



Agile- Overview

April 2017

Module Objectives

Note to the SME : Please provide the module Objectives or validate the partially updated content



What you will learn

At the end of this module, you will learn:

- The Overview of Agile and Scrum

What you will be able to do

At the end of this module, you be able to:

- Understand the need for Agile
- Describe the differences between Waterfall and Agile
- State the Agile Manifesto
- List the Agile Principles



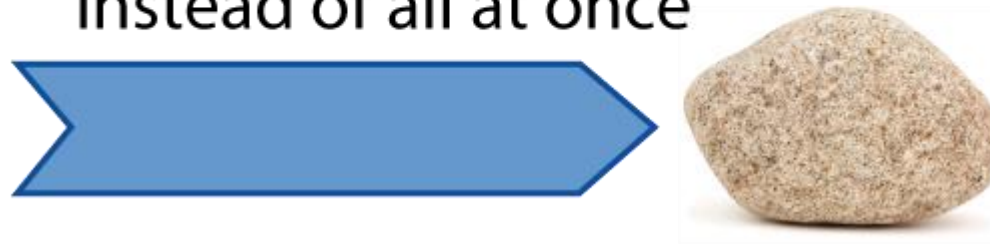
What is Agile

- Agile is a time boxed, iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver it all at once near the end.

incrementally



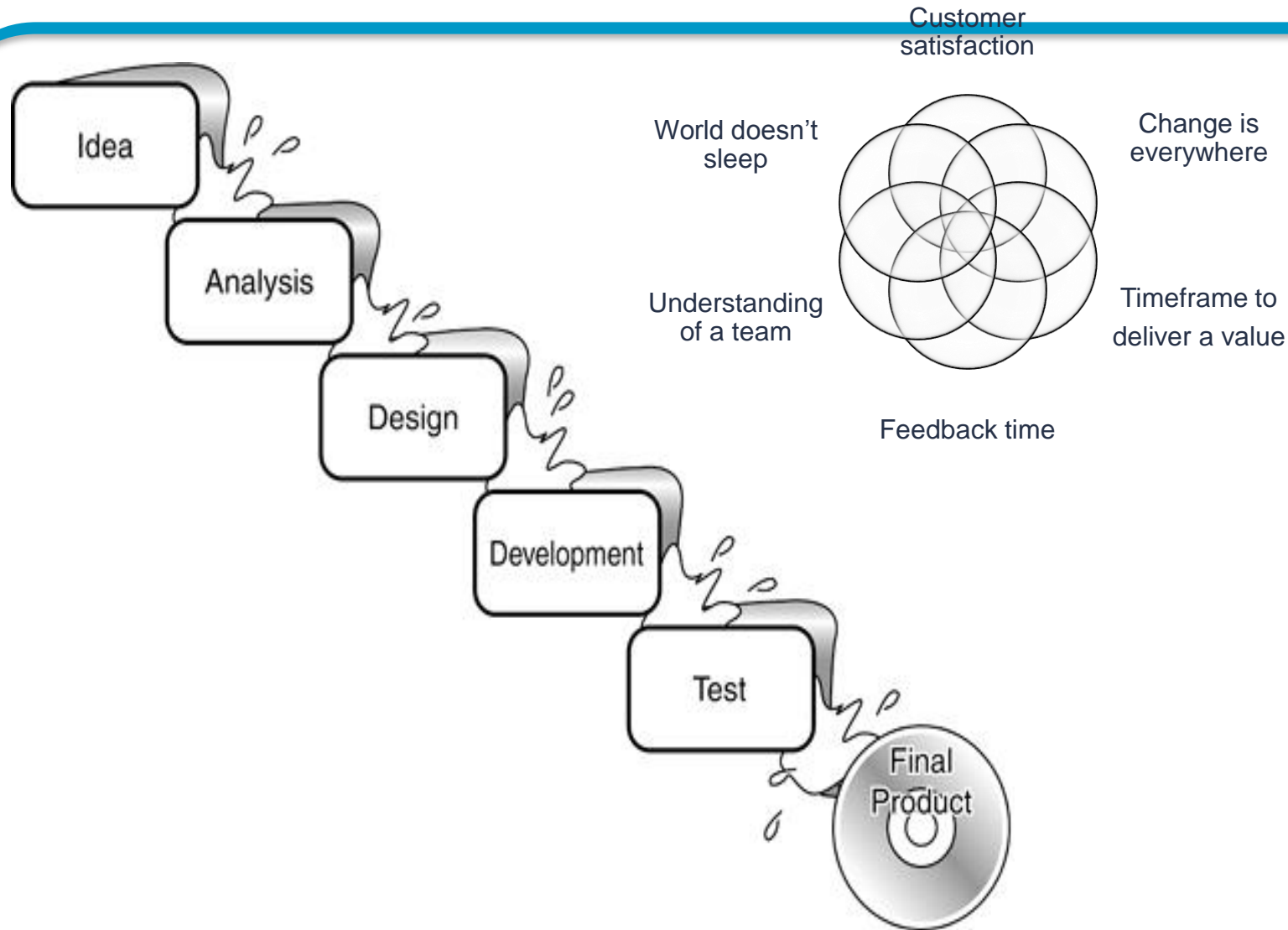
instead of all at once



Contd.,

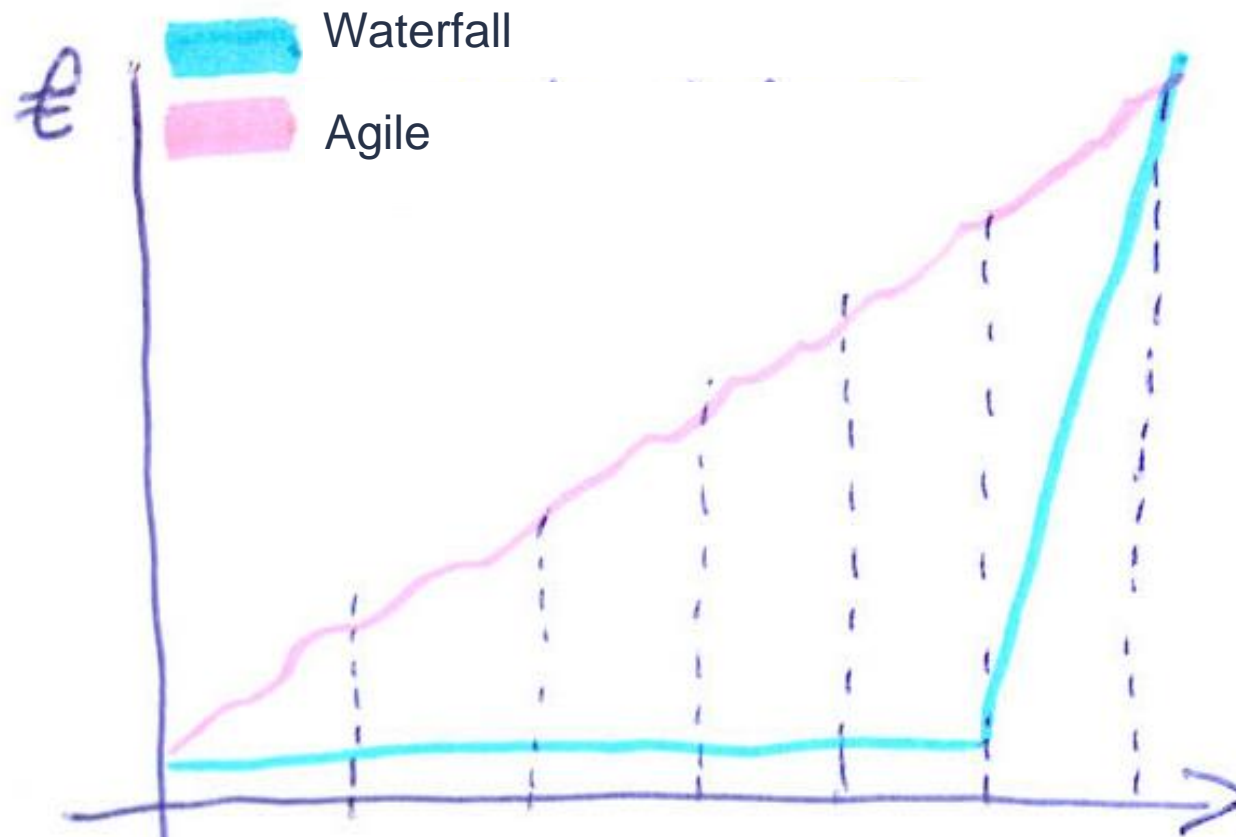
- Agile emphasizes iterative and incremental development, close collaboration with the customer, and cross-functional teams to build working software.
- **Philosophy:** People, Collaboration and Values

Waterfall Challenges



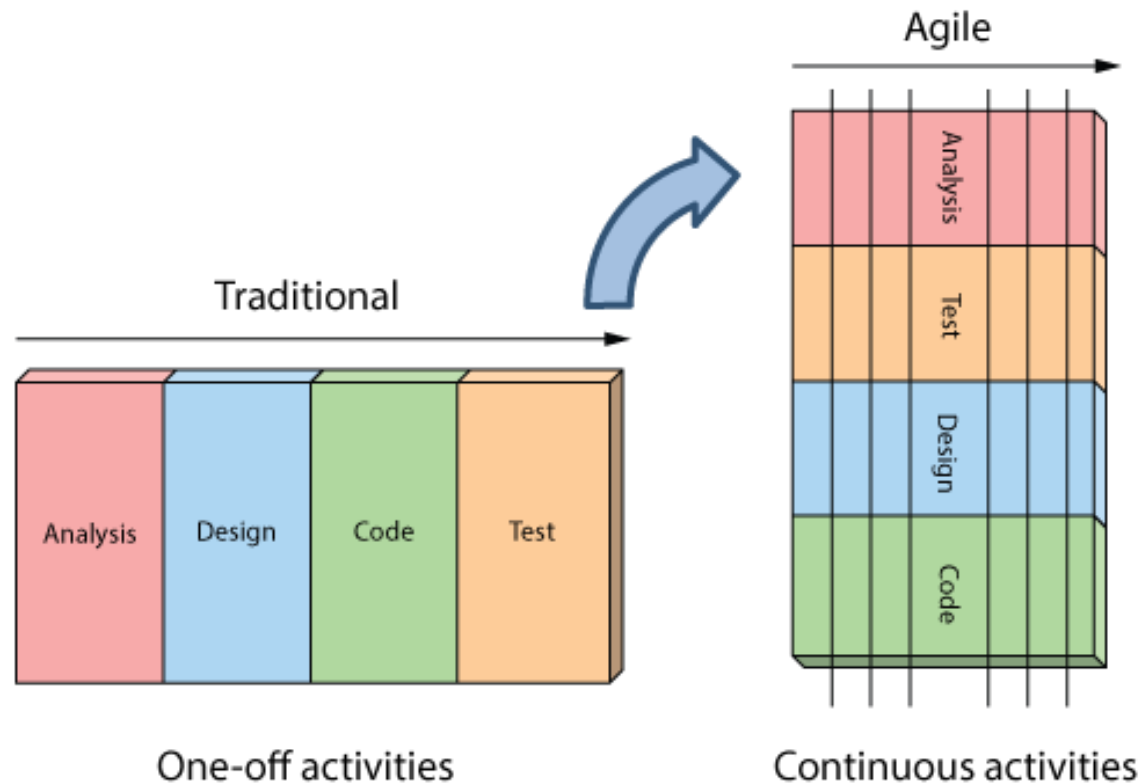
Check & Deliver

A value in Agile is delivered **during** the development, not just at the end



How is Agile Different ?

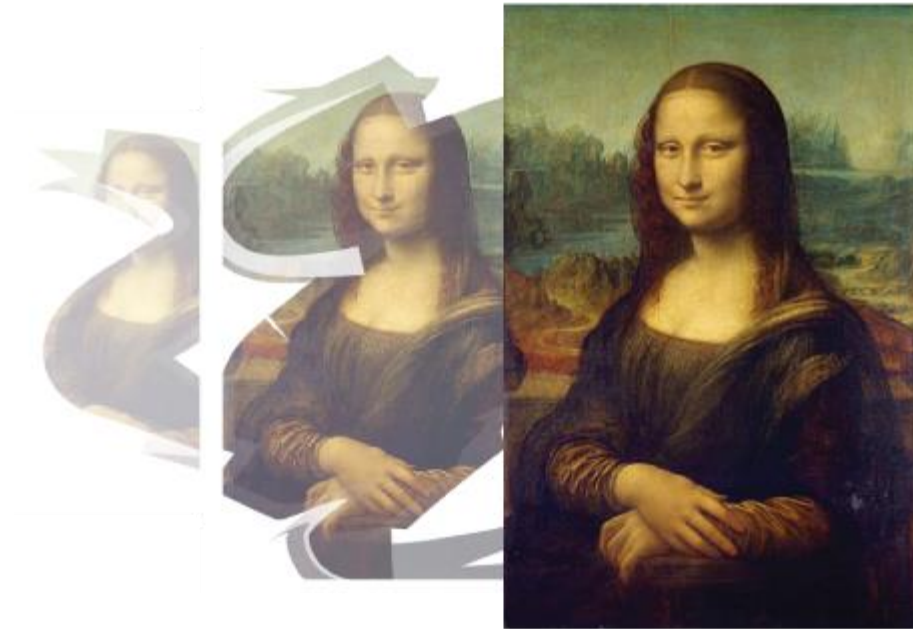
- Analysis, design, coding, and testing are continuous activities



Contd.,

- Development is iterative

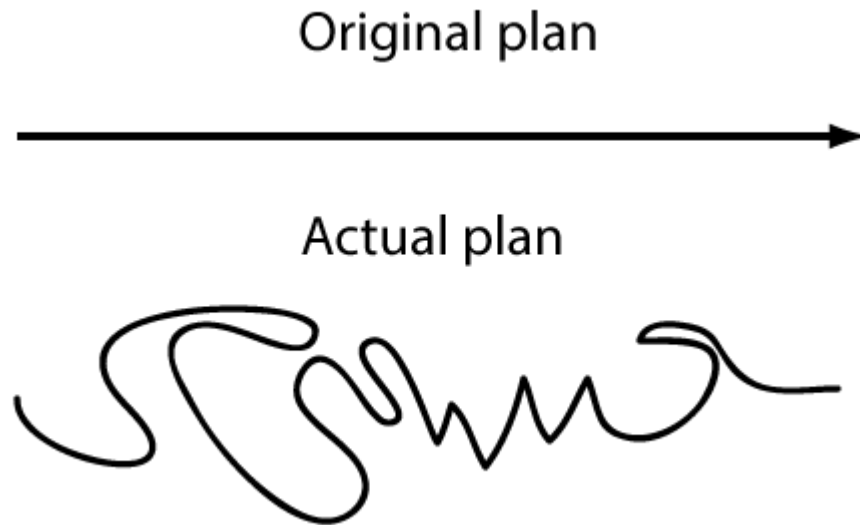
- Iterative development means starting with something really simple, and adding to it incrementally over time.



Contd.,

- Planning Is adaptive

- When reality disagrees with their plans, Agilists find it easier to change their plans than reality. They call this adaptive planning.



What is the Primary Measure of Success ?

- Working software is the primary measure of success

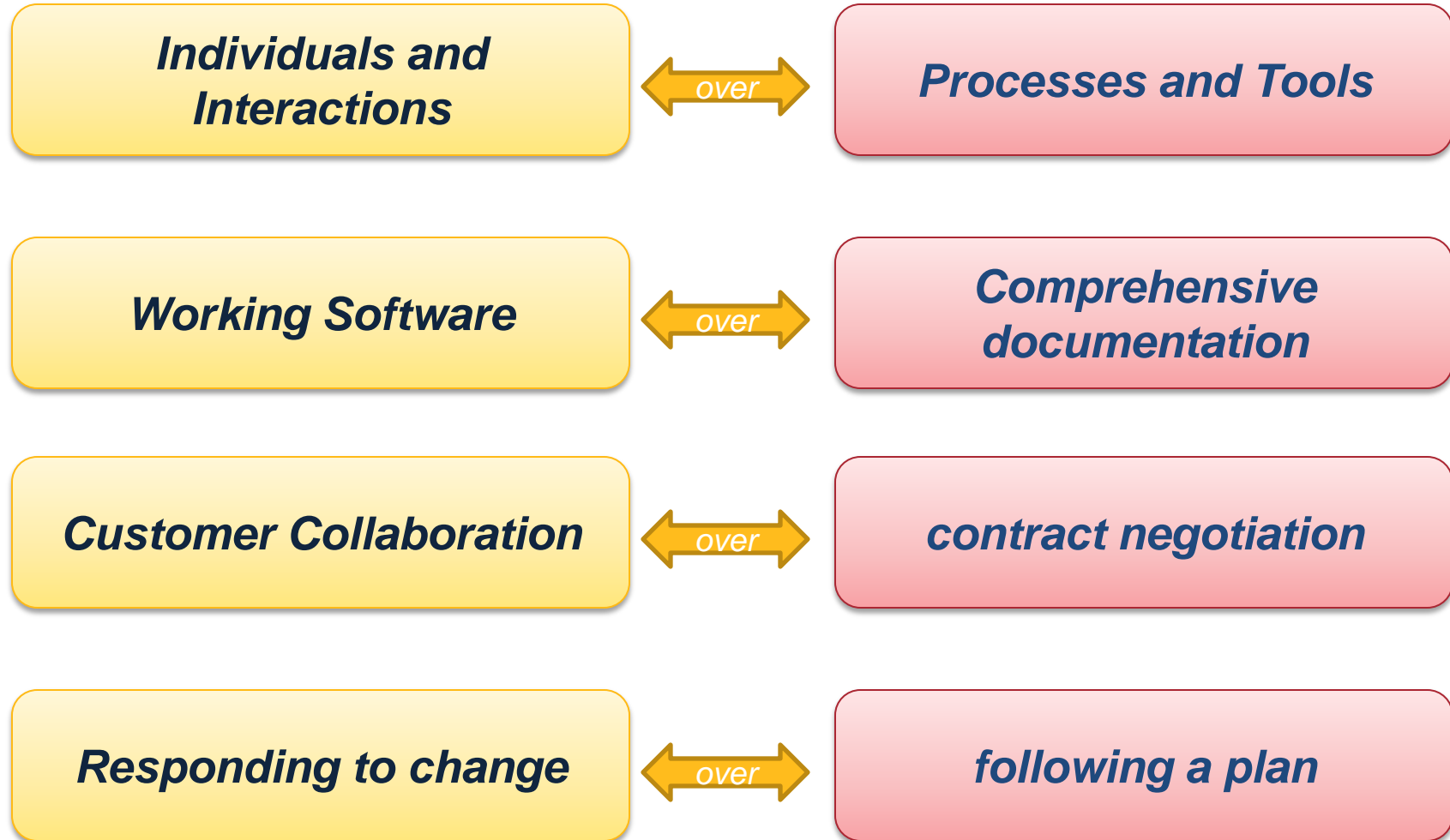


(working software)

OVER



Manifesto for Agile











While there is value in the items on the right, We value the items on the left more.

Agile Manifesto

Manifesto for Agile Software Development*

"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 & interactions	Over	 Processes & tools	 Working software	Over	 Comprehensive documentation
 Customer collaboration	Over	 Contract negotiation	 Responding to change	Over	 Following a plan

Knowledge TRAIN

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

Copyright © 2016 Knowledge Train Limited, www.knowledgetrain.co.uk. *Quoted from www.agilemanifesto.org

Value 1: Individuals and Interactions over Processes and Tools

- **Strong players:** a must, but can fail if don't work together.
- **Strong player:** not necessarily an 'ace;' work well with others!
 - Communication and interacting is **more important** than raw talent.
- **'Right' tools** are vital to smooth functioning of a team.

Value 1: Individuals and Interactions over Processes and Tools

- **Start small.** Find a free tool and use until you can demo you've outgrown it. Don't assume bigger is better. Start with white board; flat files before going to a huge database.
- **Building a team** more important than **building environment.**
 - Some managers build the environment and expect the team to fall together.
 - Doesn't work.
 - Let the team build the environment on the **basis of need.**

Value 2: Working Software over Comprehensive Documentation

- **Examples**

- Samsung.
- RedBus

- **How to train newbees** if short & sweet?

- Work closely with them.
- Transfer knowledge by sitting with them; make part of team via close training and interaction

Value 3: Customer Collaboration over Contract Negotiation (1 of 2)

- Not possible to describe software requirements up front and leave someone else to develop it within cost and on time.
- Customers cannot just cite needs and go away
- Successful projects require **customer feedback on a regular and frequent basis** – and not dependent upon a contract .

Value 3: Customer Collaboration over Contract Negotiation (2 of 2)

- **Best contracts are NOT** those specifying requirements, schedule and cost.
 - Become meaningless shortly.
- Far better are **contracts that govern the way the development team and customer will work together.**
- Key is intense collaboration with customer and a contract that governed collaboration rather than details of scope and schedule
 - Details ideally **not** specified in contract.
 - Rather contracts could pay when a block passed customer's acceptance tests.
 - With frequent deliverables and feedback, acceptance tests never an issue.

Value 4: Responding to Change over Following a Plan

- Our plans and the ability to respond to changes is critical!
- Better planning strategy – make detailed plans for the next few weeks, very rough plans for the next few months, and extremely crude plans beyond that.
- **Only invest in a detailed plan for immediate tasks;** once plan is made, difficult to change due to momentum and commitment.
 - But rest of plan remains flexible. The lower resolution parts of the plan can be changed with relative ease.

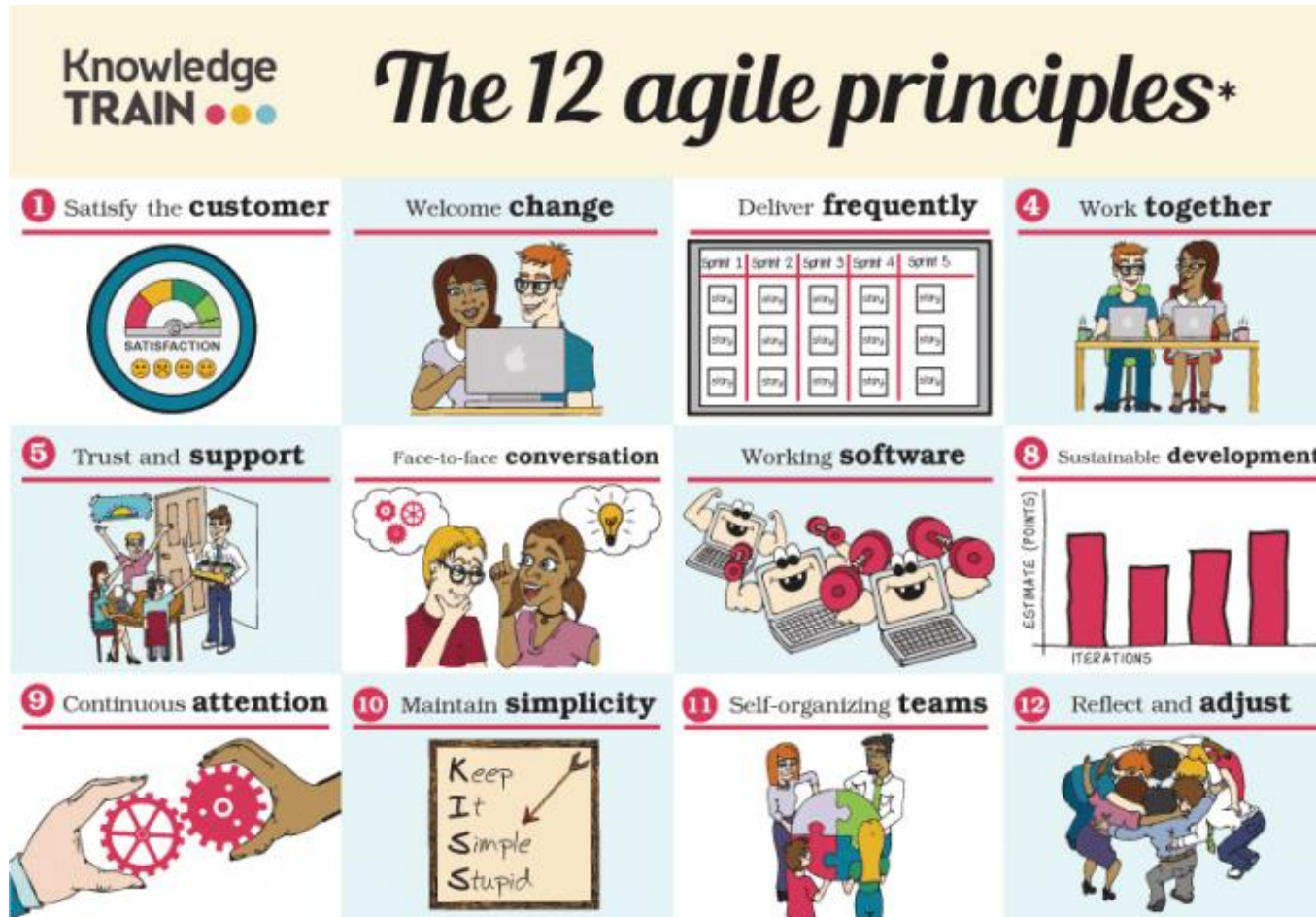
Agile Principles

1. Our highest priority is to satisfy customer through early and continuous delivery of valuable software
2. Welcome changing requirements, even in late development.
3. Working software is delivered frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, give them environment and support their needs, and trust them to get the job done.
6. Face-to-face conversation is the best form of communication (co-location)

Agile Principles

7. Working software is the principal measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work not done—is essential
11. Best architectures, requirements, and designs emerge from self-organizing teams
12. Regularly, the team reflects on how to become more effective, and adjusts accordingly

Agile – 12 Principles



Copyright © 2016 Knowledge Train Limited. www.knowledgetrain.co.uk. *Clarified from www.scrumguidelines.org

Agile Process – Characteristics

1. Incremental and Iterative delivery:

- To test assumptions and fail faster

2. Frequent Customer Feedback

- To improve the quality of product

3. Flow and Emergent design

- To improve quality of product and ability of the team to respond to changes

Agile Process – Characteristics

4. Evolving Process

- To improve quality of Product and People

5. Continuous planning

- To improve predictability

6. Autonomous Team

- To improve self-organization

7. Cross-functional Team

- To improve collaboration

Why do companies adapt Agile?

- Accelerate Time to Market
- Manage Changing Priorities
- Increase Productivity
- Better Align IT/Business
- Enhance Software Quality
- Improve Project Visibility