

Retrospective – 6 steps

- Security first
- Collect facts
- Ask: What went well?
- Ask: What can be improved?
- Who is in control?
- Prioritize

Step 1: Security First

- "Regardless of what we discover, we understand and truly believe:
 - that everyone did the best job they could,
 - given what they knew at the time,
 - their skills and abilities,
 - the resources available and
 - the situation at hand." - Kerth (2001).

Everyone in the world is doing the best, given their level of consciousness. The way to improve then is to improve our level of consciousness i.e. to become more aware.

12/13/2023 3:01 PM

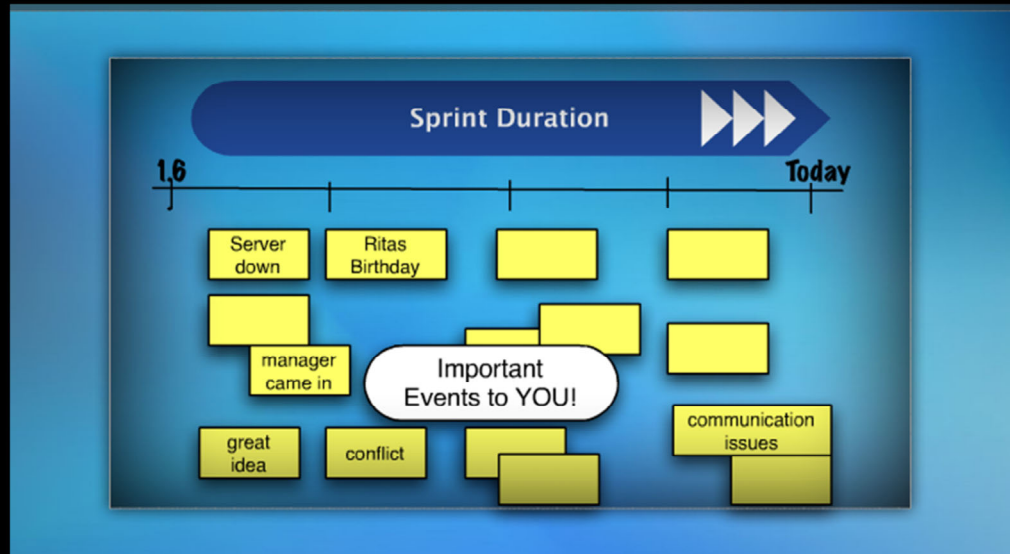
vijaynathani.github.io

Page 2

Keep this kind of message on a flipchart at the start of meeting.

Keep some distance for this meeting: Do not do it immediately after the sprint review. Don't invite senior management. Use a dedicated room (no disturbances).

Step 2: Timeline



12/13/2023 3:01 PM

vijaynathani.github.io

Page 3

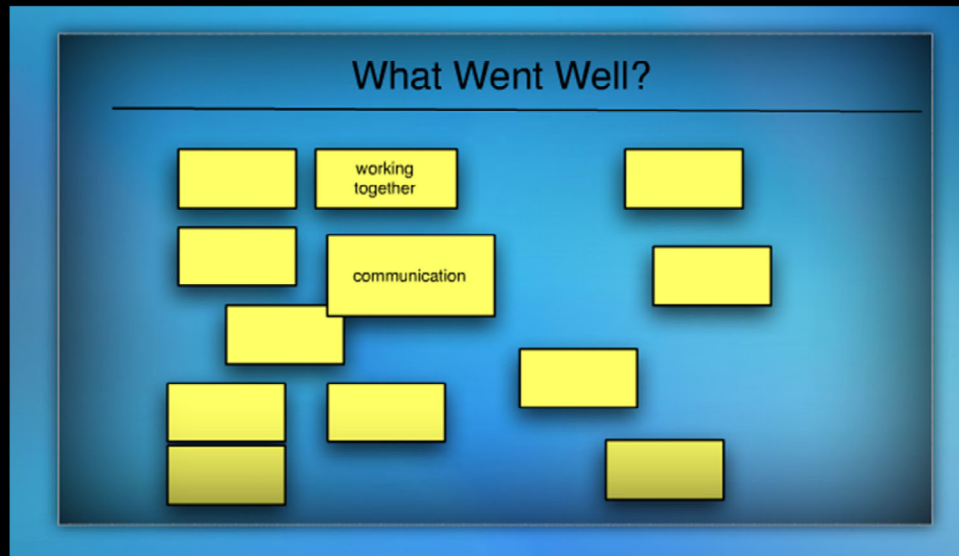
This step takes 10 to 30 minutes.

Here we talk about facts or events, not emotions. Everyone says items that are important for them, not for others.

Why do this? Because people tend to forget the items that have occurred or we are not sure whether everyone noticed an important event e.g. One team member had to sit late to finish his/her task. Everyone else went home early.

Each person says a short story about the event and puts the post-it at that date.

Step 3



12/13/2023 3:01 PM

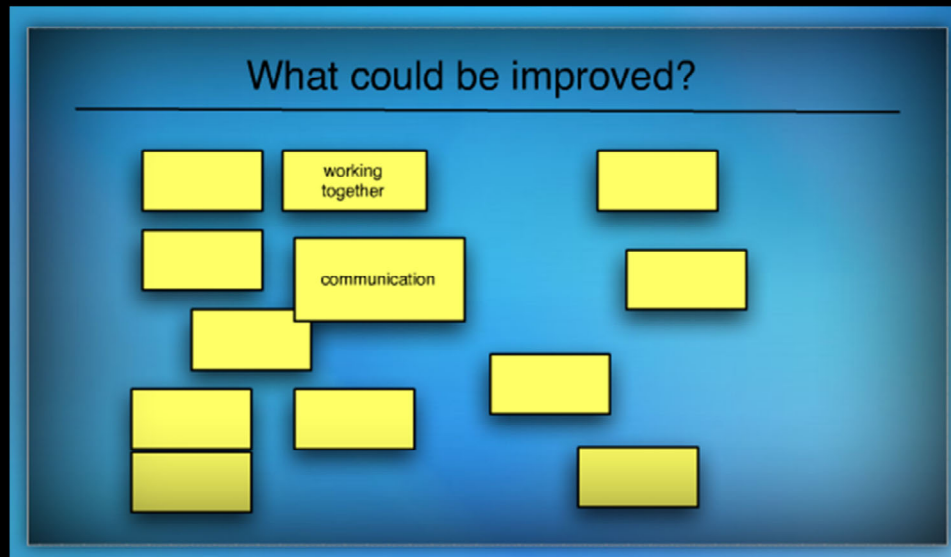
vijaynathani.github.io

Page 4

Everyone comes in good mood.

The sprint was great because

Step 4



12/13/2023 3:01 PM

vijaynathani.github.io

Page 5

Is there anything that we would repeat, if we have the opportunity. If yes then "No improvement"

If no, then what can be improved.

Mark every item

- C – Caused by Scrum
- E – Exposed by Scrum
- U – Unrelated to Scrum

12/13/2023 3:01 PM

vijaynathani.github.io

Page 6

At the end of the meeting, mark each item on the What's Working / What Could Work Better lists as either:

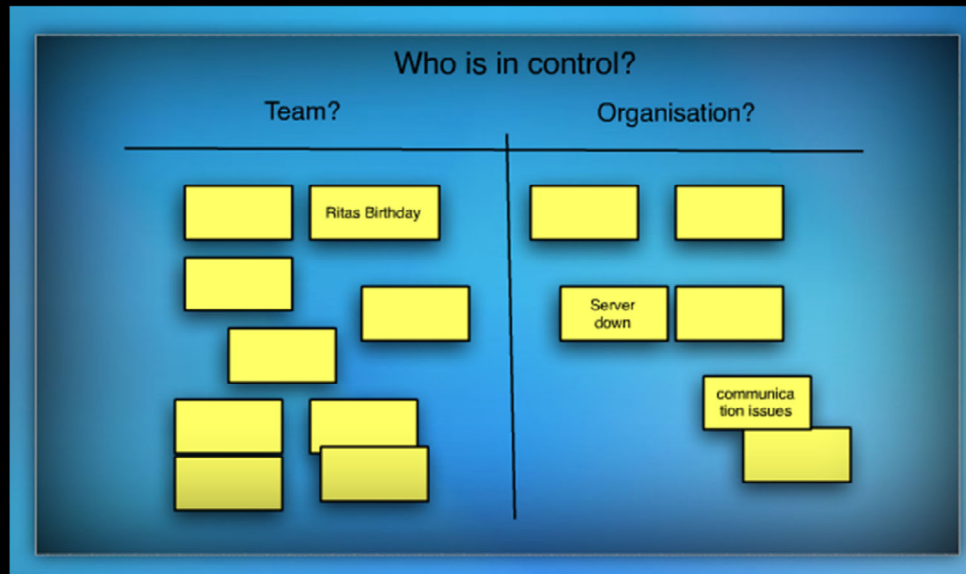
C = caused by Scrum (would not be happening without Scrum)

E = exposed by Scrum (would be there even if team were not using Scrum)

U = unrelated to Scrum (like the weather)

Then, add up the C, E, and U in each column

Step 5: Items to Improve



12/13/2023 3:02 PM

vijaynathani.github.io

Page 7

Who can make that improvement? Team or Organization.

Sample Picture



If the team is distributed, then consider tools like IdeaBoardz, Pointing Poker, Stromboard.

Part 1

Create 3 large lists (whiteboard or flipchart)

What's working

What's could work better

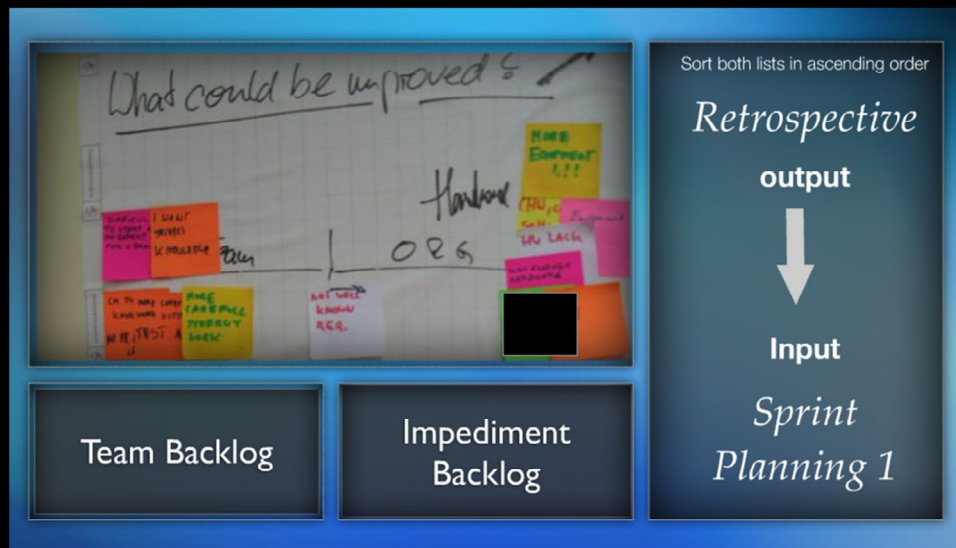
Things to try in the next Sprint

Go around the room, and give each person an opportunity to add 1 or more items to the 3 lists

If people agree with something already on the lists, put a tick mark next to them

Select a subset of the "Things to try..." list to try in the next Sprint (ScrumMaster responsible for tracking this)

Output should be used



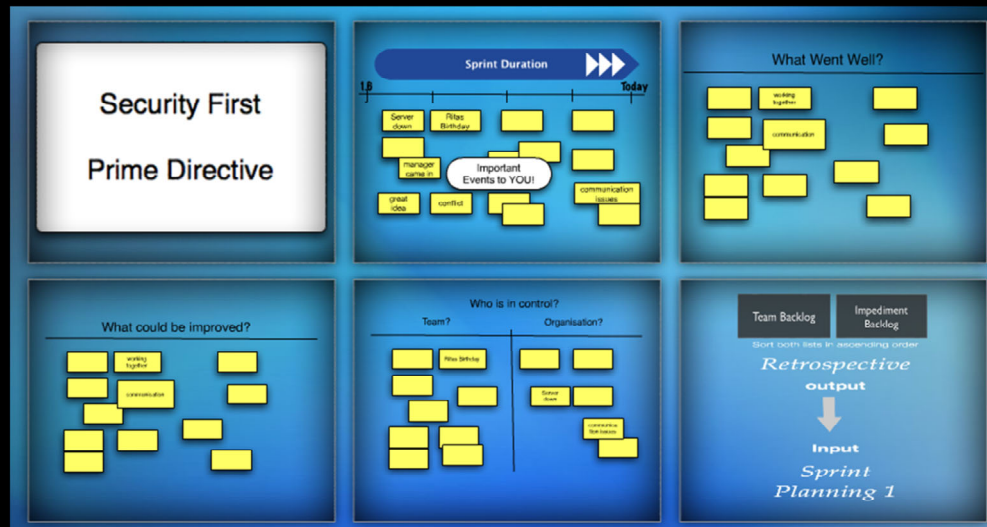
12/13/2023 3:02 PM

vijaynathani.github.io

Page 9

Team backlog is for the team to improve in the next sprint. E.g. Move one story to done and then move to another. So that testers can begin testing the done story. Impediment backlog is for the ScrumMaster to follow up e.g. If team is changing again and again, the velocity will be low.

Retrospective



12/13/2023 3:02 PM

vijaynathani.github.io

Page 10

Evaluation?



Product Owner



Scrum Master



The Team



Consultants /
Part time members

12/13/2023 3:02 PM

vijaynathani.github.io

Page 11

At cisco, the business person decides the features and evaluates the Team.

At ScrumWorks, the team is evaluated as a unit.

At Google, there is no evaluation of anyone. Each employee is supposed to maintain a web page which says

- What am I doing now?
- What are my objectives in the next months?

Menlo Innovations: Company takes a % of total revenue. The rest is decided by team regarding how to distribute among themselves.

Module: Using Scrum in Contract Software Development

A new doctrine goes through three stages. It is attacked and declared absurd; then it is admitted as true and obvious but insignificant. Finally, its true importance is recognized and its adversaries claim the honor of having discovered it. — William James

Contract Terms

```
the party in the first part shall remunerate the party in  
3.1.1 Vendor will demo potentially shippable software to  
Customer every 30 days. There will be no  
additional charge for this.  
3.1.2 Customer can replace any requirements that Vendor  
hasn't yet started working on with one or more of  
equal total size (in the estimate of Vendor) at any  
time. There will be no additional charge for this.  
3.1.3 Customer may request interim releases at any time,  
and will be charged an agreed-upon time and  
materials cost.  
3.1.4 If Customer's business goals are satisfied early,  
Customer may terminate contract early for 20% of  
the remaining unbilled contracted amount.  
notwithstanding the foregoing clause, the party in the  
first
```

12/13/2023 3:02 PM

vijaynathani.github.io

Page 13

Different Contracts:

- 1) T&M
- 2) Rolling Contract: Renew every 3 months.
- 3) Fixed price, fixed duration but variable features.
- 4) Price, Duration and Features fixed. Then use the above clauses.

Module: Managing Scrum Adoption in the Enterprise

You don't transition your projects to agile projects; you have to transform your people to think in agile ways.

The Big Picture

- The full and complete adoption of Scrum will result in a profound transformation of an entire organization
- Very few organizations are willing to change
 - Change is risky
 - Change hurts
 - Change requires work
 - Change involves loss and waste
- Very few organizations are able to successfully pull it off

12/13/2023 3:02 PM

vijaynathani.github.io

Page 15

A well documented process that everyone follows without question is the path to excellence

OR

The reason to standardize a process is so that people doing the work have something solid to question and change?

Companies with first view point will take much more time to go agile.

=====

Does it make you nervous when someone suggests that the development team should decide what can be done within a timeframe and tell managers, rather than other way around?

Does it seem unnatural for people to figure out for themselves what to do next rather than being told?

If yes for any of above, then agile is still far away.

Stages

1. Scrum adopted for a team.
2. Automation within the team.
3. Organizational changes.

12/13/2023 3:02 PM

vijaynathani.github.io

Page 16

In Step 1, many parts are manual

In Step 2, usually TDD and CI are setup.

In Step 3, support is needed from the organization to improve productivity.

Google Strategy

- Adopted company wide Agile process in 2001

When Rosing started at Google in 2001, "we had management in engineering. And the structure was tending to tell people, No, you can't do that." So Google got rid of the managers. Now most engineers work in teams of three, with project leadership rotating among team members. If something isn't right, even if it's in a product that has already gone public, teams fix it without asking anyone.

"For a while," Rosing says, "I had 160 direct reports. No managers. It worked because the teams knew what they had to do. That set a cultural bit in people's heads: You are the boss. Don't wait to take the hill. Don't wait to be managed."

And if you fail, fine. On to the next idea. "There's faith here in the ability of smart, well-motivated people to do the right thing," Rosing says. "Anything that gets in the way of that is evil."

Scrum is Hard

- Scrum surfaces all sorts of nasty stuff
 - Make sure people are prepared
 - Make sure they understand that this is Scrum working, not failing
- Help teams learn from each other
- Be ready to stage a rescue mission
 - There are some problems teams can't solve by themselves

12/13/2023 3:02 PM

vijaynathani.github.io

Page 18

Problems are normal. Don't panic. Don't despair.

Visible problem = Killable problem. Opportunity for improvement

Prioritize and fix problems. One by one.

Look back once in a while and pat yourself on the back.

Be Prepared to Use Guerilla Tactics to Get Things Done

- Some of the obstacles to Scrum are big
 - Organizational, policy, management...
- Many are small
 - Conference room table
- Focus on the big ones

Good and Bad News

- Culture eats Strategy for Breakfast

12/13/2023 3:02 PM

vijaynathani.github.io

Page 20

The Good News –very few of your competitors will be able to compete with you after you change, and very few of them have the guts to change.

The Bad News –very few enterprises succeed, even with compelling reason. Think of Toyota's warning and GM, Ford, and Chrysler's response.

Remember –Culture eats strategy for breakfast.

Find an Executive Champion

- Hold management accountable at board level to remove impediments.



12/13/2023 3:02 PM

vijaynathani.github.io

Page 21

The executive sponsor or champion should clear the path for the pilot teams.

Many champions may feel that once the team has been kicked off their work is done, but this is not so in an agile environment. These executives have to be available to facilitate and drive ongoing improvements in the organization.

There should be an open audience with the executive to inform them of problems that the agile team uncovers. As this is all about continuous improvement, the communication between the teams and the executives is also constant and never-ending.

The Urge to Tinker is Great

- If it isn't Scrum, don't let people call it Scrum
 - Everyone has a way to improve Scrum
 - Protect Scrum's good name!



12/13/2023 3:02 PM

vijaynathani.github.io

Page 22

If it is Scrum, it should have

- Automated testing for new code written / existing code modified. Code coverage for them should be between 80 to 95%.
- Continuous Integration and Tests running multiple times a day.
- Team is self-organizing. No command&control management.
- Production grade software is delivered to the customer regularly at short intervals e.g. every fortnight.
- The team constantly thinks about how to improve itself e.g. Retrospective meeting.

Break any of the above rules and it is not Scrum.

Scrum will Always Be Messy

- Scrum is about people, and people are messy
 - Inconsistent, insensitive, erratic, make mistakes
- For this reason, it will never be perfect
 - Idealists will always be disappointed
 - Opponents will always have ammo
 - Just keep asking, is it better than before

12/13/2023 3:02 PM

vijaynathani.github.io

Page 23

Common Sense:

Problem: Two weeks is too short

Solution: Use one week iteration. Once you have mastered that, two weeks will be a breeze.

Problem: There is not enough room to write requirements on cards.

Solution: Use smaller cards.

Basic Bootstrapping process

- Identify PO and SM
- Channel all requirements through PO
- Create a simple Product backlog
- Identify Scrum Team & seat them together
- Decide on Sprint length and definition of Done.
- Start sprint

12/13/2023 3:02 PM

vijaynathani.github.io

Page 24

Focus and efficiency yields much higher productivity than many heads and long working hours.

Don't worry. Don't wait. Just go.

Don't worry too hard. Just Experiment.

Listen to the team.

Scrum will uncover problems. Don't shoot the messenger.

Prioritize the problems. Fix them one by one.

Last but not the least

- Jump right in, the Water is fine



12/13/2023 3:02 PM

vijaynathani.github.io

Page 25

Don't try to fix everything before you start agile adoption. Just dive right in—you'll find a way to work within the current constraints.

The road to innovation is unpaved

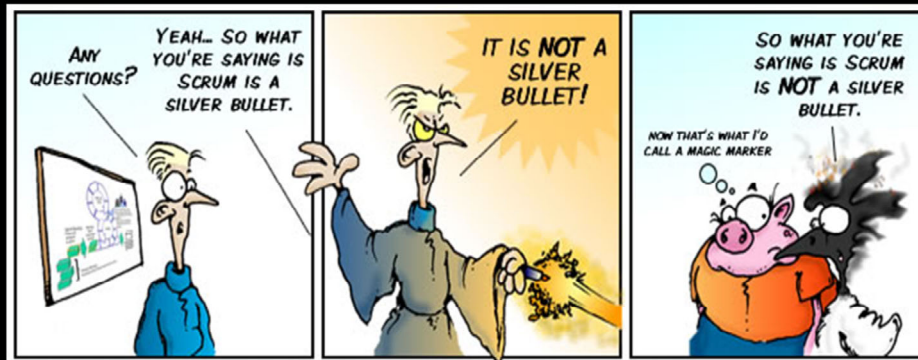
Use the iteration reviews and retrospectives to make change recommendations and implement incremental improvements as you go.

Be reasonable about this by focusing on only the top two or three things for the next iteration.

It's a backlog of recommendations and, like the backlog of product features, they all can't be implemented at once.

It is like in the movie matrix, Morpheus says "Unfortunately no one can be told what the matrix is. You have to see it for yourself."

Take this Home



"Scrum is like your mother-in-law. It's constantly pointing out your shortcomings." - Ken Schwaber.

The trick is that we're supposed to learn from that feedback and fix our problems.

12/13/2023 3:02 PM

vijaynathani.github.io

Page 26

Make a choice:

Use Scrum OR

Lose business/employees to competition.

Scrum is an effective technique for improving the software development efforts of many professionals. That's it, nothing more.

If you work hard; if you stay focused; if you take Scrum to heart; then you will likely improve your effectiveness as a developer

Agile is facing two issues:

1. Some people are looking for a silver bullet that makes software development easy. They hype Agile beyond its true value to support their hopes and dreams.
2. There is gold in the word Agile. You can make money by touting a development process that improves productivity and lessens the pain of development. If the true definition does not fit, create one that will attract customers. I believe some consultants have done this.

Agile is more like a raft on the raging currents. Work is still required to negotiate the hazards, but the raft was built with the hazards in mind.