

Agile Fundamentals for Project Managers

Saturday Workshop PMI Lakeshore Chapter

Aleem Khan

Agile Coach and Trainer

aleem.khan@360pmo.com

 @tweetaleem  <https://ca.linkedin.com/in/aleemkhan>



Aleem Khan – Agile Coach and Trainer

About me...

18+ years experience in managing complex projects & programs in financial, banking, manufacturing and telecom verticals.

Led multiple agile transformations

Methodology development and PMO expertise

Coach high performance agile teams

Author of agile case studies

Agile/PMI-ACP® course designer

Passionate about agile coaching, training and believe in life long learning.....



M.S in Project Management



Diploma in Computer Science



PMP, PMI-ACP



CSM



SAFe Agilist

Work as an agile coach, trainer, facilitator, program manager, project manager, and consultant in various organizations.....





15 Minutes

Explain, Explore - Ice Breaker

- Need is a piece of paper and pen per person
- Take a minute and write down **a word** or **phrase** that is true for you
- Find some stranger, and introduce yourself, exchange names and then explain to each other, why your description is true for you
- Now **swap** your cards for someone else's at least 4 times
- Take a look of card you have, find different partner, introduce yourself and explore how this could be true for you

Begin with End in Mind

Understand the **Fundamentals** of agile

Differentiate between various **agile methods**

Learn many **agile practices** and

Most importantly.....**Have fun!**

Workshop Logistics

Aha moments

Parking lot items

Rules of engagements

Any other business (AOB)



4 Minutes

Rules of Engagement / Team Contract

1. Be open to new approaches and listen to new ideas
2. Give everyone the opportunity for equal participation specially encourage introverts to be part of a team
3. Avoid blame or name, instead discuss the process and explore how it can be improved
4. Always find new ways to improve by exploring, inspecting and adapting
5. Seek first to understand, and then to be understood
6. Listen openly to other point of view
7. Keep discussion on track
8. Parking lot will be used to capture "off topic" questions, ideas and concerns

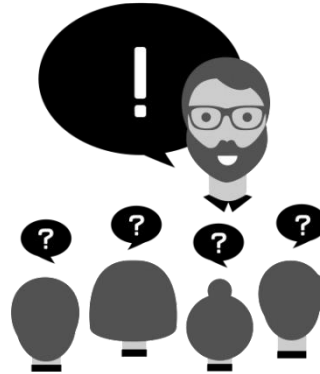
LEARNING 1.0

FOCUS ON BEING TAUGHT



LEARNING 2.0

FOCUS ON ASKING



LEARNING 3.0






FOCUS ON SHARING



Source: <http://www.learning30.co/learning-3-0/>

Process Miniature

Run the entire process in a very short time period (a few minutes to a few weeks)

#	Activity
1	Explain, Explore - Ice Breaker 
2	Learning Patterns 
3	Process Miniature 
4	What & Why of Agile 
5	Agile History 
6	Agile Values 
7	Pocket Size Principles 
8	Methodology Selection 
9	Traditional phases vs. Increments 
10	Agile teams 

#	Activity
11	Team Collaboration 
12	Various Agile methods 
13	Scrum 
14	Daily Stand-up Simulation 
15	Extreme Programming 
16	Kanban 
17	Lean 
18	Agile Myths and Facts 
19	Waterfall & agile differences  
20	Parking lot / Q&A 

 Group Activity

 Lecture

What is Agile?

Methodology?

Framework?

No Discipline

No
Design...

Another Fad

Iterative

No Planning!

No

Documentation !

Process?

*...Silver
Bullet*

Approach?

Definition

Agile is a an approach of building products or services by **EMPOWERING** and **TRUSTING** people, acknowledging **CHANGE AS NORM**, and promoting **CONSTANT FEEDBACK.**

Definition

Agile is a **PHILOSOPHY** that uses organizational models based on **people**, **collaboration** and **shared values**.

Agile uses rolling **wave planning**; **iterative** and **incremental delivery**; **rapid and flexible response to change**; and **open communication** between teams, stakeholders, and customers.

Definition

Agile is a **MINDSET**.....

established through **4 VALUES**,

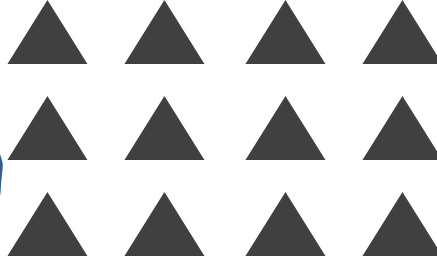
grounded by **12 PRINCIPLES** &

manifested through many **DIFFERENT PRACTICES**

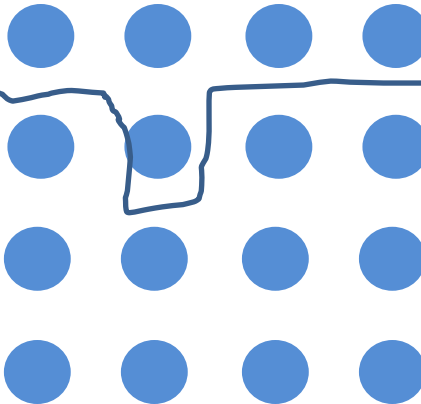
4 agile values



12 agile principles



Many agile practices



Scrum

4 AGILE VALUE

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

12 AGILE PRINCIPLES

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.

MANY AGILE PRACTICES

- | | | |
|-------------------------------|--------------------------------|---------------------------------|
| • Time-boxing | • Limit Work in Progress (WIP) | • Pair Programming |
| • Retrospective | • Avoid Waste | • Face to Face Conversation |
| • Spike Solution | • Short Iterations | • Osmotic Communication |
| • Planning Poker | • Sprint Goals | • Test Driven Development (TDD) |
| • Backlog Prioritization | • Servant Leader | • Velocity |
| • Progress Elaboration | • Self -organization | • Unit Testing |
| • Minimal marketable Features | • Team Agreements | • Test First Development |
| • Personas | • Release Goals | • Technical Debt |
| • Story Mapping | • Release Plan | • Task board |
| • User Stories | • Project Chartering | • Swarming |
| • Product Backlog | • Quality Assurance | • Regression Test |
| • Visualize Workflow | • Refactoring | • Minimum Viable Product |
| • Wireframe | • Relative Sizing | • Last Responsible Moment (LRM) |
| • Daily Stand-up | • Product Vision | • |

Why Agile?

Accelerate time to market

Enhance software quality

Reduce cost

Managing change priorities

Project visibility

Enhance software maintainability

Better align IT/Business

Reduce risk

Improve team morale

Increase productivity

Simplify development process

Improve/increase engineering discipline

1960

- **Project Mercury – NASA** First time used iterative and incremental development (IDD) in software
- Time-boxed Half day iterations
- Applied XP test-first development practice

1976

- **Tom Gilb** Introduced the terms “evolution” and “evolutionary” to the process lexicon
- Adaptive iterations
- Fast time to value

1985

- **Barry Boehm** Created spiral model
- Team prioritization based on risk
- Risk driven/exposing risk early

1985

- **Hiroataka Takeuchi & Ikujiro Nonaka** Introduced human driven/knowledge work
- The new production development game
- Cross-functional team
- Self-organize
- Legitimate power
- Sense of mission

1995

- **Ken Schwaber and Jeff Sutherland**
- First co-presented Scrum that they were using from previous few years
- Time-boxed iterations
- Small and co-located team
- Transparency, inspect and adapt

1980

- **Gerald Weinberg** wrote in Adaptive programming
- Small increments with customer driven feedback

• **Rum Baugh & Jacobson**

- Rational unified process (RUP)
- User cases, UML, architecture centric

1997

- **Jeff de Luca**
- Feature Driven Development
- Deliver Tangible working Software in timely manner
- Focused on value-driven approach

1994

- **16 RAD practitioner formalized RAD**
- Dynamic System Development Method (DSDM)
- Used most commonly in Europe

1995

1996

- **Kent Beck, Ward Cunningham and Ron Jefferies**
- Extreme Programming
- Focused on Engineering practices

1998

- **Alistair Cockburn**
- Crystal Family
- Situational specific practices
- People and communication, design, principles, domain

2000

- **Robert Charette**
- Lean development
- Strategic focus, lean production, Stretch goals

2001

Agile Manifesto

Meet the Agile influencers



Kent Back



Mike
Beedle



Arie
Bennekum



Alistair
Cockburn



Ward
Cunningham



Martin Fowler



James
Grenning



Jim
Highsmith



Andrew Hunt



Ron Jeffries



Jon Kern



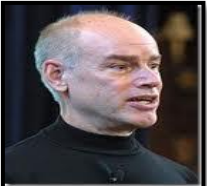
Brian Marick



Robert C. Martin



Steve Mellor



Ken Schwaber

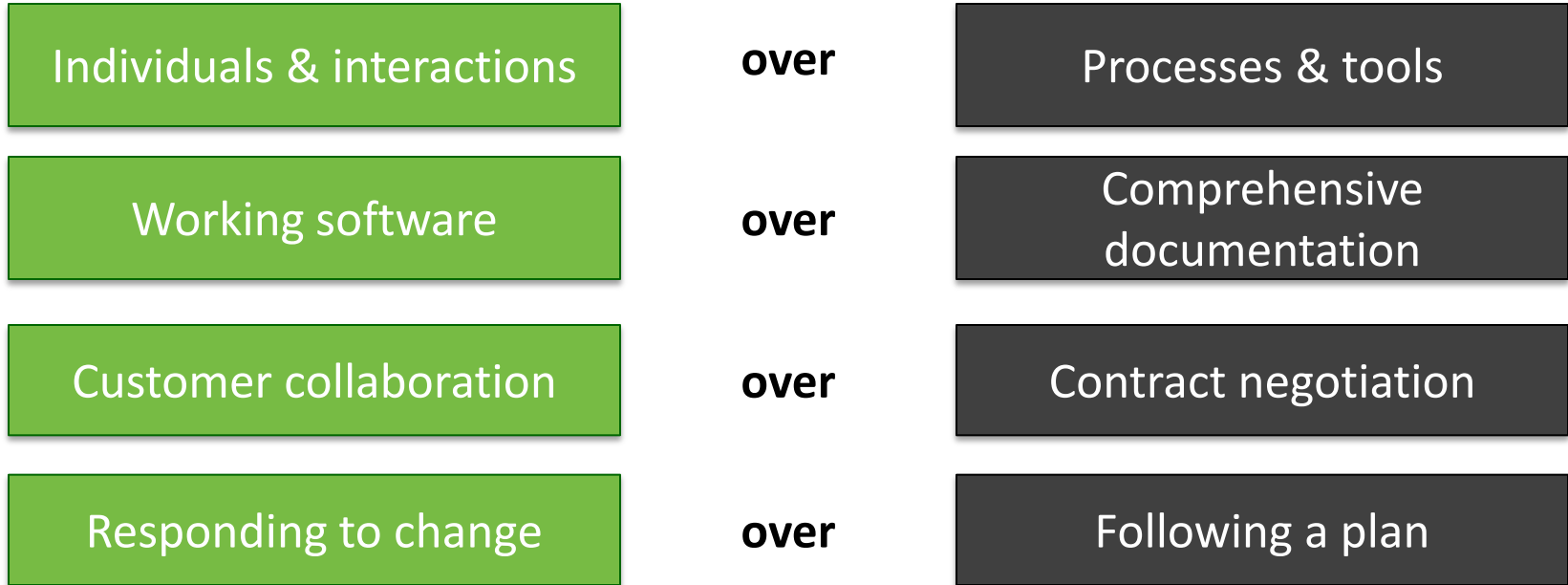


Jeff Sutherland



Dave Thomas

Agile Values



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

Pocket Size Principles



15 Minutes

12 Principles of Agile Manifesto

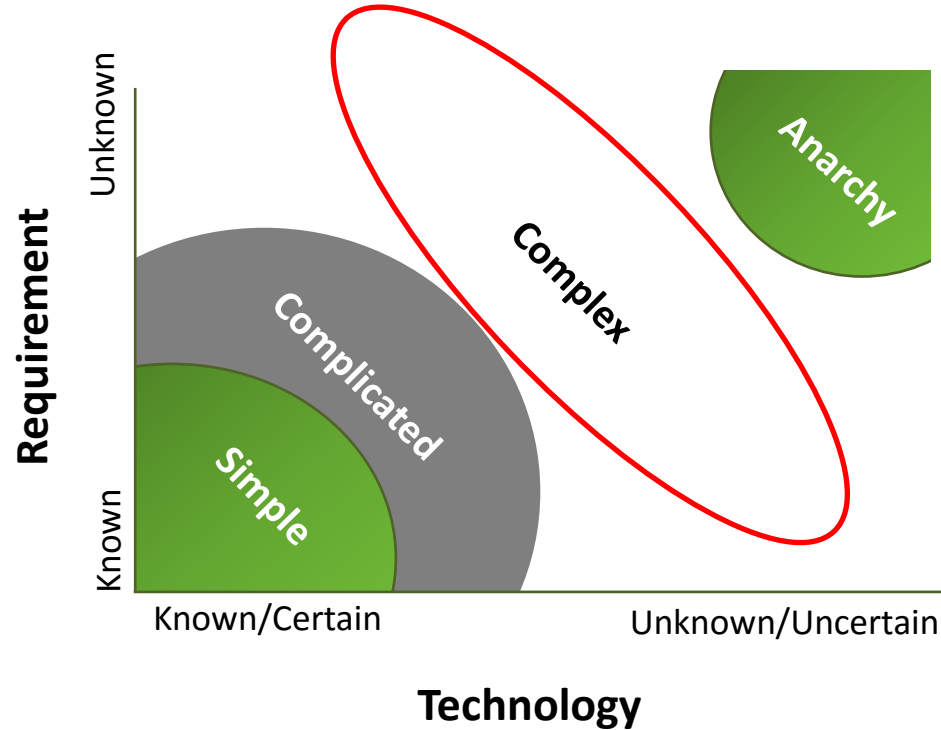
1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage
3. Deliver working software frequently, from a couple of weeks 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

12 Principles of Agile Manifesto (Continued...)

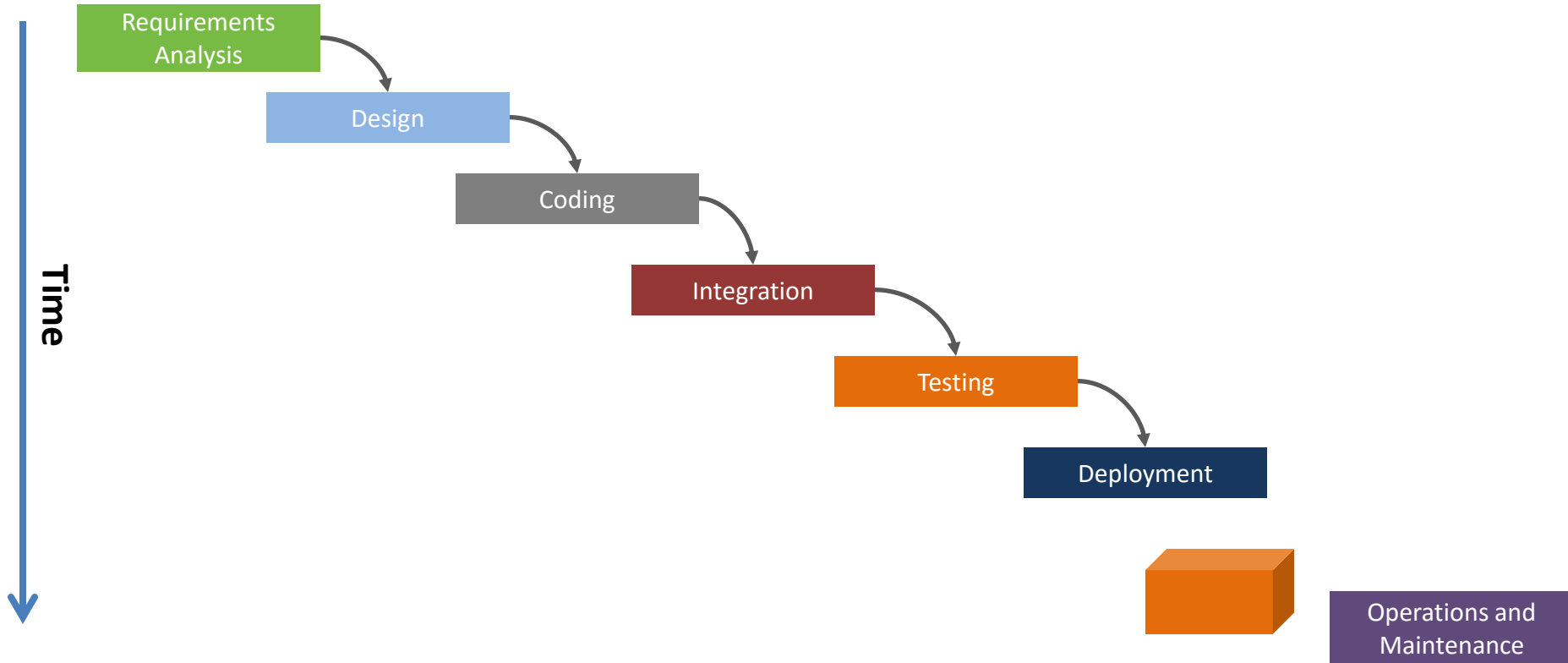
7. Working software is the primary measure of progress
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility
10. Simplicity -the 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

**Select a principle and think
how we can apply in our
traditional projects**

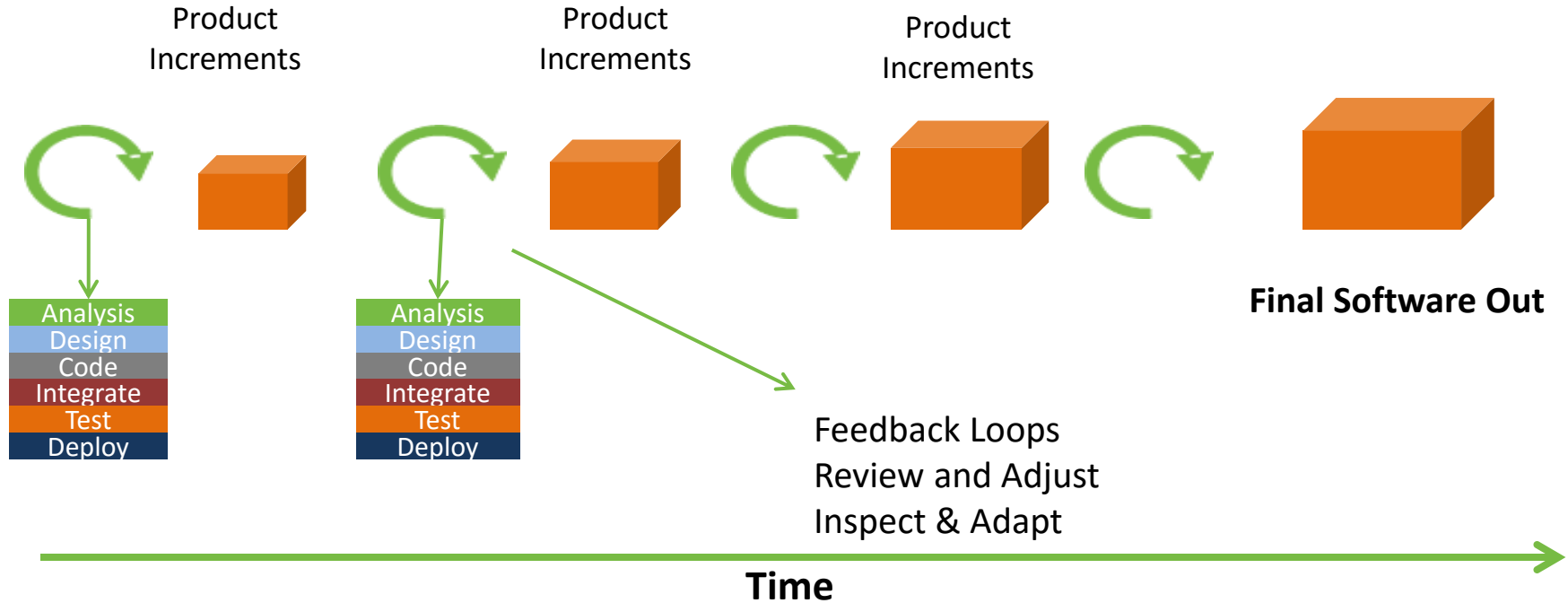
Project Noise and Method Selection



Traditional Project Phases

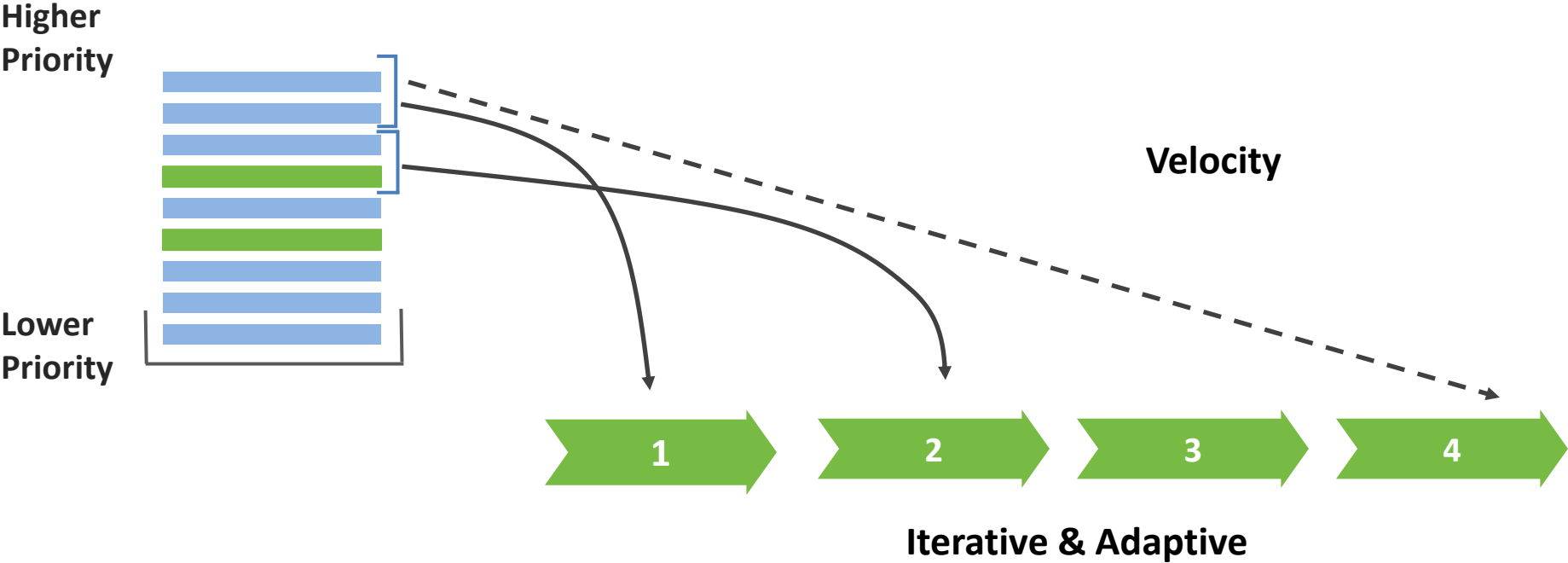


Agile Incremental Delivery



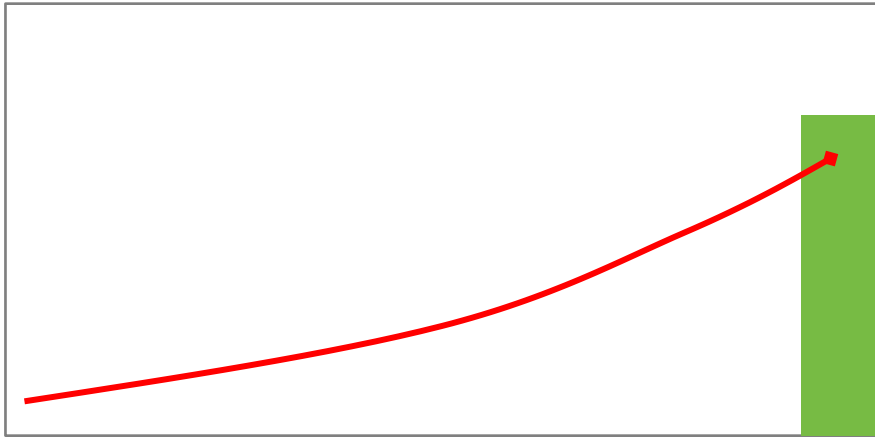
Agile adapts to frequent feedback by delivering working tested code

Agile Planning



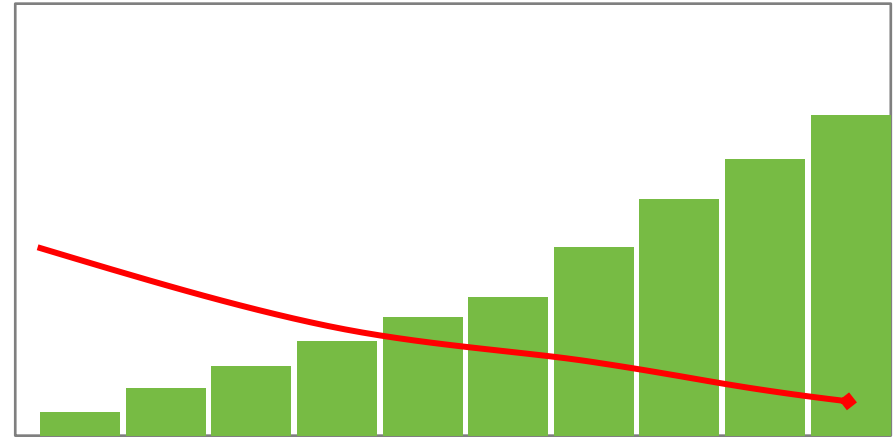
Value Delivery

Traditional

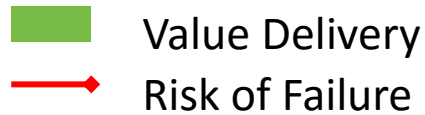


Time

Agile



Time



Sample Project Status Reporting

TRADITIONAL

100% of the system

30% done

No testing yet

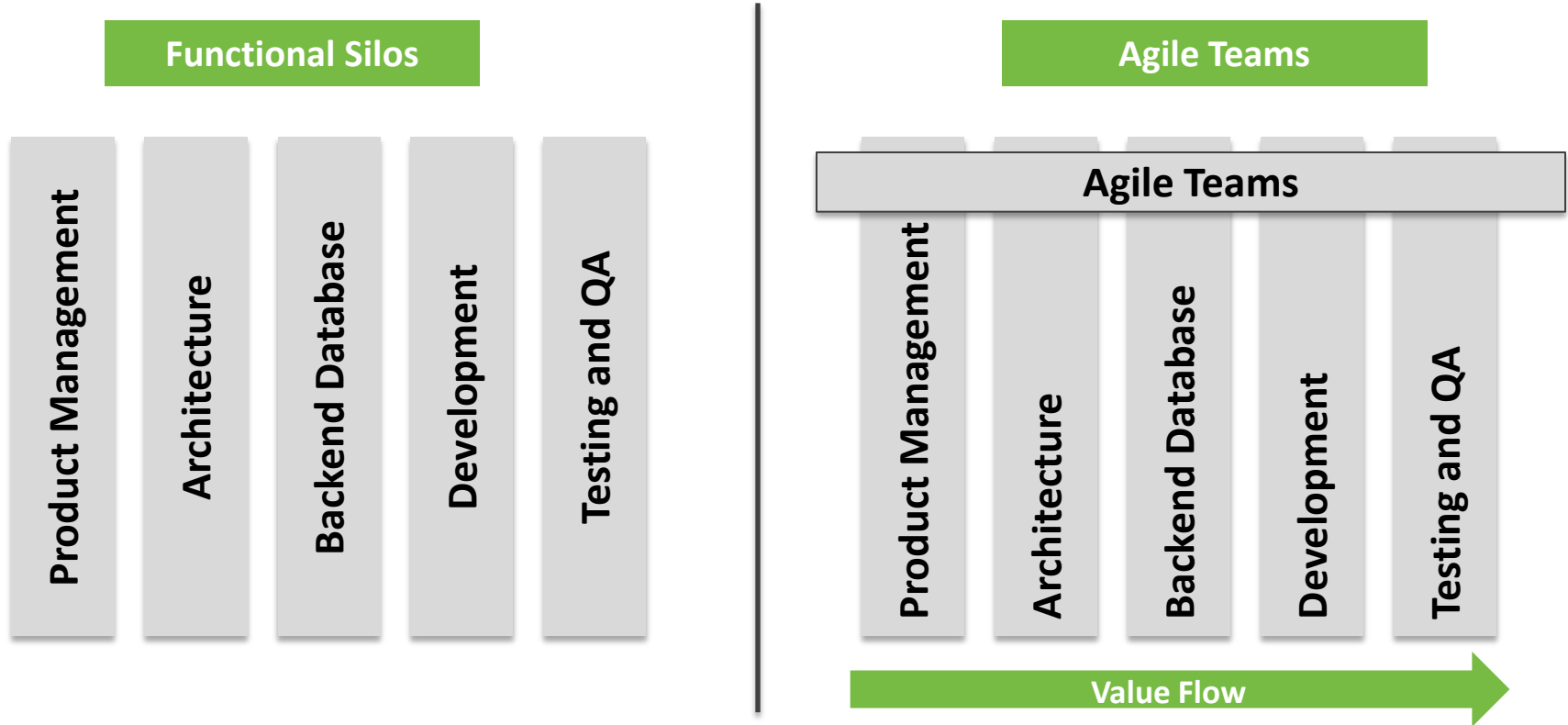
AGILE

30% of the system

100% done

With known quality

Operating Model of Agile Team



Team Collaboration



15 Minutes

Team Collaboration

- Why do we need team collaboration?
- What essential skills require to collaborate?
- Discuss and identify signs of collaboration. How can we improve collaboration in the team?

Transparent

Respect

Listening...seeking to understand

Self-Accountability

Self Organized

Truthfulness

Self-Awareness

Honesty and
Openness

Empowered

Decision
making

Trust

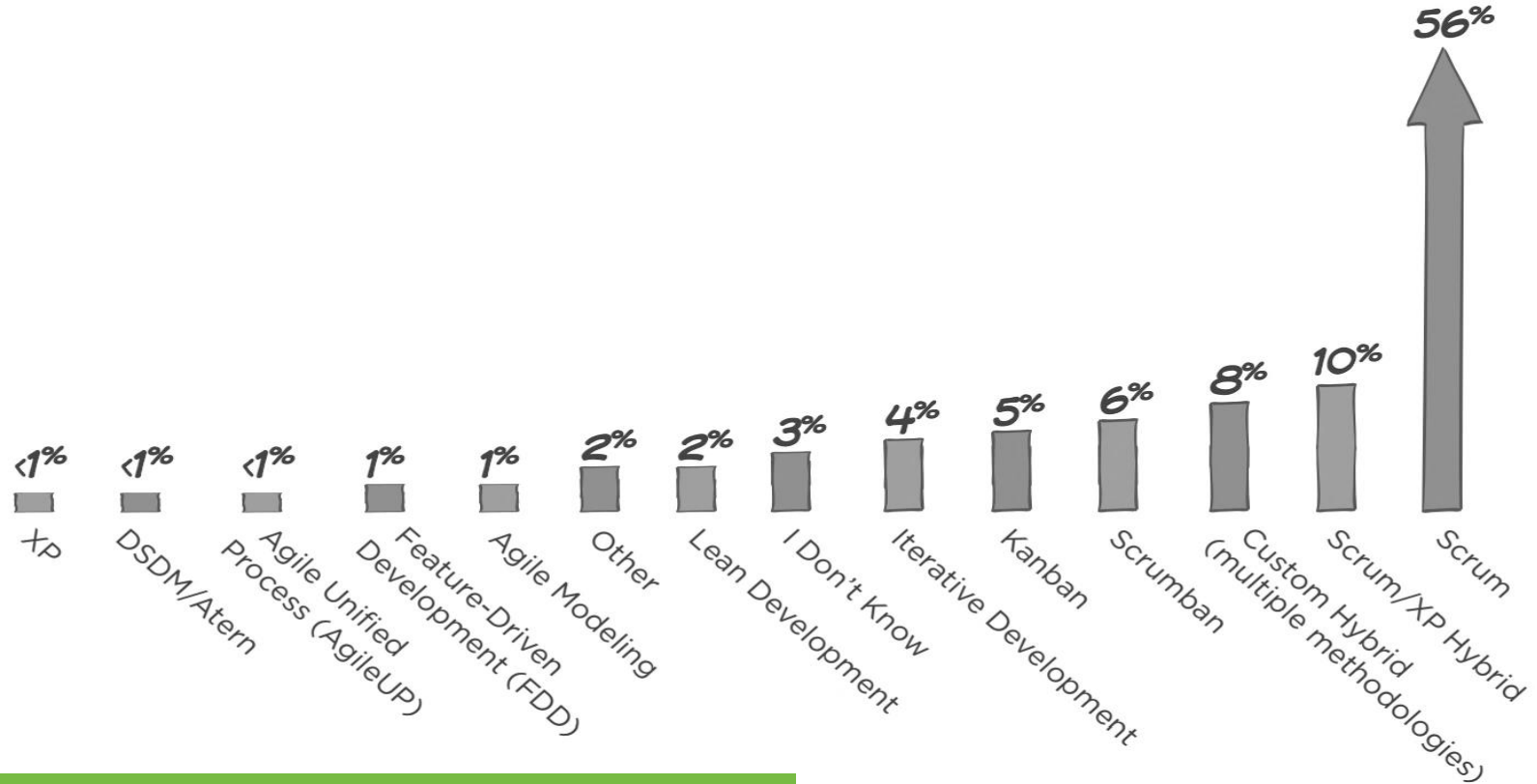
Motivation

**Problem
Solving**

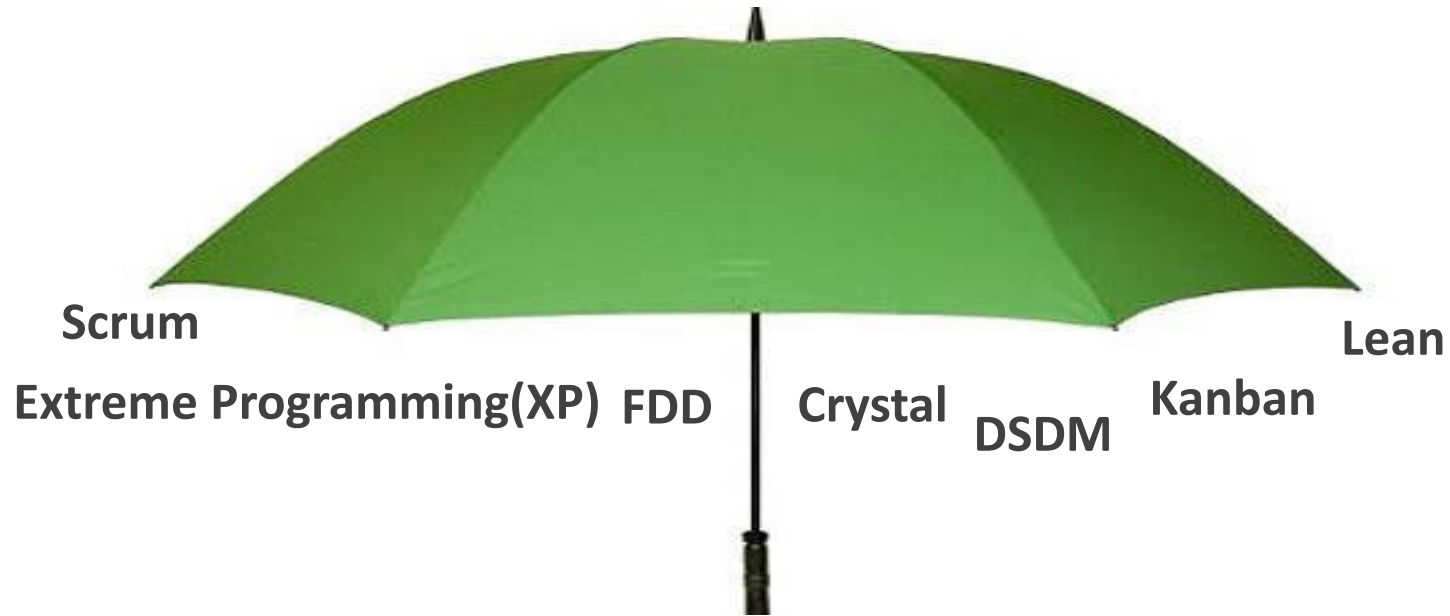
Negotiation

Various Agile Methods

Agile Methodology Used



Agile Umbrella



Prescriptive vs. Adaptive Methods

PRESCRIPTIVE (more rules to follow)

(fewer rules to follow) **ADAPTIVE**



RUP

- 3 roles;
- 20 activities; and,
- Over 70 artifacts.

FDD

- 5 Step Process
- 6 Roles

Extreme Programming (XP)

- Scrum + number of engineering practices e.g. pair programming, test first development

Scrum

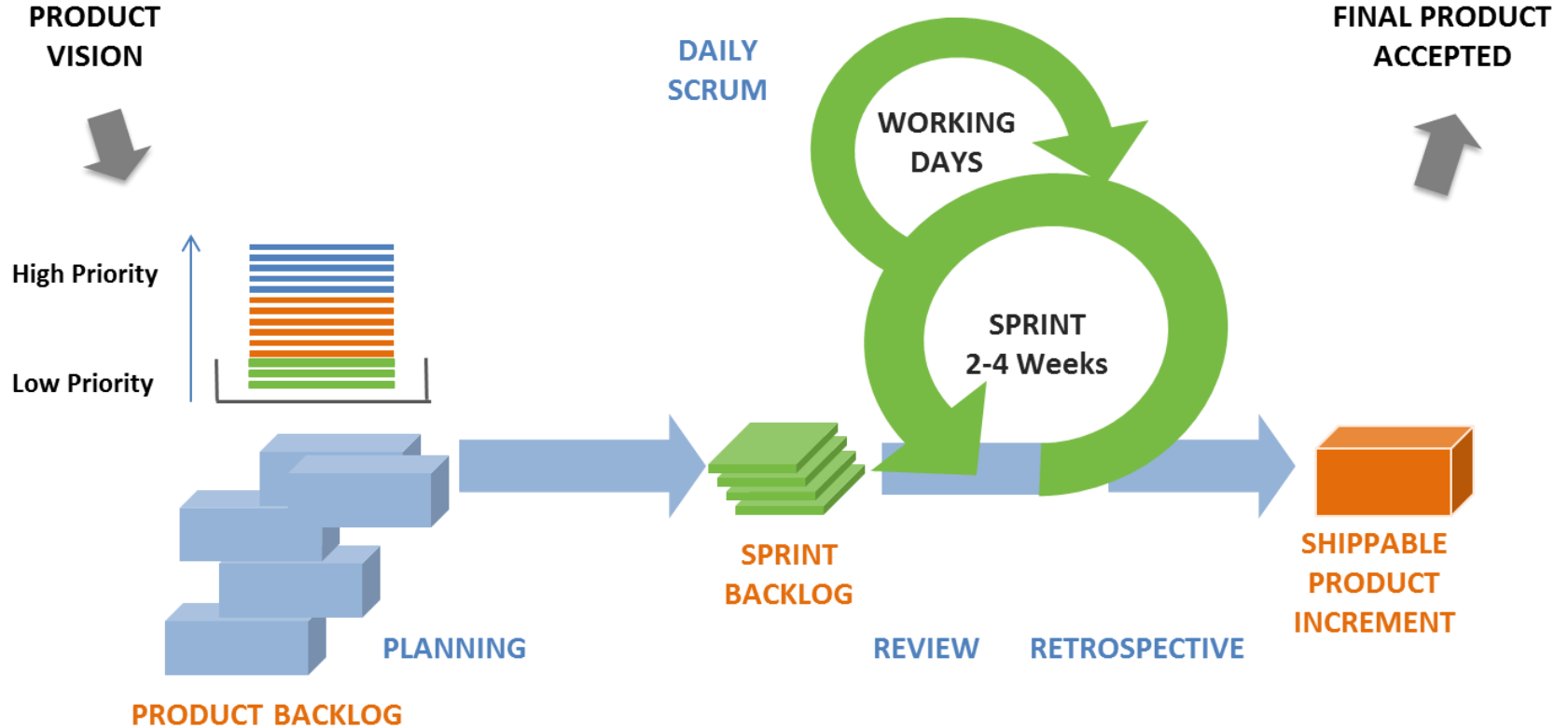
- 3 pillars;
- 3 roles;
- 4 ceremonies;
- 3 artifacts.

Kanban

- Visualize workflow;
- Limit WIP.

Do whatever

Scrum



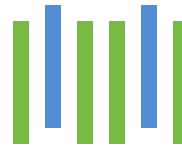
Scrum (Continued...)

ROLES



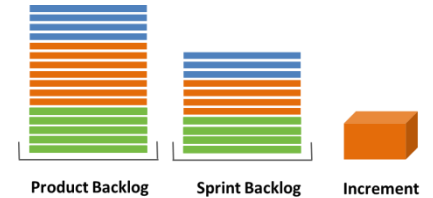
Product Owner, Scrum Master and
Development Team

CEREMONIES



Sprint Planning, Daily Scrum,
review, Backlog grooming /
refinement, release planning and
retrospective

ARTIFACTS



Product Backlog, Sprint Backlog and
Increment

Scrum (Continued...)



- The most common agile method
- Easy to understand and adapt
- Low barrier of entry
- Provide high level mechanics for complex work involving knowledge creation, and collaboration



- No engineering practice defined
- Easy to follow “Scrumbutts” path

Daily Stand-up Simulation

Daily Stand-up / Scrum

- What I did YESTERDAY?
- What I am planning to do it TODAY?
- IMPEDIMENTS – If Any?

Daily Stand-up / Scrum

- The daily stand-up is for and about the team and its commitments. In this meeting, the team checks in on how their work is progressing in the sprint, adjusts plans and gets assistance with removing impediments.
- Every day, same time, same place, same people. This provides a regular rhythm and cadence on everyone's calendar. The meeting last no more than 15 minutes.

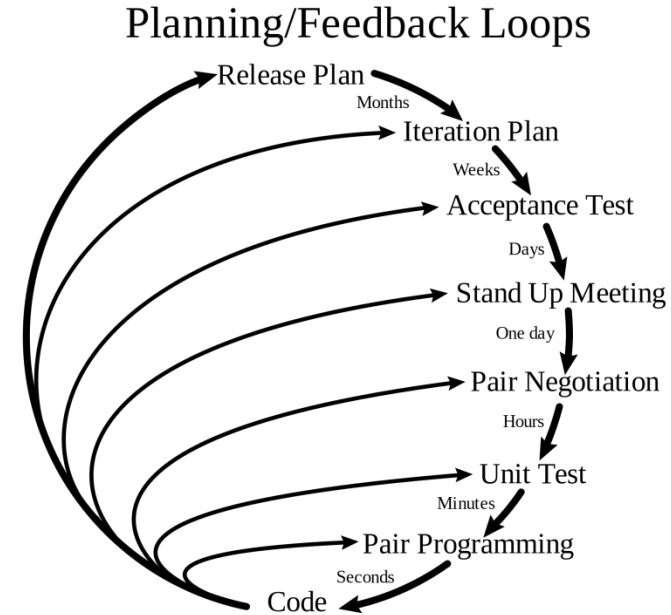
Daily Scrum or Daily Stand-up

Each team member addresses three questions:

1. What has been ~~done~~ *accomplished* since the last meeting?
2. What will be ~~done~~ *committed* before the next meeting?
3. What obstacles/impediments are in the way?

Extreme Programming (XP)

- Developed by Kent Beck when working for Chrysler in 1996.
- Software development-centric Agile method which is intended to improve software quality and responsiveness to changing customer requirements.
- It places a strong emphasis on technical practices in addition to the more common teamwork and structural practices.
- Teams apply appropriate practices in their own context.



Extreme Programming (Continued...)

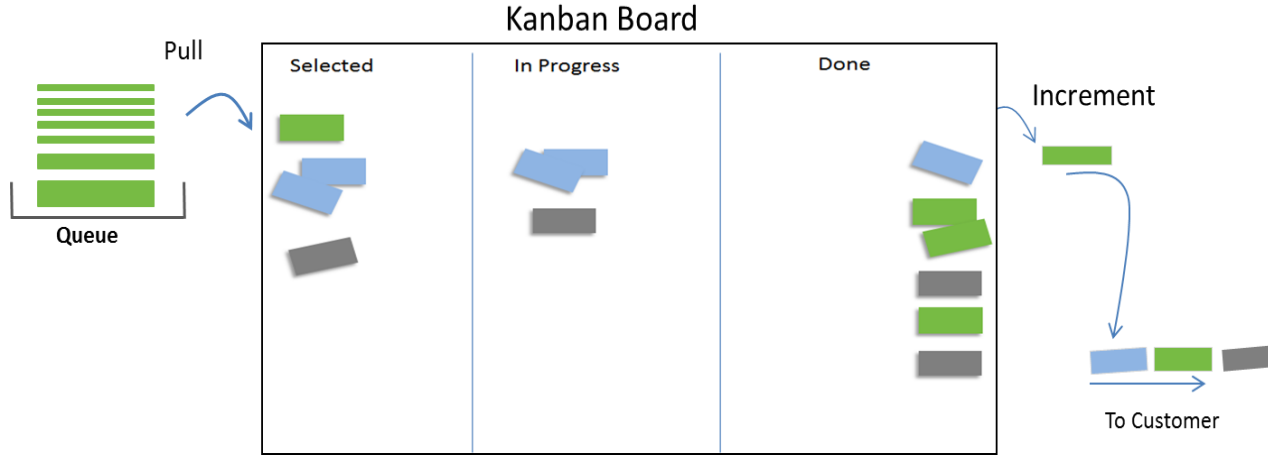


- Sound engineering practices
- First popular agile method
- Quality focused



- Software development focus makes it hard to implement in other business areas

Kanban



- Kanban is pull and flow based system
- Team process rather than individual
- Kanban focuses on how the workflow process can be improved rather than blaming an individual
- True value lies in its requirement that the team creates a workflow with explicit defined rules and limits

Kanban (Continued...)



- Lean method, focus on elimination of waste
- Starts where you are, no major process changes
- Easy to implement



- Less prescriptive
- No time-boxing
- Seems never ending flow of work

Lean Development Principles

- Lean development is a translation of well-know and accepted lean manufacturing practices to the software development domain.
- Mary and Tom Poppendieck identify seven fundamental Lean principles

**Eliminate
Waste**

Build Quality In

Empower Team

**Optimize as
whole**

Delivery fast

**Amplify
learning**

Defer decision

Lean (Continued...)



- Good list of principles
- Focus on elimination of waste
- Value stream mapping



- Difficult co-relate concepts and some practices between Manufacturing and IT

Agile Myths and Facts

Agile recommend incremental & iterative delivery



Agile has defined change management process



Agile recommend big upfront design (BUFD)



Agile teams make their own decisions



**Agile does not recommend any documentation in
the project**



Agile recommends directive teams



Agile measure progress by working software



Agile recommends accepting change during iteration



Agile recommends face to face interaction



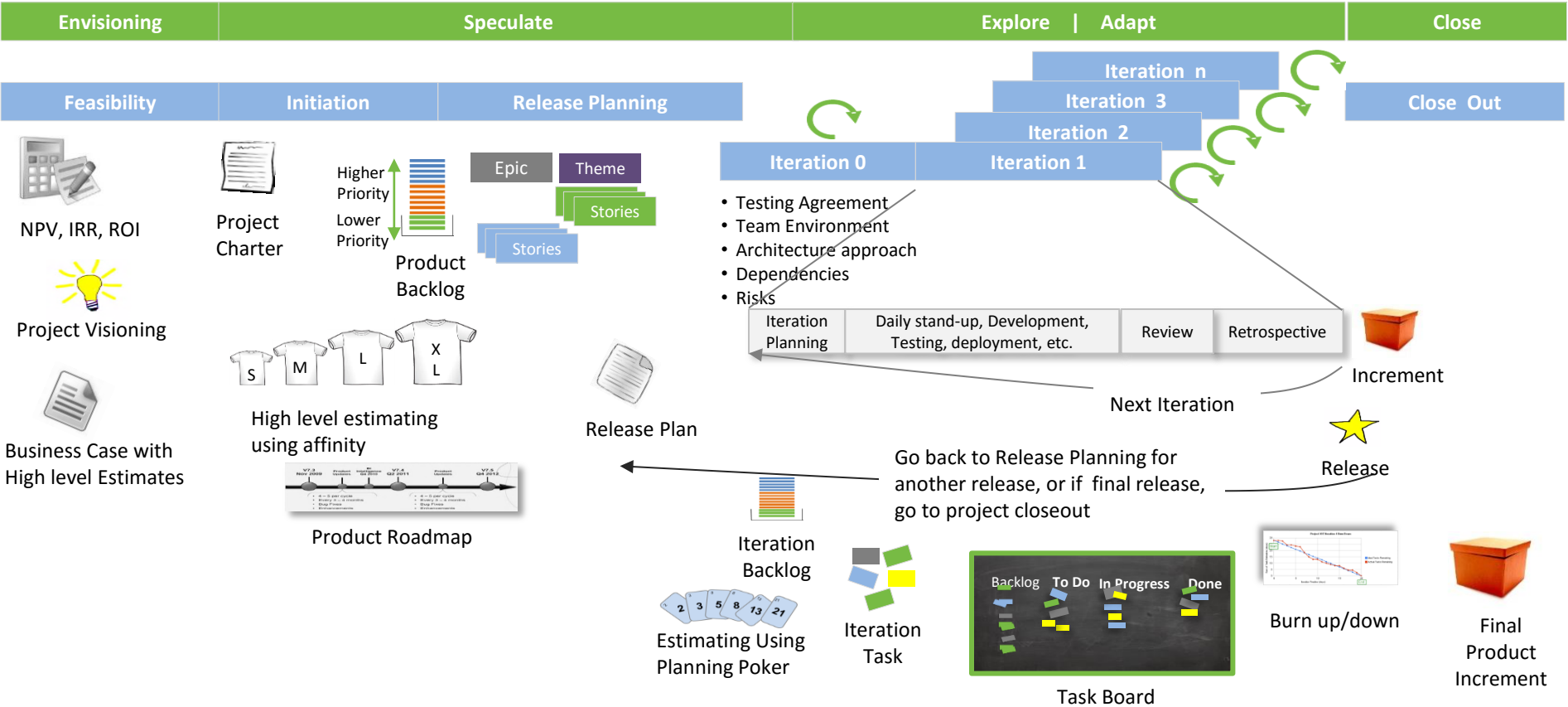
Agile follow ad- hoc process and is anti-planning



Agile base on empirical process - frequent inspect and adapt cycles



High Level Process Map



What's Different?

Traditional	Agile
Defined process: Control and Coordinated	Empirical process: Inspect and Adapt
Work is organize around the team	Team organize around work
Work is assigned or push to the team	Work is store in queue and team pull the tasks
Plan all in advance	Plan as you go
Work breakdown structure	Feature breakdown structure
Functional specs	User stories
Gantt chart	Release plan
Status report	Information radiators/deliver as you go
Learn at the end	Learn every iteration
Follow the plan	Adapt everything
Manage task	Manage team
Conventional project team	Self-organized project teams
Avoid change	Embrace change
Prescriptive	Adaptive

Compare Agile & Waterfall



15 Minutes

Waterfall vs. Agile

- Waterfall is more efficient than Agile because...
 1.
 2.
 3.
- Agile is more efficient than waterfall because...
 1.
 2.
 3.

Any Aha Moments to Share?

Q & A

Download this presentation

<http://www.360pmo.com/?p=2538>

Agile Boot Camp & ACP Exam Preparation

November 14, 21 and 28, 2015

<http://www.360pmo.com/?p=2271>

15 % Discount Code: **PMILS15**

* limited to first 10 registrations

Following references are used in the preparation of this workshop:

1. Abstracted from Shuh, Peter (2005). Integrating Agile Development in the Real World
2. How sustainable is your agile transformation to sustainable organizational agility, Ahmed Sidky
3. Strategic Management and Organizational Dynamics by Ralph Stacey in Agile Software Development with Scrum by Ken Schwaber and Mike Beedle
4. 7th Annual state of Agile versionone® Agile made easier development survey
5. Process Miniature; <http://c2.com/cgi/wiki?ProcessMiniature>
6. The Agile impact report, Rallydev
7. Iterative and incremental development by Craig Larman, victor Basili
8. What's the big fuss about Agile? by Ahmed Sidky
9. Manifesto for Agile Software Development & Principles behind the Agile Manifesto
<http://www.agilemanifesto.org/>
10. "Explain, Explore", Luke Lackrone, coaching a new team

All registered and unregistered trademarks (service marks, brands, icons, copyrights etc.) mentioned on this presentation are the property of their respective owners.

Content that references these trademarks is not sponsored by, endorsed by, or affiliated with the respective trademark owners. PMI-ACP®, PMBOK®, PMI®, and PMP® are either marks or registered marks of the Project Management Institute, Inc.



360PMO

www.360pmo.com