



# Getting Started with Agile

**AgileTraining.com**

*Building Lean High Performing Teams ..*

# About Me

## Andre Simones

- Enterprise Agile Coach & Trainer – Agile Transformation
- Background: Systems & Software Engineer, Software Development Manager, Director of IT Project Management, Entrepreneur, CSM
- Over 8 years of implementing Agile practices with many software teams spanning many organizations.
- Software developer by night (Ruby on Rails, PostgreSQL, etc.) working on the next shiny object.

# Agenda

- \* Understanding Agile (quick)
- \* When to Consider Agile
- \* Shu-Ha-Ri
- \* Common Patterns for Getting Started
- \* Patterns of Resistance
- \* Patterns for Success
- \* Smelly Agile

# The Agile Manifesto - 2001

## The manifesto's shared value statement:

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

<b>Individuals &amp; interactions</b>	Over	Processes & Tools
<b>Working Software</b>	Over	Comprehensive Documentation
<b>Customer Collaboration</b>	Over	Contract Negotiation
<b>Responding to Change</b>	Over	Following a Plan

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

# Agile Principles

Our highest priority is to satisfy the customer through **early and continuous** delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes **harness change** for the customer's competitive advantage.

Deliver working software **frequently**, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Build projects around **motivated** individuals. Give them the environment and support they need, and **trust them** to get the job done.

Business people and developers must **work together daily** throughout the project.

The most **efficient and effective** method of conveying information to and within a development team is **face-to-face** conversation.

# Agile Principles

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

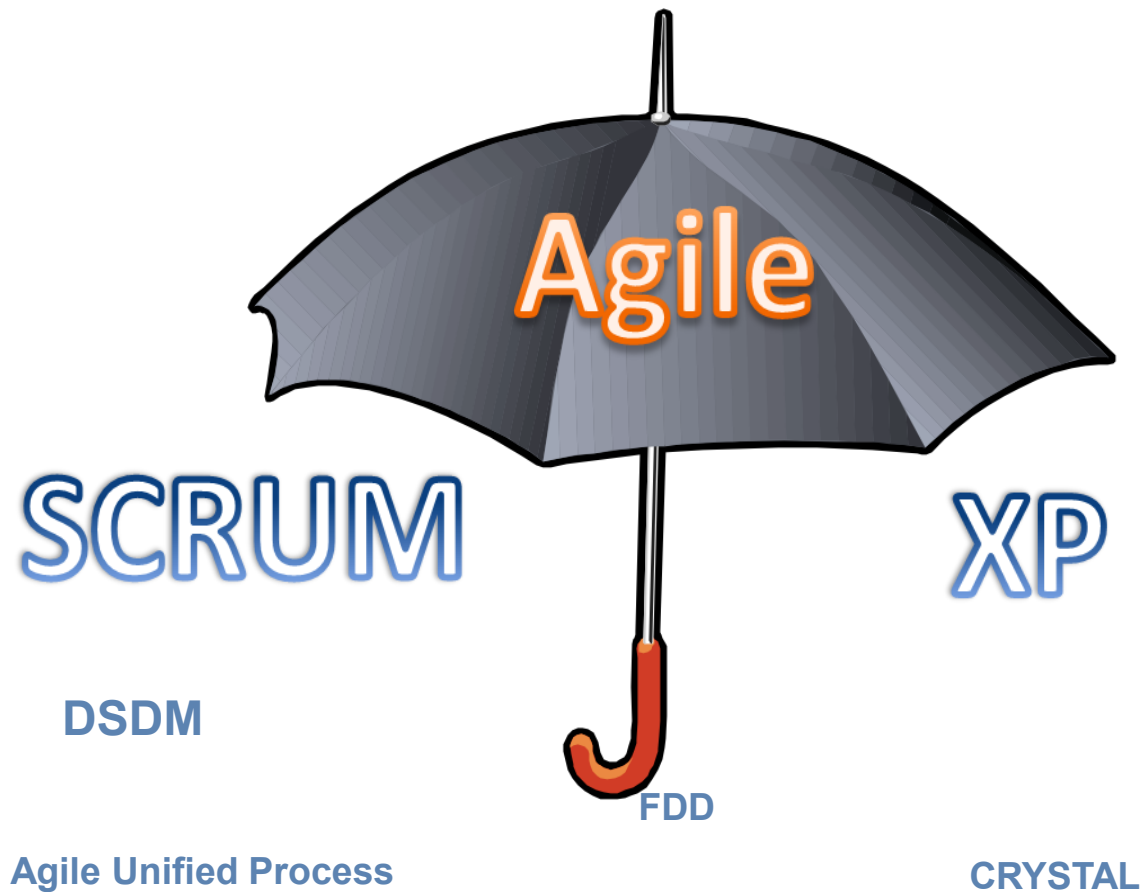
Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

# Agile is ...

An umbrella term for Agile methods



# Agile is ...

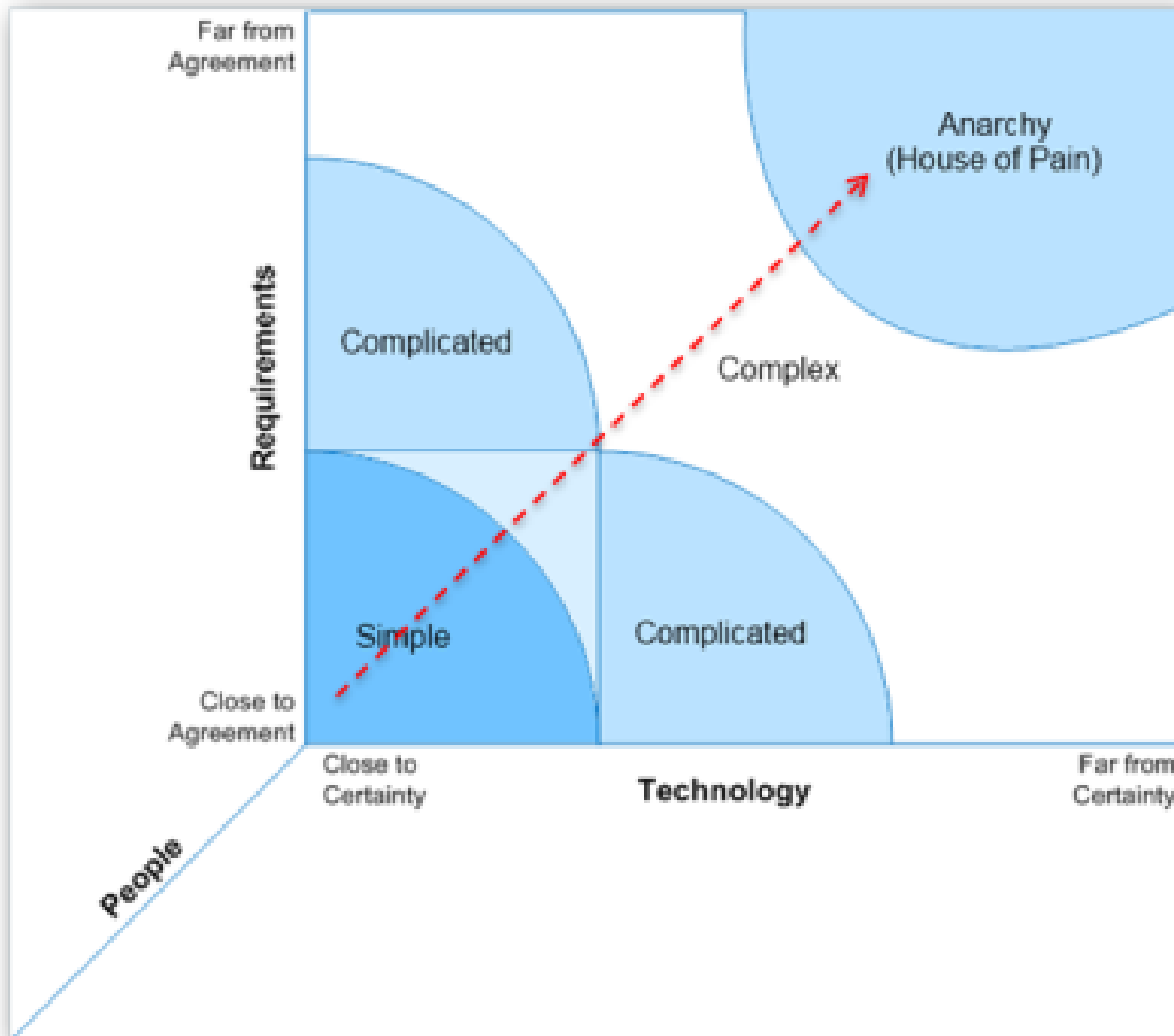
- Can be looked at as “Any method that encourages people to embrace and use Agile principles and values in their daily decision making.” (an original from yours truly)
- Goal of agile is to “deliver the highest value first with high quality as fast as possible.” (courtesy Bob Hartman, <http://agilebob.com>)



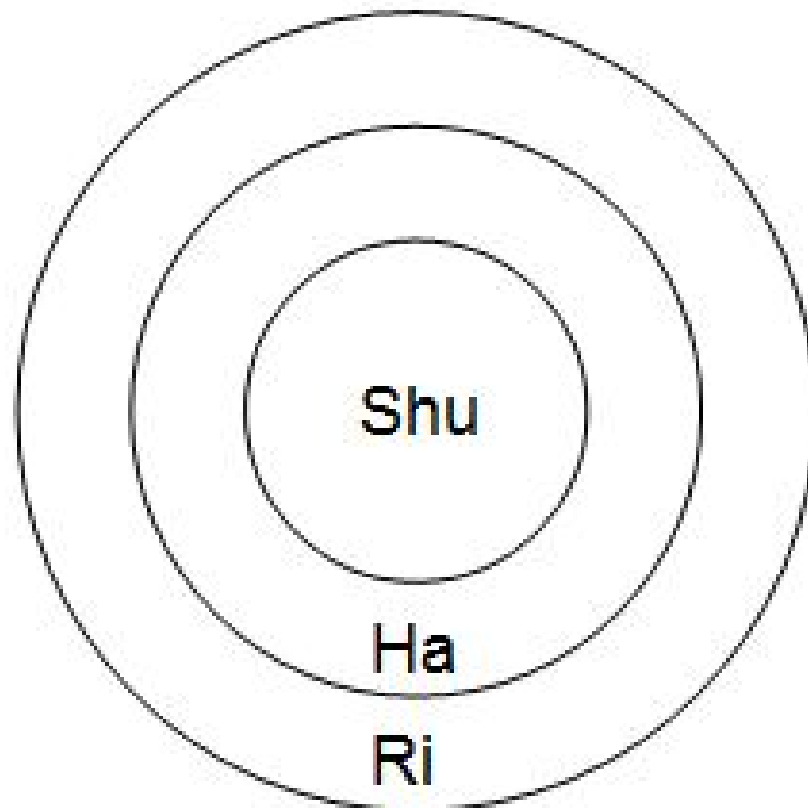
# When should we use Waterfall?

- **Good Reasons**
  - End state is well-known – no ambiguity
  - Requirements are clear and won't change or require business input or feedback
  - Technology is completely understood and there is very little or no risk
- **(Maybe) Not So Good Reasons**
  - Culture is resistant to change
  - We already kick butt! (don't need to improve)
  - There are no external business pressures for the business to become more agile, i.e. no competition (yet)
  - It won't work on "X" type projects

# When to Consider Agile



# Shu Ha Ri



# Patterns for Starting (Finally)

- All-In (Cannonball!!!)
- Grass Roots
- The Poster Child
- Call 911!

# All-In (Cannonball!!!)

The “All-In” pattern is used when the leadership (or CEO) of a company decide and decrees that their organization will be Agile and the transition will be immediate.

- Why/when To Consider
  - **Time is of the essence**
  - Won't have the problem of Agile and non-Agile teams having to sync up
  - Resistance will be reduced (not eliminated)
- What you need
  - Complete support from leadership team (in fact they are driving it)
  - Money
  - Help from the outside

# Grass Roots

The “Grass Roots” pattern is used when a team wants to “try” Agile. This is not a decree from leadership, but a desire to improve. The transition to Agile is done without fanfare or PR, i.e. it’s “covert”.

- *Why/when To Consider*
  - *Leadership does not know/understand Agile*
  - *Lots of misinformation floating around about Agile*
  - *The team collectively wants to try it*
- *What you need*
  - *Support from either the team lead or manager*
  - *Buy-in from the team*
  - *Commitment to continuous improvement*
  - *An evangelist would be great to “spread the word”*

# The Poster Child

The “Poster Child” pattern is one where someone in executive leadership has decided to “try” Agile, i.e. “dip their toe” in the water before jumping all in. It is advertised that Agile is being piloted, a team(s) is formed, and progress is monitored.

- **Why/when To Consider**
  - *Change is needed, but it is not time sensitive*
  - *Can't afford an “all-in” change*
  - *Culture change needs to happen, but can't occur all at once*
- **What you need**
  - *Support from Leadership*
  - *Agile evangelist(s)*
  - *Help*
  - *The right team and team commitment*
  - *The right pilot project*

# Call 911!

*This pattern is used when there is a critical “no fail” project and it either a) has not started or b) has started and stalled. Typically the project leader hears about Agile and is willing to try anything out of desperation.*

- *Why/when To Consider*

- *Your stuck and need to get unstuck*
- *The project is critical to the success of the company*

- *What you need*

- *Outside help*
- *An understanding that Agile only exposes problems, it does not solve them*
- *The person in charge of the project has complete control*
- *The organization is willing to make tough decisions*



# Patterns of Resistance

- Agile as Micromanagement (yes, it's common)
  - Developers get a LOT more attention up front
  - Testers are brought in up front also, when in the past they were used to “waiting” for the software at the end
- The Need to Know Everything Up Front
  - For some, it is hard to start without knowing everything, i.e. “the fat requirements doc”
- Distributed Development is a Fact of Life
- The Need for the Top Talent
  - Barry Boehm's principle of Top Talent - “Use Better and Fewer People”
  - “The top 20% of people product about 50% of the output”

# Preparing for Success

- Get an internal Agile evangelist
- Training
  - If using “all-in”, will need a lot of outside help
  - If using “grass roots”, identify that internal evangelist if you can’t bring in an external trainer.
- Agile is a culture change, and your culture may not be ready (yet)
- Agile is cross-functional, come up with a plan to introduce it to the other groups who need to be involved
- Be patient and persevere
  - Change is hard, don’t underestimate this. It will take time.

# What Smells!?!?

So you've gone "Agile", eh? Here are some common "smells" that crop up over and over. Recognize them, and figure out how to eliminate them.

- No retrospectives
- Sit down stand ups
- No product backlog
- No single person who controls the backlog
  - The backlog management can be delegated to the team!!
- The team tells the Scrum Master what to do
  - The Scrum Master is a servant leader, not just a servant
- No technical practices implemented
- No epics or themes on the backlog

# What Smells!?!? (continued)

- User stories written as “As a developer, I need to code this or that”, or “As a tester I need to test stuff”. Value is for the customer.
- Your team can’t get anything to DONE without another team
  - What could be the problem here?
- No burn up, burn down, burn anything charts
- The team doesn’t know their velocity
- Excessive requirements change for the team to be able to commit to a 2 week sprint
- Repeating problems and conversations slow everything down, Groundhog Day
- The entire team has only one skillset
- Thinking that Agile is nothing more than processes and procedures

# How to Get Rid of the Smells

- Remember Shu-Ha-Ri? Might need to go back to Shu.
- Ask the team!
- Training/coaching
- Reorganize/restructure
- Stick your neck out
  - You may have to have crucial conversations with the team and/or leadership.