Benjamin Day

Brookline, MA
Consultant & Trainer
Rental Chief Technology Officer
Therapist for Teams
Scrum, DevOps, Azure,
Team Foundation Server,
Software Architecture & Testing
Microsoft MVP
Pluralsight Author
Scrum.org Trainer
@benday

MVP Microsoft Most Valuable Professional

BDC Benjamin Day Consulting

PLURALSIGHT

PROFESSIONAL SCRUM TRAINER PST SCRUM.ORG LICENSED

Scrum.org The Home of Scrum

Azure DevOps Services Fundamentals

Azure DevOps Server 2020 Fundamentals

Scrum Master Skills

On with the show.

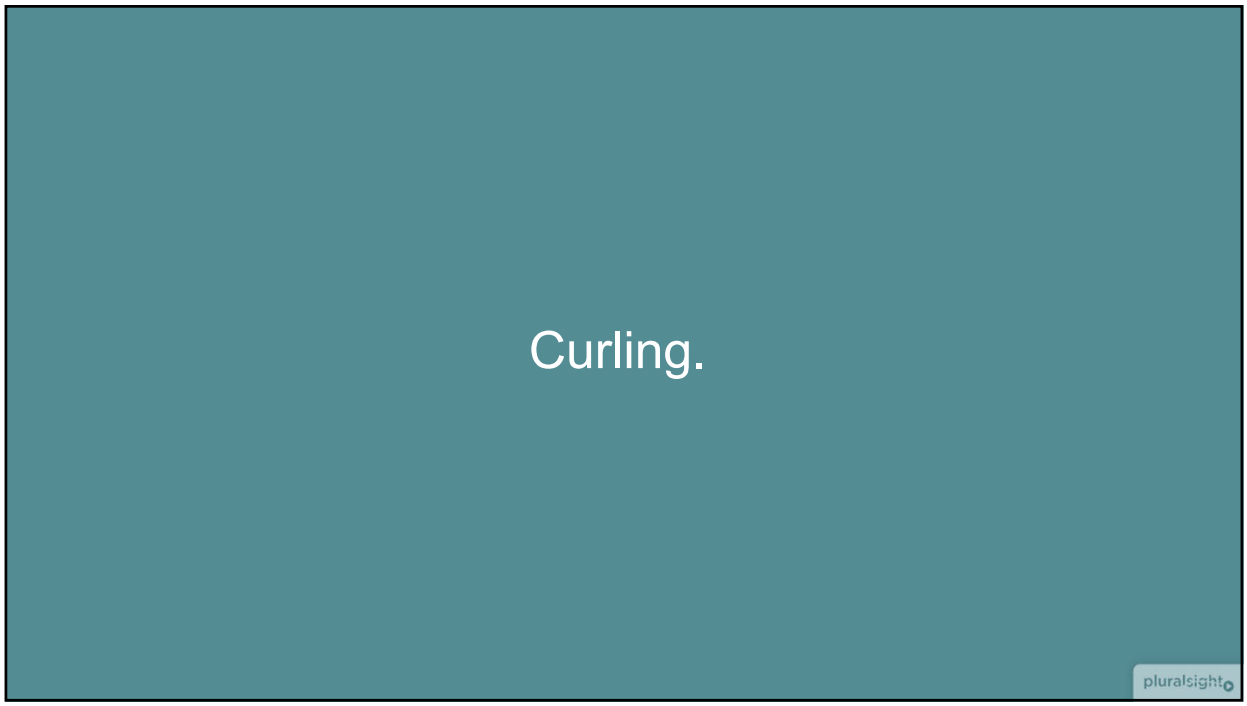
Assumptions

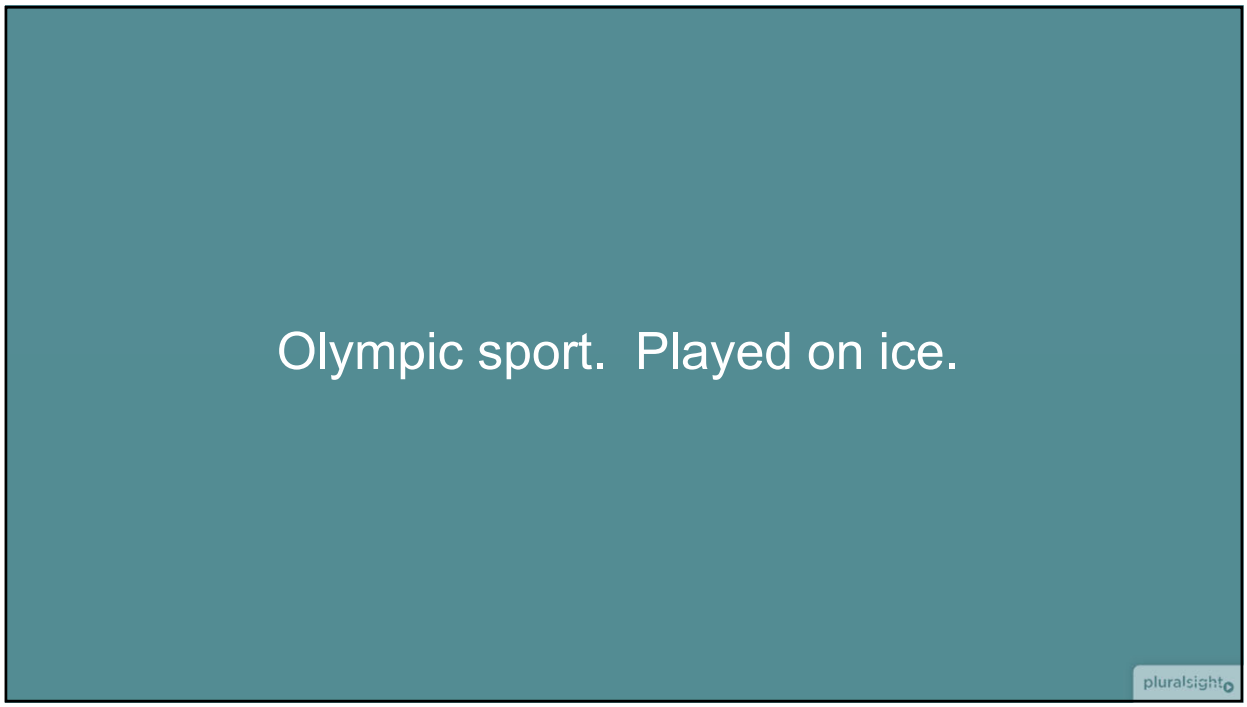
- You're technical. I'm technical.
- For today, we're talking about a fairly technical Scrum Master
  - (Don't forget the needs of the business though.)
- This talk =
  Scrum Mastering + "Getting your project started right"

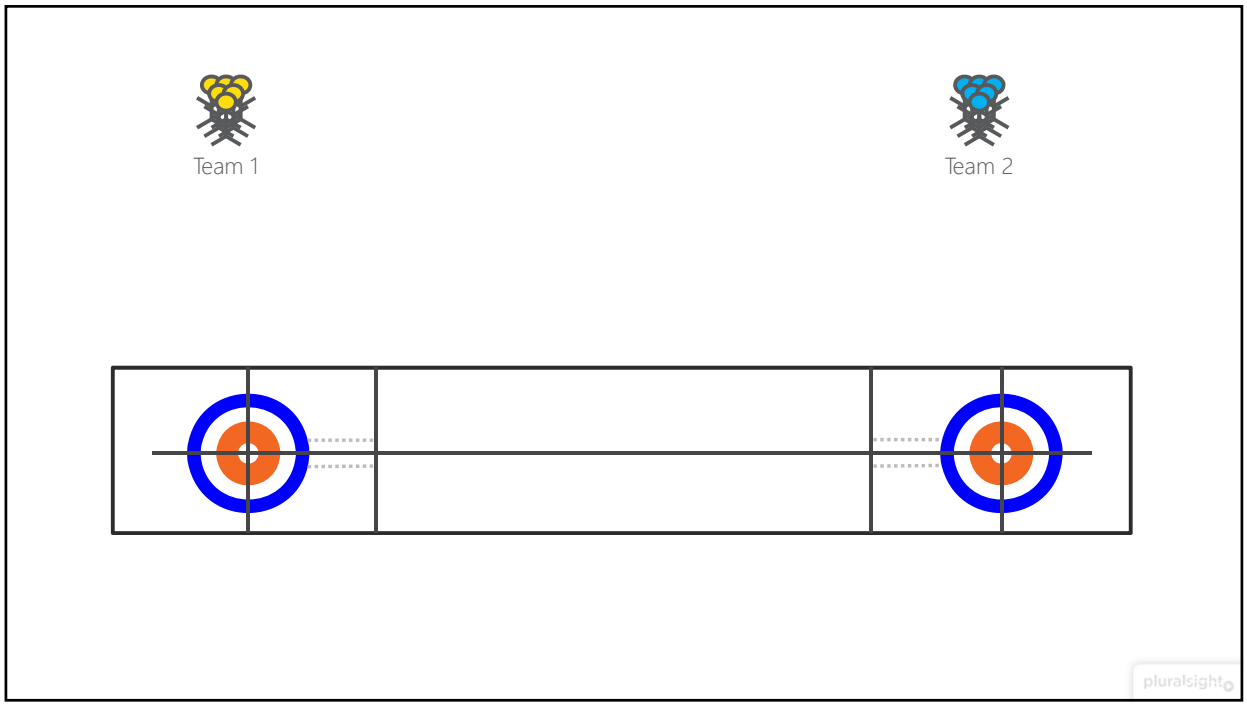
BDC

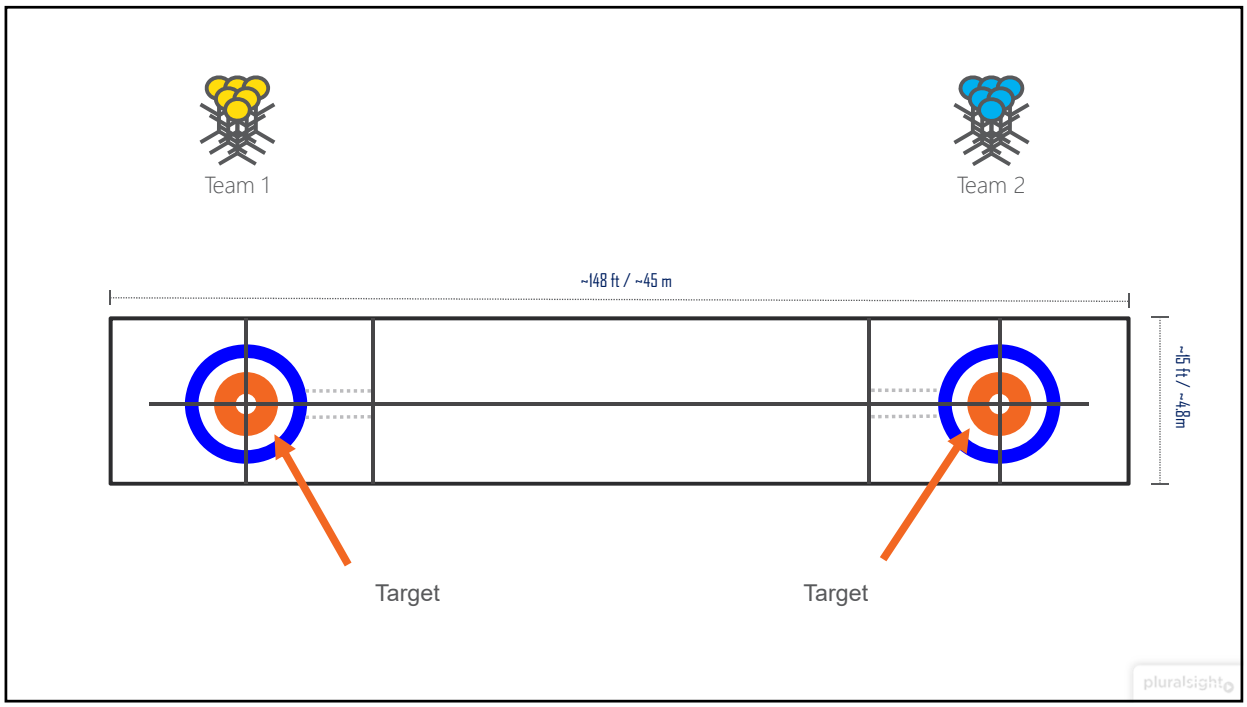
"What does the Scrum Master *do* anyway?"

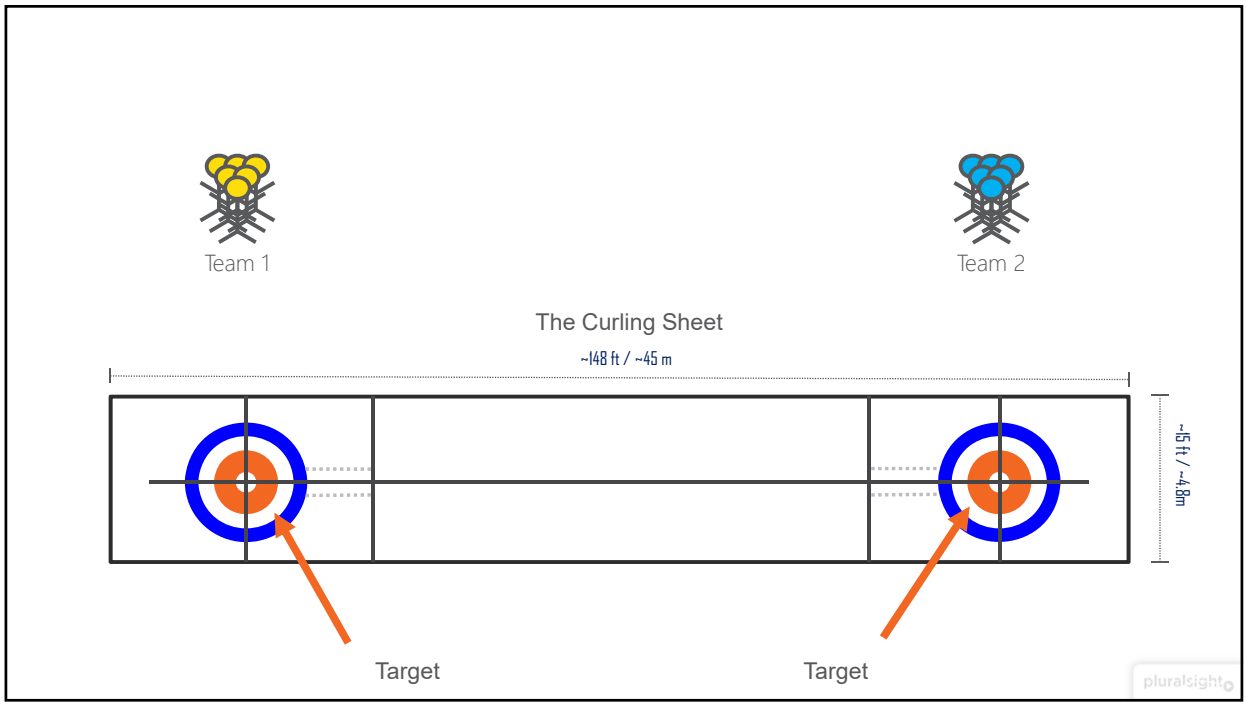
pluralsight


Curling.
pluralsight


Olympic sport. Played on ice.
pluralsight

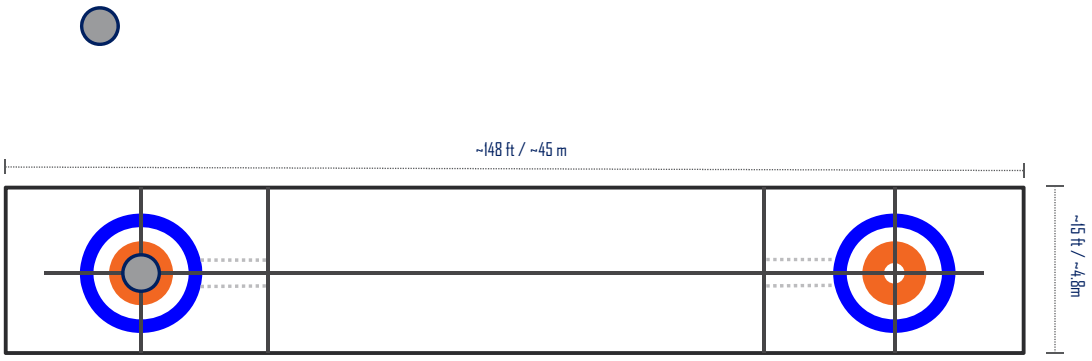

Team 1
Team 2
pluralsight


Team 1
Team 2
~148 ft / ~45 m
~51 ft / ~48m
Target
Target
pluralsight



Curling Stones

- ~40 lb. stones
- Polished granite
- It's got a handle
- 8 stones per team



- Get your stones as close as possible to the center of the target
- Each team takes turns
- Team with the most stones closest to the center of the target wins

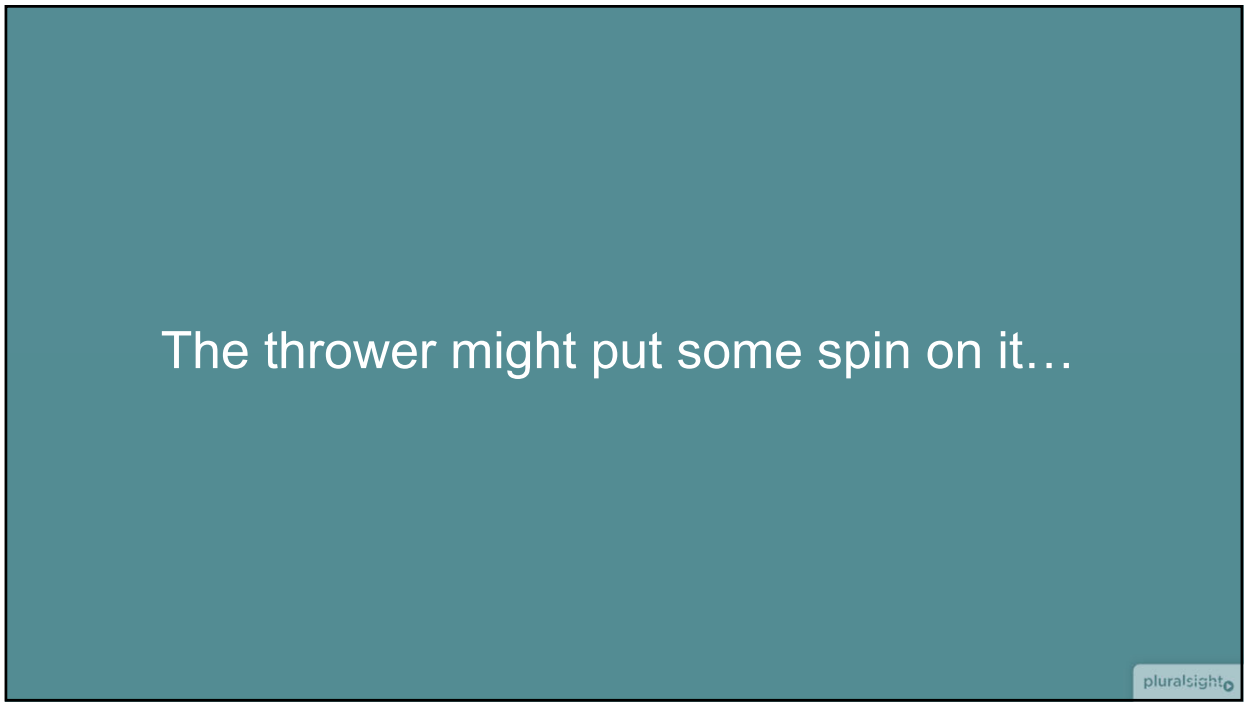
Note: that stone's not to scale

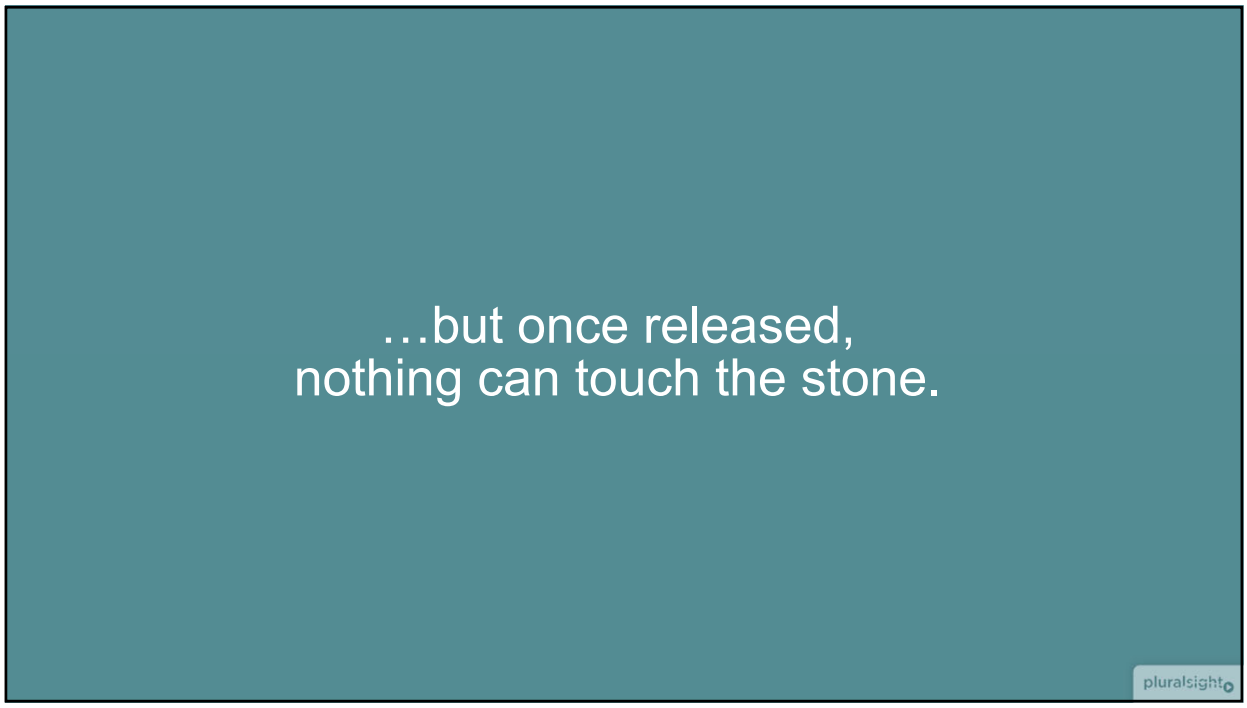
pluralsight

Throwing

- 3 team members per throw
- Thrower
  - Stone in hand
  - Pushes off with legs
- Starts glide towards target
- Thrower has to release stone

pluralsight


The thrower might put some spin on it…
pluralsight


…but once released,
nothing can touch the stone.
pluralsight

After the release,
the Sweepers take over.

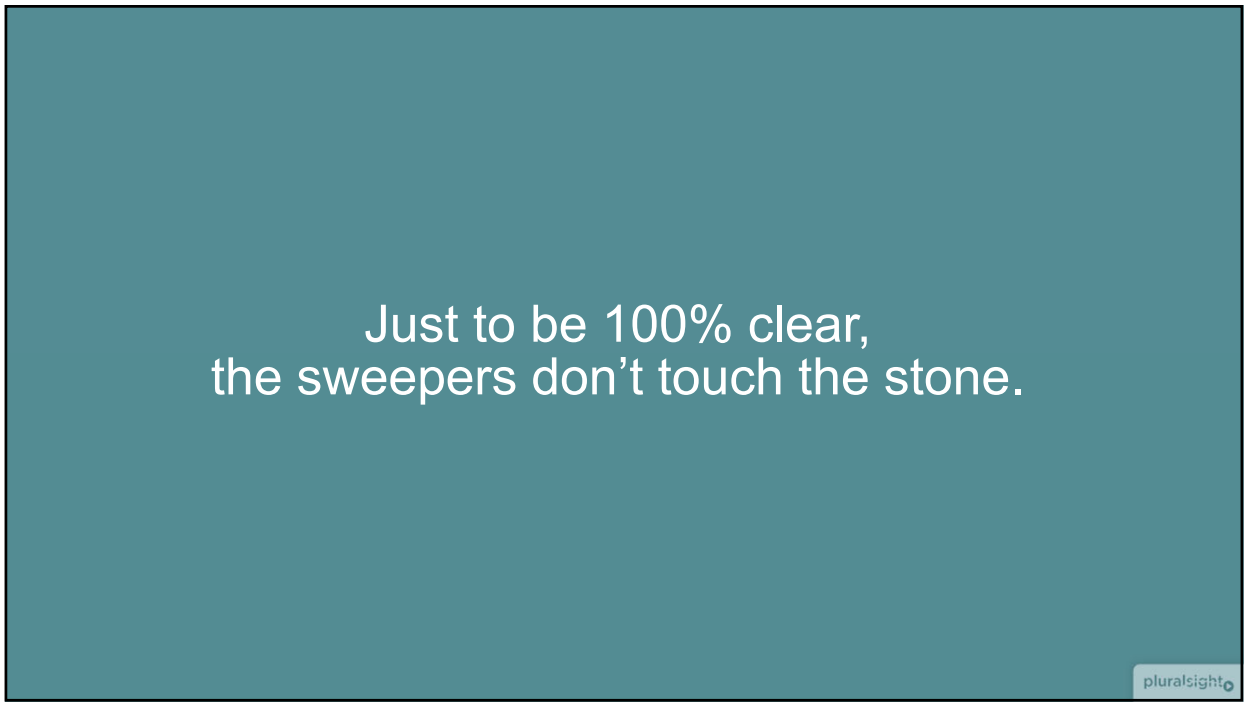
pluralsight

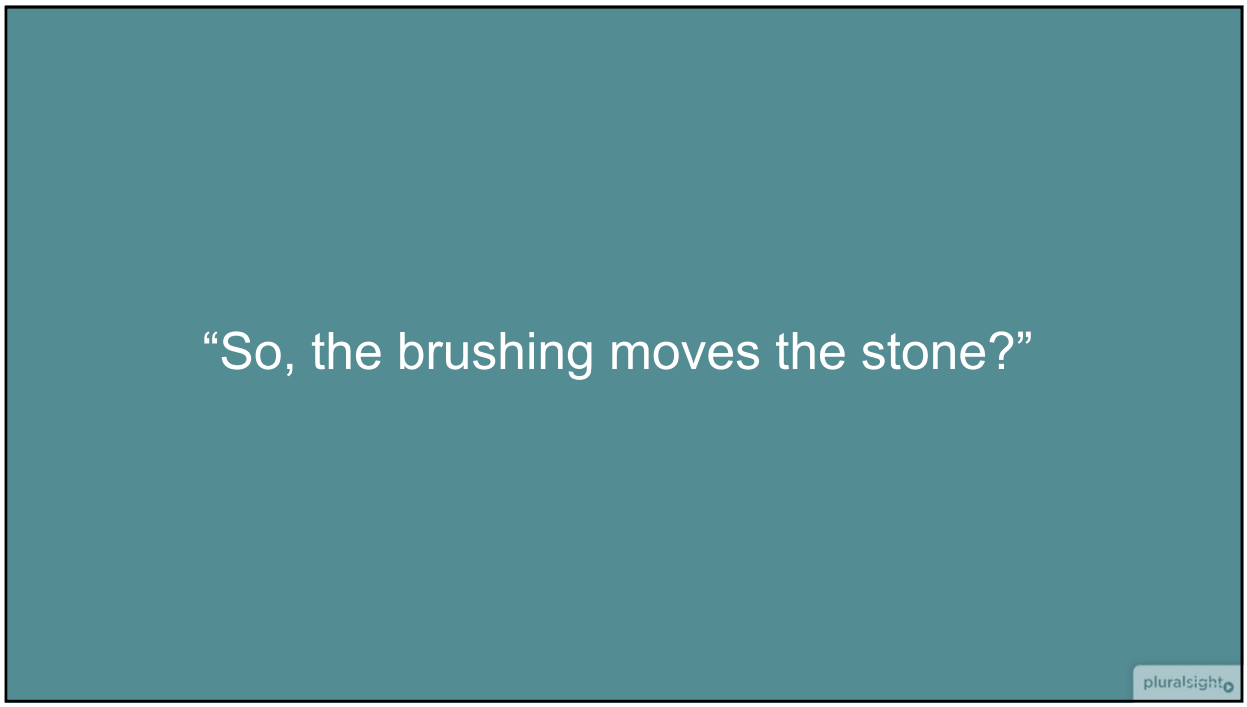
Sweepers

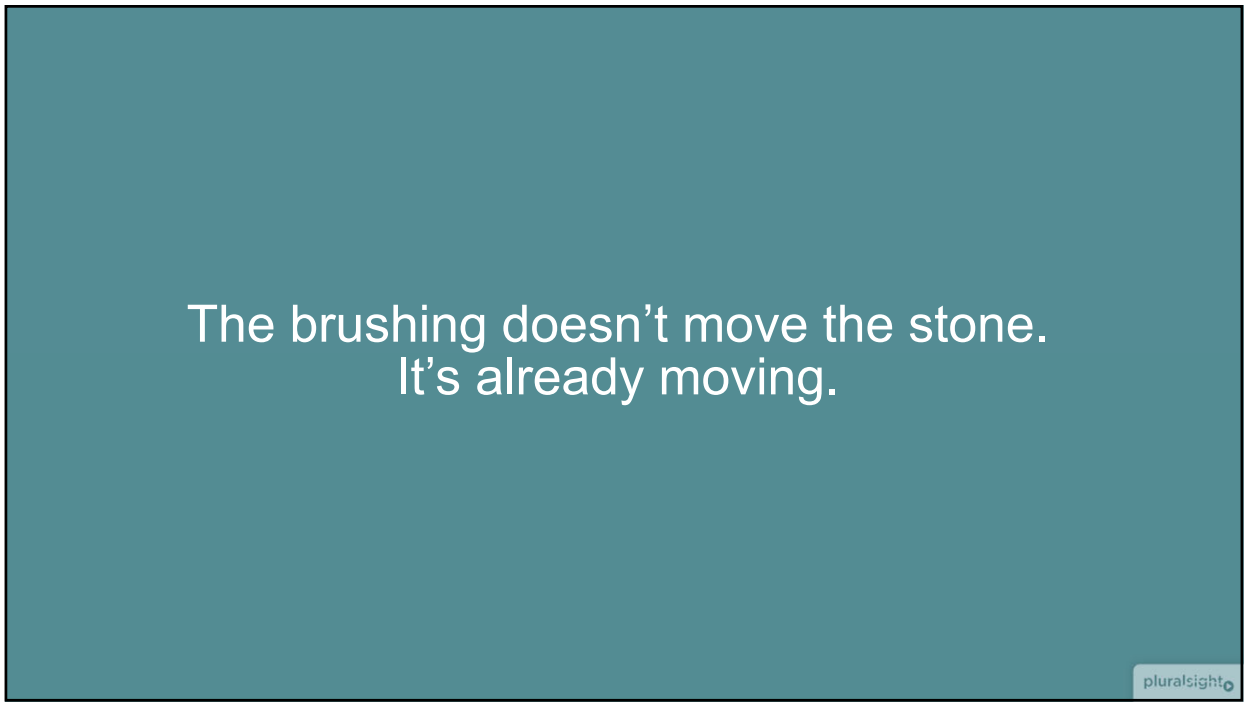
- Two sweepers stay with the stone

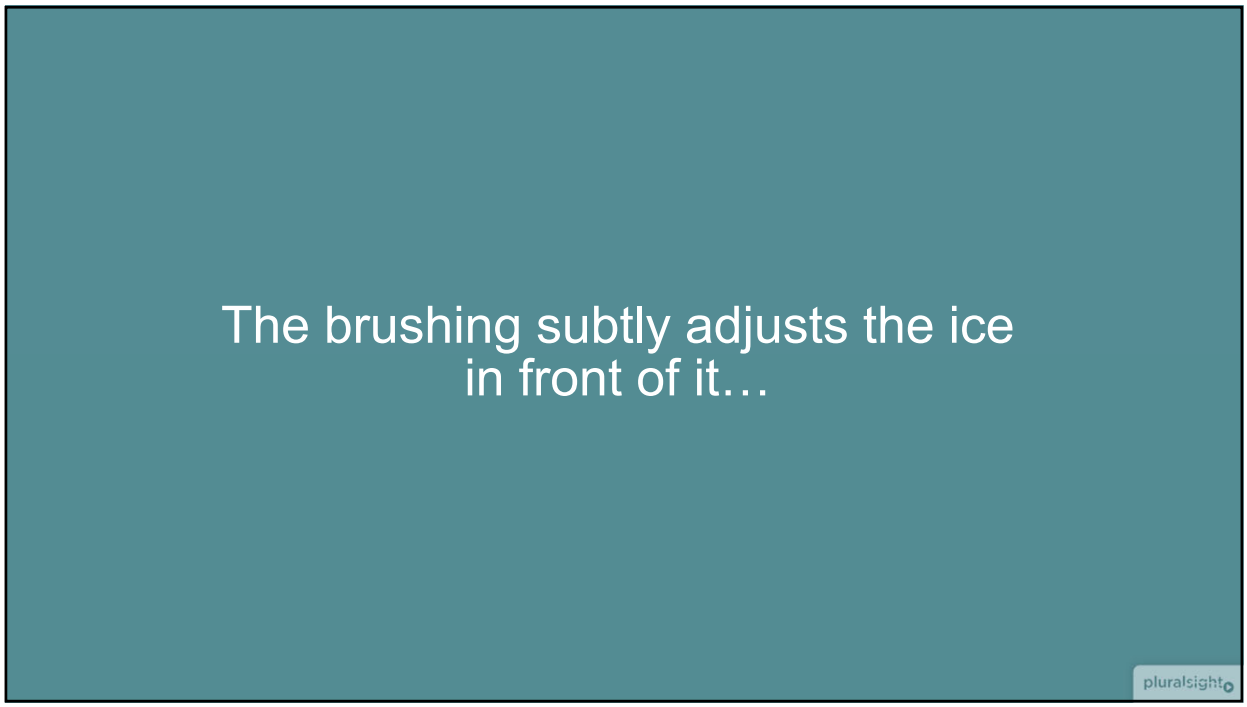
- Guide the stone to the target

pluralsight


Just to be 100% clear,
the sweepers don’t touch the stone.
pluralsight


“So, the brushing moves the stone?”
pluralsight


The brushing doesn’t move the stone.
It’s already moving.
pluralsight


The brushing subtly adjusts the ice
in front of it…
pluralsight

…and that can change the speed and direction of the stone significantly.

pluralsight

The sweepers – without touching it – are helping to coax and guide the stone so it lands correctly at the target.

pluralsight

"…and this has *what* to do with scrum mastering?

pluralsight

Scrum Master is a Coach

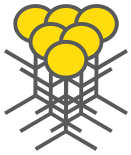
- Scrum Master is not a management role
- Team members don't report to the Scrum Master
- Scrum Master isn't the boss

pluralsight

Three Roles in Scrum

Product Owner

Scrum Master



Development Team

Three Roles in Scrum

Product Owner

- Product Owner is as close to “a boss” as you’ll get
- Provides the vision & goal

Considering the inherent bossless-ness,
Scrum expects and relies on the team
to self-organize.

pluralsight

To be successful,
the team needs to use their creativity to deliver
done, working software.

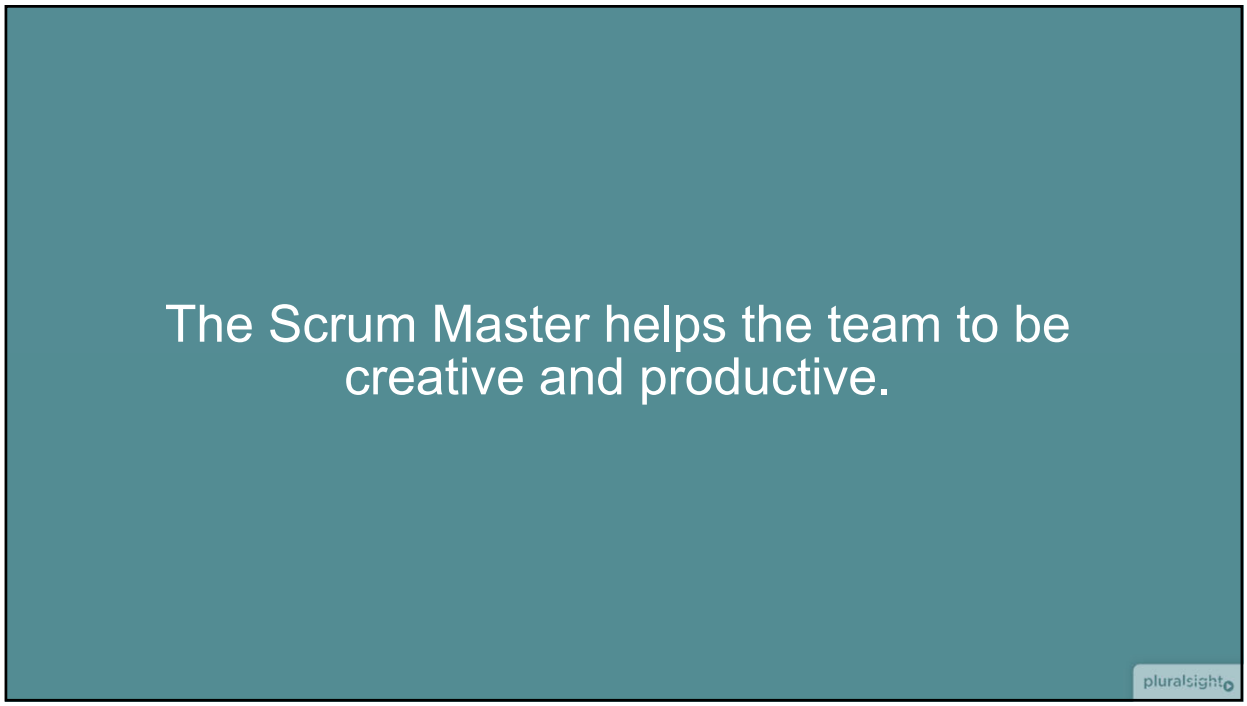
pluralsight

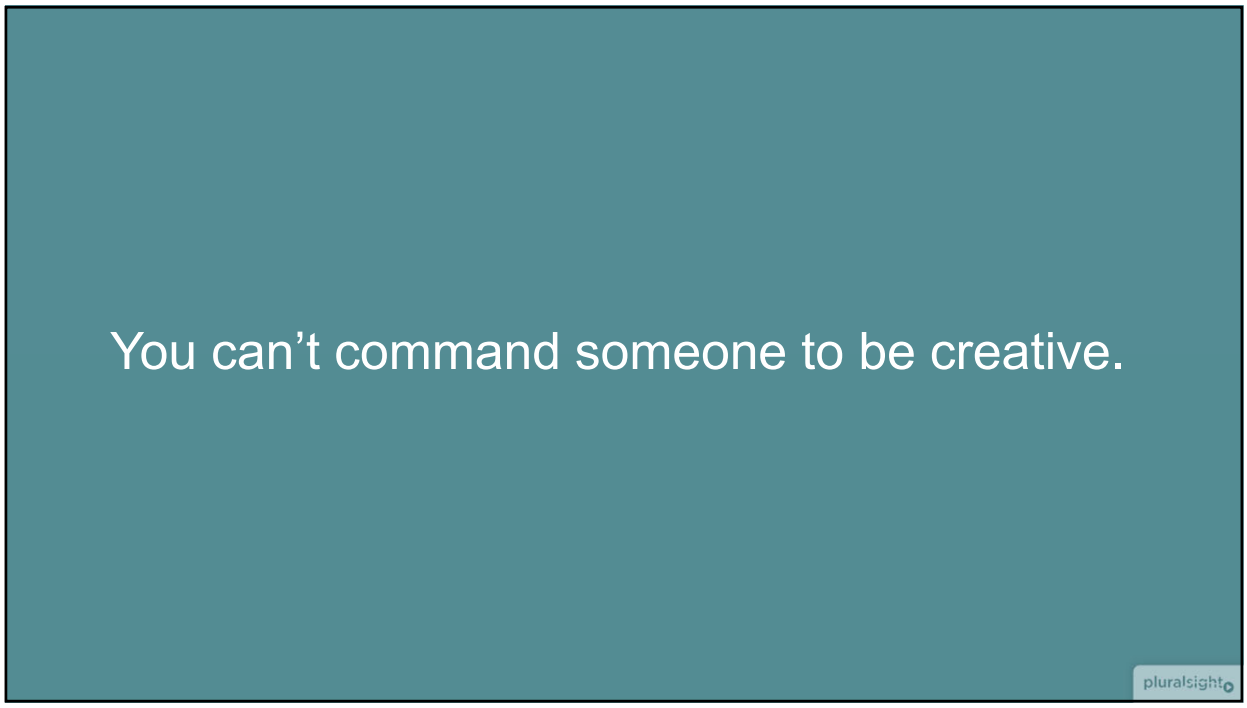
The team takes the Product Owner's goal and self-organizes to figure out how to deliver done, working software at the end of the Sprint.

pluralsight

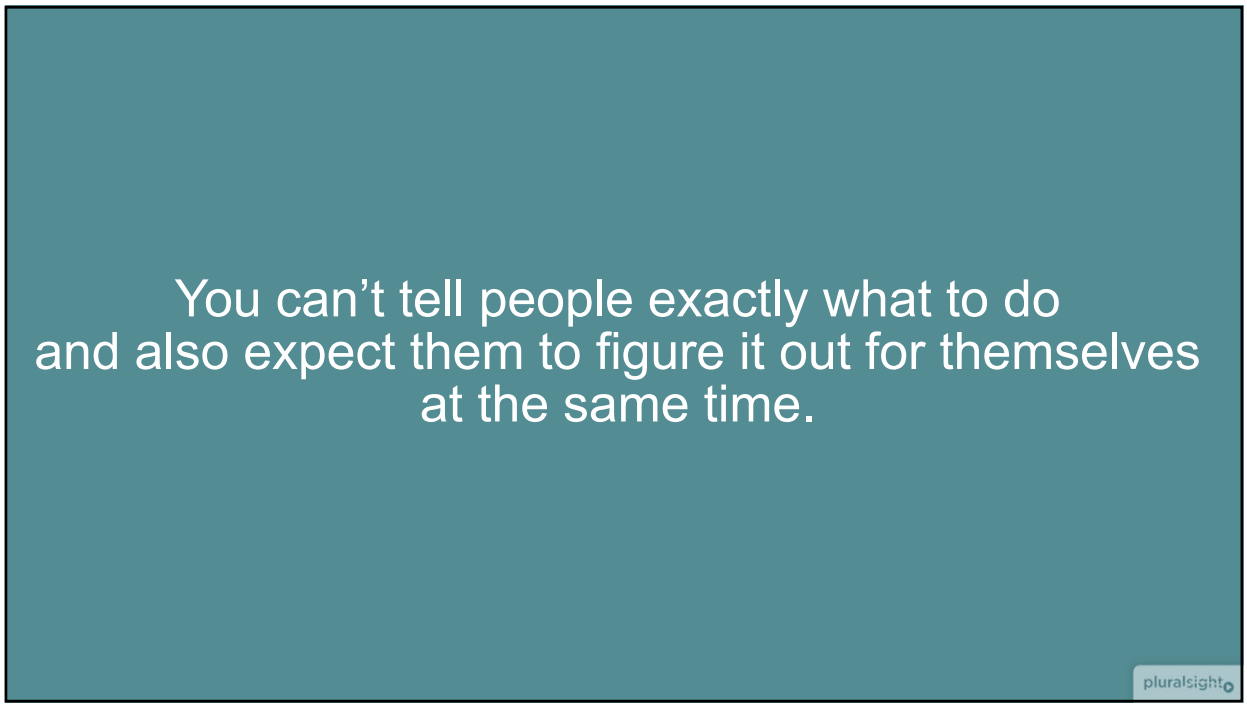
The basic idea:
Give the team their goal and then get out of the way.

pluralsight

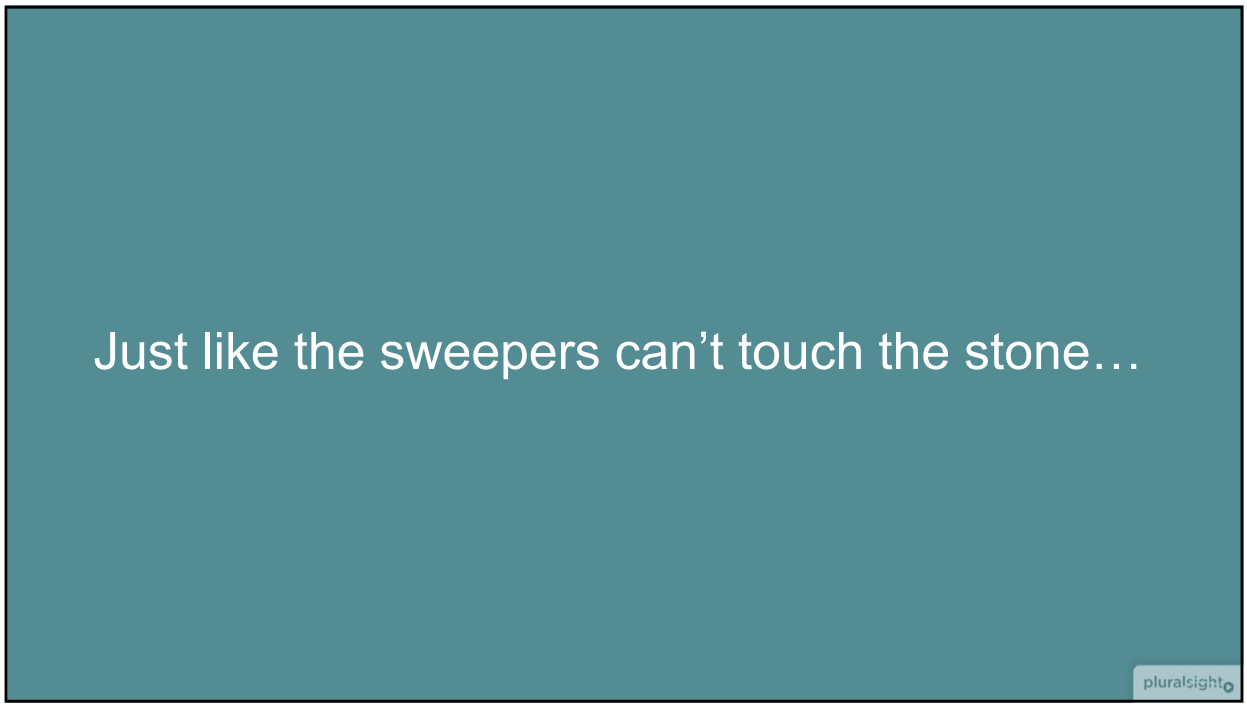

The Scrum Master helps the team to be creative and productive.
pluralsight

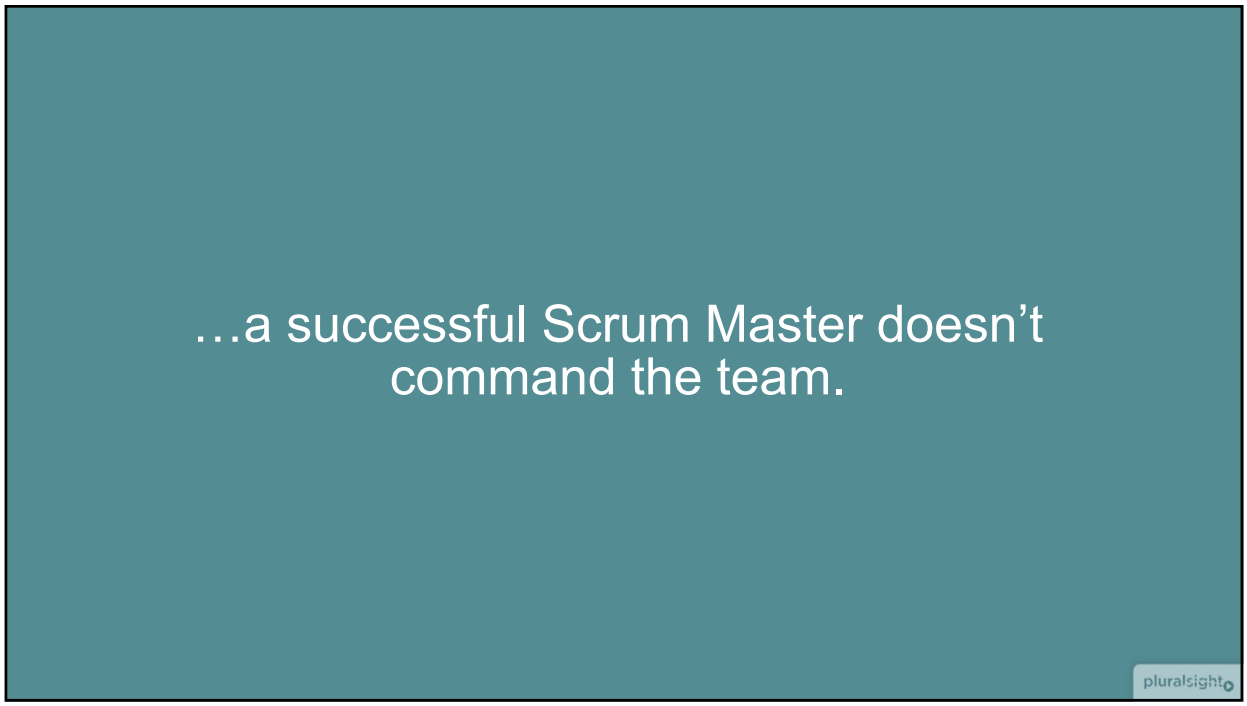

You can’t command someone to be creative.
pluralsight

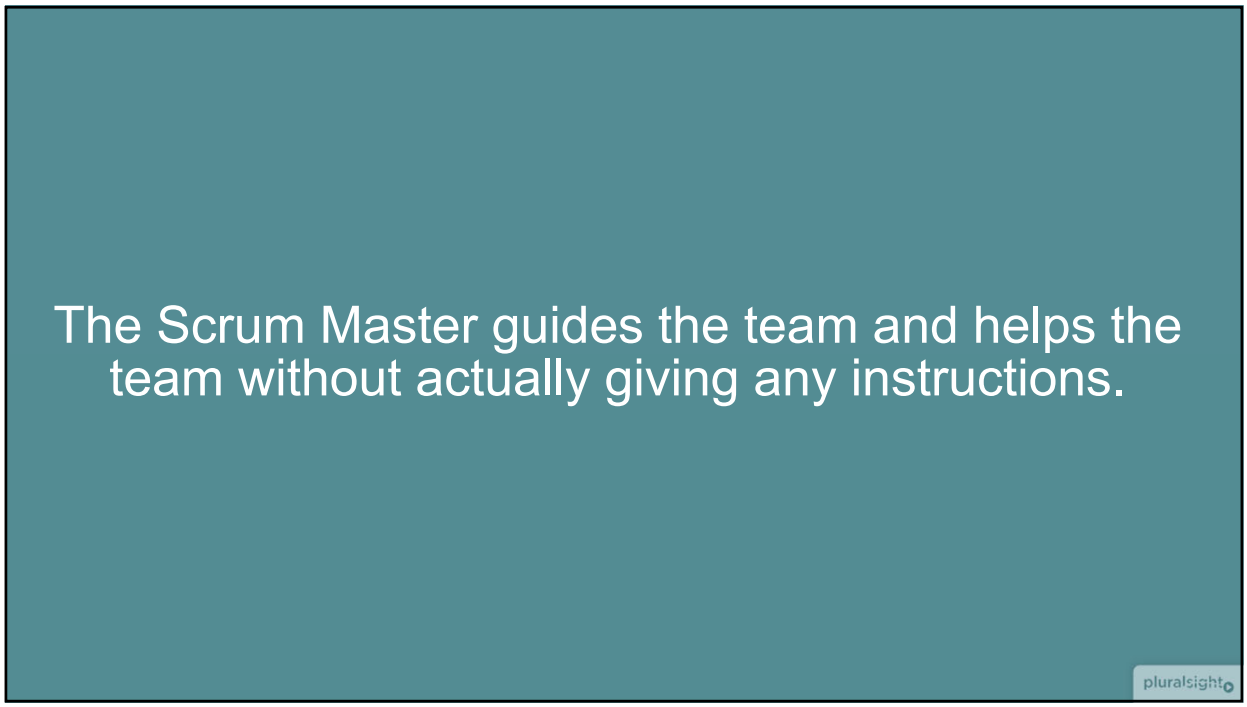

"1...2...3...BE CREATIVE!!!!"
pluralsight


You can't tell people exactly what to do
and also expect them to figure it out for themselves
at the same time.
pluralsight


And now back to curling…
pluralsight


Just like the sweepers can’t touch the stone…
pluralsight


…a successful Scrum Master doesn't command the team.
pluralsight


The Scrum Master guides the team and helps the team without actually giving any instructions.
pluralsight

- The team is the stone
- The Product Owner is the thrower
- The Scrum Master is a sweeper, guiding the team towards the goal

"What does the Scrum Master *do* anyway?"

The Scrum Master helps the team to
be productive and to use their creativity
in order to deliver done, working software.

pluralsight

The Big Lesson:
The Scrum Master leads through
influence not by command.

pluralsight

What does the Scrum Master do?

- Keeper of the process
- Help the team deliver
- Argue for and defend the team
- Help the Product Owner

BDC

Top 10 Ways: Things to Think About

- What can you do to help your team?
- What can you do to help your product owner?
- What can you do to help yourself?
- What can you do to help your organization?
- Change your Daily Scrum

BDC

Here are all the things I wanted to put in this talk.

- What can you do to help your team?
  - Written DoD
  - Encourage self-organization
  - Try to help minimize the work in progress.
  - Remind them to keep talking. One major downside of TFS is that people can sometimes think of it as an excuse to not talk.
  - Remind them about how much and how creatively you can decompose a PBI
  - Emergent Architecture.
  - Remind them they don't report to you
- What can you do to help your product owner?
  - Product owner checklist http://productowner.checklist.org by Lare Lekman
  - Remind them that they need to stay engaged and that their engagement is critical for the success of the team and the effort.
  - Remind them to plan ahead a bit and share their vision
  - Help with credibility by planning 2 to 3 sprints out
- What can you do to help yourself?
  - You can't fix everything.
  - Watch for burnout.
  - Do whatever it takes to get you to believe in self-organization
- What can you do to help your organization?
  - Get them cozy with the fact that change will be coming.
  - Changing of plans is a "feature" not a "bug."
  - Inoculate them so that they won't panic when change inevitably happens.
  - Question how products align with teams. Bring work to the teams rather than bringing Teams to the work. Let the Teams stay together and focused on a limited amount of stuff and they'll go faster.
  - Help them think about sprint lengths. Sprints help scope risk. Sprints force you to pause from time to time to see where you actually are. It helps keep you honest. Have you *actually* delivered anything? Why and why not?
  - Practice explaining Story Points.
  - Explain that Scrum doesn't replace the existing org structure. It is a layer on top of the existing org structure. While it probably won't change anyone's job, it might change how they perform their job.
  - Think like a data scientist. http://blogs.hbr.org/2013/11/how-to-start-thinking-like-a-data-scientist/
  - Watch out for fear. Fear is *everywhere*.
- Address the QA problems.
  - What does QA do at the front side of the sprint?
    - Answer collaborate with fellow team members to ensure that everyone knows how stuff is going to be tested.
    - Basically, front-load your quality focus.
- Change your daily standup
  - Consider dumping "3 questions"
  - Consider switching who "runs" the standup
  - Watch for people being bored and/or tuning out
    - Bored people implies doing too much work and/or a wobbly sprint goal
    - Bored people implies multiple teams of 1 rather than a team of X. You have people who sit near each other rather than actual functional team.
- What is your Velocity?
  - If you don't have estimates, you can't have a velocity.
  - If you don't have a DoD, velocity is problematic.
  - If you're wobbly on DoD, your velocity is suspect.
  - Be firm on DoD. No partial credit.
- Better Retrospectives
  - Write them down. Review the results.
  - Try to walk through the sprint day by day and ask people how they felt, what was going on on this day, why is the burndown going up or down? This helps them to *remember* what was happening.
- Common Objections to Scrum
- Leading vs Coaching
- Coaching organizations
- Coaching teams
- Common objections

BDC

Top 10 Ways

1. What is your DoD?
2. People Skills
3. Credibility
4. What is your Sprint Goal?
5. Do less.
6. Emergent Architecture
7. Good programming & testing practices
8. "Kick it over the wall to QA"
9. Teams : Products
10. Retrospective

BDC

Top 10 Ways

1. ***What is your DoD?***
2. People Skills
3. Credibility
4. What is your Sprint Goal?
5. Do less.
6. Emergent Architecture
7. Good programming & testing practices
8. "Kick it over the wall to QA"
9. Teams : Products
10. Retrospective

BDC

Definition of Done (DoD) = *Everything* it takes to say something is completely done.

BDC

What is your DoD?

- Closest thing to a “silver bullet” in Scrum
- Technical Debt will ruin you.
- Write it down.
- Review and discuss it regularly.

BDC

“Done vs. Done Done”

BDC

Sample Definition of Done (DoD)

Development / Coder

- Code is written with unit tests
- Unit tests have a minimum of 75% code coverage
- Code has been merged to Main
- Code compiles and unit tests pass when run as part of an automated build
- Database schema objects are under source control
- Database upgrade script is under control
- Code reviewed by someone other than the original author

Testing, Deployment, Ops

- Written QA test plan
- Tested with QA test plan by someone other than the original author
- Deployed and tested in Staging environment
- Automated UI tests are written and pass
- No Severity 1 or 2 bugs
- Reviewed by Product Owner
- Passes acceptance criteria for the PBI
- Known deployment & rollback plan
- Deployment plan reviewed by Ops
- Database changes reviewed by DBAs
- Load tested
- Deployed to Production

Do not relax your DoD.

No partial credit…ever.

BDC

Partial credit usually means
Technical Debt.

BDC

Partial Credit & Technical Debt →
"Wobbly" Velocity

BDC

Top 10 Ways

1. What is your DoD?
2. ***People Skills***
3. Credibility
4. What is your Sprint Goal?
5. Do less.
6. Emergent Architecture
7. Good programming & testing practices
8. "Kick it over the wall to QA"
9. Teams : Products
10. Retrospective

BDC

99% of the time,
it's a people problem.

BDC

Watch for interpersonal problems.

BDC

Here's a trick:
"Trust your gut."

BDC

Consider reading this book.

BDC

Want to be a super hero?
Go see a therapist.
(I'm not kidding.)

BDC

Top 10 Ways

1. What is your DoD?
2. People Skills
3. ***Credibility***
4. What is your Sprint Goal?
5. Do less.
6. Emergent Architecture
7. Good programming & testing practices
8. "Kick it over the wall to QA"
9. Teams : Products
10. Retrospective

BDC

2 to 3 Sprints of Product Backlog

- Helps give the PO "room"
- Helps everyone to know what's going on.
- Be ready to answer where a PBI (aka. "feature") is on the backlog.
  - How many sprints out?
- Discuss your *written* DoD with the PO, executives, and stakeholders
  - Why is it in their best interest?

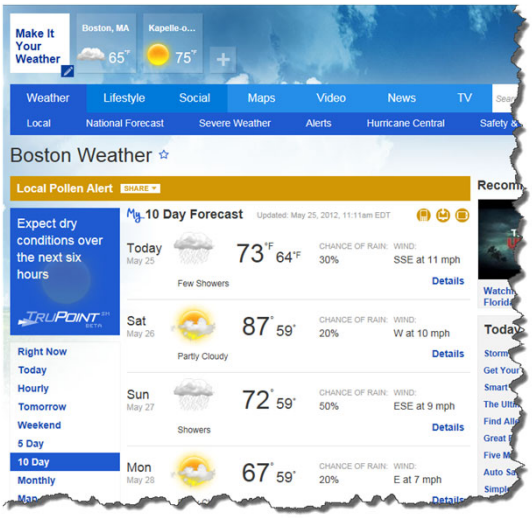
BDC

Fit for purpose.

- Remember that not everything is, needs to be, or should be a work of art.
- Balance "Time to Market" with "Long-term Maintenance"
- Communicate with the business in terms they understand
  - (Hint: this is probably money and resources.)

BDC

Train everyone to say *"forecast"* rather than *"commitment"*



BDC

Top 10 Ways

1. What is your DoD?
2. People Skills
3. Credibility
4. ***What is your Sprint Goal?***
5. Do less.
6. Emergent Architecture
7. Good programming & testing practices
8. "Kick it over the wall to QA"
9. Teams : Products
10. Retrospective

BDC

Do you have a Sprint Goal?

BDC

Is it easily understood and stated?

BDC

Hint:
Your sprint goal should not be

```
select * from SprintBacklog
```

BDC

Review your Sprint Goal
in the Daily Scrum.

BDC

And while we're talking about the Daily Scrum…

…consider dumping the "3 questions" format.

Remind your team that they don't report to you.

BDC

Consider switching up who "leads" that meeting.

BDC

Watch for people being bored
and/or tuning out.

BDC

Bored implies a lack of focus.

BDC

Bored implies that people aren't on the same team.

BDC

Top 10 Ways

1. What is your DoD?
2. People Skills
3. Credibility
4. What is your Sprint Goal?
5. ***Do less.***
6. Emergent Architecture
7. Good programming & testing practices
8. "Kick it over the wall to QA"
9. Teams : Products
10. Retrospective

BDC

Minimize work in progress.

BDC

Put another way…
don’t try to do everything at once.

BDC

If everyone on your team is working on separate PBIs,
is the team really a team?

BDC

Finish one thing. Then do the next thing.

- Try to craft the work so that multiple people are working on related things
- Complete that thing. Move on to the next thing.
- Try to drive stuff to DoD early.
- This ensures that you're delivering *something* in the sprint.

BDC

Anti-pattern:
2 days from the end of the sprint
and nothing's DoD yet.

BDC

Top 10 Ways

1. What is your DoD?
2. People Skills
3. Credibility
4. What is your Sprint Goal?
5. Do less.
6. ***Emergent Architecture***
7. Good programming & testing practices
8. "Kick it over the wall to QA"
9. Teams : Products
10. Retrospective

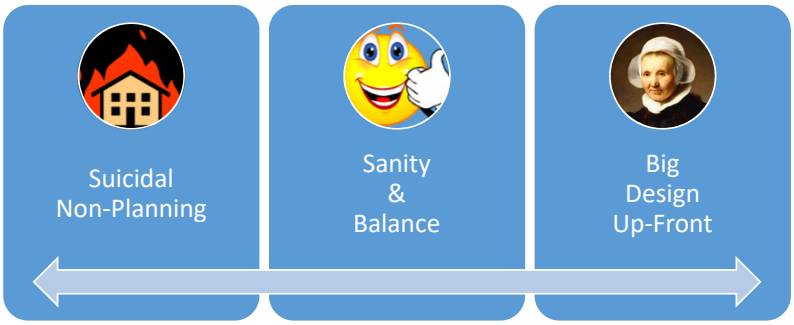
BDC

Avoid BDUF.
(Big design up-front.)

YAGNI.
(You ain't gunna need it.)

Emergent Architecture.

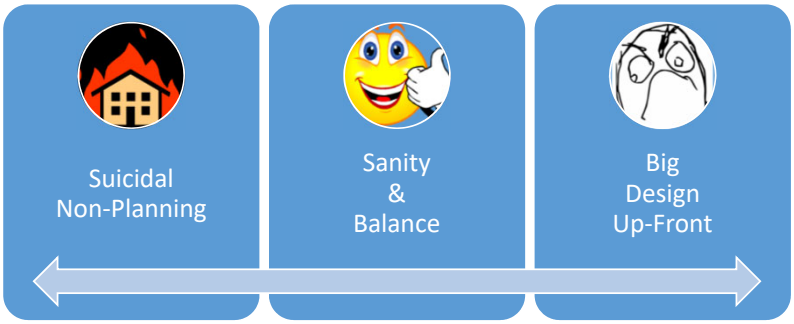
- Build what you need.
- It's a spectrum.



BDC

Emergent Architecture.

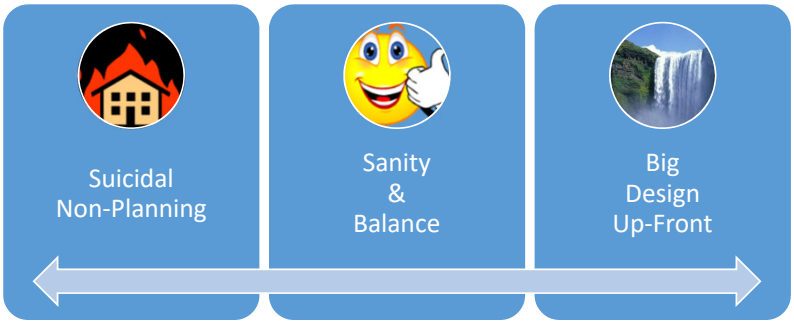
- Build what you need.
- It's a spectrum.



BDC

Emergent Architecture.

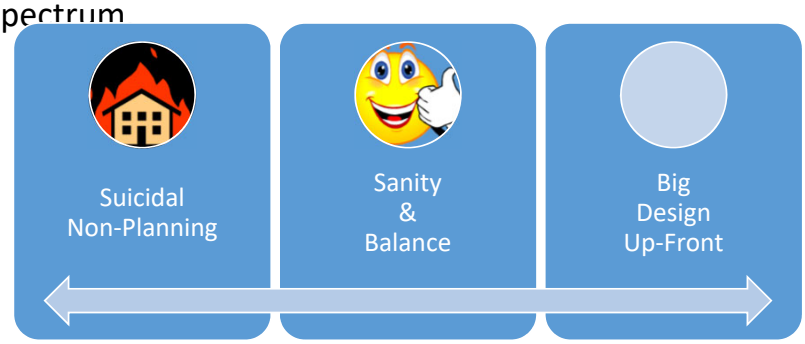
- Build what you need.
- It's a spectrum.



BDC

Emergent Architecture.

- Build what you need.
- It's a spectrum



BDC

Top 10 Ways

1. What is your DoD?
2. People Skills
3. Credibility
4. What is your Sprint Goal?
5. Do less.
6. Emergent Architecture
7. ***Good programming & testing practices***
8. "Kick it over the wall to QA"
9. Teams : Products
10. Retrospective

BDC

It's a metaphysical *certainty*
that you'll have to change stuff.

BDC

You *won't* get your "requirements" right.

BDC

Your customers
will
change their minds.

BDC

Accept that
you'll have to change.

BDC

The Goal:
Make refactoring painless.

BDC

Loose coupling.

- Code to interfaces.
- Use the Dependency Injection Pattern
  - (Pass dependencies in on the constructor.)
  - Consider an IoC Framework
- Use the Repository Pattern
- Remember Single Responsibility Principle

BDC

Build for Testability

- Unit test, unit test, unit test
- Unit test != Integration Test
- Test one layer at a time in isolation
- No database connections from a unit test
- Integration tests in a separate project
  - Keep yourself honest

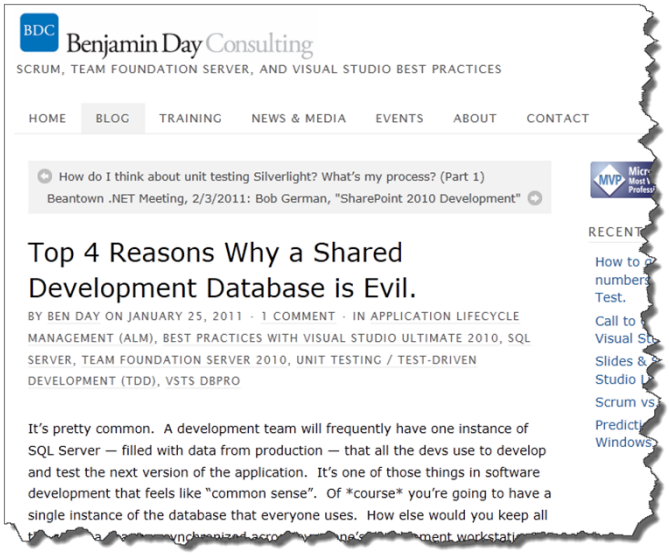
BDC

For the love of all things
precious & beautiful…

BDC

…DON'T USE A SHARED DEVELOPMENT DATABASE!!!!!!!!

BDC



http://tinyurl.com/bqextsa

Eliminate the "works on my box" problem *early*.

- Set up automated builds from the very beginning.
- If you're using TFS, use Gated Check-in builds
- Deploy your database as part of your builds.
- Run your unit tests from the builds

BDC

Top 10 Ways

1. What is your DoD?
2. People Skills
3. Credibility
4. What is your Sprint Goal?
5. Do less.
6. Emergent Architecture
7. Good programming & testing practices
8. ***"Kick it over the wall to QA"***
9. Teams : Products
10. Retrospective

BDC

Never say "Kick it over the wall to QA" again.

- Us vs. Them
- QA is part of the team
- QA's work should be part of the DoD

BDC

What does QA do at the start of a sprint?

BDC

Top 10 Ways

1. What is your DoD?
2. People Skills
3. Credibility
4. What is your Sprint Goal?
5. Do less.
6. Emergent Architecture
7. Good programming & testing practices
8. "Kick it over the wall to QA"
9. ***Teams : Products***
10. Retrospective

BDC

How do Teams align to Products / Projects?

- Watch out for partial allocations
- Do team members have more than one Daily Scrum?
- Rather than 1 team per 1 product,
  try 1 team that supports multiple products

BDC

Teams to Products

- Bad
  - Billing System has a Billing Team
  - Accounting System has an Accounting Team
  - CRM system has a CRM Team
  - Website has a Website team
  - People are 50% allocated to multiple teams.

BDC

Teams to Products

- Better
  - Team A
    - Billing
    - Accounting
  - Team B
    - CRM
    - Website

BDC

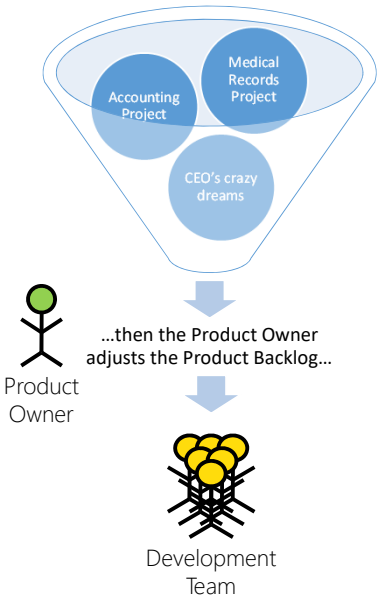
Teams to Products

- Best
  - Team A and Team B are cross-functional
  - Either team can do tasks from Billing, Accounting, CRM, or Website

BDC

The Product Owner as a work funnel.

- Product Owner has to decide priority for the team
- The team can't do everything at once
- There has to be priorities
- The team isn't qualified to balance these priorities



BDC

Entonnoir?

Top 10 Ways

1. What is your DoD?
2. People Skills
3. Credibility
4. What is your Sprint Goal?
5. Do less.
6. Emergent Architecture
7. Good programming & testing practices
8. "Kick it over the wall to QA"
9. Teams : Products
10. ***Retrospective***

BDC

Retrospectives are the curer of all ills.

BDC

Retrospectives help keep problems contained to (hopefully) a single Sprint.

BDC

Tip:
Watch for unscheduled work.

Tip:
Get people REALLY thinking by trying to recreate what happened day by day.

Tip:
If it gets heated, learn to say
“Ok…and what else?”

Tip:
Record the findings from the Retrospective

Tip:
Review the notes from
previous Retrospectives.

BDC

Top 10 Ways

1. What is your DoD?
2. People Skills
3. Credibility
4. What is your Sprint Goal?
5. Do less.
6. Emergent Architecture
7. Good programming & testing practices
8. "Kick it over the wall to QA"
9. Teams : Products
10. Retrospective

BDC

What can you do to help yourself?

BDC

Watch for burnout.

BDC

You can’t fix everything.

Do whatever it takes to get yourself
to believe in self-organization.

But don’t let your team walk all over you.

Some teams will use
self-organization against you.

"…but let's remember here.
This isn't a *&^%$# democracy."
-David Starr

How do you help the
Product Owner?

Remind them of how important they are to the success of the project.

Remind them that they need to stay engaged.

LARE lekman

Ketterän kehityksen kouluttaja ja valmentaja

Product Owner Checklist – November 2013

"This brief checklist helps you remember the most important things to become a good Product Owner.

As the Product Owner you are responsible for maximizing the value of the product and the work of the Development Team. Your way of doing this may vary across organizations, Scrum Teams, and individuals. This checklist is therefore only an example.

http://productownerchecklist.org

BDC

An Example Checklist for Product Owners

Lare Lekman | Revised in November 2013 | productownerchecklist.org

Product Vision

- ☐ I have a product vision (created with customers, end users, and investors, when possible)
- ☐ I can answer to questions about the product vision and business model in a concise and m
- ☐ I have a short tagline for the product vision, for example "1,000 songs in your pocket" (
  to communicate the essence and value of the product release.

Stakeholders

- ☐ I understand the needs of my stakeholders (for example customers, end users, and inve
- ☐ I communicate regularly with my stakeholders to understand their needs and to manage
- ☐ I can answer to questions about how each product backlog item will generate value for th
- ☐ I am motivated to work as a Product Owner, and make sure I have the stakeholders' man
- ☐ My forecasts to stakeholders are based on development team's measured velocity or th

Product Backlog

- ☐ I have a product backlog
- ☐ I have a mandate to make decisions about the product backlog
- ☐ I update the product backlog at least before each sprint planning meeting
- ☐ The product backlog items are ordered (based on value, risk, work estimates, dependenci
- ☐ The product backlog items are clearly expressed and more detailed towards the top
- ☐ The product backlog is accessible to all scrum team members

BDC

Any last questions?

BDC

Thanks.

benday@benday.com | www.benday.com

BDC Benjamin Day Consulting

BDC