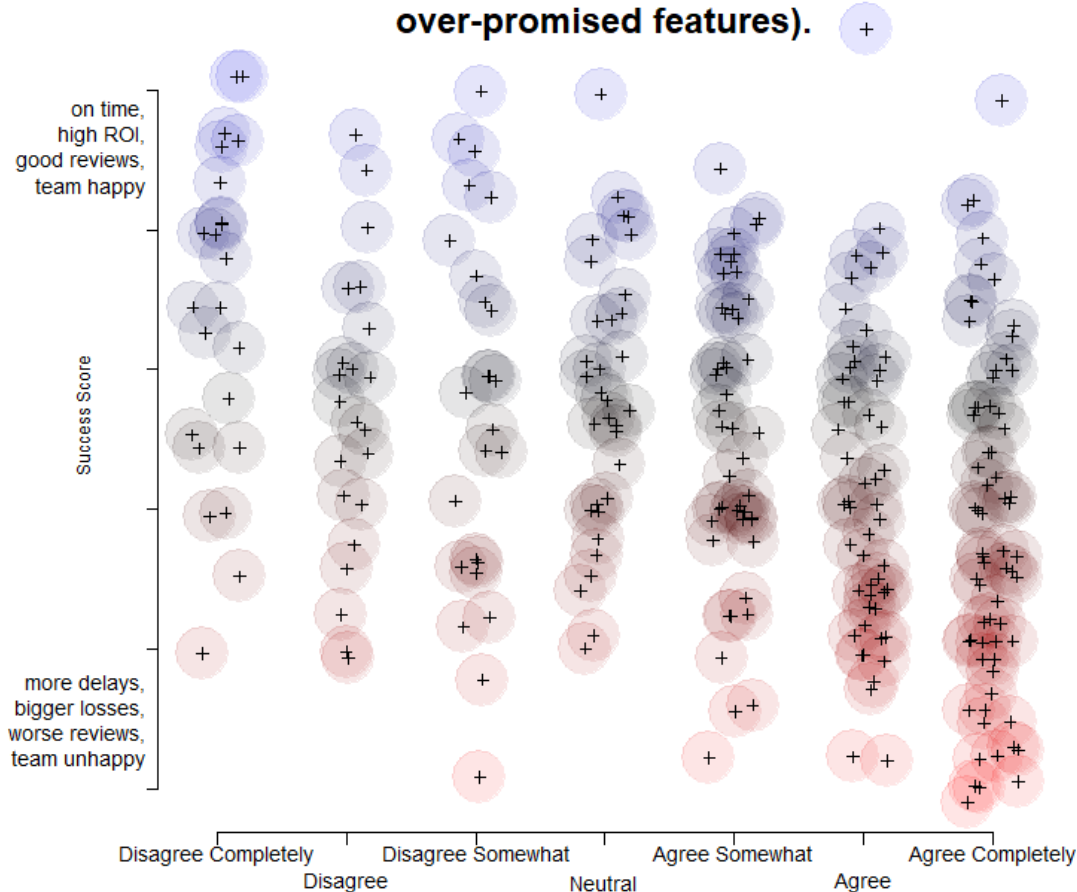
Good:

- Encouraging creative risks
- Being open about failures
- Resolve team issues
- Realistic schedule
- Communicate changes

Bad:

- Ignoring feedback

If we worked overtime, I believe it was because studio leaders or producers failed to scope the project properly (e.g. insufficient manpower, deadlines that were too tight, over-promised features).



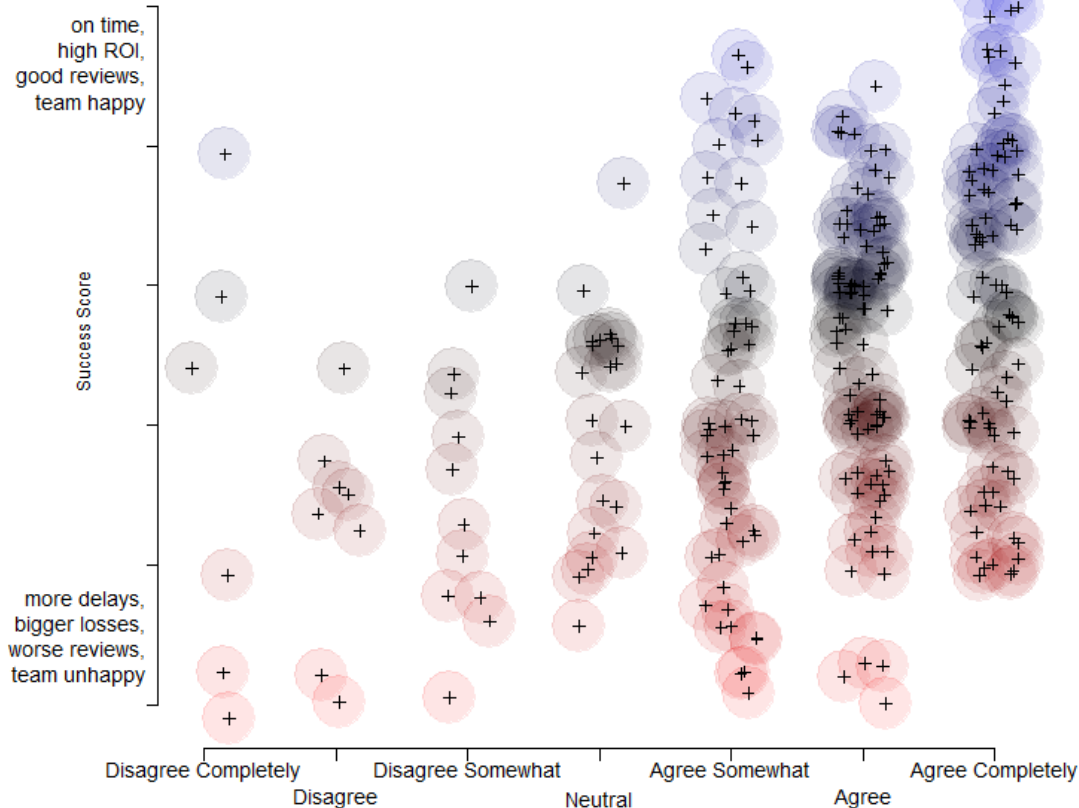
Good:

- Encouraging creative risks
- Being open about failures
- Resolve team issues
- Realistic schedule
- Communicate changes

Bad:

- Ignoring feedback

My peers on the team were committed to making a great game.



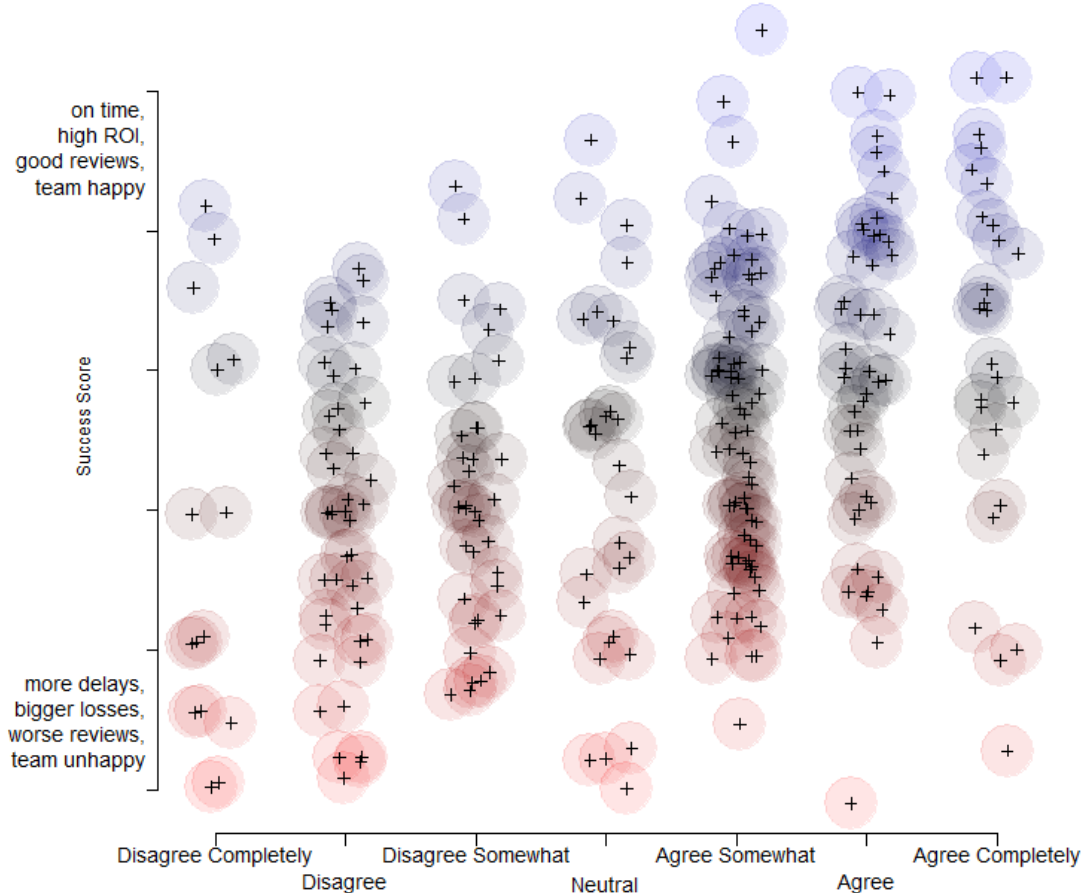
Good:

- Encouraging creative risks
- Being open about failures
- Resolve team issues
- Realistic schedule
- Communicate changes
- Engaged team

Bad:

- Ignoring feedback

The team received some form of direct feedback on a regular basis as to how well we were doing.



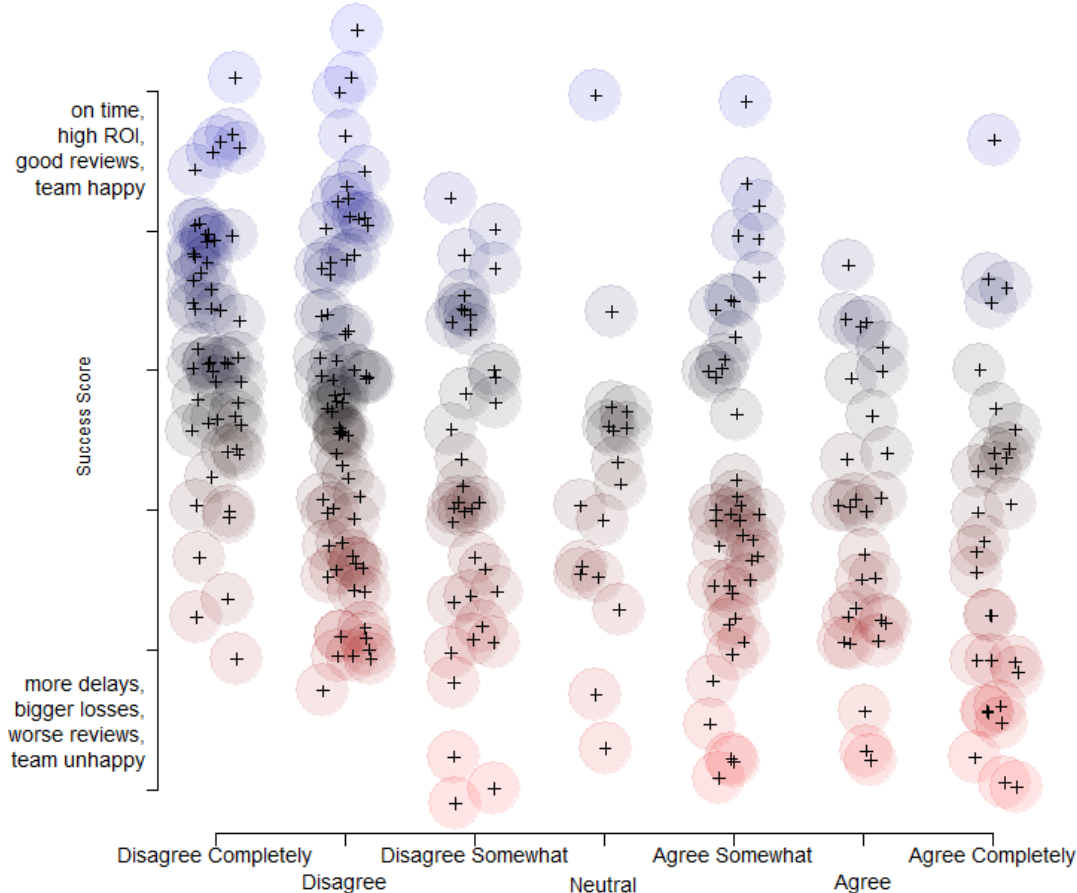
Good:

- Encouraging creative risks
- Being open about failures
- Resolve team issues
- Realistic schedule
- Communicate changes
- Engaged team
- Regular direct feedback

Bad:

- Ignoring feedback

Team members would often work for weeks at a time without receiving feedback from project leads or managers.



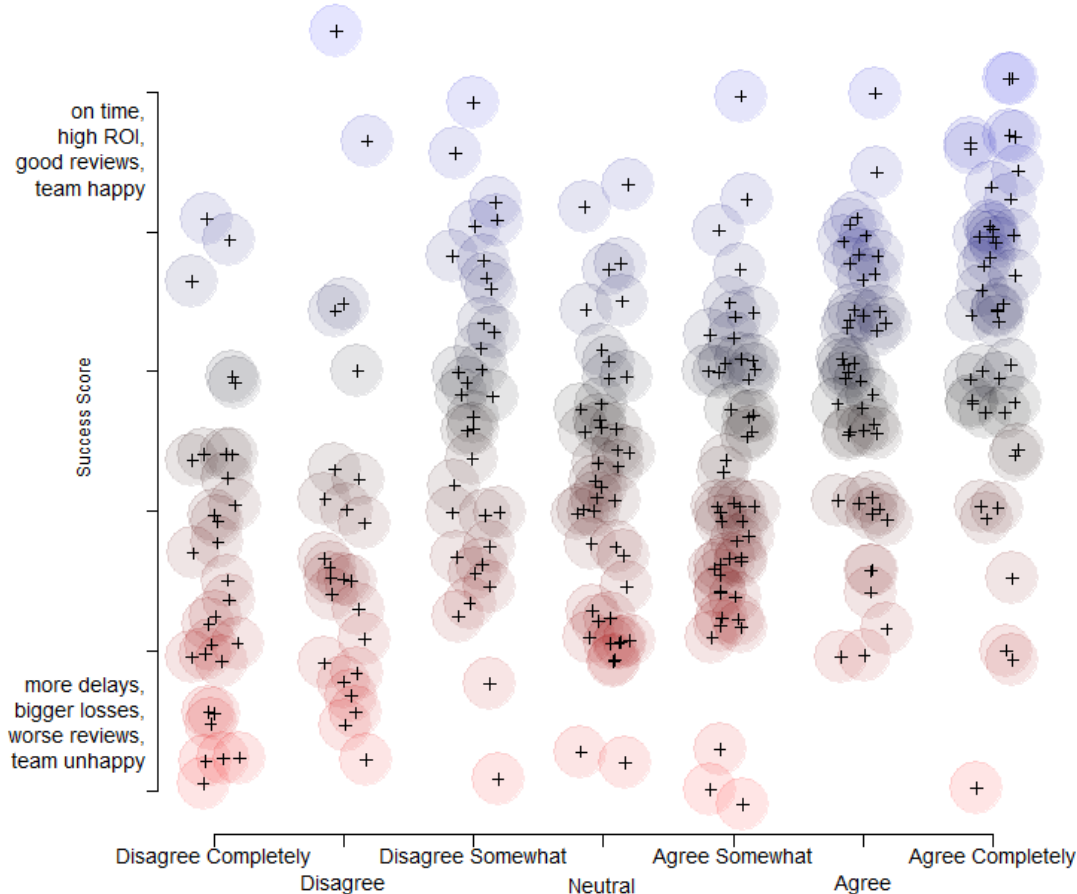
Good:

- Encouraging creative risks
- Being open about failures
- Resolve team issues
- Realistic schedule
- Communicate changes
- Engaged team
- Regular direct feedback

Bad:

- Ignoring feedback

The mission and/or values of this organization made sense to me, and I believed in them.



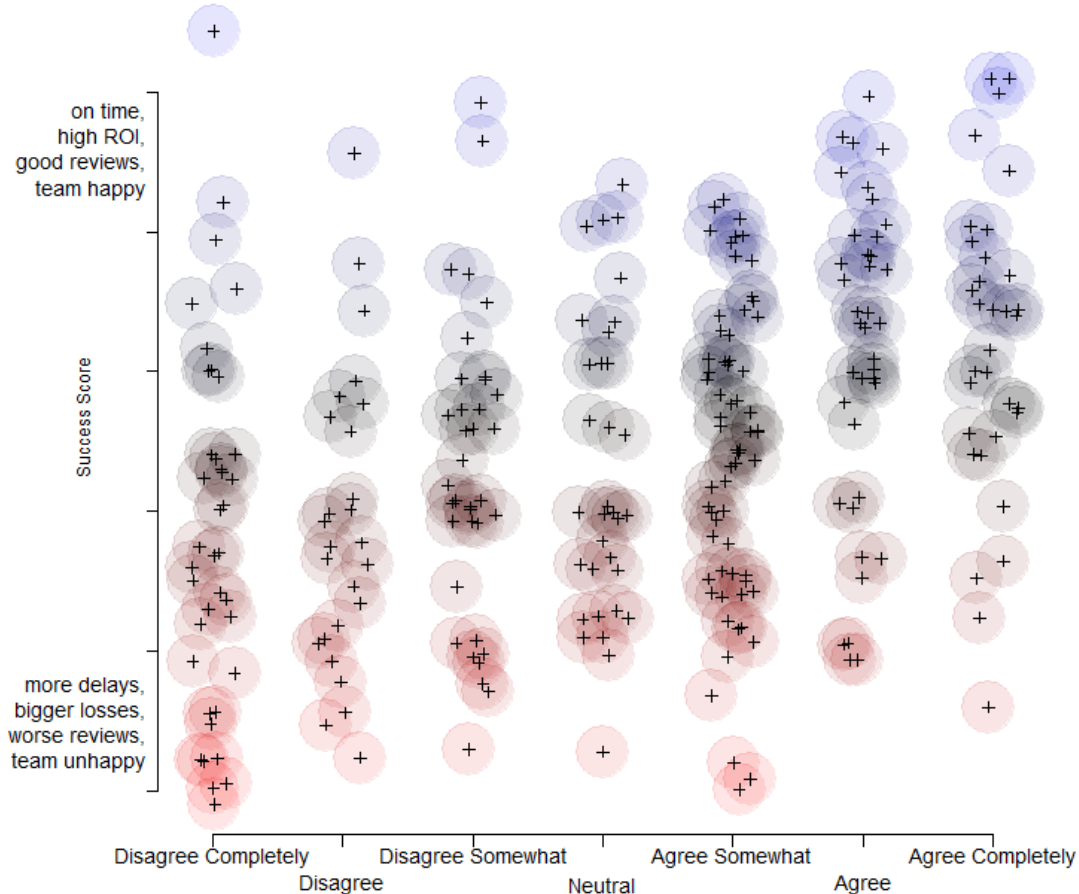
Good:

- Encouraging creative risks
- Being open about failures
- Resolve team issues
- Realistic schedule
- Communicate changes
- Engaged team
- Regular direct feedback
- Good company mission/values

Bad:

- Ignoring feedback

Novel ideas were celebrated at this organization even if they didn't achieve their intended result.



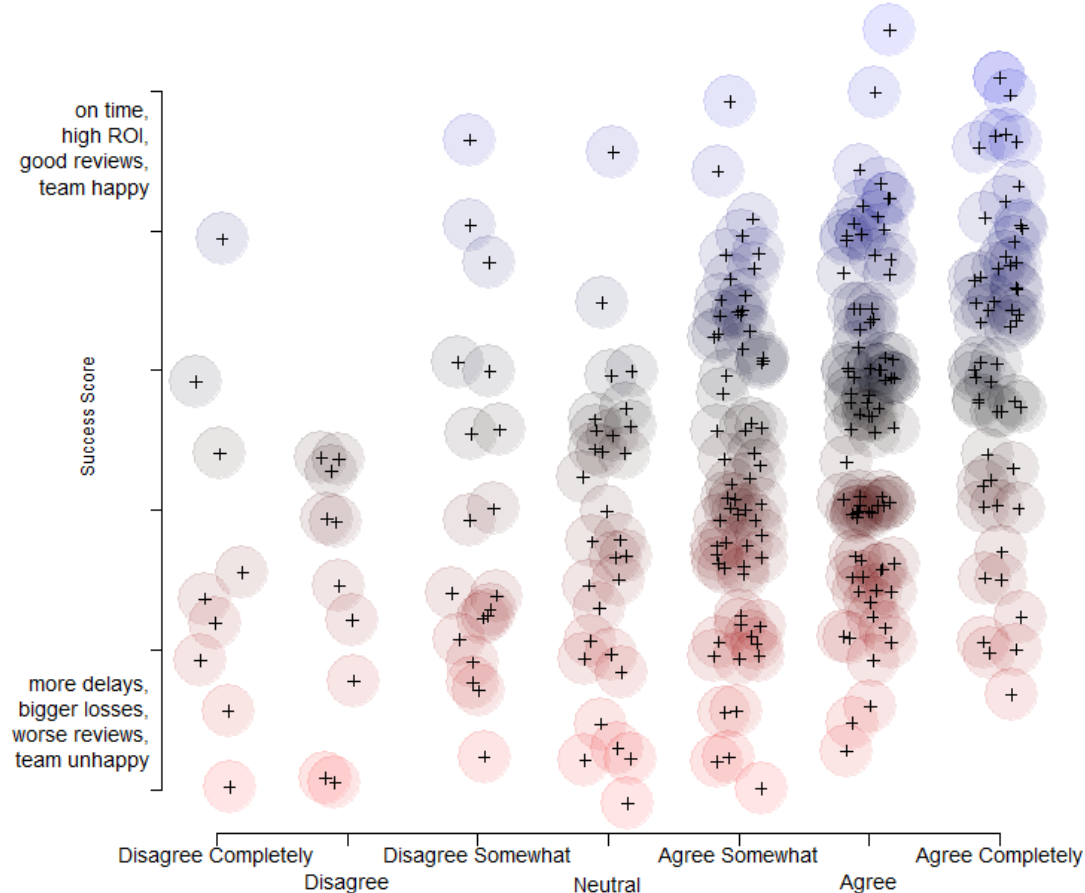
Good:

- Encouraging creative risks
- Being open about failures
- Resolve team issues
- Realistic schedule
- Communicate changes
- Engaged team
- Regular direct feedback
- Good company mission/values

Bad:

- Ignoring feedback

Team members held each other to high standards for their particular discipline (art, design, engineering, etc).



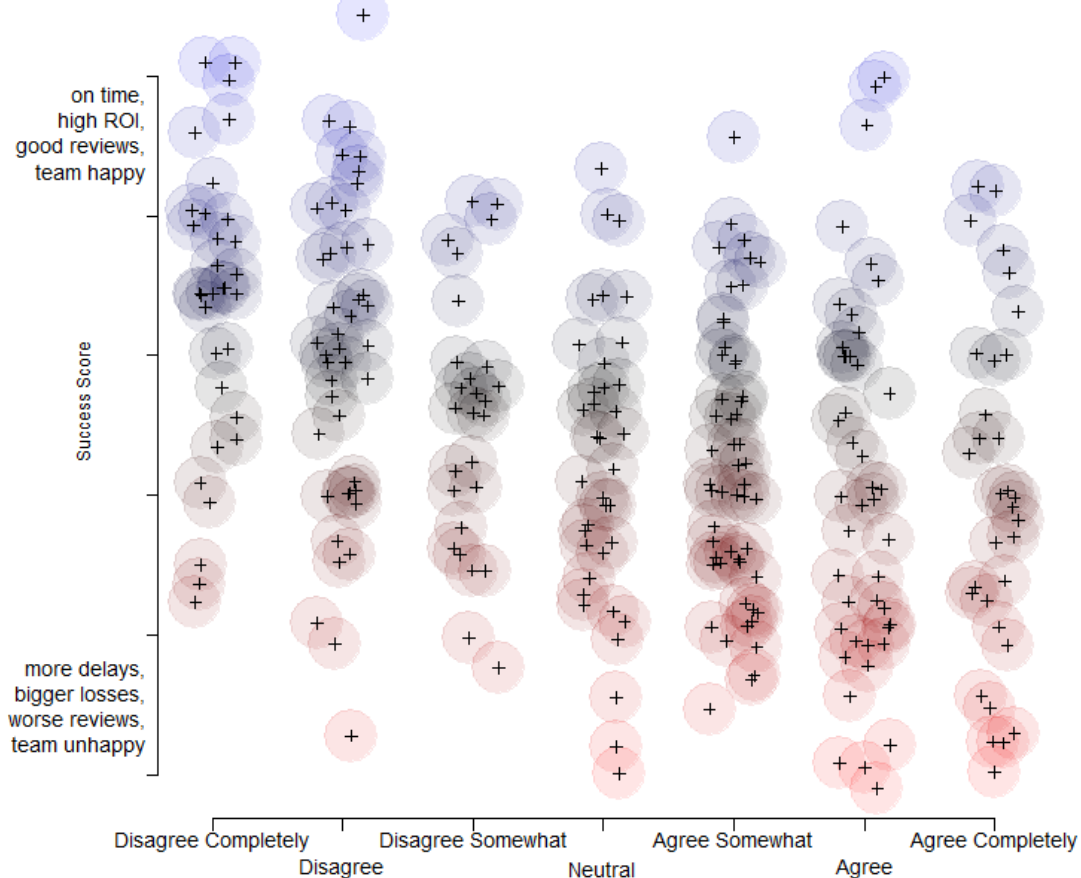
Good:

- Encouraging creative risks
- Being open about failures
- Resolve team issues
- Realistic schedule
- Communicate changes
- Engaged team
- Regular direct feedback
- Good company mission/values
- Team with high standards

Bad:

- Ignoring feedback

I was sometimes surprised by receiving feedback much later than I needed it, or seeing things in a performance review that had never been mentioned before.



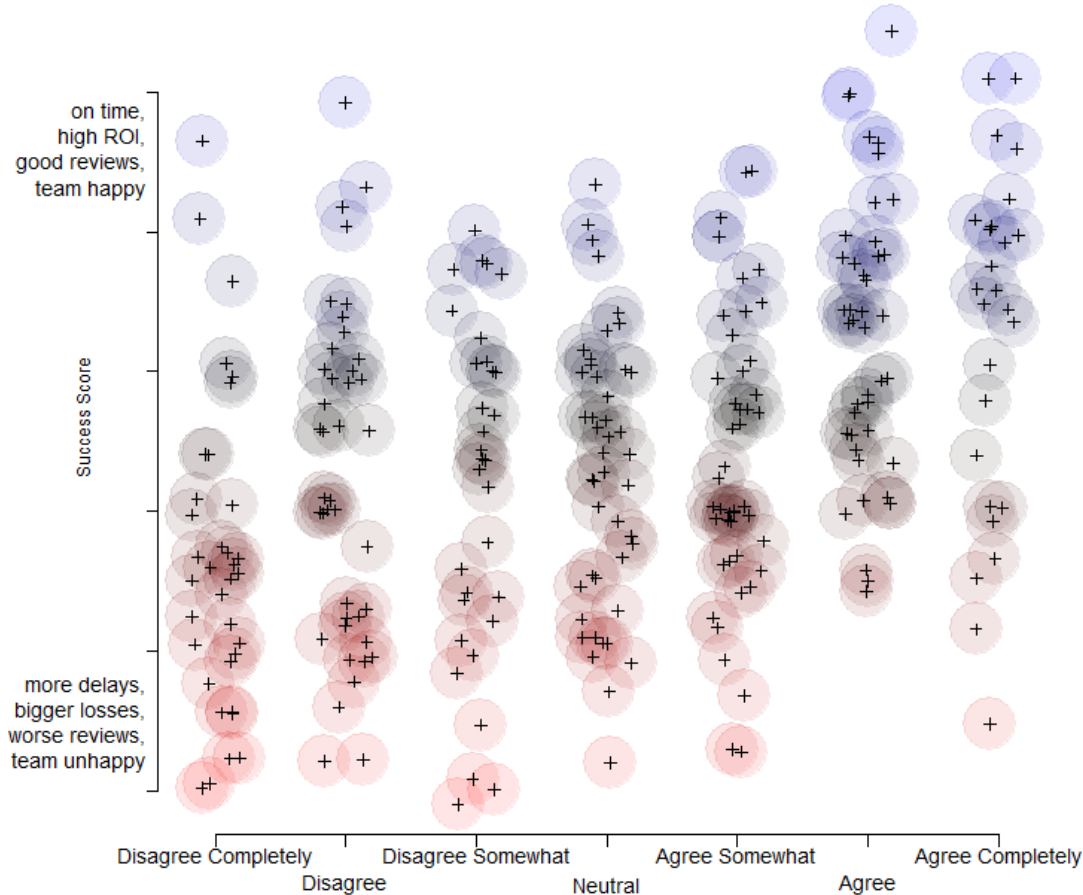
Good:

- Encouraging creative risks
- Being open about failures
- Resolve team issues
- Realistic schedule
- Communicate changes
- Engaged team
- Regular direct feedback
- Good company mission/values
- Team with high standards

Bad:

- Ignoring feedback

The organization has a clearly-defined mission statement or set of values.



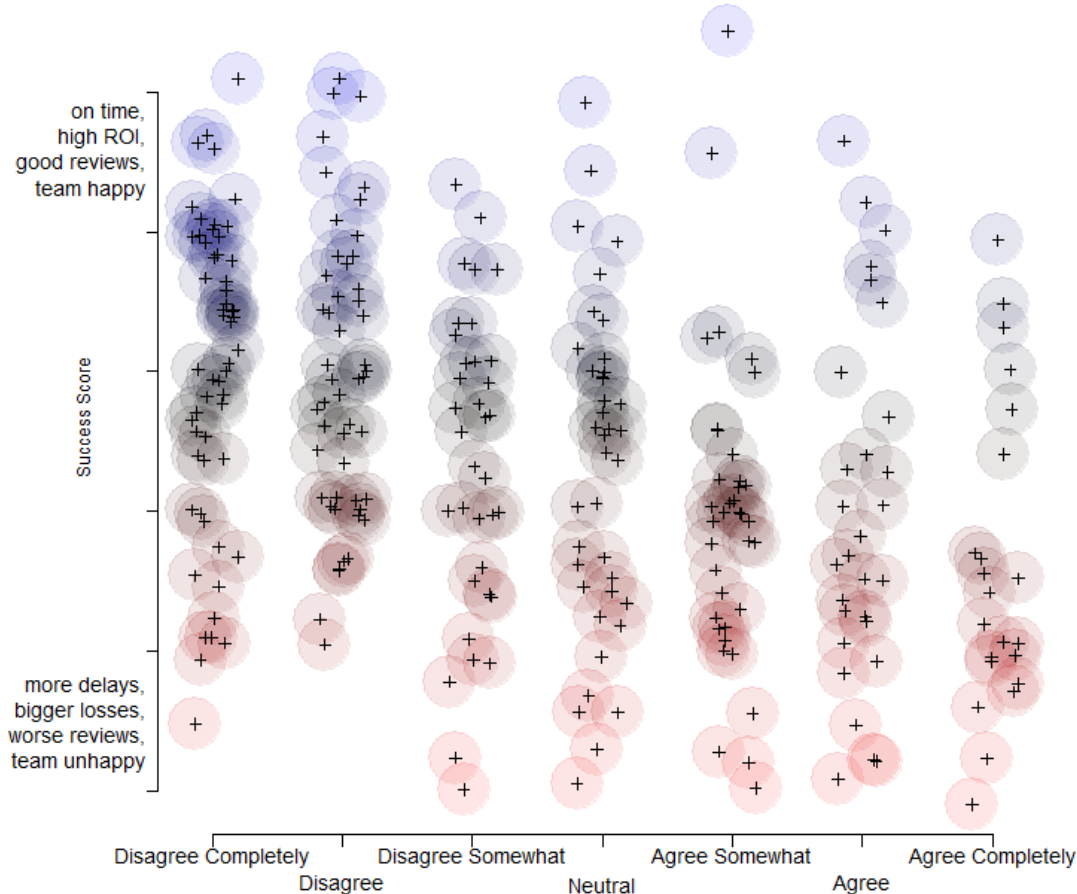
Good:

- Encouraging creative risks
- Being open about failures
- Resolve team issues
- Realistic schedule
- Communicate changes
- Engaged team
- Regular direct feedback
- Good company mission/values
- Team with high standards

Bad:

- Ignoring feedback

There was a lot of turnover on this project.



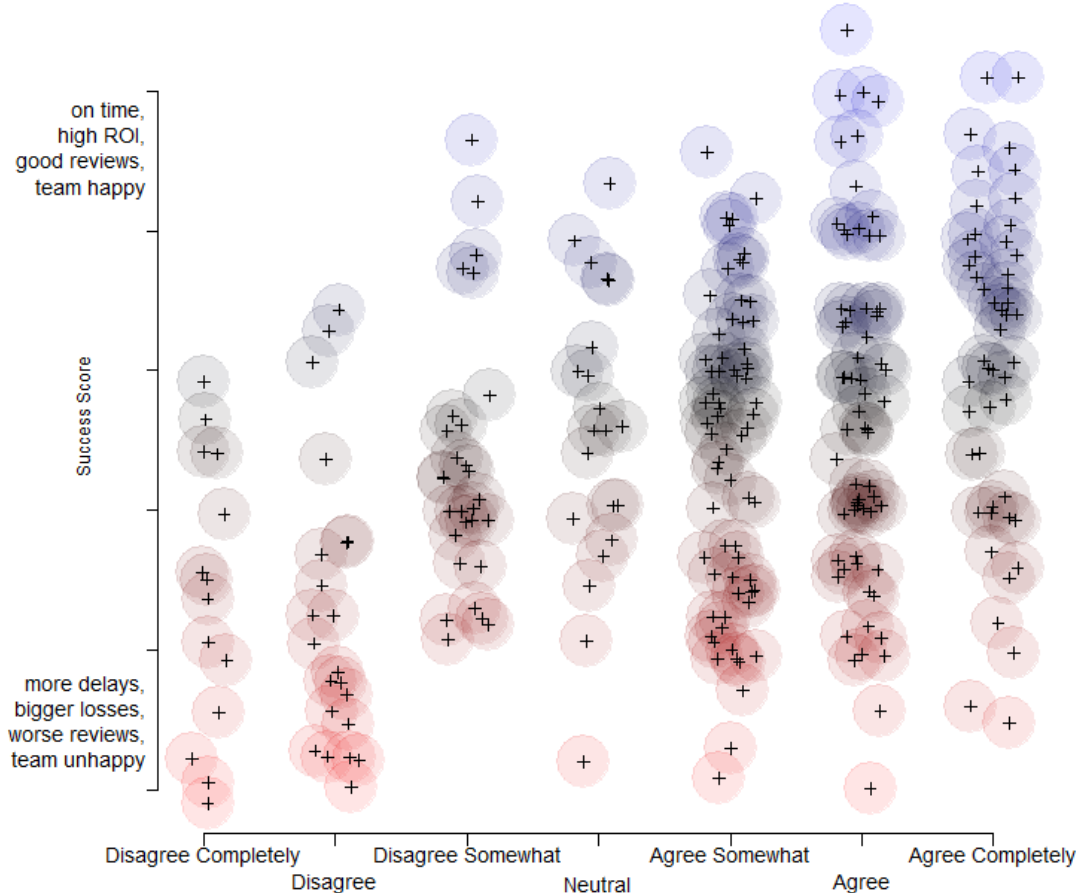
Good:

- Encouraging creative risks
- Being open about failures
- Resolve team issues
- Realistic schedule
- Communicate changes
- Engaged team
- Regular direct feedback
- Good company mission/values
- Team with high standards

Bad:

- Ignoring feedback
- Having to replace people

Most team members cared deeply about the vision of this game.



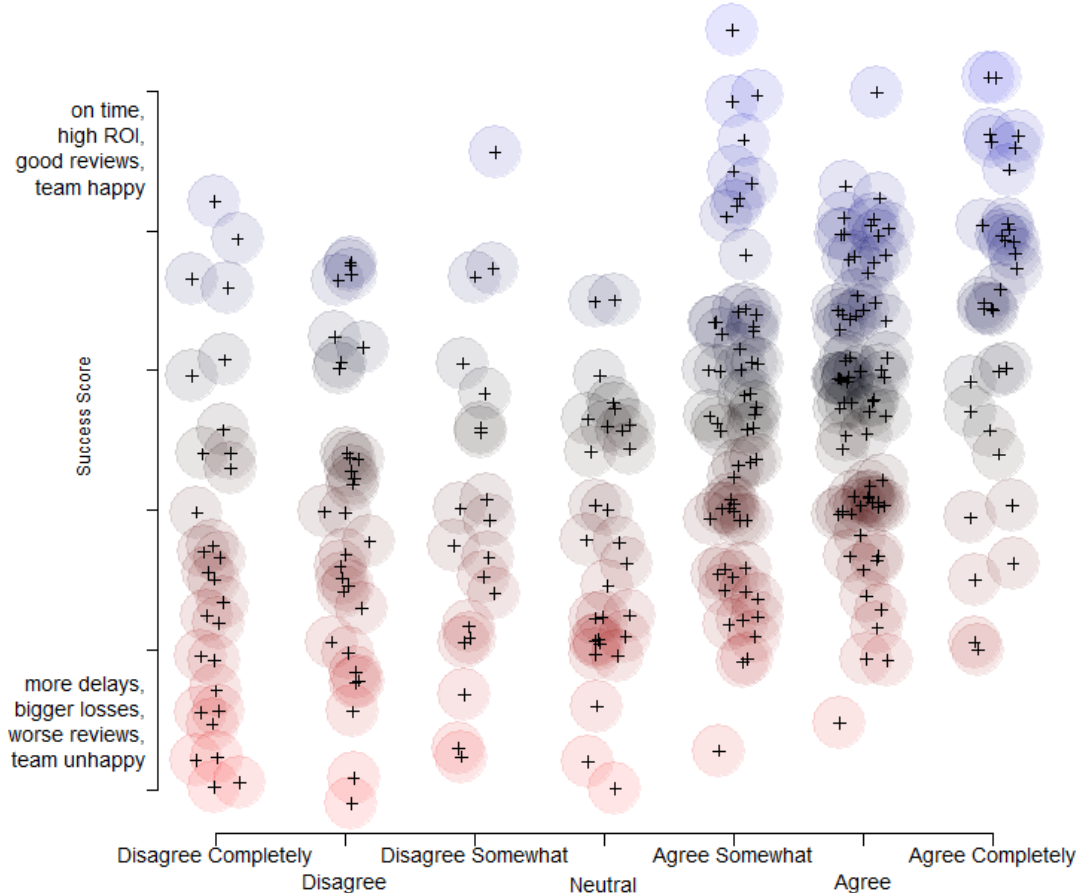
Good:

- Encouraging creative risks
- Being open about failures
- Resolve team issues
- Realistic schedule
- Communicate changes
- Engaged team
- Regular direct feedback
- Good company mission/values
- Team with high standards

Bad:

- Ignoring feedback
- Having to replace people

It was safe to take a risk on this team, and stick your neck out to say something that needed to be said.



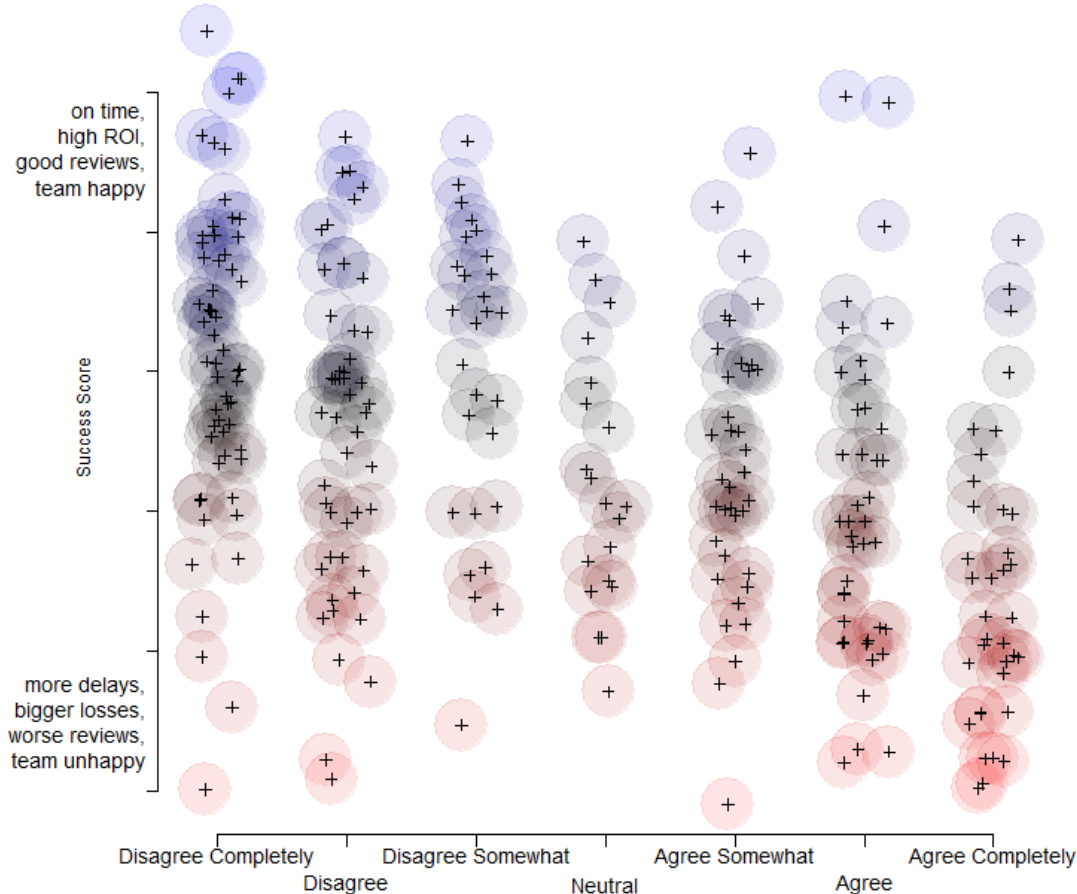
Good:

- Encouraging creative risks
- Being open about failures
- Resolve team issues
- Realistic schedule
- Communicate changes
- Engaged team
- Regular direct feedback
- Good company mission/values
- Team with high standards
- Fearless/safe communication

Bad:

- Ignoring feedback
- Having to replace people

Our team sometimes seemed to be stuck in a cycle of never-ending crunch / overtime work.



Good:

- Encouraging creative risks
- Being open about failures
- Resolve team issues
- Realistic schedule
- Communicate changes
- Engaged team
- Regular direct feedback
- Good company mission/values
- Team with high standards
- Fearless communication

Bad:

- Ignoring feedback
- Having to replace people
- Crunch

Part 5 Main Points

- These points really help:
 - Encouraging individuals to take creative risks
 - Frequent direct feedback. Don't surprise people with negative feedback a month after the change made it into the build
 - Have a realistic schedule
 - An engaged team with high standards
 - A company with values or a mission

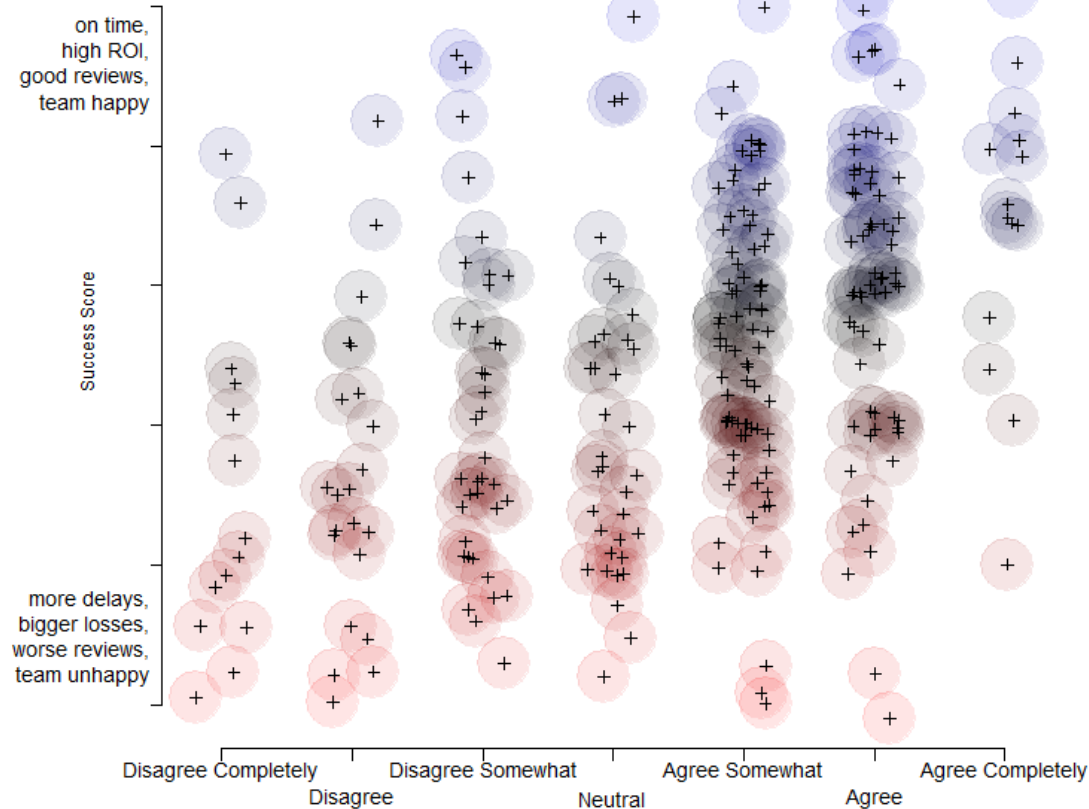
Part 6: The Top 10 Charts

10

Most team members bought in to the decisions that were made.

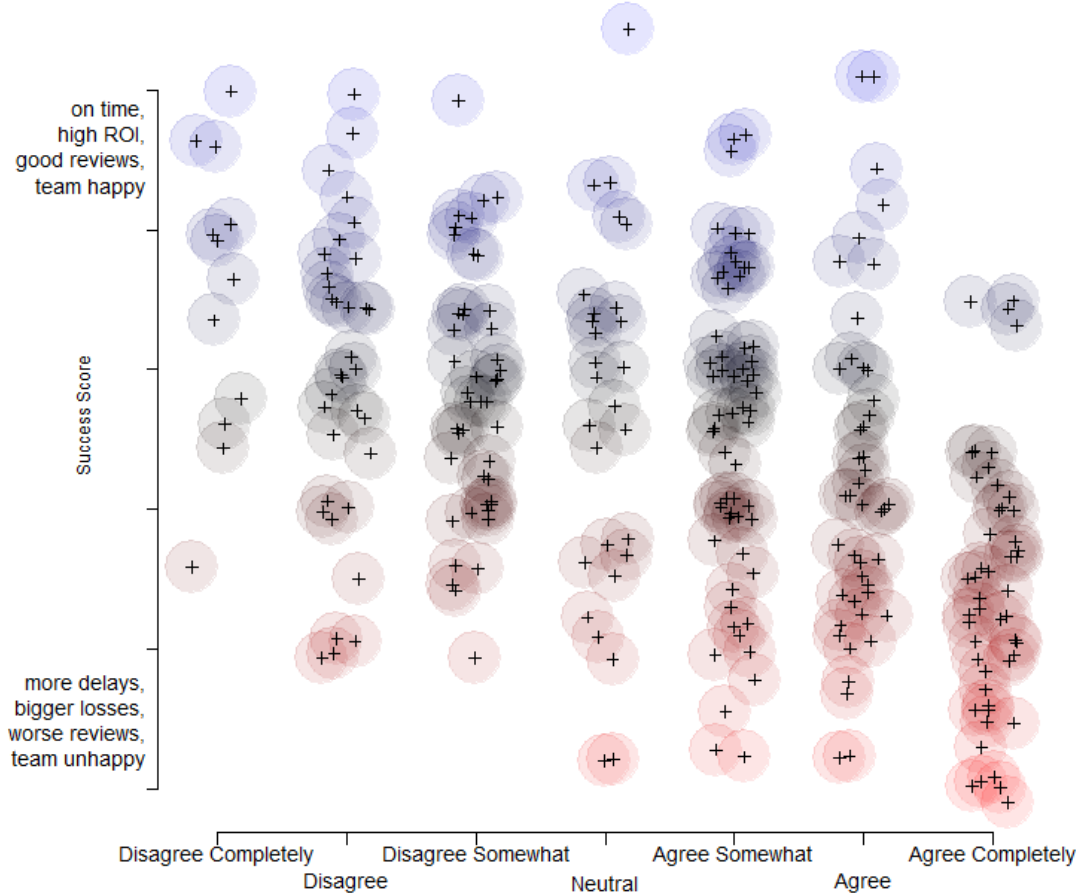
Good:

- Committed, cohesive team



9

Core elements of the game design frequently changed during the development process.



Good:

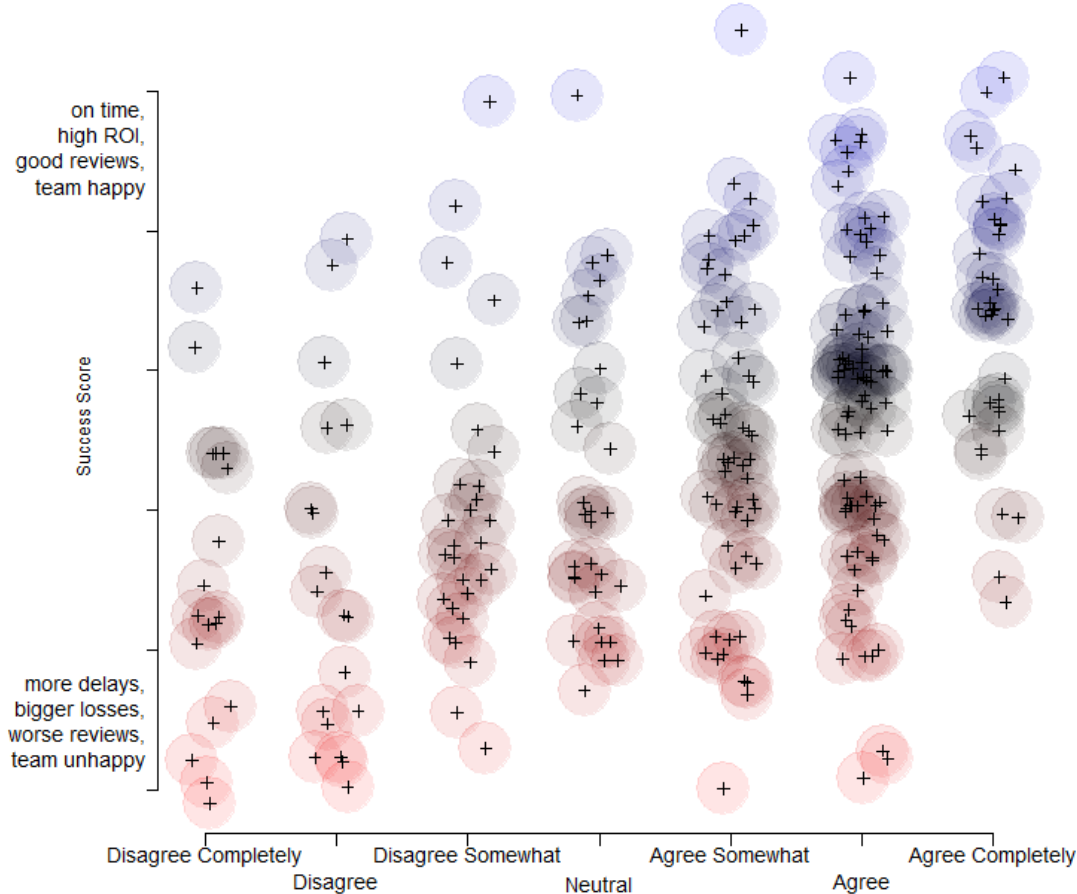
- Committed, cohesive team

Bad:

- Frequent design changes to the core of the game

8

When we disagreed about game design, we resolved the disagreement, instead of ignoring it.



Good:

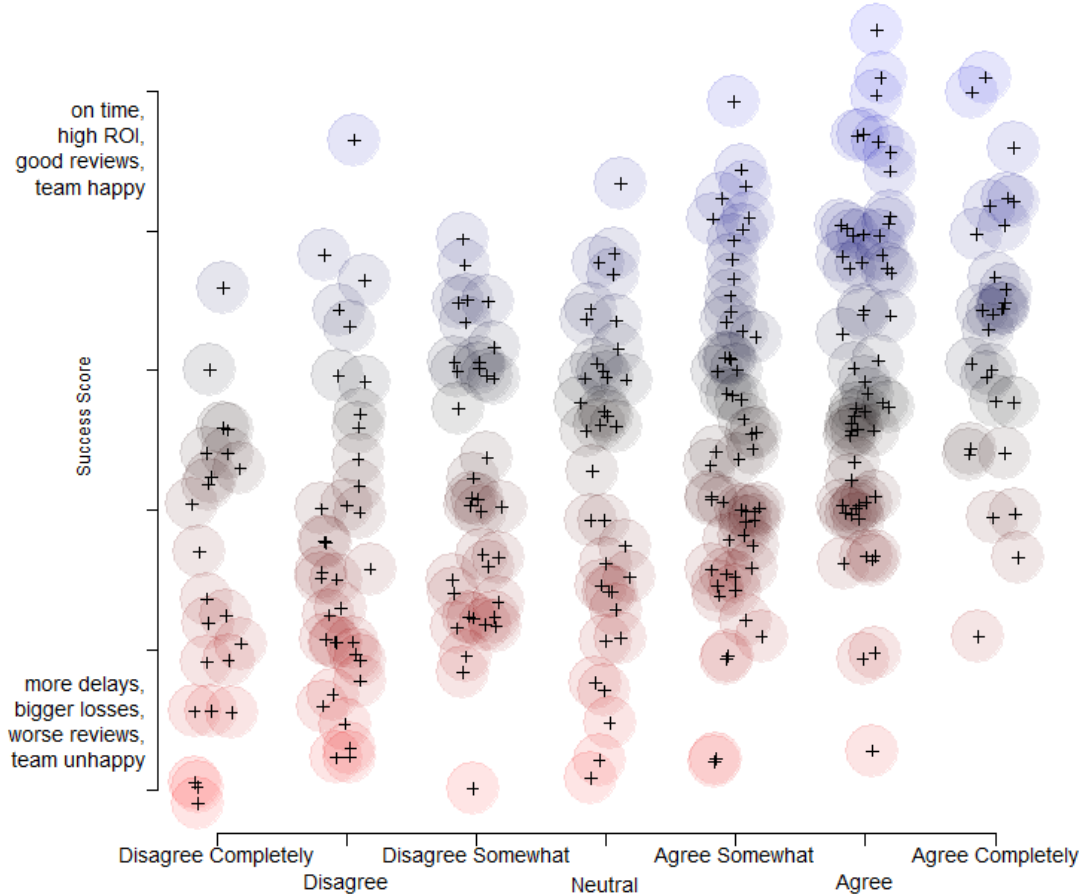
- Committed, cohesive team
- Resolving disagreements

Bad:

- Frequent design changes to the core of the game

7

If core design elements changed, studio leaders were able to clearly communicate and justify why they changed.



Good:

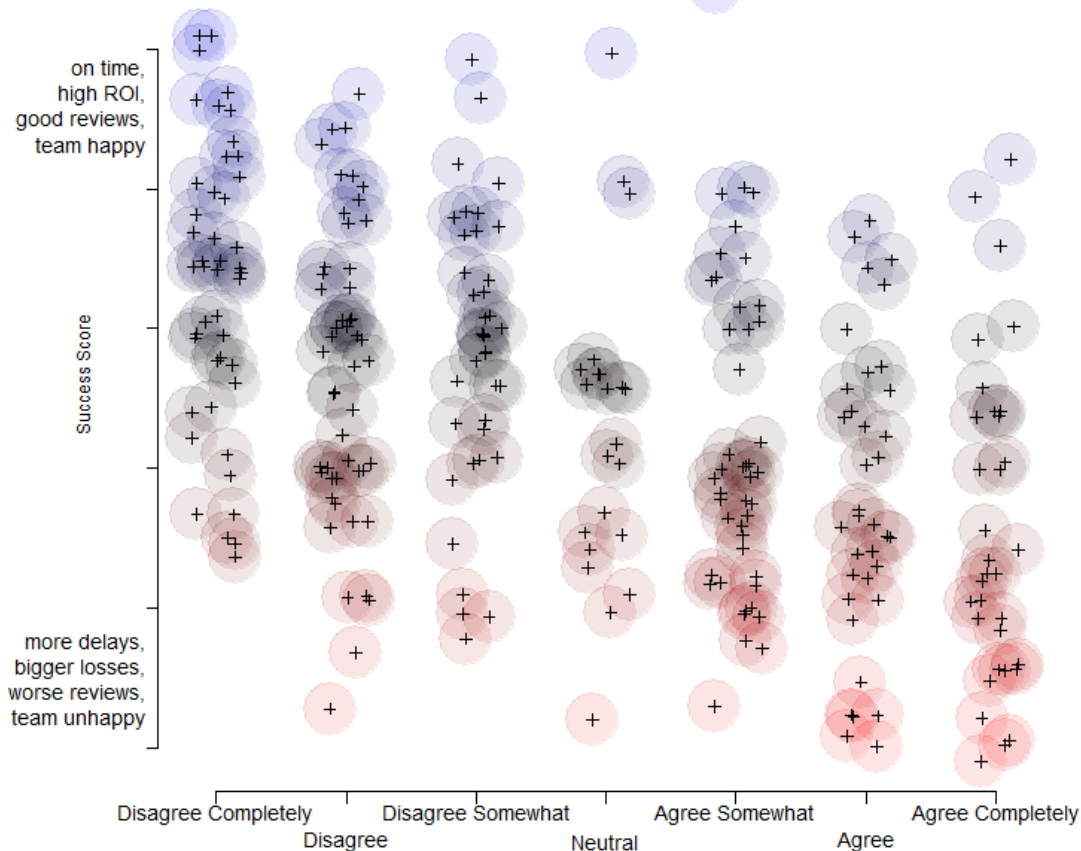
- Committed, cohesive team
- Resolving disagreements
- Clearly communicate and justify design changes

Bad:

- Frequent design changes to the core of the game

6

I was confused by the differing visions of what the major goals of this game were among our team leads (designers, producers, product managers).



Good:

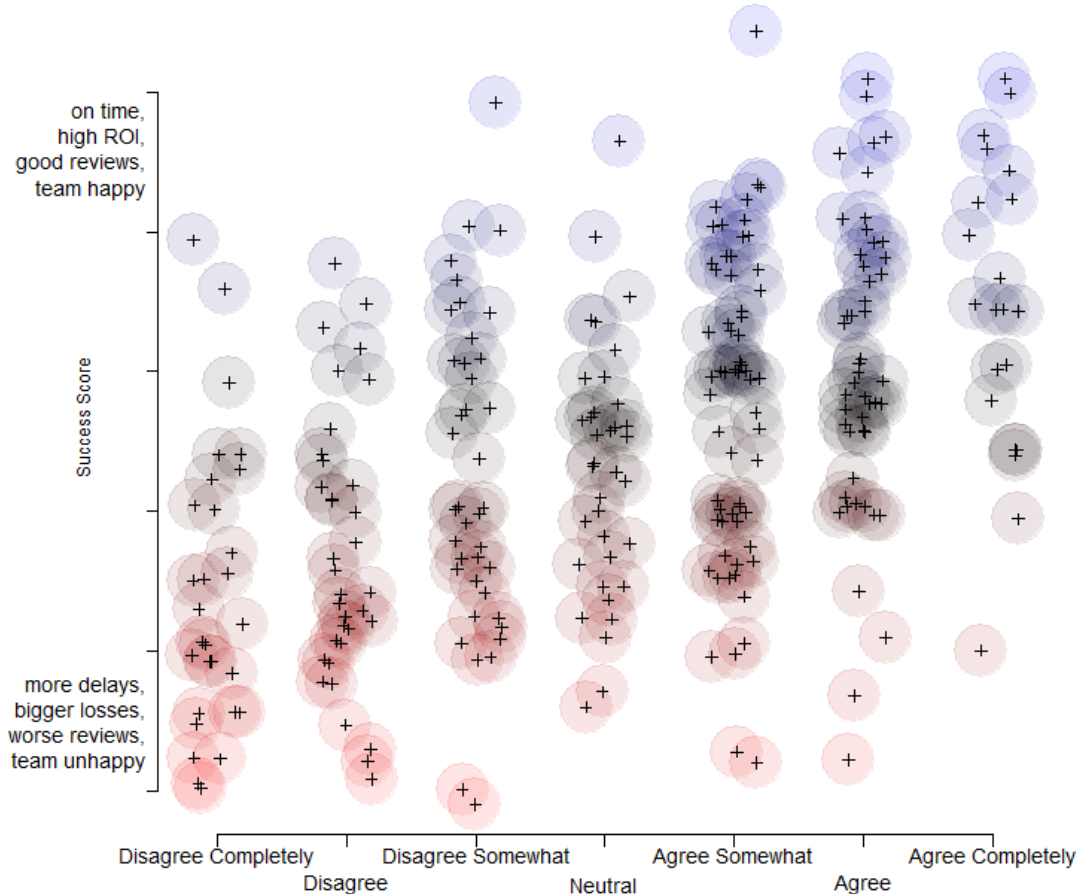
- Committed, cohesive team
- Resolving disagreements
- Clearly communicate and justify design changes

Bad:

- Frequent design changes to the core of the game
- Competing/confusing goals

5

The development plan for the game was clear and well-communicated to the team.



Good:

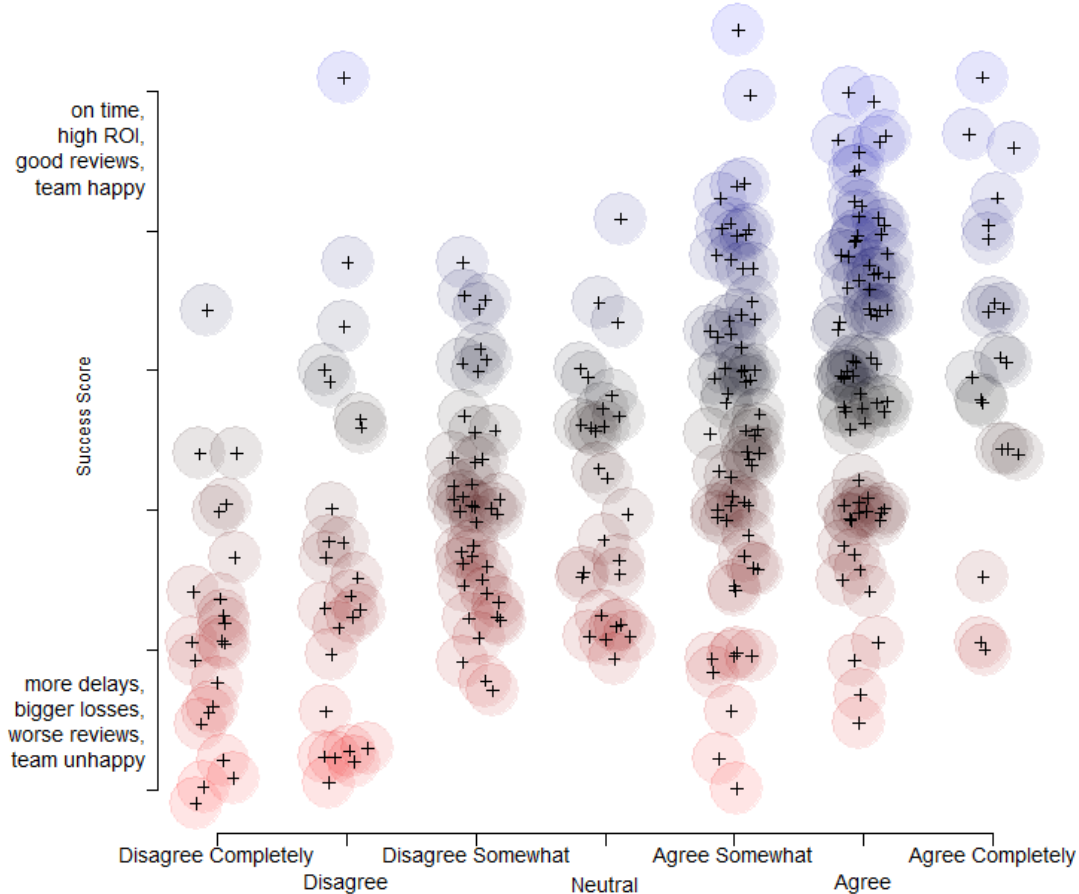
- Committed, cohesive team
- Resolving disagreements
- Clearly communicate and justify design changes
- Clearly communicated plan

Bad:

- Frequent design changes to the core of the game
- Competing/confusing goals

4

Most team members had a similar vision of what this game was throughout the process of making it.



Good:

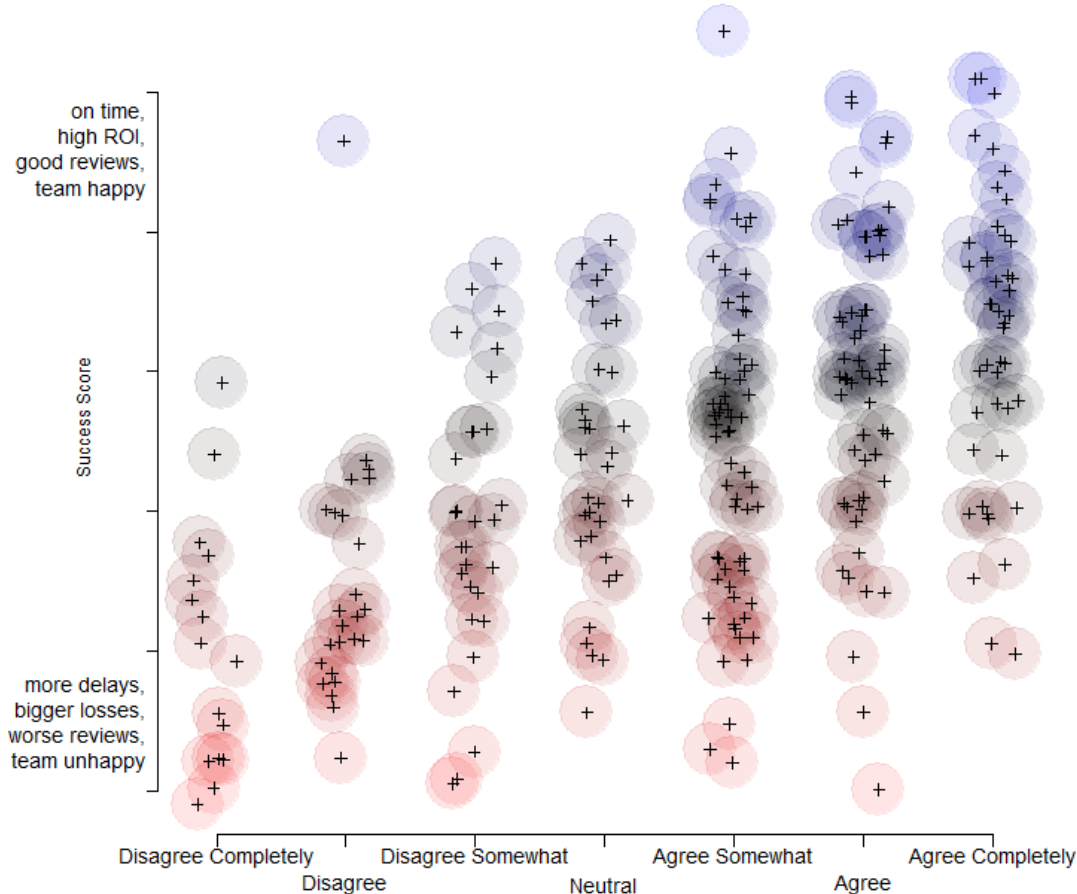
- Committed, cohesive team
- Resolving disagreements
- Clearly communicate and justify design changes
- Clearly communicated plan
- Shared vision for the game

Bad:

- Frequent design changes to the core of the game
- Competing/confusing goals

3

The team believed enthusiastically in the vision for this game.



Good:

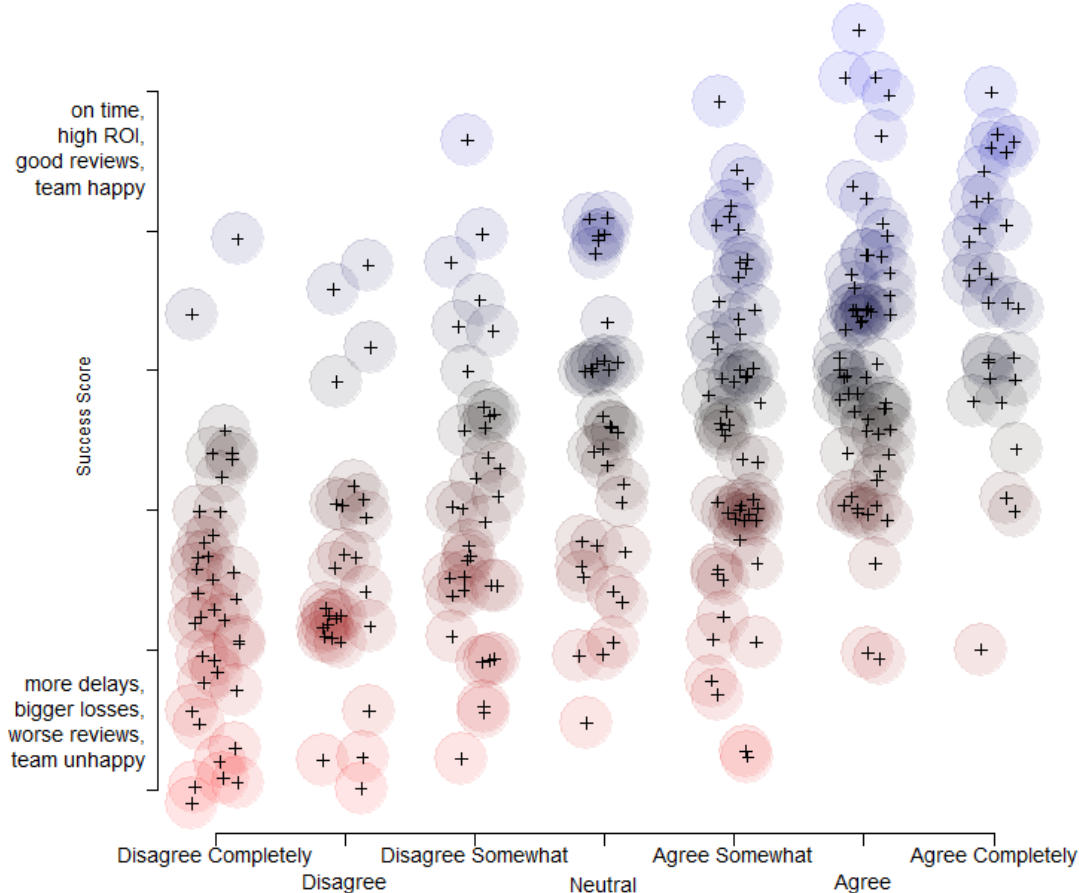
- Committed, cohesive team
- Resolving disagreements
- Clearly communicate and justify design changes
- Clearly communicated plan
- Shared vision for the game

Bad:

- Frequent design changes to the core of the game
- Competing/confusing goals

2

Our final game closely resembled the initial vision for the game.



Good:

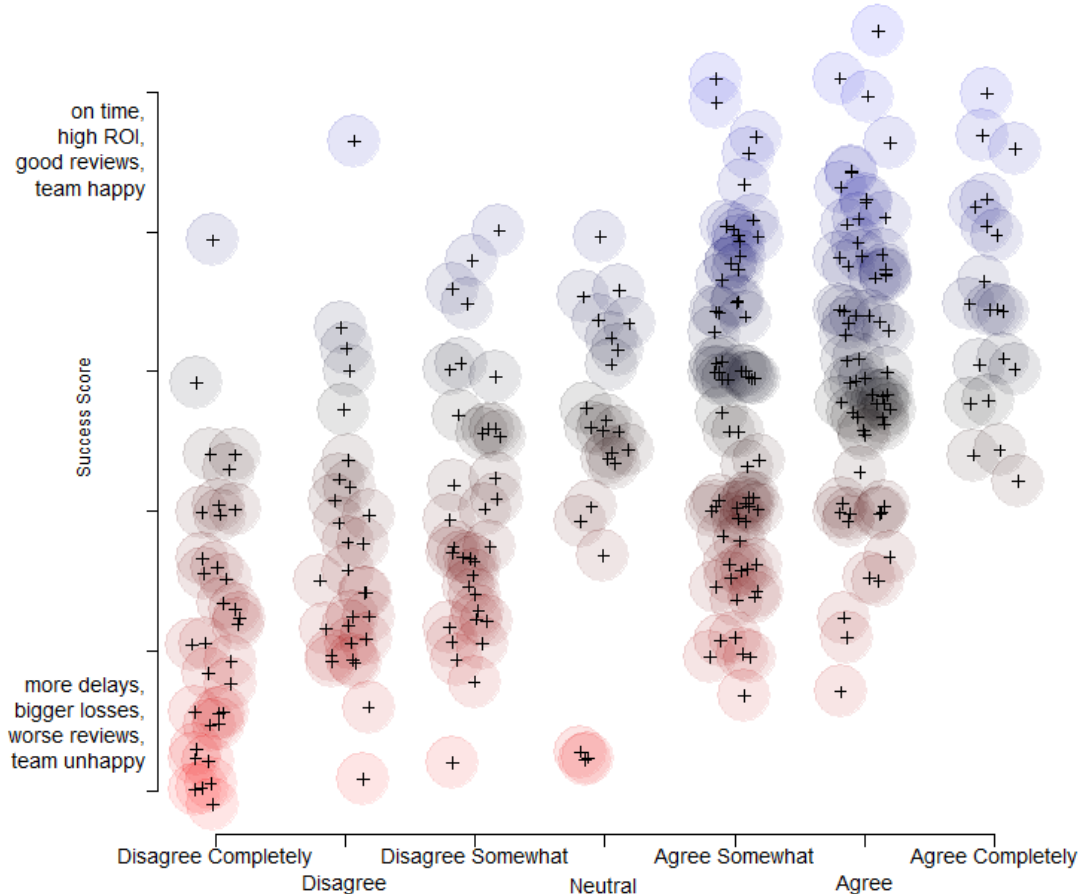
- Committed, cohesive team
- Resolving disagreements
- Clearly communicate and justify design changes
- Clearly communicated plan
- Shared vision for the game
- Stable vision for the game

Bad:

- Frequent design changes to the core of the game
- Competing/confusing goals

1

The vision for the final version of the game was clear and well-communicated to the team.



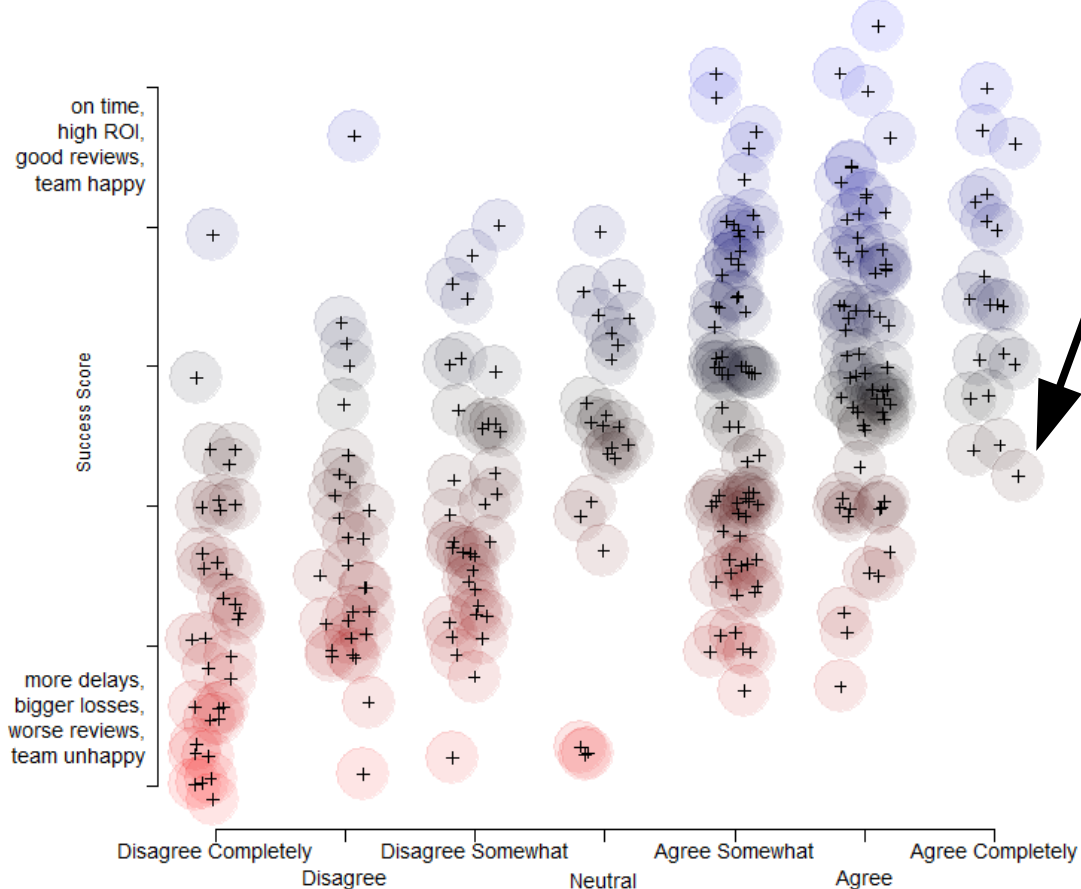
Good:

- Committed, cohesive team
- Resolving disagreements
- Clearly communicate and justify design changes
- Clearly communicated plan
- Shared vision for the game
- Stable vision for the game

Bad:

- Frequent design changes to the core of the game
- Competing/confusing goals

The vision for the final version of the game was clear and well-communicated to the team.



What did this team do wrong?

Our team sometimes seemed to be stuck in a cycle of never-ending crunch / overtime work – Agree completely

There was a lot of turnover on this project – Agree Somewhat

Team members would often work for weeks at a time without receiving feedback from project leads or managers – Agree somewhat

If we worked overtime, I believe it was because studio leaders or producers failed to scope the project properly – Agree

Internal politics played a big role in the dynamics of the team – Agree

Members had easy access to communicate with senior leadership about the project, raise concerns, or discuss personal issues – Disagree

Some team members put their own careers above the collective goals of the game project – Agree completely

The entire team met frequently to openly discuss topics of interest and ask questions – Disagree

What sort of technology solution did the team use to create the game? -
New engine

What did this team do wrong?

Our team sometimes seemed to be stuck in a cycle of never-ending crunch / overtime work – Agree completely

There was a lot of turnover on this project – Agree Somewhat

Team members would often work for weeks at a time without receiving feedback from project leads or managers – Agree somewhat

If we worked overtime, I believe it was because studio leaders or producers failed to scope the project properly – Agree

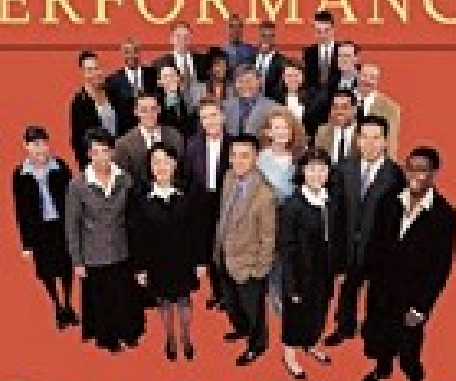
Part 6 Main Points

- It's all about the vision for the game.
 - Have a clear vision
 - Communicate that vision to the team
 - Make sure they believe in it
 - Resolve disagreements
 - Make sure there are no competing/confusing goals
 - Don't change the vision too often.
 - If you do change it, communicate the change clearly and make sure everybody is on board

Part 7: OK How The Heck Do We Do That?

LEADING TEAMS

SETTING THE STAGE FOR
GREAT
PERFORMANCES



J. RICHARD HACKMAN

NEW YORK TIMES BEST-SELLER

The FIVE DYSFUNCTIONS of a TEAM

A LEADERSHIP FABLE



PATRICK LENCIONI
AUTHOR OF THE NATIONAL BEST-SELLER *THE ADVANTAGE*

Copyright © 2009 by McGraw-Hill
THE NEW YORK TIMES BESTSELLER

12

The Long-Awaited Follow-Up
To The Bestseller
**FIRST, BREAK
ALL THE
RULES**

THE ELEMENTS OF GREAT MANAGING

Based on Gallup's ten million workplace interviews—
the largest worldwide study of employee engagement

RODD WAGNER & JAMES K. HARTER, PH.D.
Copyright © 2009 by McGraw-Hill

“What Went Right and What Went Wrong”: An Analysis of 155 Postmortems from Game Development

Michael Washburn Jr.¹, Pavithra Sathiyarayanan¹, Meiyappan Nagappan¹,
Thomas Zimmermann², Christian Bird²

¹Rochester Institute of Technology, Rochester, NY, USA

²Microsoft Research, Redmond, WA, USA

{mdw7326, ps2908}@rit.edu, mei@se.rit.edu, {tzimmer, cbird}@microsoft.com

ABSTRACT

In game development, software teams often conduct postmortems to reflect on what went well and what went wrong in a project. The postmortems are shared publicly on gaming sites or at developer conferences. In this paper, we present an analysis of 155 postmortems published on the gaming site Gamasutra.com. We identify characteristics of game development, link the characteristics to positive and negative experiences in the postmortems and distill a set of best practices and pitfalls for game development.

Keywords

Games, Postmortems, Qualitative analysis.

1. INTRODUCTION

Over the past thirty years, the importance and market-share of video games in the world of software has grown by leaps and bounds. In lockstep with this growth, the scale of work required to develop games, whether in terms of budget, size of codebase, or team makeup, has ballooned and is on par with or exceeds any other software endeavors [13]. Games are arguably the most sophisticated and complex forms of software [18].

Indeed, games have been the driving factors behind many

created from scratch, and deadlines are incredibly tight [12].

Therefore it is important to understand both the challenges that game development efforts face as well as the best practices that teams use to build games more effectively. The challenges are real problems faced by complex software efforts and represent avenues for research for our community. Successes and best practices embody knowledge that can aid future game development efforts and in some cases may generalize to or can be adapted for software development in non-game contexts. Because game development makes up a large slice of commercial software, a non-trivial proportion of students in computer science and software engineering programs will work on games during their careers. An understanding of game development can help educate and prepare such students.

Interestingly, game development has received very little attention in the academic community, as only three of the 116 open and closed source projects studied in the major software engineering conferences in two years were games [15]. Thus, one might reasonably expect that getting an inside view of game development is limited to a select few. Fortunately, the game development community has a unique practice that belies this assumption. Development teams often conduct postmortem retrospectives and share them publicly on gaming sites such as Gamasutra.com and at gaming conferences such as the Game Developers Conference (GDC).

Game Industry Specific Lessons

- Everybody has problems with scoping. Avoid feature creep like the plague
 - Allowing individuals to take creative risks is important
 - But not if it blows the schedule
 - If you have a realistic schedule, it's easier to take creative risks
- Repeated pattern of using prototyping and playtests to clarify and to communicate
 - Preproduction is not about generating ideas. It's about clarifying your ideas and making sure everyone has the same idea
 - Half the point of playtesting is realizing “oh *that's* the game we're making”

Prototyping

- Easy to fool yourself
 - “Well I think the prototype is fun”
 - “It will be great if we do it with higher production values”
- Easy to test the wrong thing
 - Make a fun 2D prototype that sucks in 3D
 - Make a prototype that’s fun in isolation, but you have no idea how it interacts with the rest of the game

What Actually Works

- Find your sound
 - Every rock band has their own sound. Every game studio has their own style
 - What game can this team make?
 - What is this tech good at?
- Be incremental
 - Make a sequel
 - Make something in an existing genre
 - Do a new take on an old game
 - Make a bigger version of a small game

Incrementalism

- Incremental doesn't mean boring
- Gone Home couldn't have happened without Bioshock
- Bioshock couldn't have happened without System Shock 2



Making a New Game or Feature

- If you have to make a new thing, your number one job is to clarify what that new thing is
- Reduce uncertainty in as many ways as you can
- What is clear in your head may be uncertain to other team members
- Braid used the game mechanics of classic platformers
 - That removes a huge amount of uncertainty from the design
 - The time traveling mechanic also had a prototype



Reuse Existing Ideas

- If your team can play an existing game where an idea works, that is better than any prototype
- You can take something that's a small idea in an existing game and make it a big idea in your game
- Gameplay with rich vocabularies are great for reuse
 - Shooters
 - Platformers
 - Third person combat
 - RPG mechanics
 - ...

Summary for Vision

- How to achieve a clear, well communicated vision that everybody believes in and that doesn't have to change too often:
 - Find your sound
 - Be incremental
 - If you have to make a new game, find ways to be incremental
 - Avoid feature creep
 - Prototype new ideas
 - Playtest to communicate and clarify

Main Points from the Survey

- Build a team where people feel safe. Safe to take creative risks and safe to speak up
- Give frequent direct feedback. Listen to feedback
- Have a realistic schedule
- Have an engaged team with high standards
- Have collective goals, avoid competing goals, resolve disagreements
- Work hard to clarify and communicate your vision

Thanks!

malte.skarupke@avalanchestudios.se

www.probablydance.com