AGILE FOR ALL

Making Agile a Reality.®

Certified ScrumMaster Training

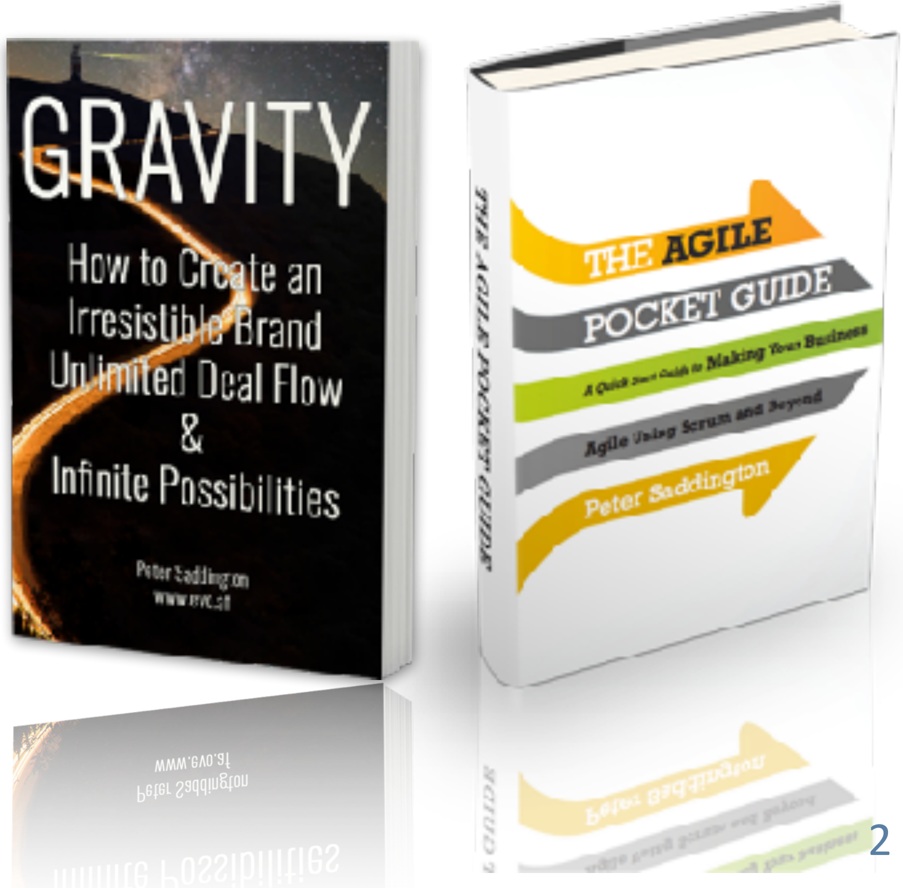
ScrumAlliance®
transforming the world of work

CSM
ScrumAlliance
CERTIFIED ScrumMaster

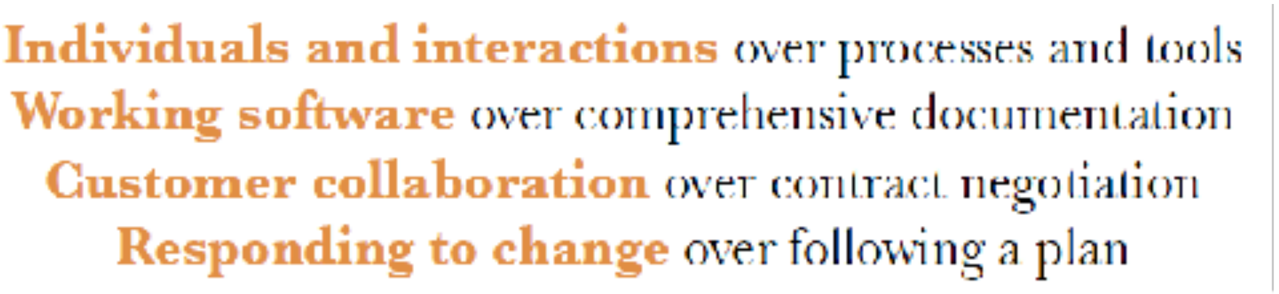
Peter Saddington MDiv, CST

- 3X Startup Founder, Author, Agile Trainer, and Brand Consultant
- 20+ years of software development experience
- Social Scientist - M.A. Counseling, M.A. Education, M.Div. Theology
- Peter’s Community - vchunting.com
- *The Agile Pocket Guide - A Quick Start to Making Your Business Agile* (2012)
- *Gravity - How to Create an Irresistible brand, Unlimited Deal Flow and Infinite Opportunities* (2020)

Email: *peter.saddington@agileforall.com*
Web: *agileforall.com*
Twitter: *@AgilePeter*
Book: www.vchunting.com


GRAVITY
How to Create an Irresistible Brand Unlimited Deal Flow & Infinite Possibilities
Peter Saddington
www.vc.st
THE AGILE POCKET GUIDE
THE AGILE POCKET GUIDE
A Quick Start Guide to Making Your Business
Agile Using Scrum and Beyond
Peter Saddington

The Agile Manifesto Principles



Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

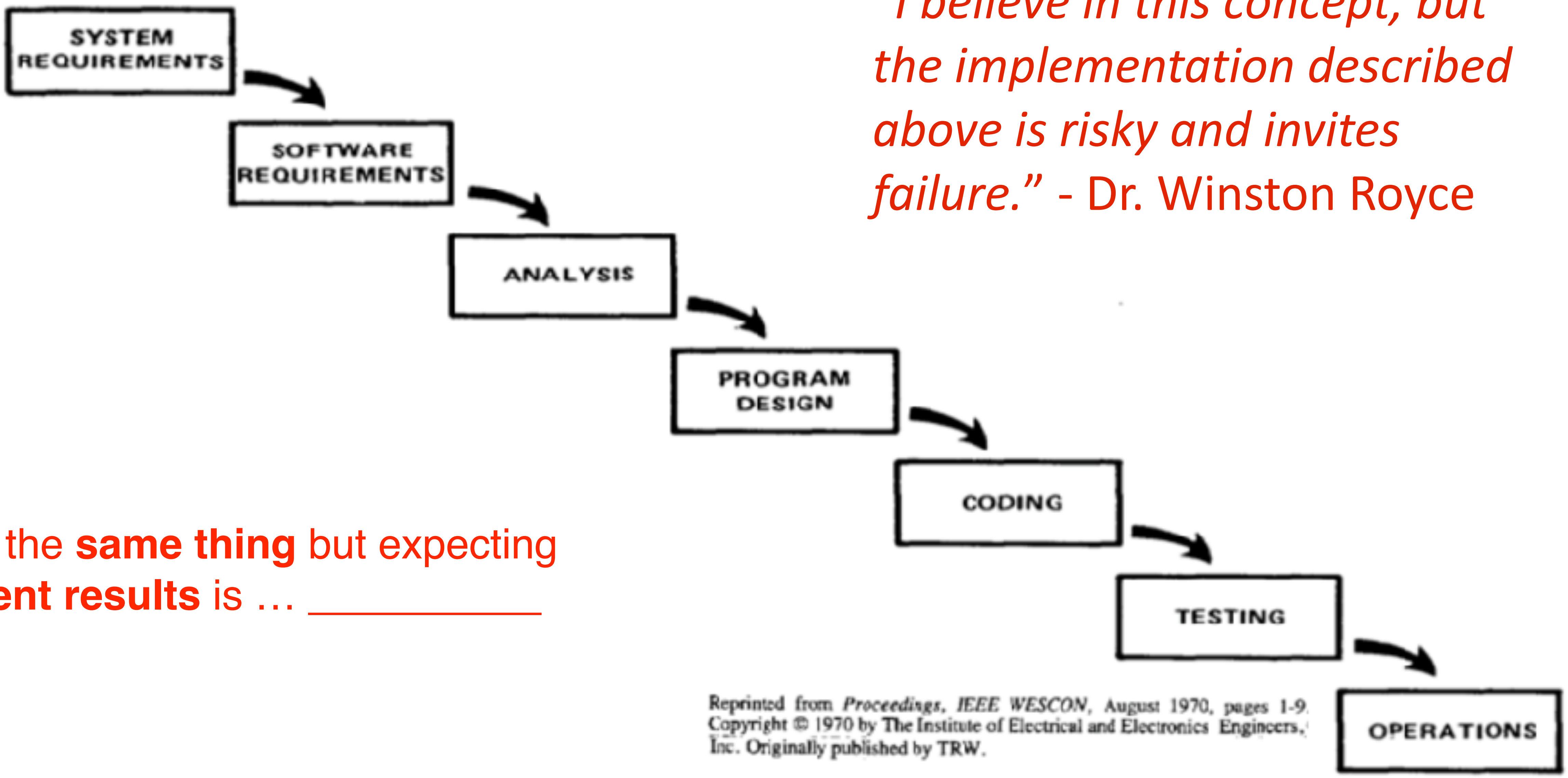
1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

AGILE FOR ALL
Making Agile a Reality®

www.agilemanifesto.org

Plan Driven Development - Not a Proposal

"I believe in this concept, but the implementation described above is risky and invites failure." - Dr. Winston Royce

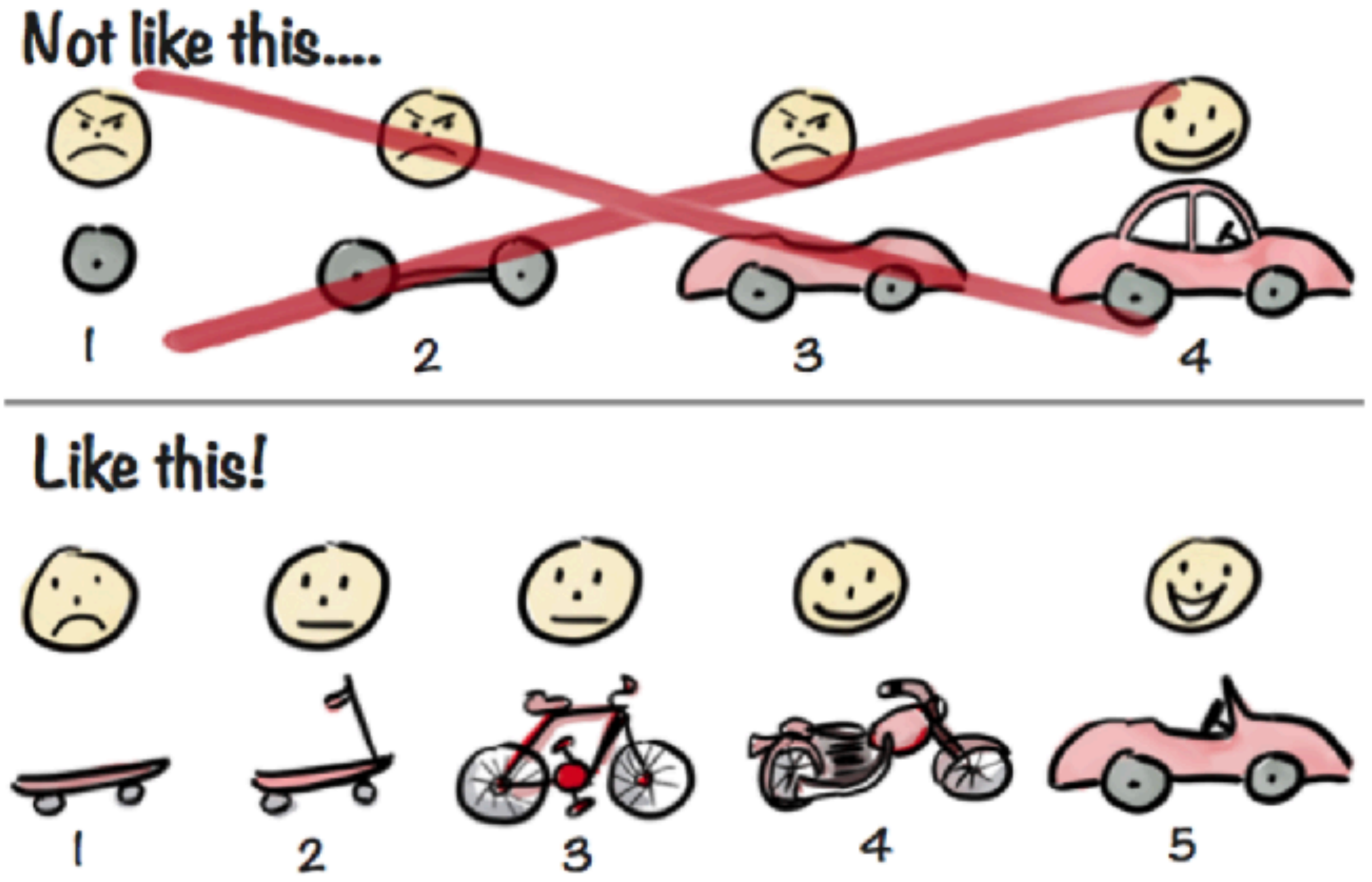


Reprinted from *Proceedings, IEEE WESCON*, August 1970, pages 1-9. Copyright © 1970 by The Institute of Electrical and Electronics Engineers, Inc. Originally published by TRW.

Figure 2. Implementation steps to develop a large computer program for delivery to a customer.

Doing the **same thing** but expecting **different results** is … ___________

Iterative, Incremental Methods



Only Incremental

Iterative & Incremental

This Photo by Unknown Author is licensed under CC BY-SA

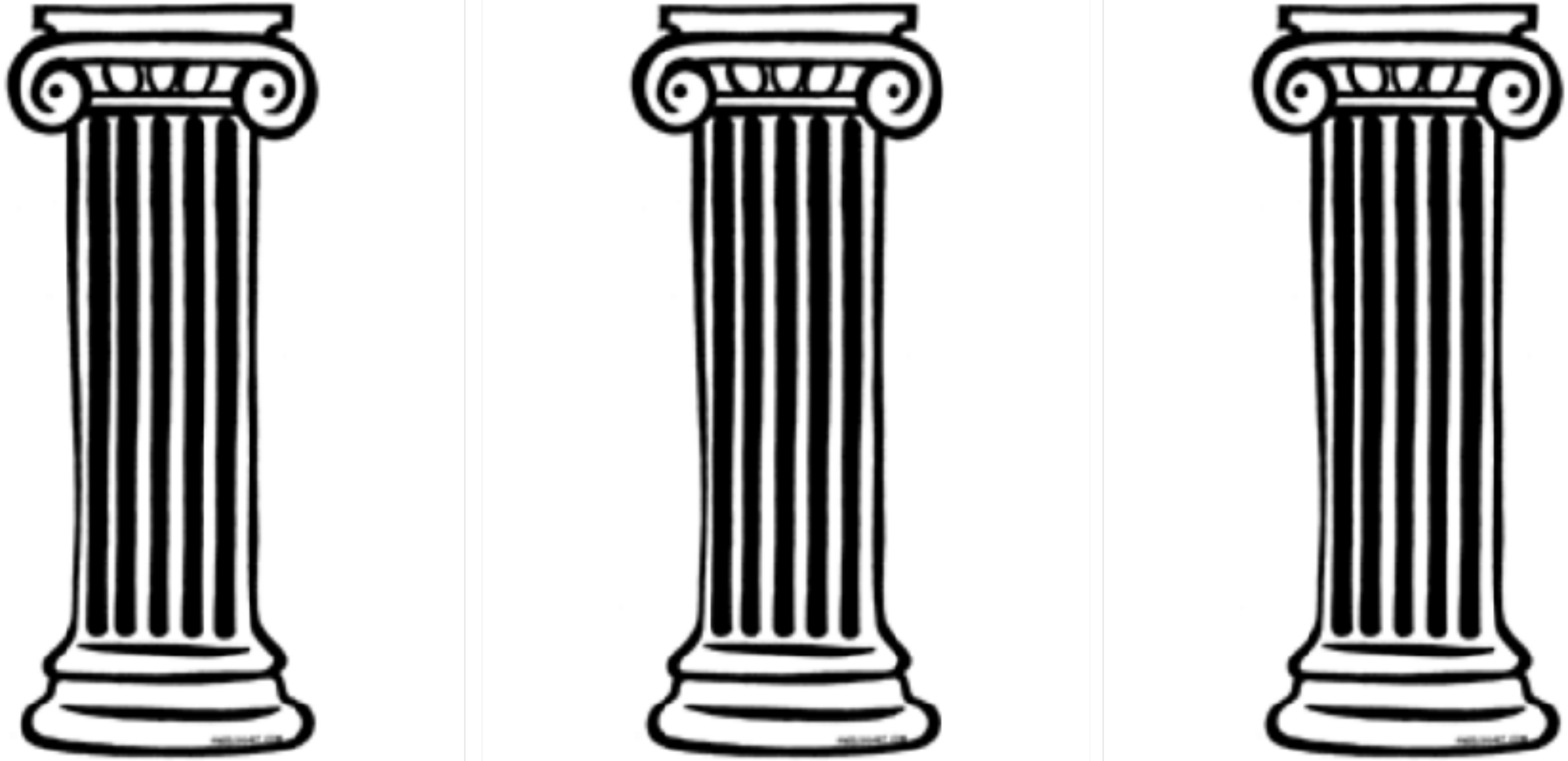
Incremental means building something piece by piece

Iterative means building something through successive refinements, a whole MVP that will do the job, then refining it in ways that enhances the end-to-end product. Helps with **IKIWISI**!

Sprints are timeboxes in which we do both incremental and iterative development.

The Scrum Empirical Process

____________ _____________ _____________

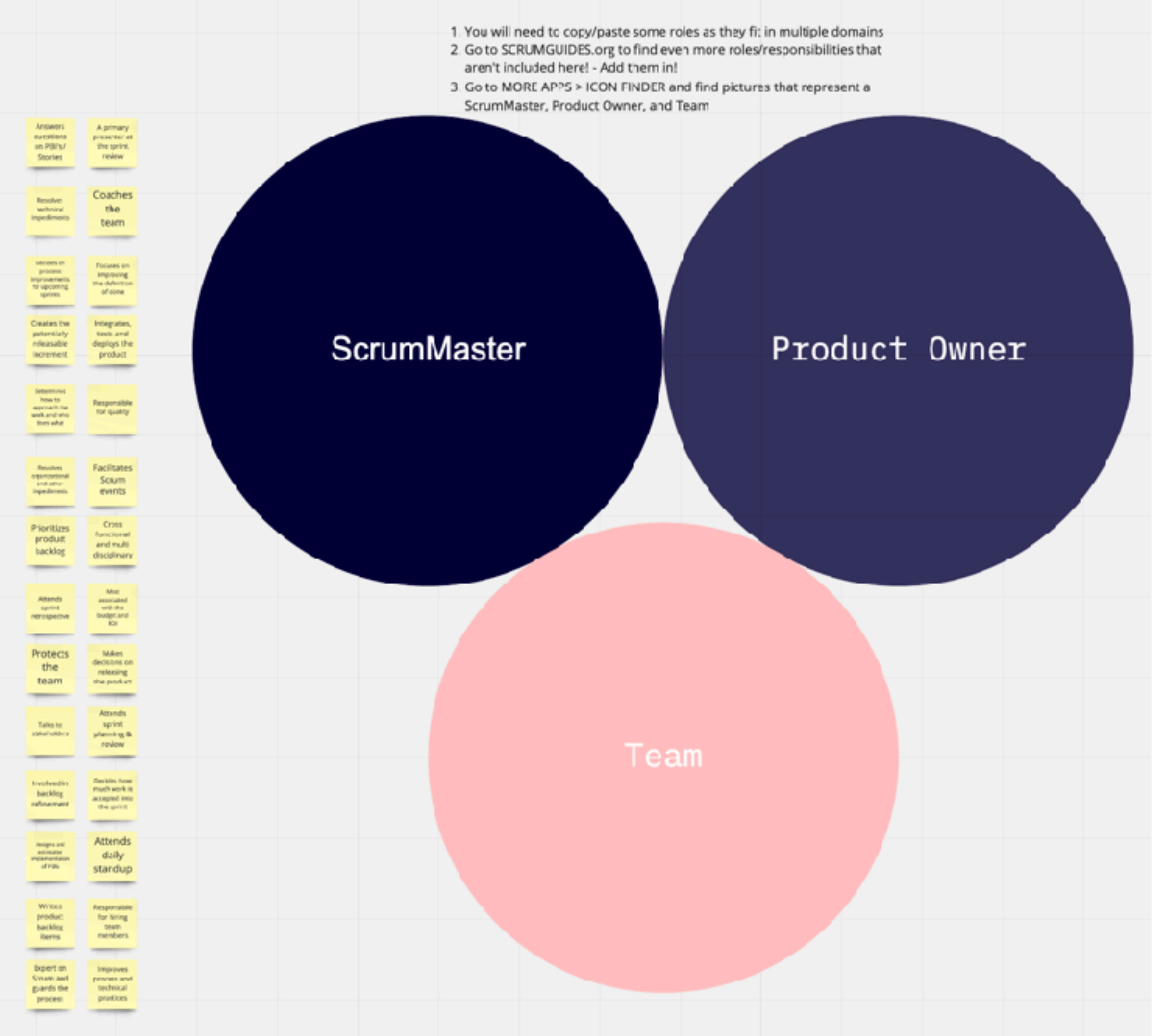


Foundation:__________________

AGILE FOR ALL
Making Agile a Reality.®

BUILD YOUR OWN SCRUM NOTES

3 SCRUM ACTORS NOTES



1. You will need to copy/paste some roles as they fit in multiple domains
2. Go to SCRUMGUIDES.org to find even more roles/responsibilities that aren't included here! - Add them in!
3. Go to MORE APPS > ICON FINDER and find pictures that represent a ScrumMaster, Product Owner, and Team

Answers questions on PBI's/ Stories

A primary presenter at the sprint review

Resolves technical impediments

Coaches the team

Decides on process improvements to upcoming sprints

Focuses on improving the definition of done

Creates the potentially releasable increment

Integrates, tests and deploys the product

Determines how to approach the work and who does what

Responsible for quality

Resolves organizational and other impediments

Facilitates Scrum events

Prioritizes produt backlog

Cross functional and multi disciplinary

Attends sprint retrospective

Most associated with the budget and ROI

Protects the team

Makes decisions on releasing the product

Talks to stakeholders

Attends sprint planning & review

Involved in backlog refinement

Decides how much work is accepted into the sprint

Designs and estimates implementation of PBIs

Attends daily standup

Writes product backlog items

Responsible for hiring team members

Expert on Scrum and guards the process

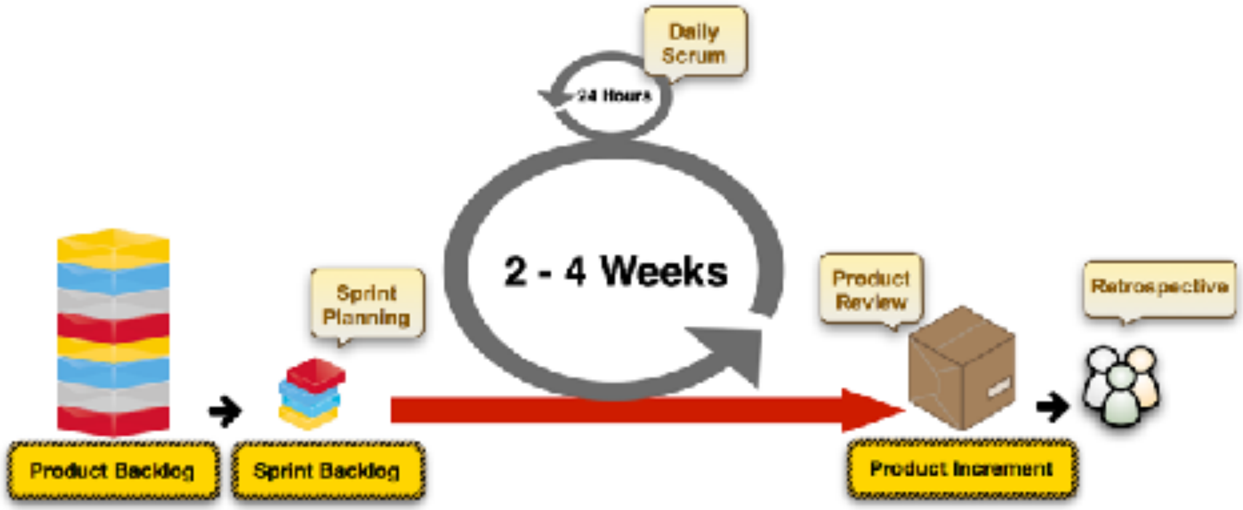
Improves process and technical practices

Sprint Planning Meeting

WHAT (PO) + HOW (TEAM)

What do we want? (PO)

- **“What items are candidates for the Sprint?”**
- Backlog items and acceptance criteria are finalized



How will we do it? (Team)

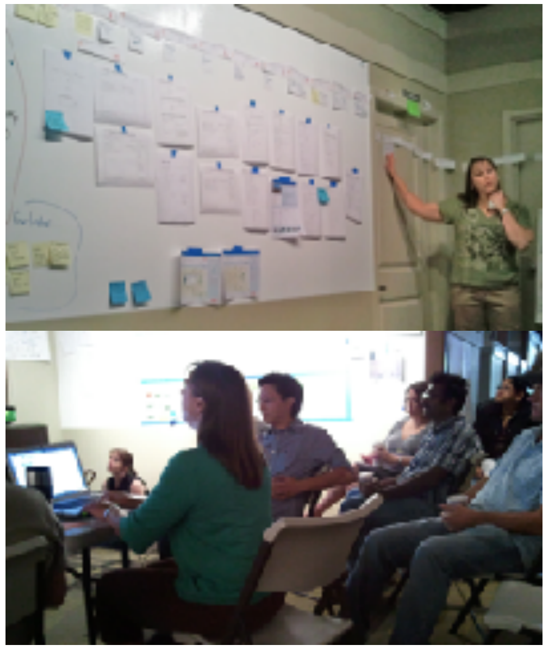
- Stories broken down into **tasks**
- Stories are discussed and (re)estimated
- **PLAN TOGETHER**
- Definition of Done is reviewed
- Timeboxed (~1hr/week)

Daily Scrum

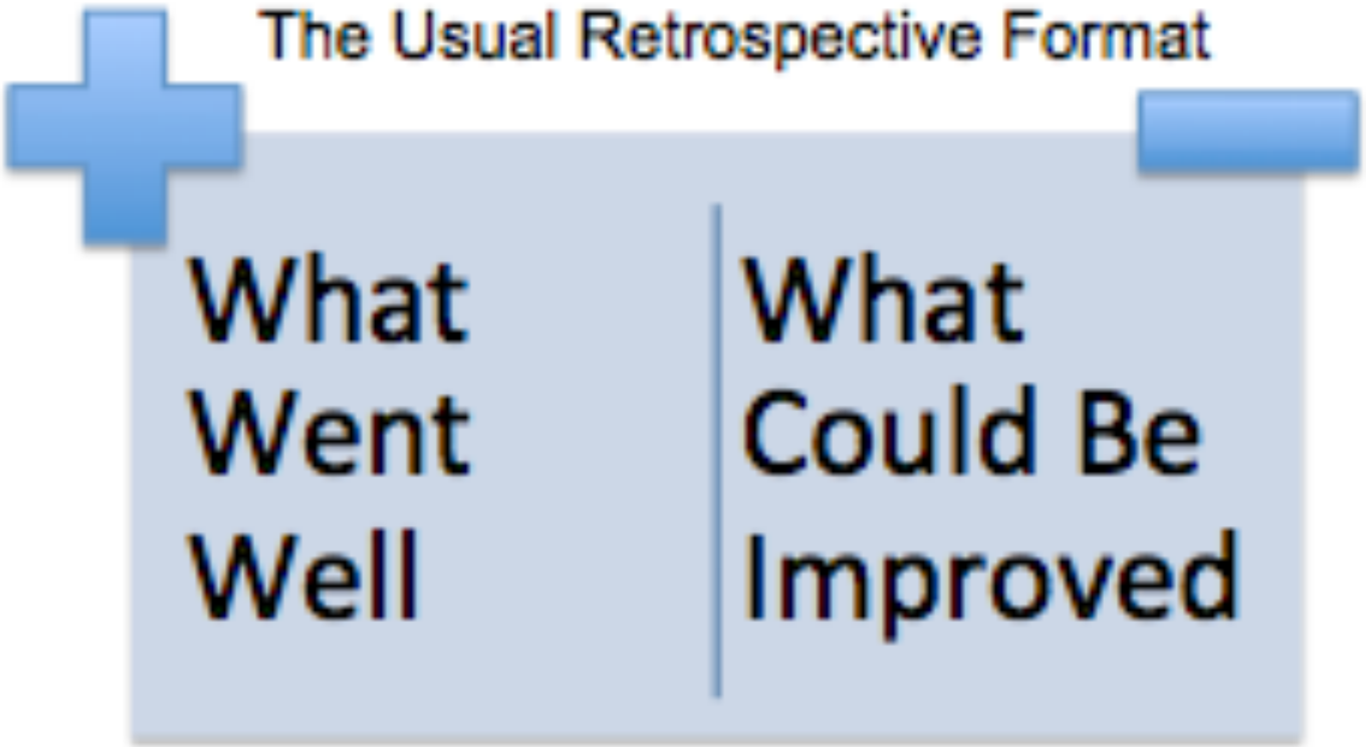
- 15 minute meeting every day
- Same time, same place
- Three questions
  - What did I do since we last met?
  - What do I plan to do by our next meeting?
  - What impediments do I have?
- Focus is on Sprint commitments
- Not a status update! - **ALIGNMENT Meeting**

Product Review

- Demonstrate functionality
- Show “Done” work
- **Get sign-off from PO on A/C and DoD**
- **Gather feedback from stakeholders**
- ~1 hr/week
- Celebrate!



Retrospective



The Usual Retrospective Format

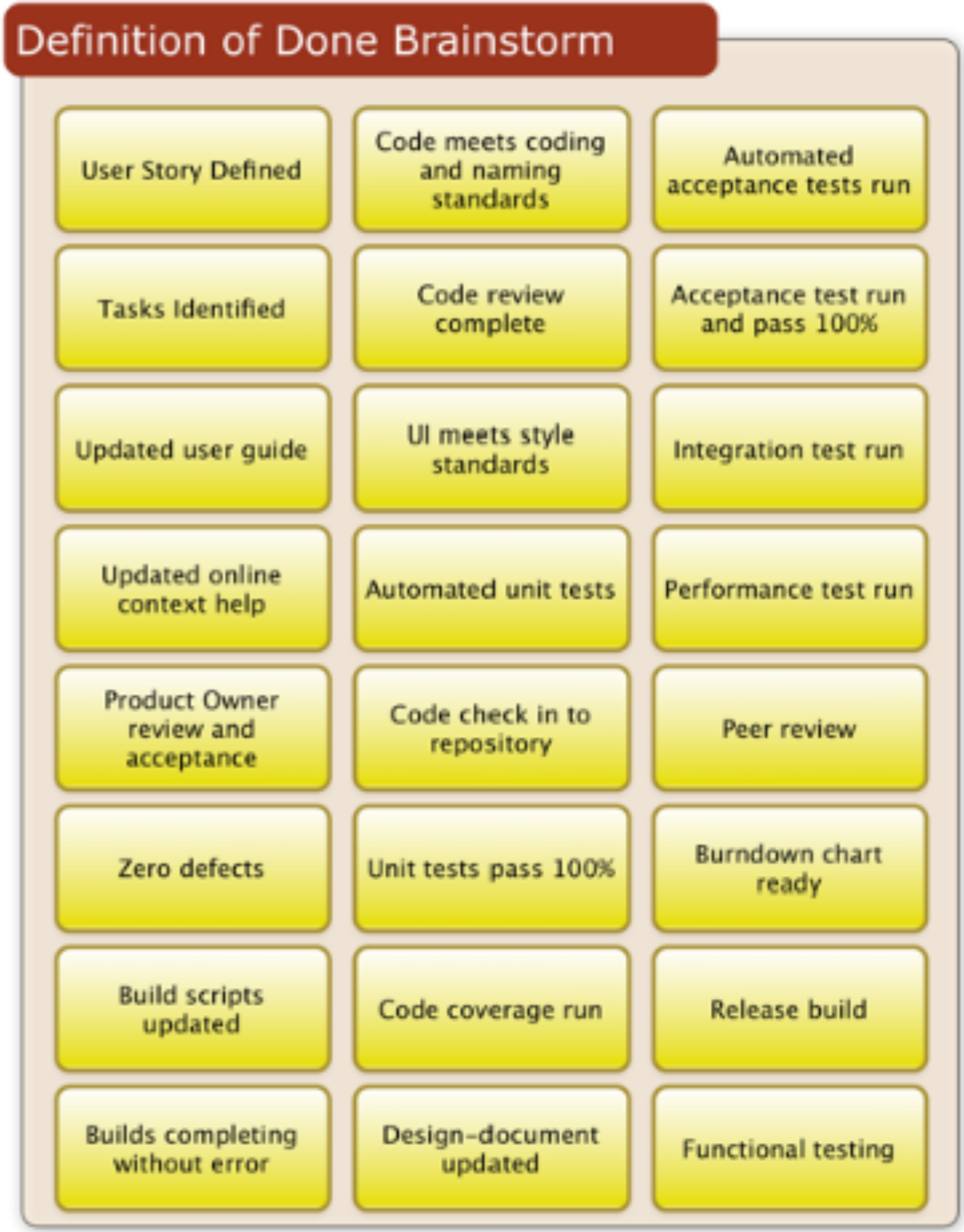
What Went Well

What Could Be Improved

ACTION ITEMS TO DO DURING NEXT SPRINT (Review these during the next retrospective)

Definition of Done

- **Defines the work products that will be delivered with each item as it is ready for acceptance**
- **"Quality checklist"**
  - *These must pass before we deploy our product to our customers*
  - *"Potentially releasable"*
  - *If multiple teams are working on one product, one DOD is shared across teams*
  - *Can include enterprise standards*
- Sprint cancelation?



Definition of Done Brainstorm

User Story Defined	Code meets coding and naming standards	Automated acceptance tests run
Tasks Identified	Code review complete	Acceptance test run and pass 100%
Updated user guide	UI meets style standards	Integration test run
Updated online context help	Automated unit tests	Performance test run
Product Owner review and acceptance	Code check in to repository	Peer review
Zero defects	Unit tests pass 100%	Burndown chart ready
Build scripts updated	Code coverage run	Release build
Builds completing without error	Design-document updated	Functional testing

- **ScrumMaster**
  - Serves the organization
  - Planning Scrum and implementation
  - Teach and train Scrum
  - Helping employees/stakeholders
  - Causing change that increases productivity
  - Working with other ScrumMasters
  - Supports development team
  - Removes impediments
  - Personas/Characteristics:
    - Humble, respective, empathetic, persuasive, connected.
    - Coach, protector, facilitator, agile advocate

- **Delivery Team**
  - 3-9 People
  - Cross-functional - *The team learns from each other*
  - Self-organizing
  - Structured & empowered to organize & manage own work
  - No titles
  - No sub-teams - testers, architects, ops, BA, etc.

ScrumMaster as a Coach

What is the difference between coaching, facilitating & mentoring?

- **Coaching** – a trusted advisor, trying to work themselves out of a job. An agile coach helps a team or individual adopt and improve agile methods and practice. A coach helps people rethink and change the way they go about development. May include observations and often Socratic questions. Yet it can occasionally be directive.
- **Facilitating** – a content-neutral role where the person is experienced at enabling groups to work more effectively and collaborate in an open environment
- **Mentoring** – relationship-based advice or guidance, often from someone more senior or experienced
- There's also **training** – expertise-based, teaching specific skills or knowledge, with well-defined learning objectives, relatively brief

Invoke Socrates: Ask, Don’t Tell

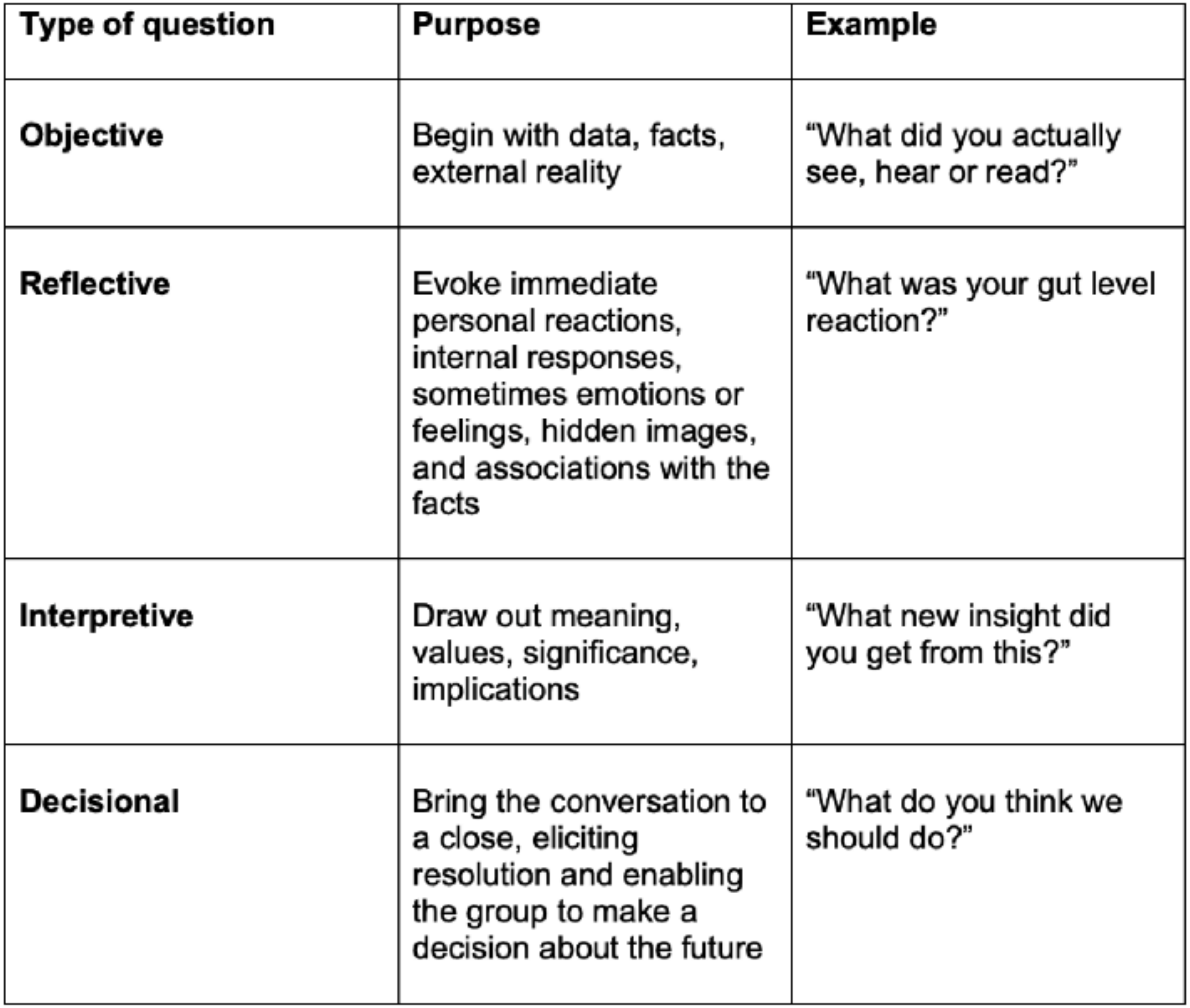
The best SMs follow the mantra “Trust the Team”

Usually, it is better to ask powerful questions:

- I noticed that <situation>; what shall we do?
- I feel <feeling>; do you share that?
- Shall we try to find out why <situation>?
- What do you think we should do?
- What have you decided?
- What should I do?
- What do you prefer?

AGILE FOR ALL
Making Agile a Reality.®

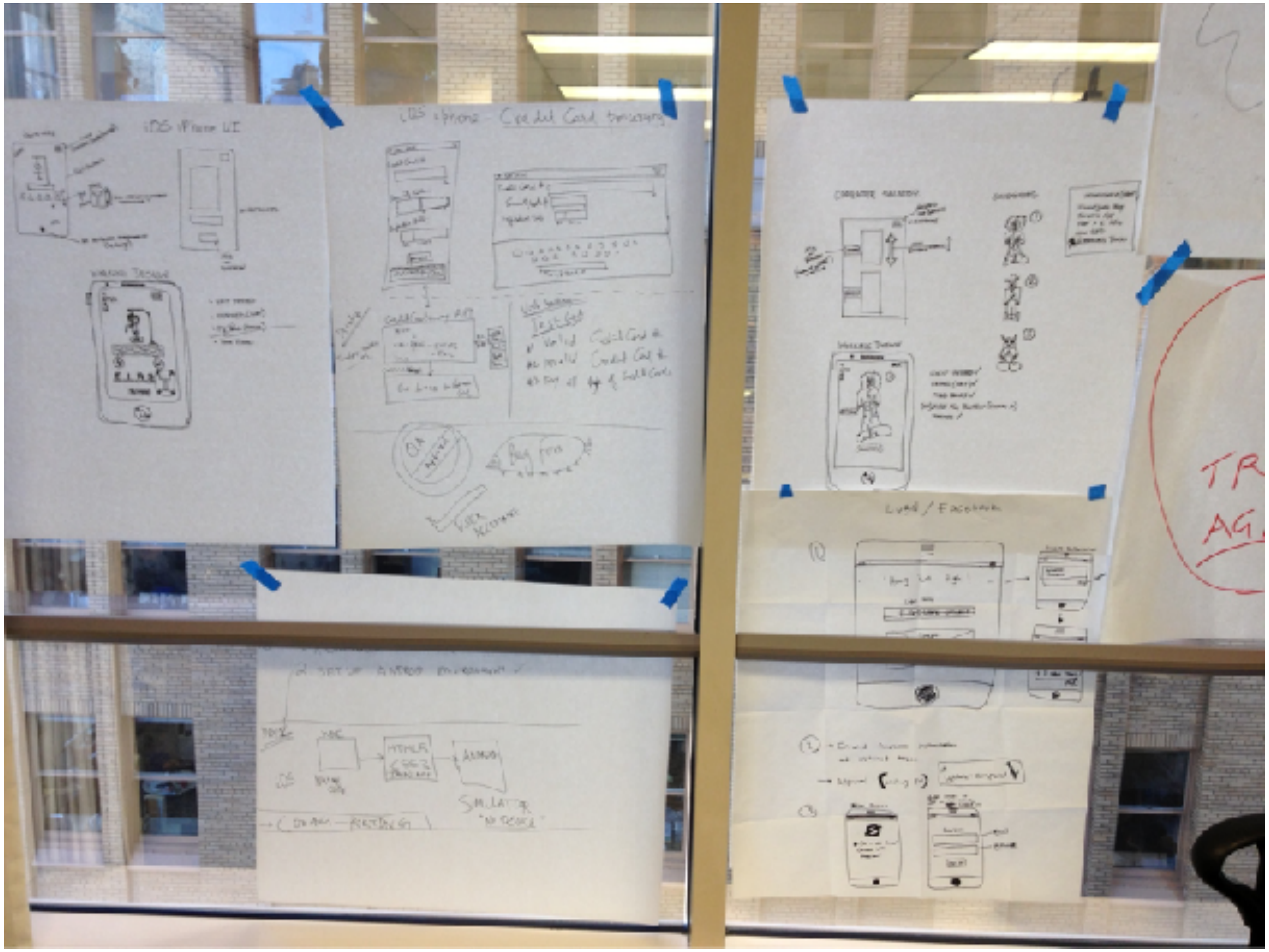
ScrumMaster Role Deeper Dive
Better Communication Using Focused Conversations



Type of question	Purpose	Example
Objective	Begin with data, facts, external reality	"What did you actually see, hear or read?"
Reflective	Evoke immediate personal reactions, internal responses, sometimes emotions or feelings, hidden images, and associations with the facts	"What was your gut level reaction?"
Interpretive	Draw out meaning, values, significance, implications	"What new insight did you get from this?"
Decisional	Bring the conversation to a close, eliciting resolution and enabling the group to make a decision about the future	"What do you think we should do?"

Scrum Simulation - Million Dollar Mobile App

- Think of your favorite mobile apps - Games, productivity, camera, location-based apps, food, social, chat, informational, utility, etc.
- Project Goal: Develop a product in 3 Sprints of 3 “days” each.



8 minutes:

- **Choose a Product idea + Product Name**
- **Decide who will be your PO and SM**

AGILE FOR ALL
Making Agile a Reality.®

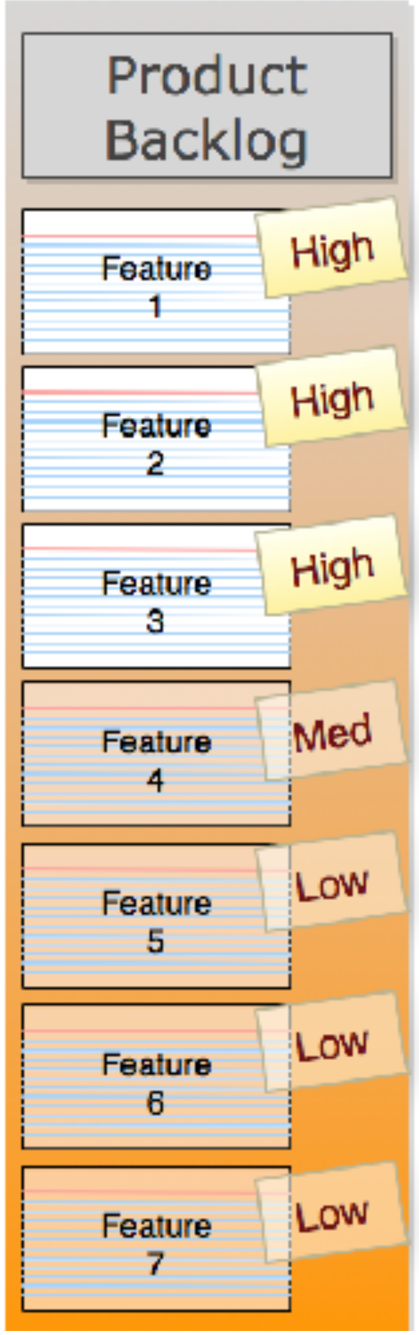
Product Owner

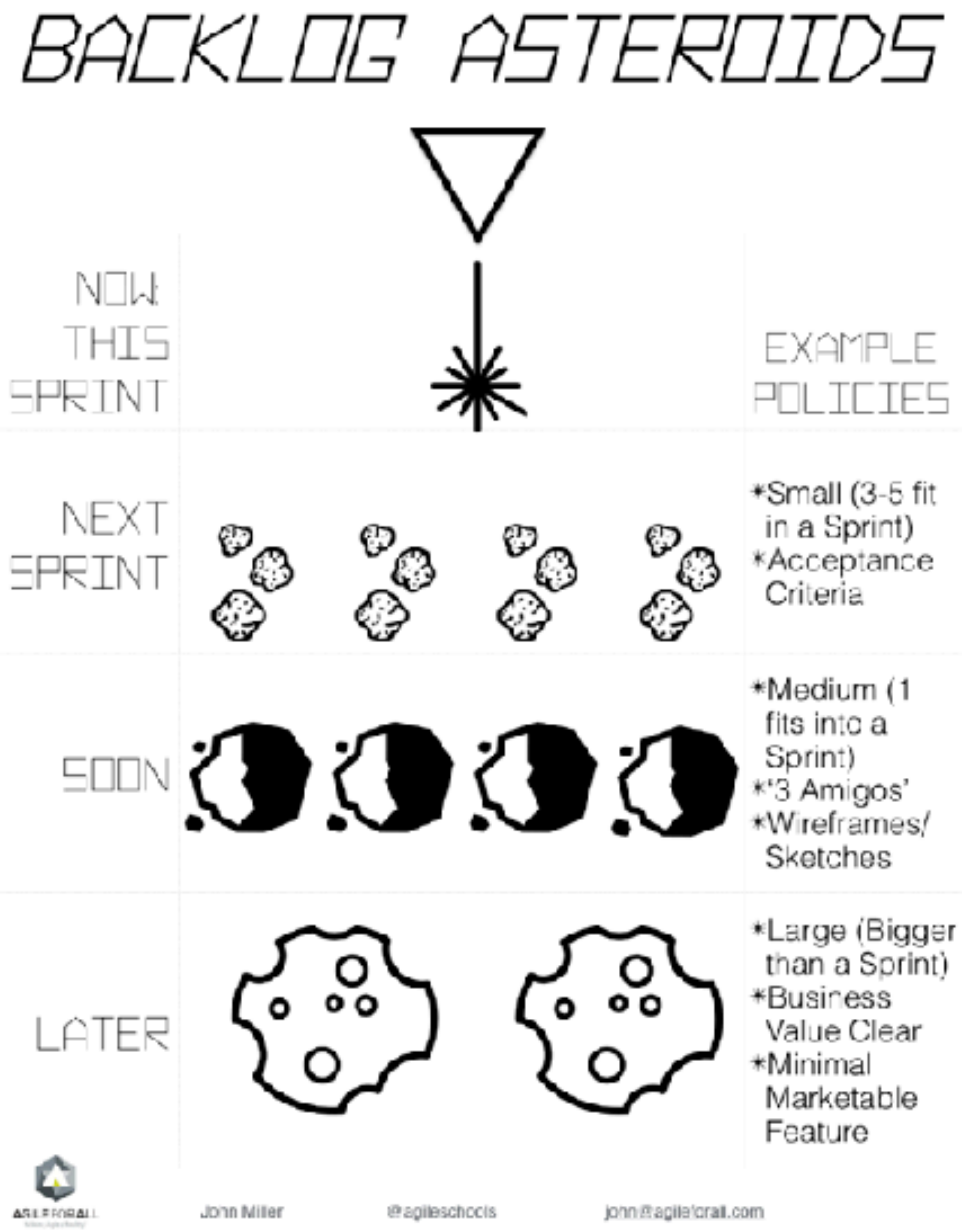
- Clearly expresses the ROI and VALUE of the PBI
- Create the Product Vision - End goals and missions
- Creates and maintains the Product Backlog in priority order
- Refines the backlog - Visible, transparent, clear
- Characteristics _______ ________ _______ ________
- Engages with _________ __________ __________

- *Single Product Owner per team?*

Product Backlog Refinement

- Items can split, removed, changed, added, re-prioritized anytime
- A Product Backlog Item (Story) is ready if it is understandable by both the **Product Owner** and the team - Work together!
- Product Backlog Items cannot be groomed appropriately if the delivery team(s) haven't helped





User Stories in the Product Backlog

AS A <type of user>

I WANT TO <goal>

SO THAT <value/why>

Acceptance Criteria:

<Verification>

Acceptance Criteria

- Acceptance Criteria help to set scope and define what "Done" means
- *First pass done by the PO before Sprint Planning, but can be **negotiated** by Team and PO!*

AS A user, I can cancel a reservation...

AC:
1. **Verify that** a premium member can cancel the same day without a fee.
2. **Verify that** an email confirmation is sent.
3. **Verify that** the hotel is notified of any cancelation

3C's + I.N.V.E.S.T. of User Stories

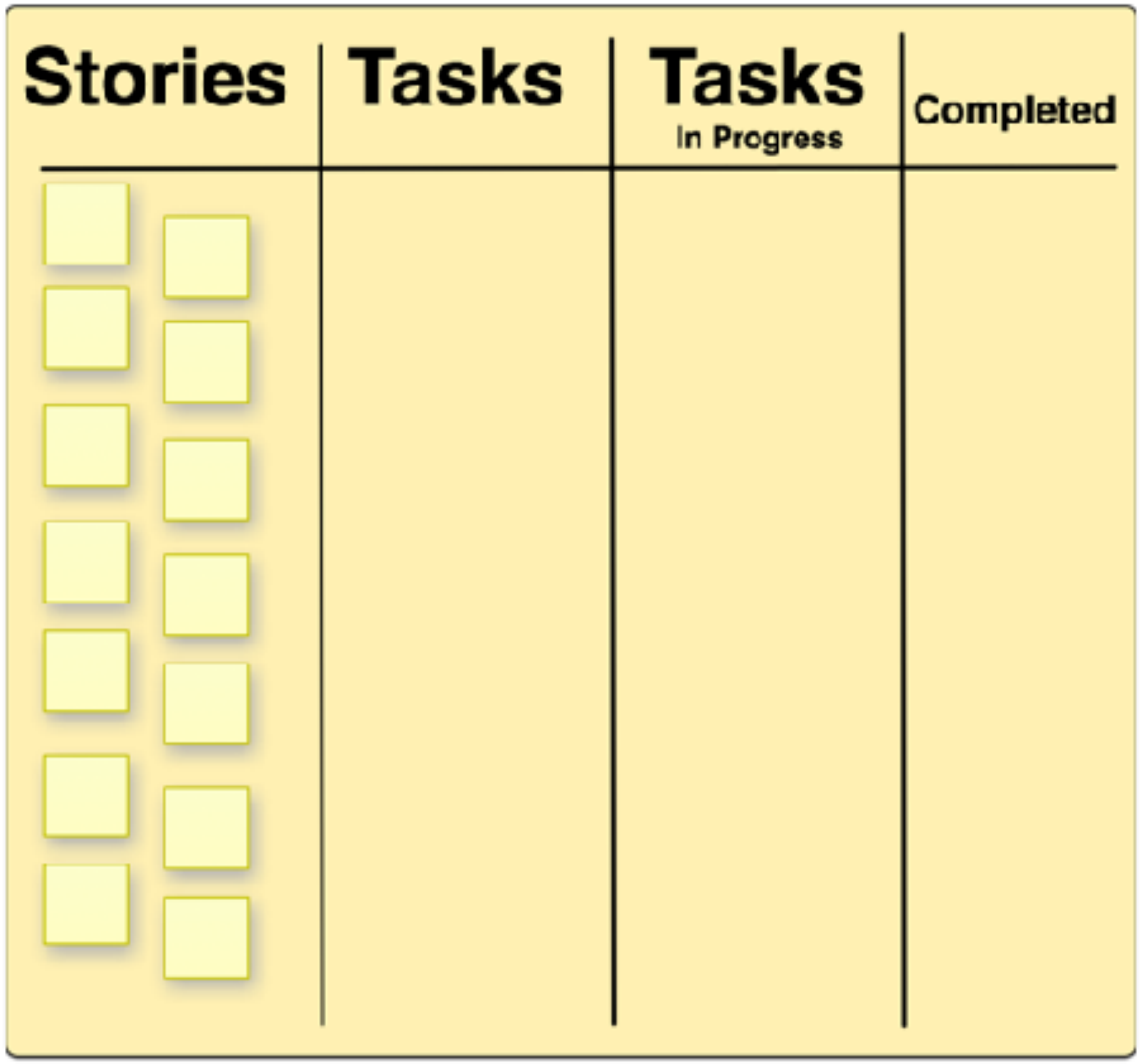
- **CARD**: Token representing the requirement. Notes are written on it, reflecting priority and cost
- **CONVERSATION**: Requirement is communicated from customer to team to refine details (often supplemented with documents)
- **CONFIRMATION**: Acceptance tests make it possible to understand what needs to be done to complete the requirement

- **Independent** - schedule and implement in any order
- **Negotiable** - able to change
- **Valuable** - must be valuable to customer
- **Estimable** - to help customer rank based on value and cost
- **Small** - consumable by a team for implementation
- **Testable** - shows story is clear and defines done state

THEME -> EPIC -> USER STORY

User Story Writing - Scrum Simulation

- Write stories that represent the items (20 stories)
- Divide and conquer on different ideas/themes
- All - place the stories on Backlog
- 20 minutes



AGILE FOR ALL
Making Agile a Reality.®

Estimation

Now Let’s Plan!

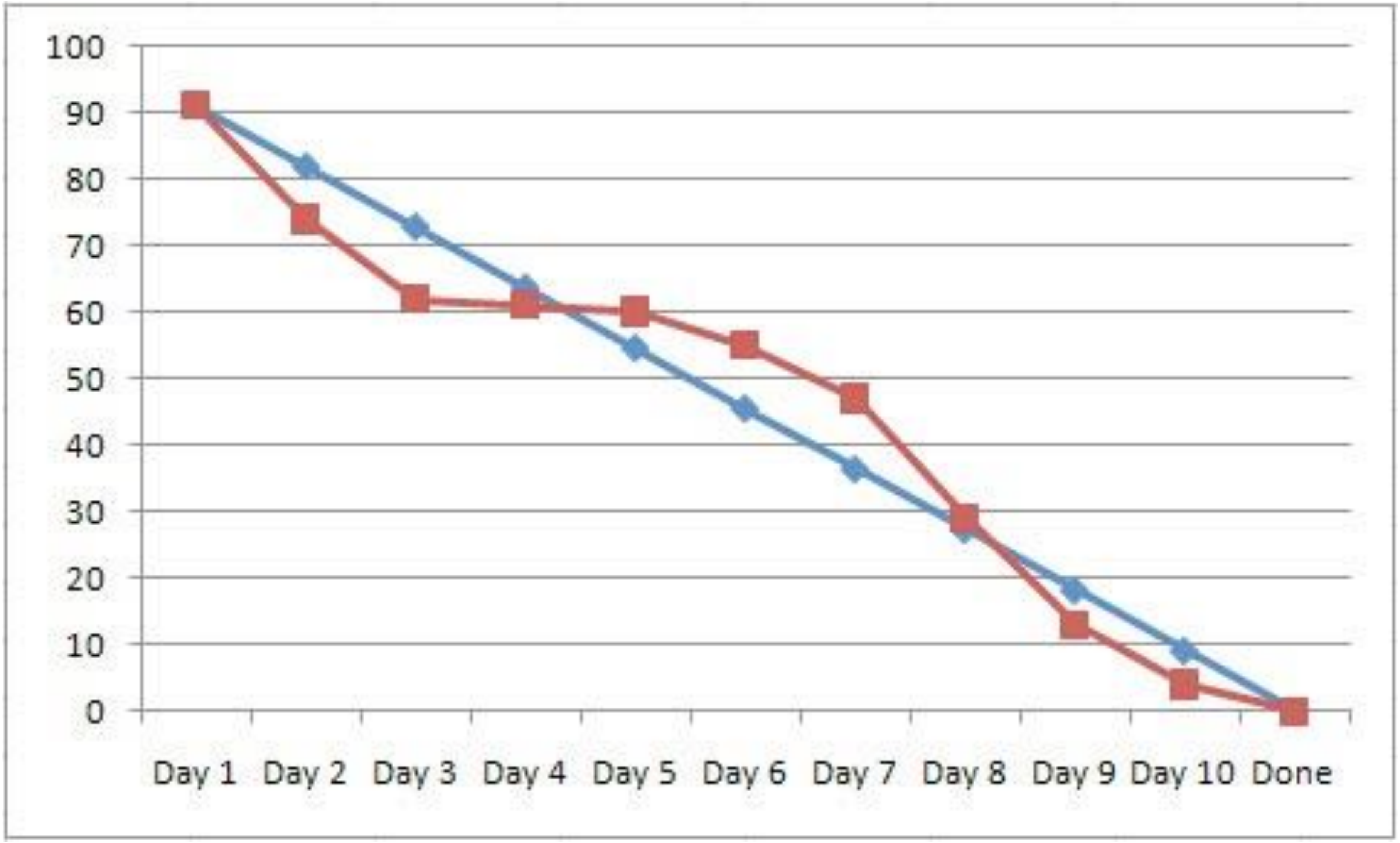
- Release Planning
  1. Requirements -> User Stories
  2. Affinity estimation
  3. Prioritize
- Sprint Planning - (24 min sprints)
  1. Highest priority
  2. Questions
  3. A/C
  4. Tasks / Plan together
  5. Incremental commit

Teams - Execute the Sprint 1, 2, 3

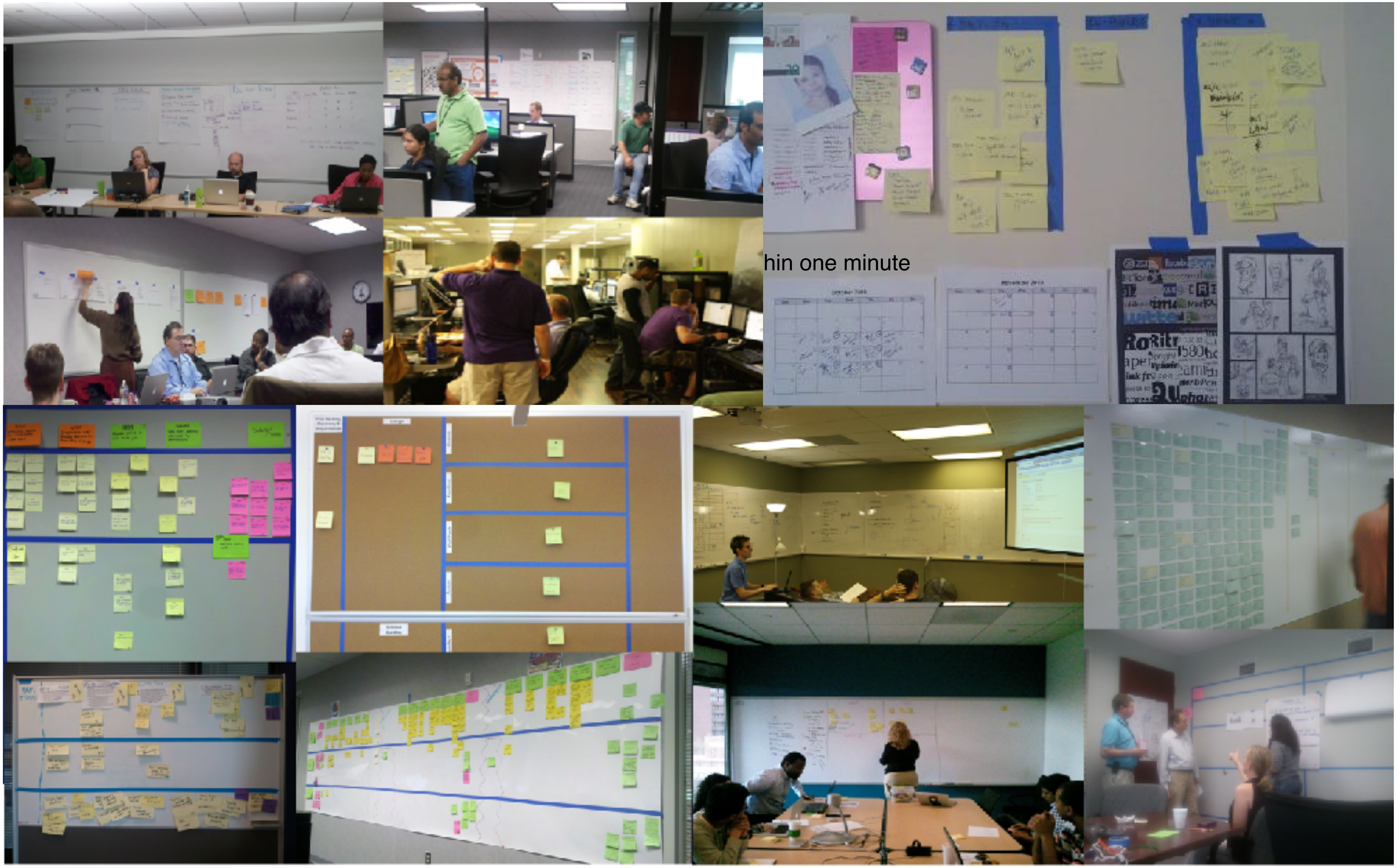
- Sprint Schedule:
  - 8 min - Day 1
  - 8 min - Daily Scrum + Day 2
  - 8 min - Daily Scrum + Day 3
  - 3 minutes - Review / Demo
  - 10 minutes - Retrospective
- Team self-organizes
  - Use task board during Daily Scrum
  - Volunteer for Tasks
  - Pair and collaborate
- Product Owner
  - Write, refine, and help team understand User Stories
  - Add them to Product backlog
- And then... back to Sprint Planning for the next Sprint.
  - Priority, commitment, A/C, tasks, review DoD, burndown, re-estimate from experience?

Burn Down Charts

- **Days** on horizontal axis / **User story** units on vertical
- Does not track effort, only completed work [units]
- Theoretical trend line



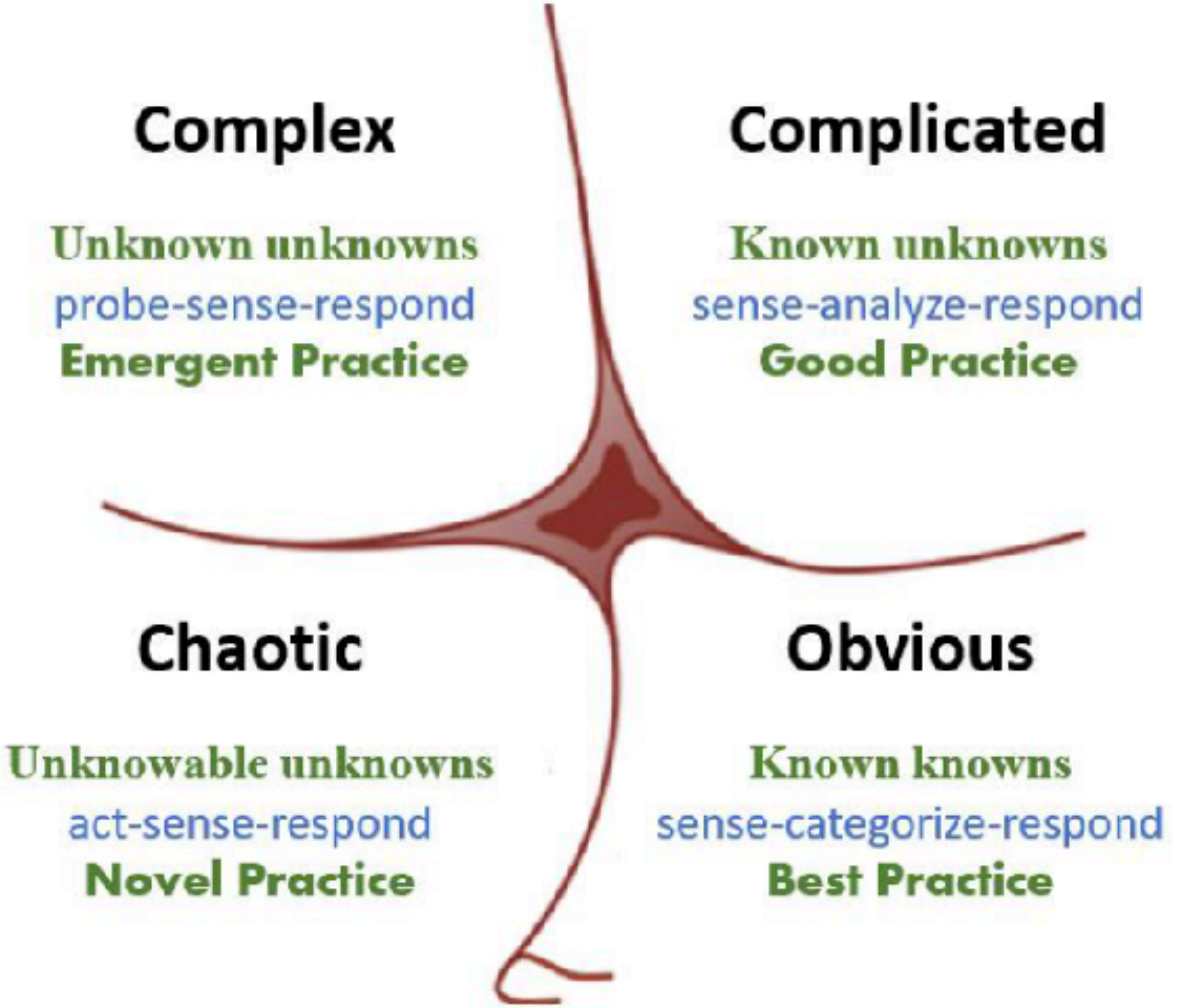
Collaborative Common Space + Info Radiators



http://sloanreview.mit.edu/article/why-showing-your-face-at-work-matters/

The Current World & Cynefin (kun-EV-in)

~~Slow~~
Fast
~~Isolated~~
Global
~~Simple~~
Complex
Humans
~~Disconnected~~
Networked
~~Reclusive~~
Social



Source: Dave Snowden www.cognative-edge.com

Use Cynefin to:

- Help understand how you perceive situations
- Make sense of other's behavior
- Help reach decisions
- Create a sense of place in a complex world

AGILE FOR ALL
Making Agile a Reality.®

The 5 Scrum Values

How a person behaves reveals what they value. Take a value and describe the behaviors you'd expect to see in an organization who lives out these values. After you're done, grab a partner and review.

Example: *A organization that values Focus does not force a team to work on multiple projects at the same time.*

- Focus ____
- Respect ______________________________________
- Openness _____________________________________
- Commitment ___________________________________
- Courage ______________________________________

Have you run into cognitive dissonance?

Book Resources

- The Agile Pocket Guide, Peter Saddington, Wiley, 2012
- Agile Software Development with SCRUM by Ken Schwaber, Mike Beedle
- Agile Project Management with Scrum by Ken Schwaber
- User Stories Applied by Mike Cohn
- Agile Estimating and Planning by Mike Cohn
- Collaboration Explained, Jean Tabaka, Addison Wesley, 2006
- Agile Retrospectives, Esther Derby and Diana Larsen, Pragmatic Programmer, 2006
- Lean Software Development, Poppendieck and Poppendieck, Addison Wesley, 2003
- Agile & Iterative Development, Craig Larman, Pearson Education, 2004
- Agile Estimating and Planning, Mike Cohn, Pearson Education, 2005
- User Stories Applied, Mike Cohn, Pearson Education, 2004
- Extreme Programming Explained, Kent Beck, Addison Wesley, 2000
- Project Retrospectives, Norman Kerth, Dorset House, 2001
- The Knowledge Creating Company, Nonaka and Takeuchi, Oxford University Press, 1995
- The Pragmatic Programmer, Hunt and Thomas, Addison Wesley, 2001
- The Blind Men and the Elephant: Mastering Project Work, David Schmaltz, Berrett-Koehler Publishers; , March 2003
- Complexity and Management, Ralph D. Stacey, Routledge, 2000
- Corps Business, David Freedman, HarperCollins, 2000
- Industrial Dynamics, Jay W. Forrester, MIT Press, 1961
- The Art of Focused Conversation, Brian Stanfield, New Society Publishers, 2000