# Executive/Manager Introductory Workshop on Agile-Scrum

By: Joe Little, CST and MBA
August 2019

# Situation…

- This is a 4 hour introduction.
- “Scrum is simple, but very hard to do well.”
- Anything to do with people is inherently complicated.
- So, much more to say and do later, but time is limited today, so…

# Joe Little

- Agile Coach & Trainer (CST), MBA
- 20+ years in senior level consulting to well-known firms in New York, London and Charlotte, and elsewhere.
- Focus on delivery of Business Value; interest in Lean
- CST (CSP, CSM, CSPO)
- Was a Senior Manager in Big 4 consulting
- Head of Kitty Hawk Consulting, Inc. since 1991
- Head of LeanAgileTraining.com
- Started trying to do [Agile] before reading The Mythical Man-Month

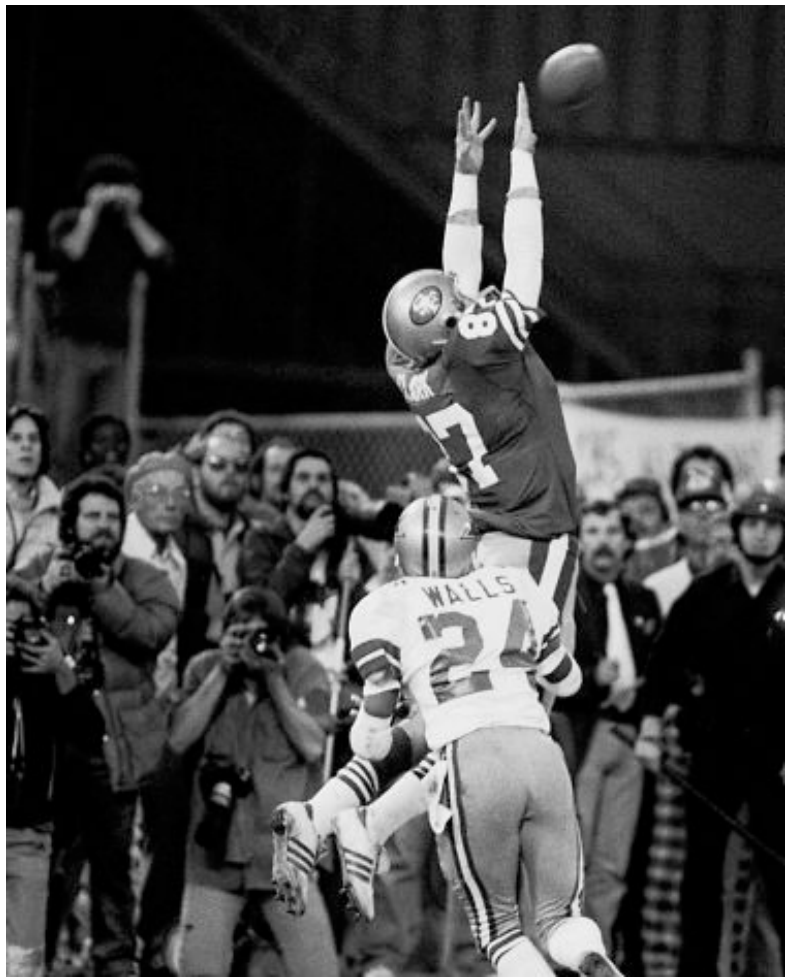
  - **www.LeanAgileTraining.com/blog**
  - **jhlittle@leanagiletraining.com**

# Why me?

- CST and MBA (not a common combination)
- Co-trained 8x with Jeff Sutherland
- Did waterfall for 20+ years. Agile full-time since 2005. CST since early 2008.
- Two books (Agile Release Planning, A Scrum Introduction)
- Wide experience with many types of firms.
- Reside in Charlotte, NC.  Have taught on most continents. (Not Antartica.)

# Agile is… easy and difficult

The Catch
(See wikipedia)
1982



# Why do Agile?

- Some organizations seem to do Agile because it is the latest fad.

- Agile-Scrum is a means to achieve key business goals. You all need to make those connections.

# What do you want Agile to do for your firm?

- What benefits do you want?
- What benefits do you expect?

# Some Benefits

- More frequent delivery (can respond more quickly)
- Better TTM (time to market or speed to market)
- Higher productivity (velocity per Sprint)
- Happier customers
- More business value delivered per period
- Happier workers (useful!)
- Higher quality

# The benefits are big

- Even if you do not transform very well, benefits are still big and well worth it
- …if you can change enough to do Agile-Scrum truly well, benefits are potentially huge

# Scrum in Practice

## "Transforming the World of Work"

ScrumAlliance
transforming the world of work

# A simple game

- We are building ….
- …a real team (of 7)…
- …who want to win together…
- …under challenging circumstances…

# Myths of Agile

- There are many.
- Problem: in some sense the myths (the bad things) are done under the name of “agile”
- They are NOT true with professional agile.
- We will discuss some along the way.

# What is ‘Agile’?

- Two quick ways to describe:
  - Agile is the opposite of Waterfall
  - Agile is any ‘method’ that subscribes to the Agile Manifesto and Agile Principles
    (a pretty good definition)
- Alternate: Agile is any approach that gives your business a better ability to adapt successfully in a changing, almost chaotic, environment

# Some Agile Methods

- There are many…
  - Extreme Programming (XP)
  - Scrum (defined earlier than XP; includes a kind of Kanban)
  - Kanban Method (David Anderson, kind of from Taiichi Ohno)
  - others

# Some differences

- In the next few slides…

# Autonomy / Independence

- “Although project teams are largely on their own, they are not uncontrolled.” Takeuchi and Nonaka.
- Part of being a real Team. They make decisions. What is the scope of decision-making?
- A high level of independence.
  - Self-organizing
  - Self-managing
  - Self-directing
- But not completely.

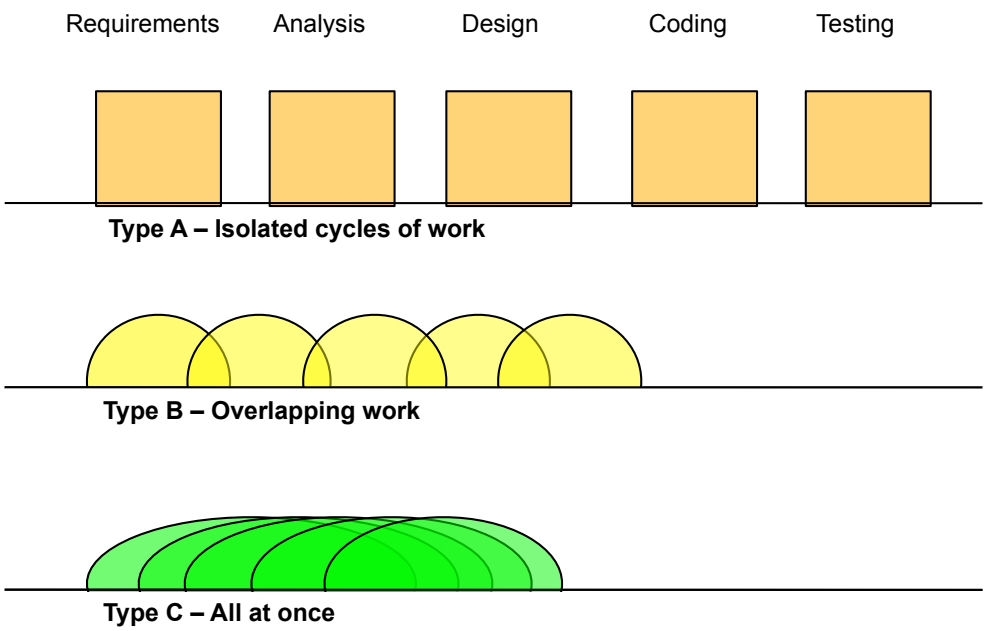
# Three dysfunctions

- Managers leave the Team alone too much, and especially do not fix impediments for the Team.
- Managers under-cut the autonomy of the Team. Make too many decisions, reverse the PO's decisions, force them to do other work (w/o explanation).
- Managers do not support self-organization enough. (Understand, talk, explain, encourage.)

# Other differences

- High “known” situations vs high “unknown” situations.
- Machine intensive process vs people intensive process.
- Particularly knowledge worker intensive — about creation and innovation.

# Agile vs. Waterfall



# Change or Die

- Maybe I over-state the case…., but…
- Other firms are learning how to use agile…
- I think: if you do not learn agile faster than they do, you will get beaten in a relatively few years.
- And the pace of change is NOT slowing down. You MUST HAVE the ability to adapt to change faster.
- Still: I am not sure you must agree with me now. But think about it.

# A too-fast Introduction to Scrum.

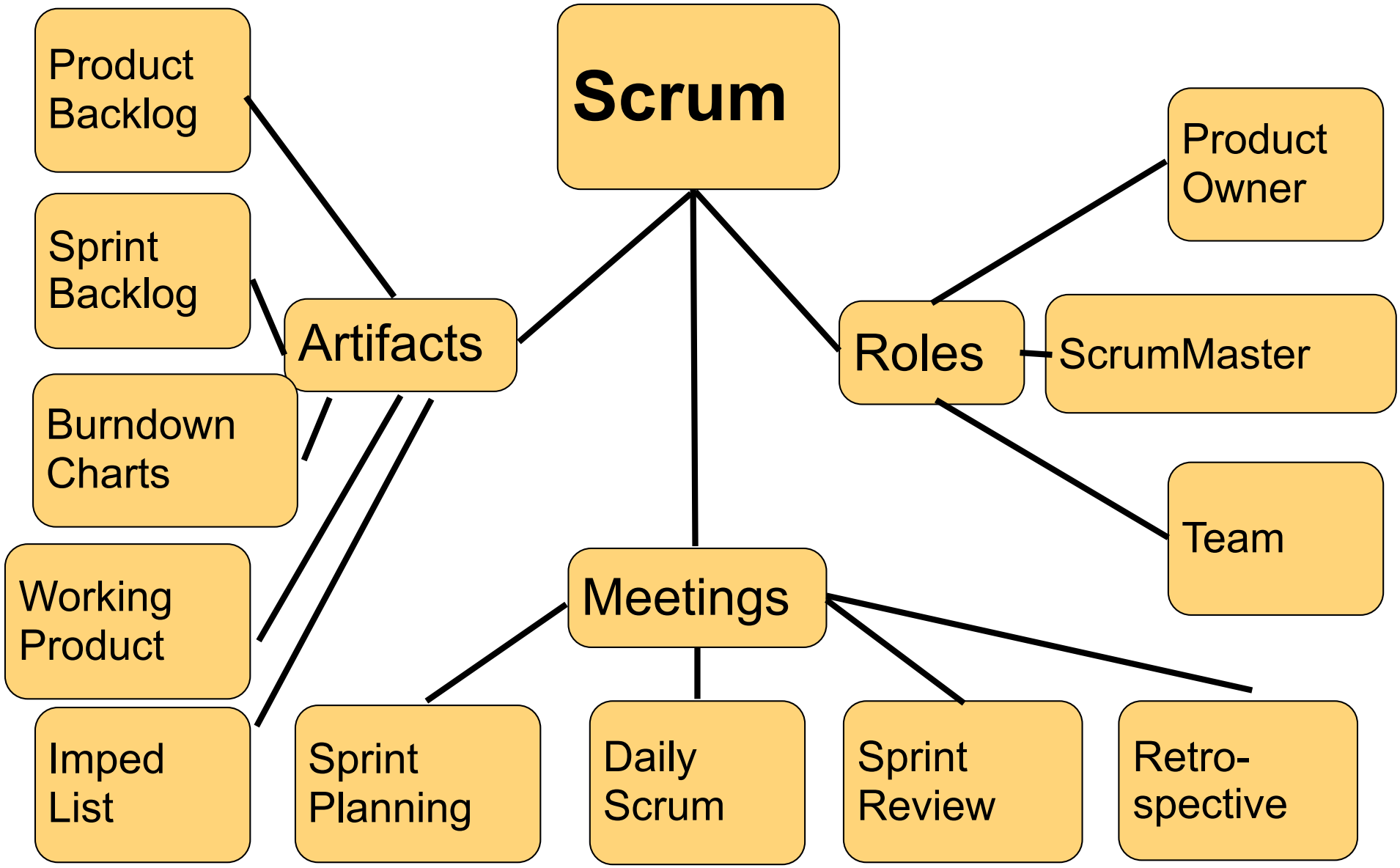
- Let's do a super-fast Introduction to the basics of Scrum

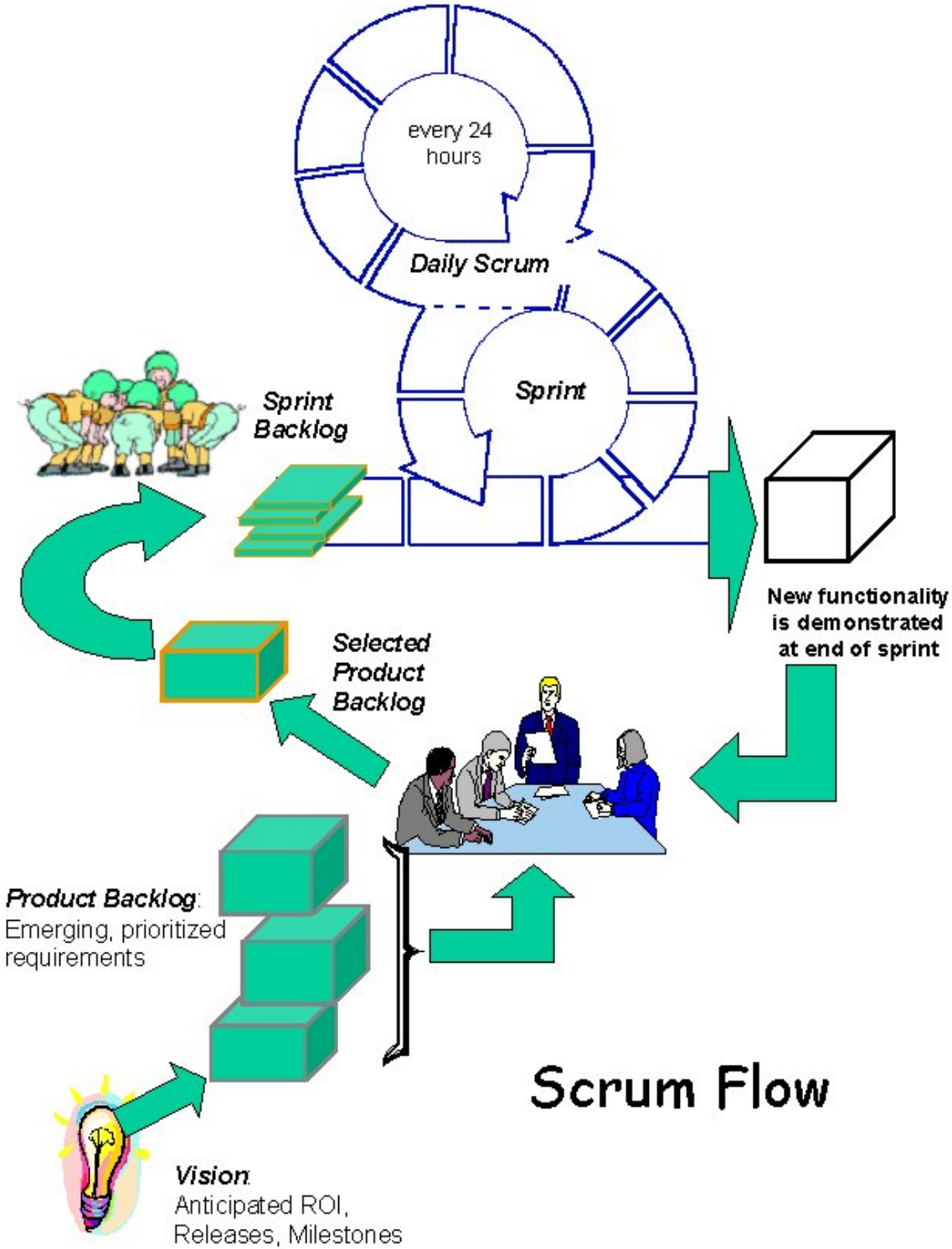
- It is much too fast, but a start…  You must learn more later.

# Scrum is…

- …a bare framework.
- It is NOT a full methodology.
- Each Team MUST add to Scrum, using “common sense”, to do its work professionally.
- “Common sense is not very common.”

# Scrum is a Simple Framework






Scrum Flow

1. What did you do yesterday?
2. What will you do today?
3. What got in your way?

# Empirical Process

- Transparency - Important that multiple people can see X at the same time.  (The product increment, the inputs, the process.)

- Inspection - of the product and the process and the inputs. You expect problems.  You then adapt. For complex situations, by multiple people.

- Adaptation - They know what “should be” for the inputs and the process. They can diagnose quickly what went wrong. They have the power and ability to adapt quickly (successfully).

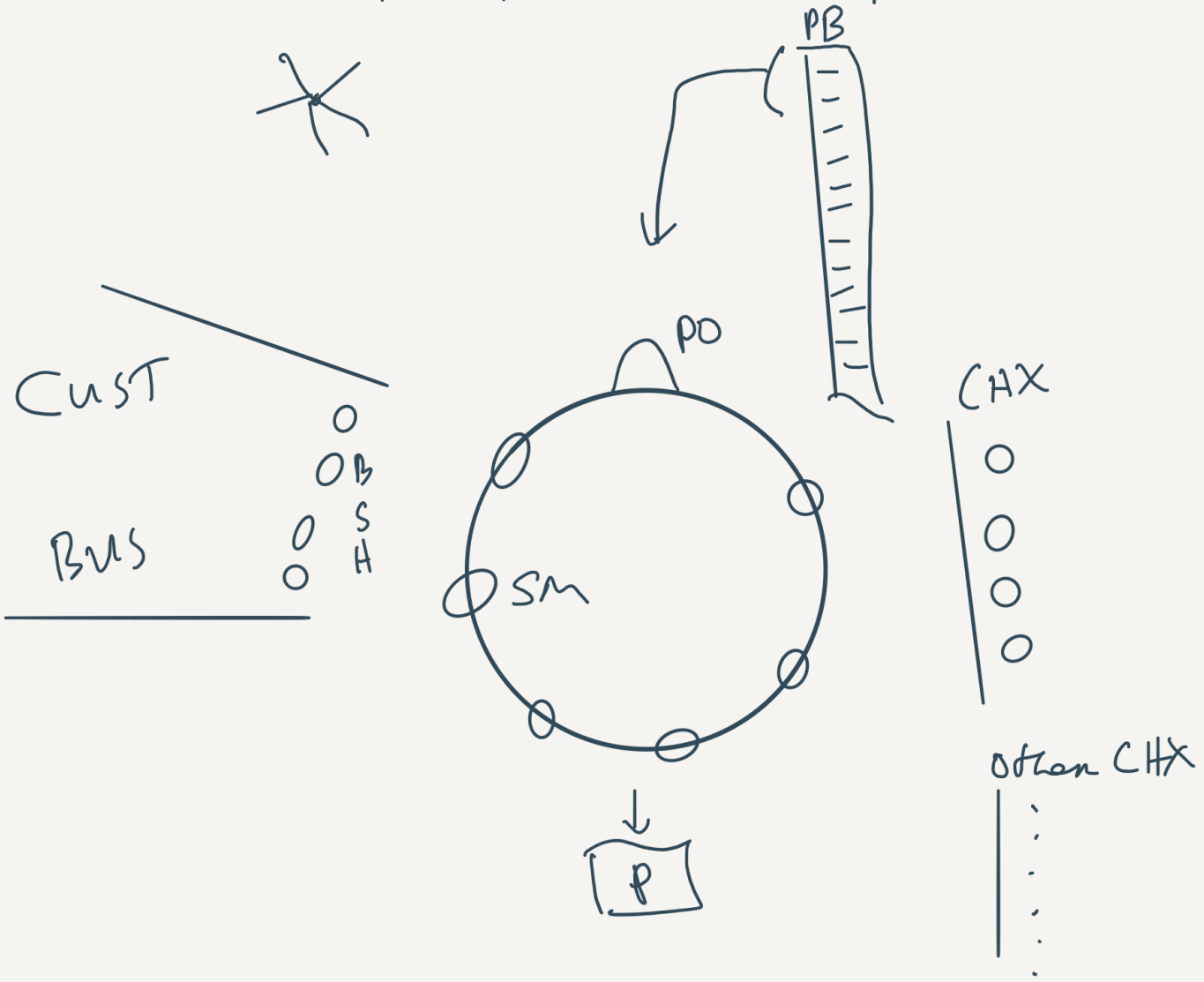
# Agile Values

- Can you explain one of these now?
  - Commitment
  - Courage
  - Focus
  - Openness
  - Respect

# Optimize Value

- Always “too much to do”
- So, prioritize.
- Try to optimize using the Pareto rule. 20% of the work delivers 80% of the value.
- Timeboxes help.

# The Fuller Picture



# Continuous Improvement

- Scrum has a strong focus here.
- Key to SM.
- Key in Daily Scrum and Retrospective
- Shown (partly) via increases in Velocity, and we expect notable improvement, mainly by overcoming notable impediments.
- Similar to Lean and many other approaches.

# Key Issues (can be addressed)

- Team size (normally 7 is good)
  - How to use the “chickens”?
- What is/are the product(s) for a Team?
- What about urgent/important work identified during the Sprint?
- How to become a great team if not collocated? (harder)
- How to break down the silos over time.

# Scope of Scrum

- Sweet Spot: New product development (broadly)
- Any work (with a team)… ?
- Maximum Sprint is 4 weeks
- Best known: regular software development
- But all kinds of work and products have been delivered using Scrum.

# Advantages of Real Team of 7 people

1. More self-reliant
2. Less risk (eg, more knowledge redundancy if one person leaves)
3. More heads to learn from; to see the fuller elephant
4. More people to help you get unstuck
5. Less wait time
6. Not too many to communicate/collaborate with
7. More ideas about how to improve
8. Better ratios (for PO and SM)

# Why is “understanding the Customer” so hard?

1. We do not know the Customers well enough
2. They and we do not understand the problem well enough
3. They and we do not understand the root cause(s) well enough
4. They and we do not articulate the solution well enough

# To understand the Customer…(better)

- We all identify PB Items (user stories) at high level “up-front”
- We get details in a rolling wave (the DOR).
- The PO represents the customer (full-time) (final decision-maker)
- The (4?) Business Stakeholders represent the customer (part-time), and are willing to call out problems.
- The Sprint Review
- Frequent Releases/Deliveries (more feedback)

# It’s hard - but why?

- First two…
- Change is hard…
- Being as completely honest and transparent as Scrum requires is hard.

# Planning - Some CHANGES!

- We accept the relative chaos of life (when doing innovation).
- Agile views planning as very valuable, and all plans as very inaccurate predictions of the future.
- Yogi Berra: To predict is difficult, especially of the future.
- Mike Tyson: Everybody's got a plan until they get punched in the mouth.

# Culture Change

- For many companies and maybe especially for engineering cultures, there is discomfort with the imprecision of planning.  Sorry!
- Agile Myth: Agile people do zero planning!
- Fact: We spend more time planning in agile than with waterfall.

# The Truth (re planning)

- We do a fast initial plan, prioritize our “stupidity” and learn FASTER!
- As we learn (including about changes), we re-plan.
- We use the planning process to adapt faster.
- In Agile we spend MORE time planning.
- We expect a new plan every sprint.  We want to learn that fast.

# Agile Release Planning - first day

Example: 6 months of work for Team of 7.
Process:

- Articulate Vision
- Develop Product Backlog
- Identify Business Value
- Identify Effort
- Consider benefits-costs
- Risks, dependencies, Learning, MVP, other
- Lay out stories in Sprints; identify first release (more?)
- Complete the Day Zero plan

Next day: Start release plan refactoring - every sprint.

# Simple yet Hard

- It is simple, and the benefits are huge.
  - In one year from starting, we expect each Team to double their productivity. For example.
- BUT…
- It is hard.
  - Lots of companies fail to successfully adopt agile. Not good.
- It requires effort, and it requires some investment ($, but more mentally).

# Agile is a paradigm shift.

# Agile Manifesto and Agile Principles

- We will now review these.
- Try to imagine which ones YOU can accept and which ones seem concerning.
- Try to imagine which ones THE CULTURE can accept and which ones seem concerning.

# Agile Manifesto

www.agilemanifesto.org

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

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

That is, while there is value in the items on the right, we value the items on the left more.

# The Principles behind the Agile Manifesto - 1

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.

# The Principles behind the Agile Manifesto - 2

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.

# Ok, there you have it!

- You have the basics.
- What questions do you have at this stage?

- Maybe two types of questions or concerns:
  - I don't quite understand "x"
  - I think I understand it, but I have concerns about how "x" will work

# What are your biggest questions or concerns?

- Let's list them and discuss as many as we can in X minutes [15?]

- Later we will also talk about "impediments".

# Other: Extreme Programming (XP)

- Much more involved than Scrum, more pieces are parts
- Recommended: Add it to Scrum (where appropriate) later.
- Add ideas and practices one at a time.
- Includes most of Scrum, but with some different names.

# Other: Kanban Method

- Originates from Piggly-Wiggly in 1960's (?)
- Kanban is Japanese word for "card".
- Taiichi Ohno uses Kanban practices to rigorously improve within context of "The Toyota Way"
- David Andersen recommends Kanban if you can't change anything else.
- Using cards to visualize useful things about the work and making it better.
- To be much more effective, why not add the Scrum things?
- Scrum includes Kanban: The Scrum Board (and the PBL and SBL).

# Lean Ideas (three)

- Mura - unevenness of flow
- Muri - over-burdening the system
- Muda - waste

- Implement in this order.

# Changes for the Managers

- New lingo - easy
- Really understanding the meaning of the new lingo - hard
- Practicing the new paradigm at a winning professional level - very hard (Tom Brady is one example)
- Helping the culture change to the new paradigm - very hard
- Helping teams with impediments - **much easier**!! (Yay!)

# How Management Changes with Agile

- Self-management by the Team
- One manager (for 4 teams?)
- One Steering Committee (for 20 teams?)
- Help them fix the impediments, one at a time.

# What are the biggest impediments?

- Let's list them….
  - To Agile Transformation
  - To a team doing the work more effectively

# Key Issues

- Collaboration between Business and technology
- Insufficient effort in making the change happen (so common)
- Resolving things with HQ.
  - They will see it differently
  - They have a different culture
  - They want you to do it “the right way” (I expect), and actually they are partly right - and also partly wrong (I suspect). You must own it, which means you must do it your way, to some degree.

# New ideas

- There are many more new ideas with lean-agile-scrum.
- Next: I recommend you read the two articles that discuss these two sets of ideas.

# Common mistakes in adopting Agile - 1

1. Waiting until everything is ready.
2. Not good enough competence in making Big changes happen.
3. Unwillingness to really change.  AKA “agile in name only”.  Or a LOT of scrum-ish (or Scrum-Butt). (Related: Resistance)
4. Insufficient sense of urgency about agile-scrum (said earlier?)
5. Not girding for the long march
6. Insufficient energy (resources) supporting the change
7. Lack of defined actions to support the change
8. Badly formed teams (too small, no full-time PO or SM, not enough coders-testers).

# Common mistakes in adopting Agile - 2

1. Insufficient training (recurrent)
2. Insufficient coaching
3. Complacency when a Team is doing half-baked Scrum
4. Lack of aggressiveness in attacking impediments
5. Insufficient collaboration with the Business side
6. Not good support from the Top
7. Sense that the Top is forcing us to do Agile
8. And many more…

# Actions for you - 1

- Support the change.
  - Learn what it is.
  - Change is hard. Accept. Get ready mentally.
  - Get your hands dirty some, learn the lingo and the concepts.
  - Support as best you can. (Don’t just repeat slogans.)
  - Accept that there will be some pain for the gain.

# Actions for you - 2

- Help fix one impediment at a time
- Support self-organization by the Team; trust the Team
- Help prioritize the releases
- Give each Team only one mission at a time (more?)
- Business engages continuously.  Really about the same effort, but more continuous, with much better results
- Don’t disrupt the Team

# Scrum: The challenge is: can you adopt it more successfully than usual?

- This is better.  This can be much much better.
  - For customers, for the Teams, for the individuals.
- It is more fun!
- But are you ready and willing to take it on?
  - There will be many mis-understandings, and some mistakes and failures.
  - Change is hard.
  - In a sense, Agile will change ‘everything’.
- It is easily worth it. If you all have the guts for it.  And, oddly, it will still be more fun too.

# Biggest learning

- What is the biggest thing you learned today?

(You have more to learn about this.)

# THANK YOU!

(This will help many people; they thank you.)

# More learning needed

- It is a major paradigm shift and culture shift.  Respect that in yourself and others.
- It is not easy to do well.  Like … Rugby. Or other sports.
- I am happy to help (train, teach, coach, etc).
- You can get help from other places too.

# Contact Info

- Joseph Little
- www.leanagiletraining.com
- jhlittle@leanagiletraining.com

# Appendix

# The 4 common impediments (Joe's opinion!)

- Not a real team (yet)
- People on multiple 'projects' (teams)
- Not stable teams (yet)
- Insufficient PO or business side commitment to the Team
- Weak automated testing / continuous integration

# The New New Product Development Game

1. Built-in instability
2. Self-organizing project teams
3. Over-lapping development phases
4. “Multi-learning”
5. Subtle control
6. Organizational transfer of learning

# 6 Myths of Product Development

1. High utilization is good
2. Large batches are good
3. Just stick to the initial plan
4. People working on multiple projects is good
5. The more features per release, the better
6. No mistakes are allowed!

# Additional benefits

- More transparency
- More adaptability to change (good and bad change)
- More innovation
- Lower cost per unit of business value
- More profitability for the firm