

Part of the "Understanding Agile" series of talks and presentations.



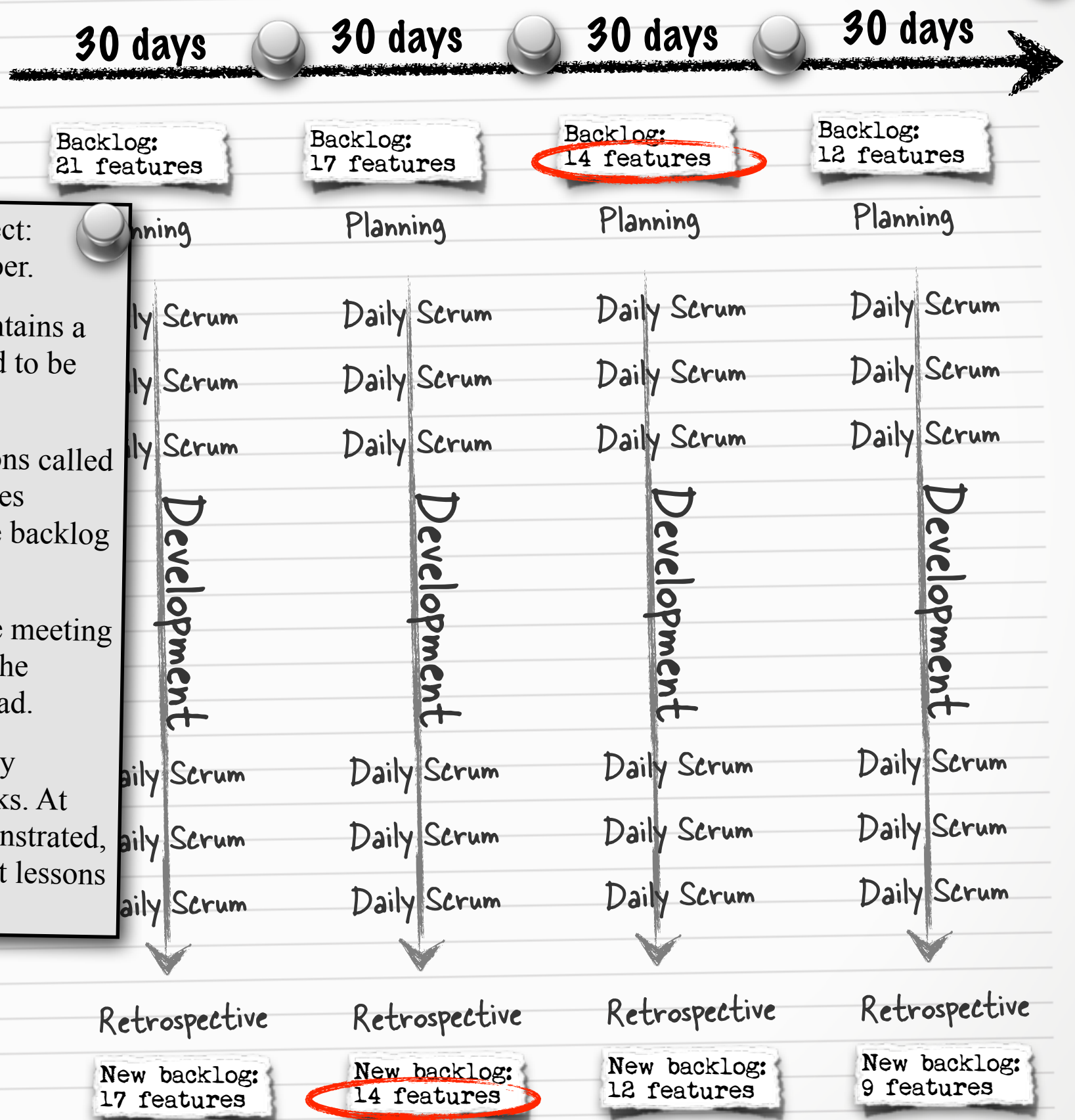
Scrum and Self-Organizing Teams

Courage, Openness, Pigs, and Chickens

Andrew Stellman and Jennifer Greene <http://www.stellman-greene.com>

Basic pattern for a Scrum project

- There are **three main roles** on a Scrum project: product owner, Scrum Master, and team member.
- The team (including the product owner) maintains a **backlog** of features and requirements that need to be built, organized by value and difficulty.
 - The software is built using timeboxed iterations called **sprints**. At the start of each sprint, the team does **planning** to determine which features from the backlog to build.
 - Every day, the team holds a short face-to-face meeting called a **Daily Scrum** to update each other on the progress they've made, and the roadblocks ahead.
- The **Scrum Master**, keeps the project rolling by helping the team identify and remove roadblocks. At the end of the sprint, working software is demonstrated, and the team holds a **retrospective** to figure out lessons they've learned so they can improve.



The basic Scrum pattern is really simple, and most teams have no trouble adopting it. If it's so easy, then why aren't all of us doing Scrum?

Scrum values give direction to the practices and roles.

“Grand principles that generate no action are mere vapor. Conversely, specific practices in the absence of guiding principles are often inappropriately used.”

-- Jim Highsmith, *Agile Project Management*

Practices

- Project backlog
- Sprints, sprint planning, sprint backlog
- Daily Scrum meeting
- Task boards and burn-down charts
- Sprint and project retrospectives
- User stories, story points, velocity

Roles

- Product owner
- Scrum Master
- Team

FUN FACT: DID YOU KNOW THAT SCRUM DOESN'T ABSOLUTELY REQUIRE TEAMS TO USE USER STORIES?

Scrum Values

- * **Commitment:** Each person is committed to the project's goals
- * **Respect:** Team members respect each other
- * **Focus:** Everyone is focused on the work
- * **Openness:** Each team member is aware of the work everyone else is doing
- * **Courage:** Team members have the courage to stand up for the project

Scrum has its own set of values. If they clash with the values built into the culture of the company, that's a major barrier to adoption.

Effective Scrum teams are self-organizing

Ken Schwaber wrote this in "Agile Project Management with Scrum" – that's the book that introduced Scrum to the agile community.

For Scrum to work, the team has to deeply and viscerally understand collective commitment and self-organization. Scrum's theory, practices, and rules are easy to grasp intellectually. But until a group of individuals has made a collective commitment to deliver something tangible in a fixed amount of time, **those individuals probably don't get Scrum**. When the team members stop acting as many and adopt and commit to a common purpose, the team becomes capable of self-organization and can quickly cut through complexity and produce actionable plans.

For a Scrum team to become effective, they need to do more than just follow the basic Scrum pattern. Effective scrum teams are self-organizing; without that, team members feel like they're just going through the motions.

Different team members can be motivated by different things



Developer/Architect

WORKING WITH
THIS
TECHNOLOGY
WILL LOOK
GREAT ON MY
RESUME.

NOW I'LL
DEFINITELY GET
THAT
PROMOTION.



Scrum Master

ONCE I PROVE
MYSELF HERE,
I CAN MANAGE
A LARGER
TEAM.

IF I LAND
THAT BIG
CLIENT, MY
BONUS WILL
BE HUGE.



Product Owner



Team Lead

Every one of us has our own self-interest, and there's nothing wrong with that. But motivating each team member as an individual isn't an effective way to bring a team together.

Teams self-organize around elevating goals

- Teams can be motivated by many things: getting an opportunity to work on a new technology or in a domain that they want to learn about, the possibility of a promotion, a performance bonus, working from home.
- Teams can be motivated by negative things, too: the boss will be mad, you'll get yelled at (or less money, or fired!).
- Those things, positive or negative, motivate people to work for themselves and their own interests, not come together as a team.
- Teams are motivated best by elevating goals, which bring them together for a higher purpose that they all care about. It's only with an elevating goal that a team can self-organize.
- Delivering value can be a very effective goal.
- When a team has an elevating goal that they truly believe in, and they're given the room to decide for themselves how to accomplish it (and the freedom to take risks) they will work hard to use the tools they have available to solve any problem standing in the way of that goal.

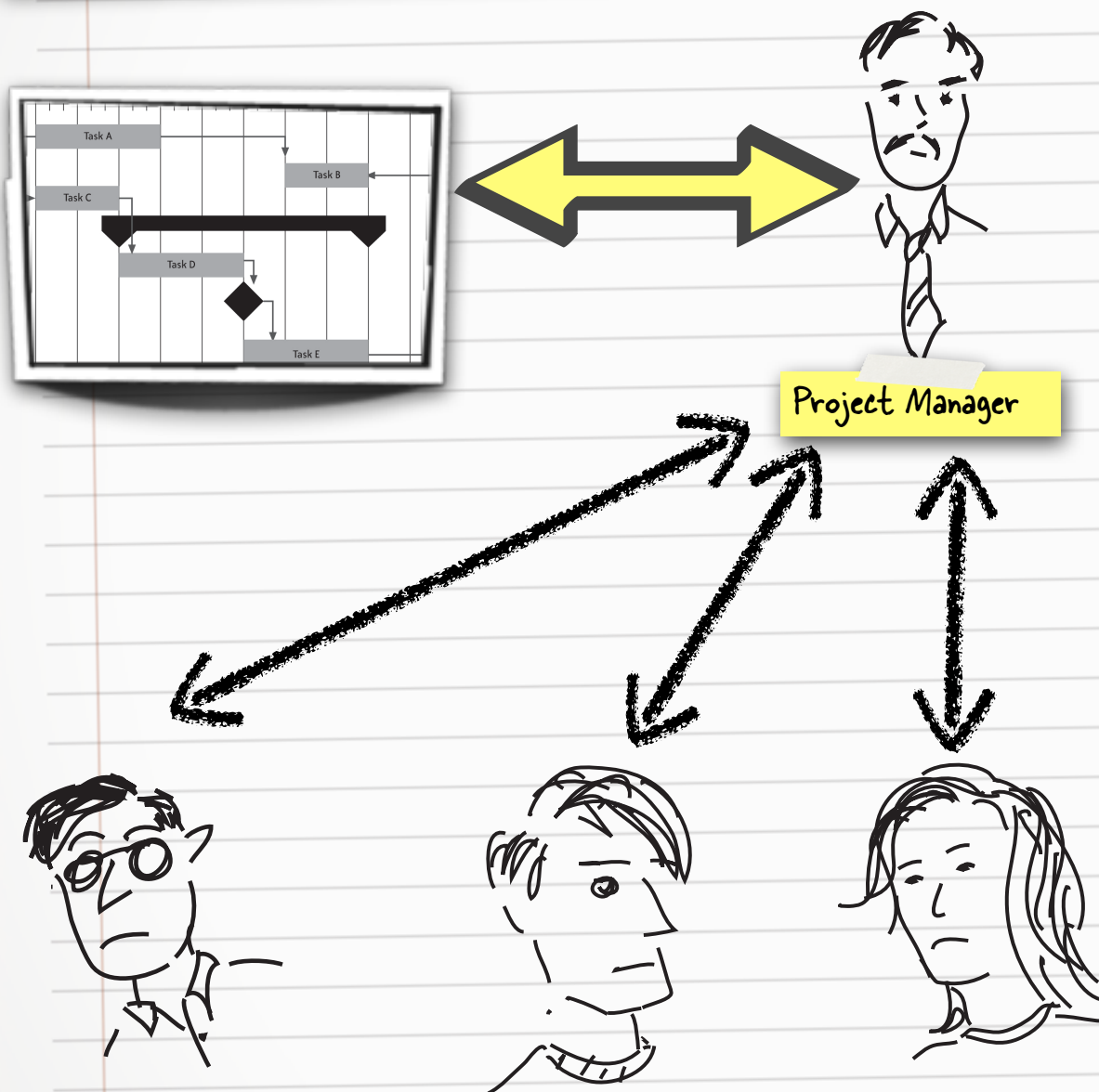
Is "the boss really needs this done" an effective elevating goal that the team can rally around?

"If you're out digging ditches, that's not very elevating or inspiring. But if you're digging ditches to protect your town that's about to be attacked by an enemy, well, that's more inspiring, even though it's the same activity. And so the leader's job, really, is to try to determine or frame the activity in such a way that people can understand what the value is." -- Steve McConnell, *Beautiful Teams* (Chapter 16)

"You can't always tell when a team is dysfunctional. If there's negative shouting and screaming, that's a problem. But sometimes, you can be on a team where everyone's trying, but nobody's communicating, and nobody's reaching their goals. The easy analogy is that during the day you've got people digging a hole, and at night other people are filling it up. Everybody's working really hard digging and filling holes, but in the end nothing of value is actually occurring." -- Scott Ambler, *Beautiful Teams* (Chapter 26)

"One of the signs that I have for the health of an organization is that they're reticent to fail. Organizations that are totally anal and refuse to fail at all tend to be the least innovative organizations, and they're hardly any fun, because these people are so fearful of failing they take the most conservative actions. On the other hand, organizations that are freer in failing, not in a way that will destroy the business, but are given some freedom to fail, are the ones that are more productive, because they're not in fear for their life with every line of code." -- Grady Booch, *Beautiful Teams* (Chapter 9)

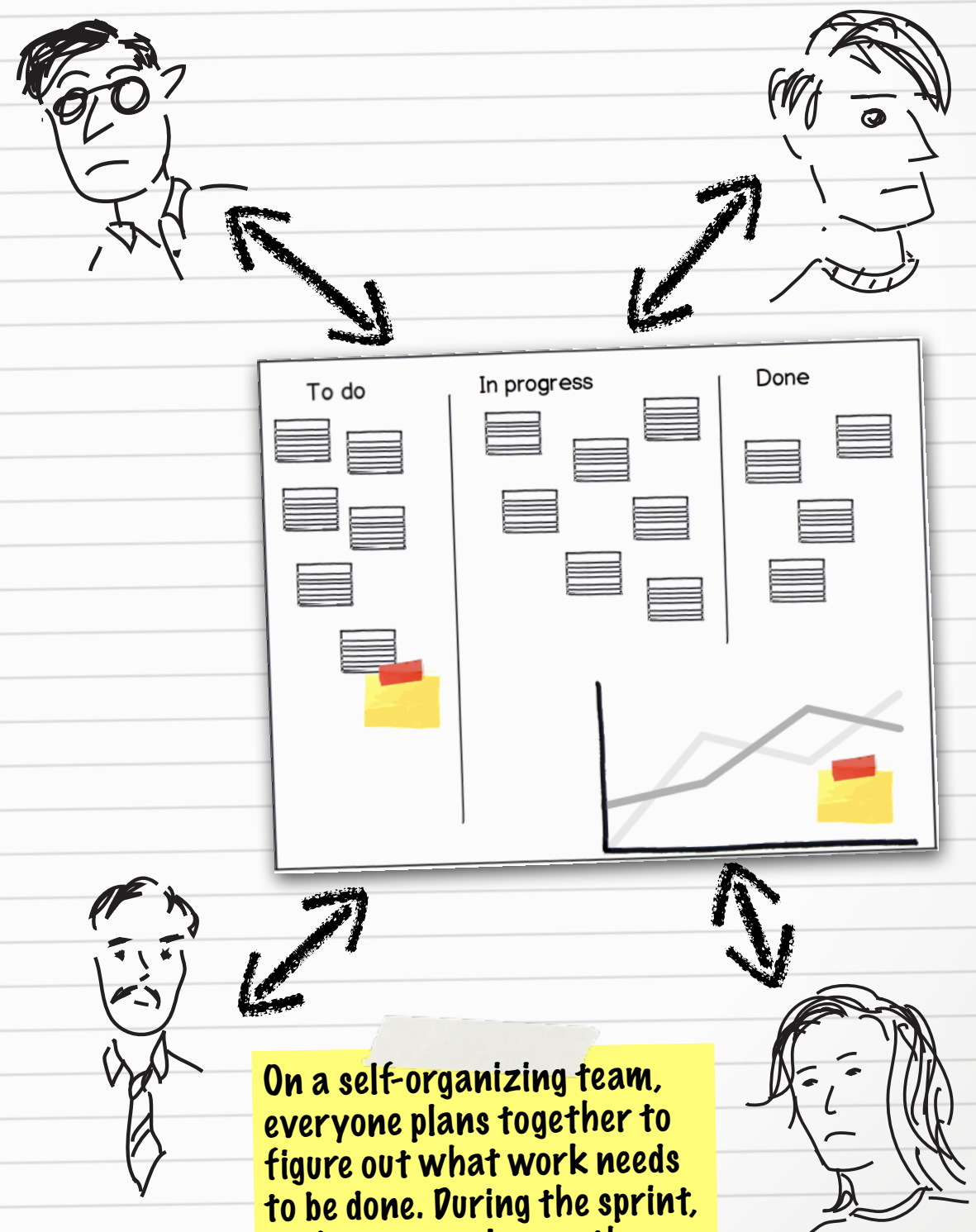
Command and Control



"Command" refers to the way work is assigned to the team (rather than the team deciding together who does what).

"Control" refers to the way changes are handled, with a project manager constantly monitoring the project and updating the plan when changes are discovered.

Self-Organizing



On a self-organizing team, everyone plans together to figure out what work needs to be done. During the sprint, each person takes on the work that the project needs.

Benefits of Command and Control

- * If you know up front what you want to build, command-and-control is very efficient.

- * You can define the goals and plan up front, and track your project's progress against it.

Drawbacks

- * Decision-making for the whole project rests on the shoulders of the project manager.

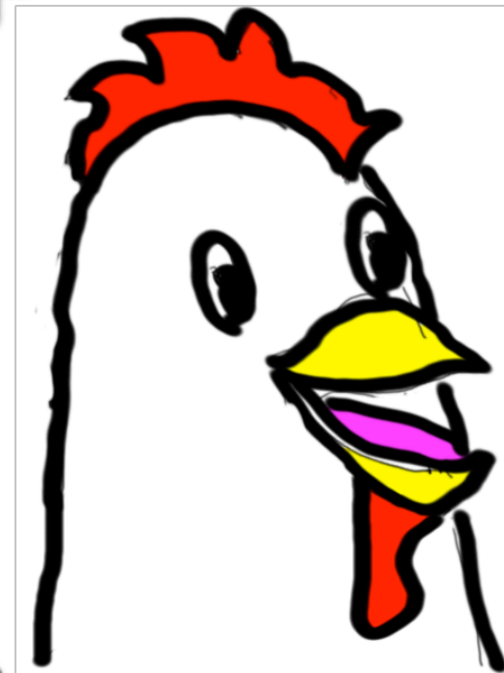
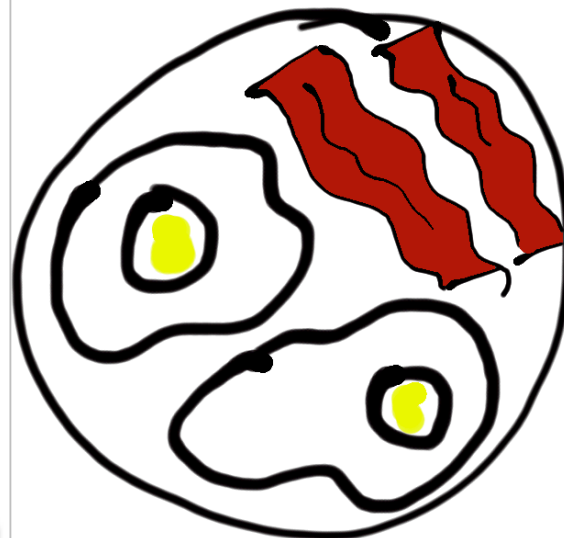
- * Adjustments to the plan have to be made through an often cumbersome change control process

How often are projects planned perfectly from the beginning?



Pigs

- Pigs genuinely feel that in order for them to succeed, their project needs to succeed.
- When all team members are pigs, it means that everyone is committed.
- It also means that everyone will do any job, if that's what the project needs.



Chickens

- Chickens are simply assigned to or interested in projects, but not committed to their success.
- And that's okay! Every project needs its chickens: people who sell the product, support it, or run the company.
- But it's very easy for even an experienced Scrum team to fall back into a chicken mentality.

One thing that Scrum is asking you to do is to be a pig: while you're working on a sprint, making the project successful is more important than any other professional goal that you have.

It's easy to talk about courage, but it's a lot harder to show it when it means saying no to your boss about an important feature.

The Daily Scrum is all about openness because it gives everyone on the team all of the information. That can be uncomfortable for some people.

The plan doesn't commit us. Our commitments commit us, the plan is just where we write those commitments down.

Scrum Values

REVISITED

- * **Commitment:** Each person is committed to the project's goals
- * **Respect:** Team members respect each other
- * **Focus:** Everyone is focused on the work
- * **Openness:** Each team member is aware of the work everyone else is doing
- * **Courage:** Team members have the courage to stand up for the project

A team that's truly focused is not asked to multitask. If you want someone to get two things done, ask for the first thing, then the second.

When a manager questions the team's estimates, what does that say about respect? What effect does it have on their sense of commitment?

The Scrum values are the values of self-organizing teams.

Imagine a conversation between the CEO and the Scrum Master about a project. The CEO asks what he's going to get with a two million dollar investment in software over the next year. The Scrum Master tells the CEO that he's not sure yet; they'll have an update every month, and at the end of the project they'll have at least two million dollars worth of software. No CEO in his right mind would ever approve that project. Why not?

ARE YOU OKAY WITH---

It's easy to get behind the Scrum values as abstract concepts, but a lot harder to stick with them if your company's culture clashes with them.

COMMITMENT

- * Relinquishing control of the project and trusting the team to decide what gets delivered?
- * Not going off on your own and building something without talking to the rest of the team, and just integrating it at the end?
- * Not having one "wring-able" neck?
- * Listening to comments and feedback, and not telling people to mind their own business?

RESPECT

- * Trusting the team to deliver each feature at the best possible date based on relative value and how the project progresses?
- * Giving the team enough time to do the work, and not demanding overtime of them?
- * Trusting the team to choose the tasks that are right for them and for the project, rather than relying on strict roles, RACI matrices, etc.?
- * Not being able to ever say again that you don't know why the team decided to do something the way they did?

FOCUS

- * Never asking anyone on the team to do work that's not part of the current sprint?
- * Never asking anyone on your team to do work that the whole team hasn't agreed to take on ("but I need this done now!")?
- * Putting what's most valuable to the company ahead of all other project or work concerns?
- * Thinking about project planning or users if you never have before?

OPENNESS

- * Not being able to predict what date each feature is going to be released?
- * Thinking about technical details if you never have before?
- * Thinking about what the programmer next to you is working on, and whether it really fits into the overall goal?
- * The programmer next to you thinking about your work in exactly the same way?

COURAGE

- * Not being able to blame lack of planning on your project manager?
- * Not being able to blame poor requirements that don't hit the mark on your product owner or senior managers?
- * Taking the time to really understand your users?
- * Building something that isn't perfect, because what the users need most is something that's simply good enough?

ARE YOU OKAY WITH---

What if you're not okay with some of these things?

Not every team in every company can self-organize from day one, and that's okay. If your company's culture doesn't match the Scrum values, your job as a Scrum adopter is to help the people around you understand its values.

Don't just adopt some practices and then call it a complete Scrum adoption. Adopting a new methodology is a process, and it takes time for people to learn a new way of doing things. Becoming a pushy agile zealot can make it harder to bring agile to your company!

Give your team new practices that they can do today, but also give them a vision of what they can accomplish tomorrow. Then they won't just assume that they're doing Scrum because they've adopted a few simple practices.

Many teams adopt Scrum practices without the Scrum values first, and that's okay--but only if everyone on the team shares the goal of self-organizing. When they work towards a real Scrum adoption together, everyone can help change the culture around them to accept the Scrum values.