1. Introduction to Agile Software Development

IT 4406 – Agile Software Development

Level II - Semester 4

Intended Learning Outcomes

- At the end of this lesson, you will be able to;
  - Identify the significance of meeting deadline for organizational success.
  - Explain how Agility becomes a successful way.
  - Understand the Principals behind the Agile Manifesto.

List of sub topics

1.1 Rational of Agile

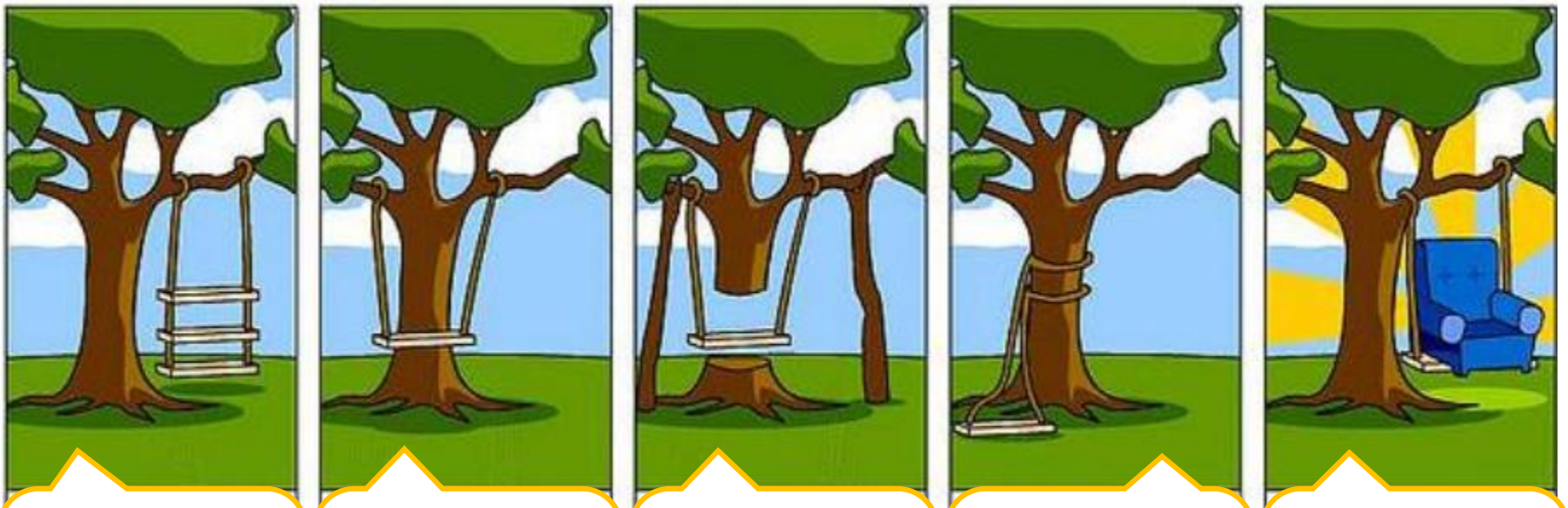
1.2 How to use Agile

1.3 Agile Manifesto

1.4 Scrum, Lean, Kanban, Extreme Programming

1.1 Rational of Agile

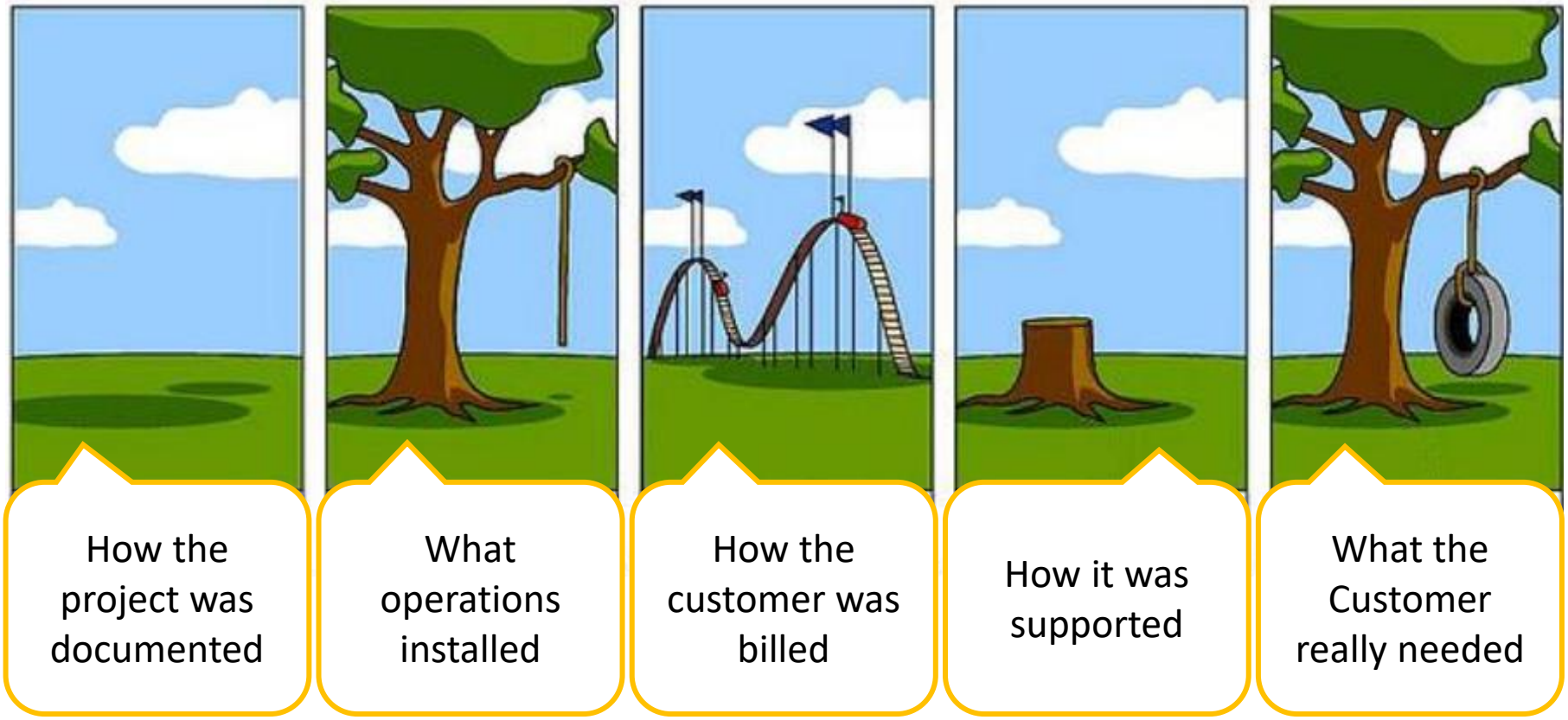
- The problem…



https://blogs.perficient.com/2011/07/22/how-to-build-a-tire-swing-a-case-for-agile-development/

1.1 Rational of Agile

- The problem…



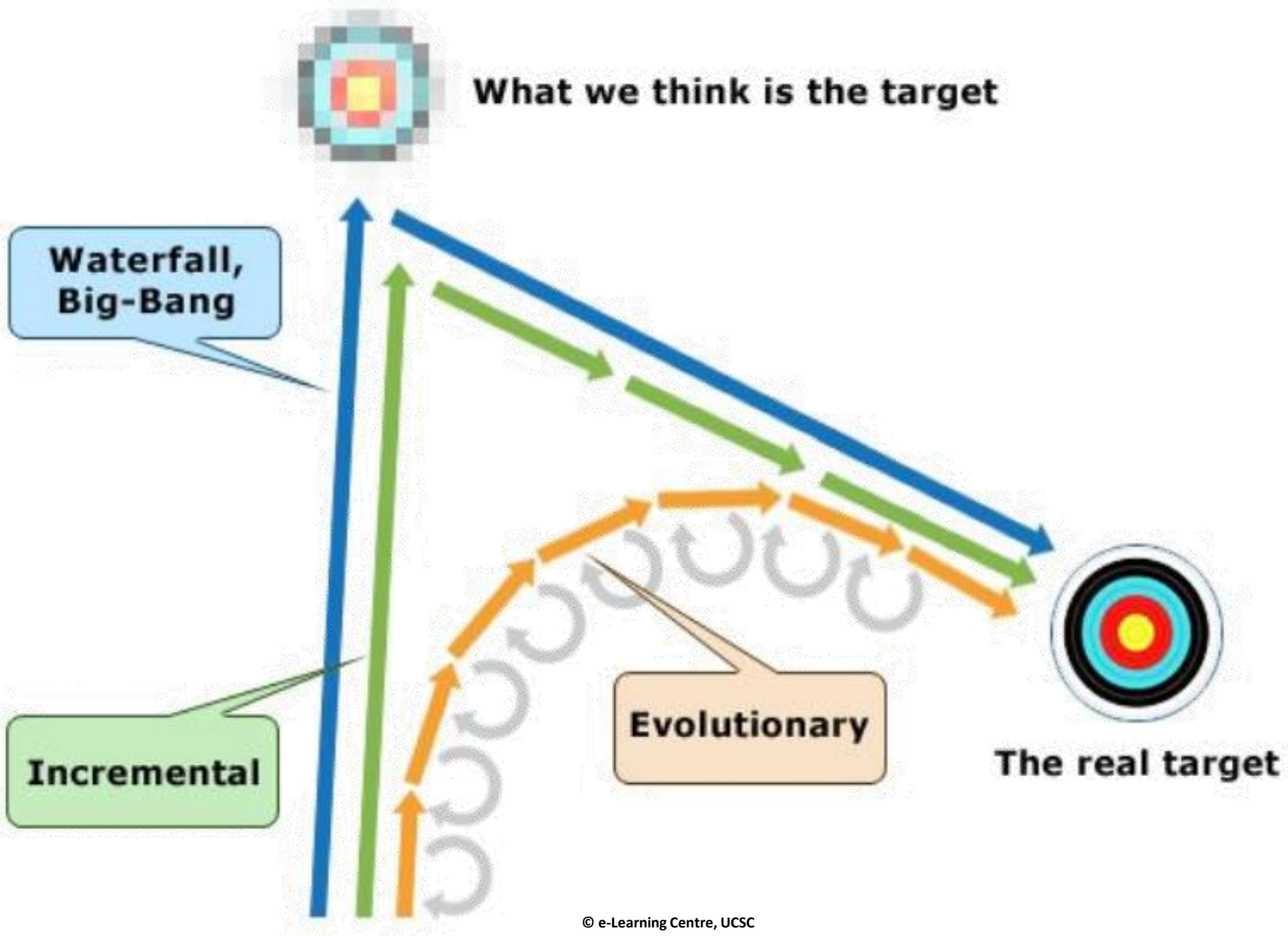
https://blogs.perficient.com/2011/07/22/how-to-build-a-tire-swing-a-case-for-agile-development/

Here is a Worrying Stat…

- An average of 70% software development projects **fail**!!
- Reasons for FAILURE?
  - Not meeting the timelines
  - Costs overrun
  - Customers have NOT got what they asked for

https://www.zdnet.com/article/study-68-percent-of-it-projects-fail/

Missing the target



Agility

- In general, **agility** is defined as *"the ability to both create and respond to change in order to profit in a turbulent environment"*
  - Changes in Requirements
  - Changes in Design, Implementation
  - Changes in Technology
  - Changes in Team
  - Changes in users/client contacts

https://www.informit.com/articles/article.aspx?p=25980

What is Agile?

- Philosophy + a set of Guidelines for software development
- P:
  - Customer satisfaction
  - Early incremental delivery
  - Small, highly motivated project teams
  - Overall development simplicity
- G:
  - Active continuous communication & collaboration between Developers and users

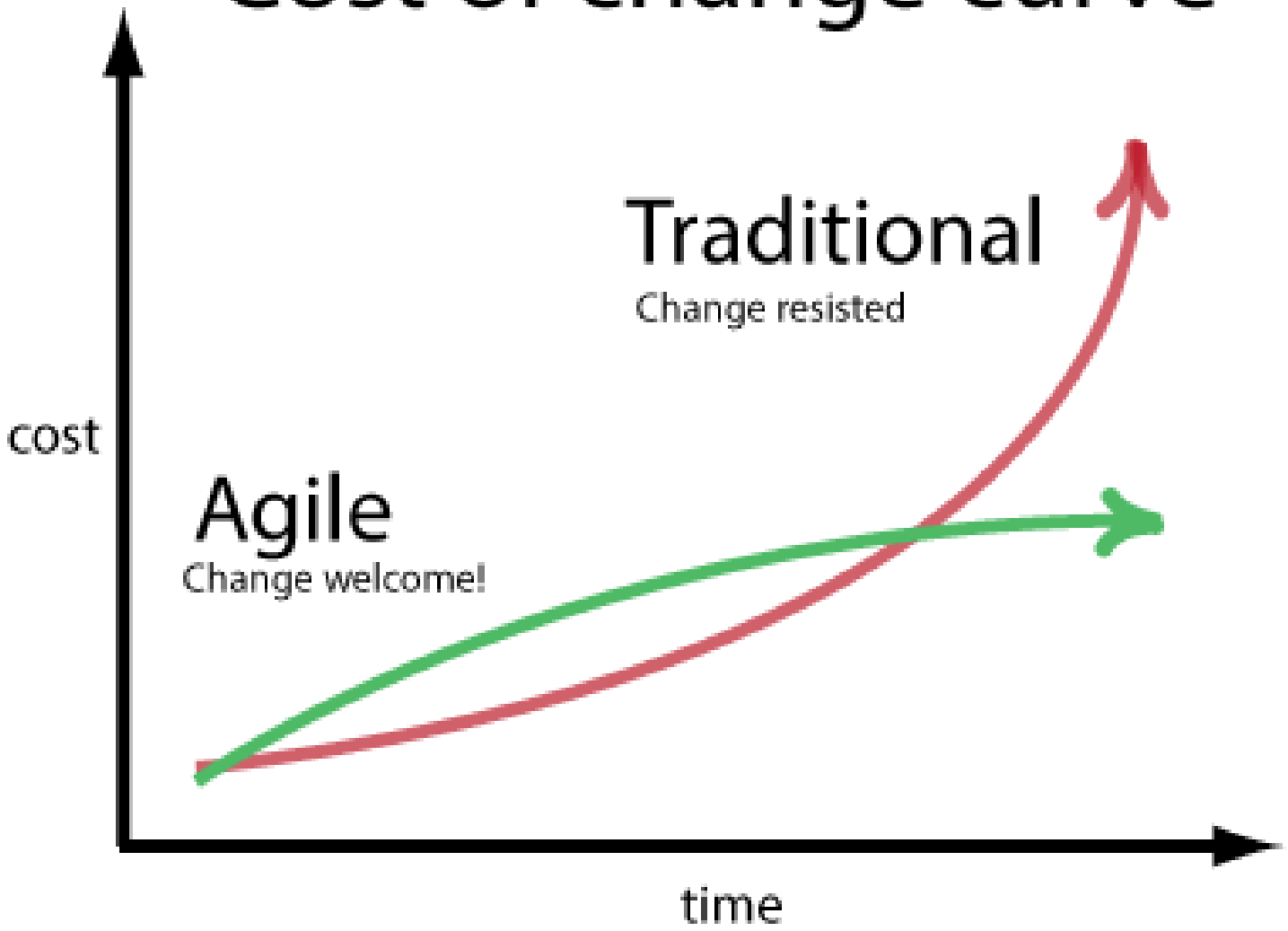
So why Agile?

- Customer satisfaction by rapid delivery of useful software
- Welcome changing requirements, even late in development
- Working software is delivered frequently (weeks rather than months)
- Working software is the principal measure of progress
- Close, daily co-operation between business people and developers
- Face-to-face conversation is the best form of communication (co-location)

So why Agile? CONT...

- Projects are built around motivated individuals, who should be trusted
- Continuous attention to technical excellence and good design
- Simplicity
- Self-organizing teams
- Regular adaptation to changing circumstance

Cost of change curve



http://www.agilenutshell.com/how_is_it_different

AGILE PRINCIPLES – in a nutshell

- Eliminate Waste
- Build Quality In
- Deliver Fast
- Improve the system
- Defer Commitment
- Respect People
- Create Knowledge

Evolution of Agile

- SCRUM – Jeff Sutherland, Ken Schwaber
- Xbreed - Mike Beedle
- DSDM - Arie van Bennekum
- XP – Kent Beck, Ward Cunningham, Ron Jeffries , Robert C. Martin
- Martin Fowler (Thoughtworks)
- FDD – Peter Coad, Jon Kern
- Testing - Brian Marick
- Adaptive Software development – Jim Highsmith
- Crystal Family – Alistair Cockburn
- Pragmatic programmers - Andrew Hunt , Dave Thomas

Agile methods

- The Agile movement proposes alternatives to traditional project management.
  - "A method, or process, is a way of working. Whenever you do something, you're following a process."
- Agile methods are processes that support agile philosophy
  - Ex: Extreme Programming I Scrum
- Agile methods consist of individual elements called practices

Refer to the Ref.2

Agile practices

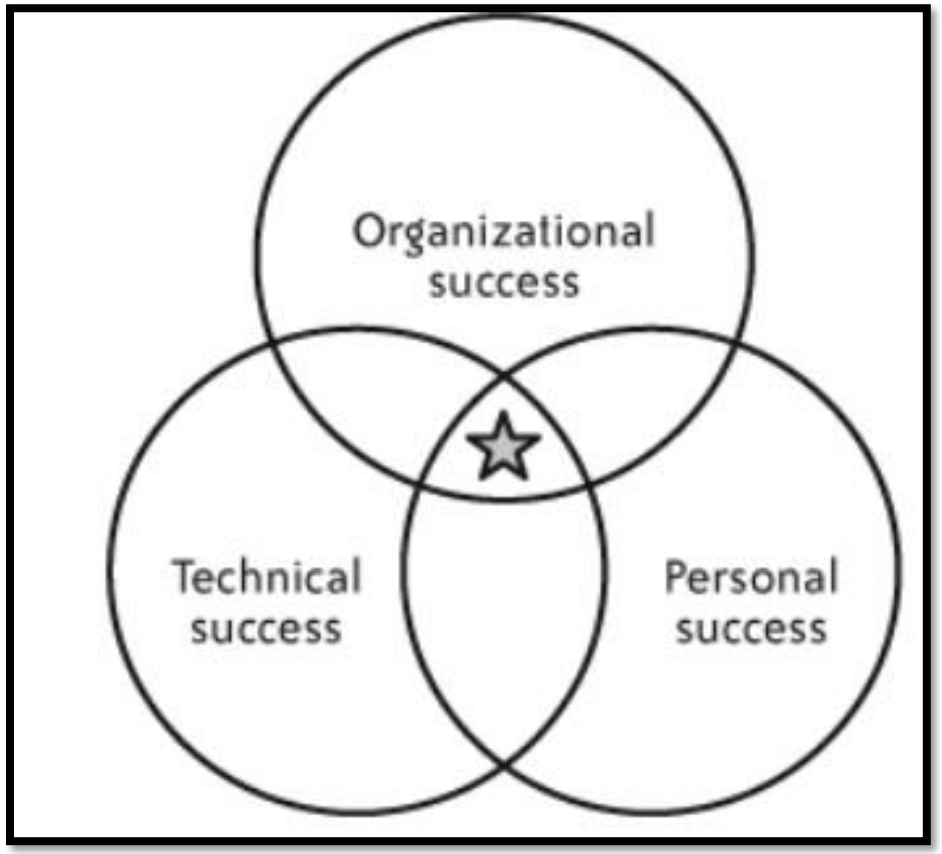
- Practices include using
  - Version control,
  - Setting coding standard
  - Giving weekly demos to your stakeholders

Agile practices often perform double- and triple-duty, solving multiple software development problems simultaneously and supporting each other in clever and surprising ways.

Refer to the Ref.2

Agile development

- Agile development focuses on achieving,
  - Personal successes
  - Technical successes
  - Organizational successes



Refer to the Ref.2 pg.5

The importance of “Three types of Success”

- Without personal success
  - troubles motivating yourself and employees.
- Without technical success
  - source code will eventually collapse under its own weight.
- Without organizational success
  - team may find that they’re no longer wanted in the company.

Refer to the Ref.2 pg.5

Organizational Value and Agile

- Aside from revenue and cost savings, sources of value include:
  - Competitive differentiation
  - Brand projection
  - Enhanced customer loyalty
  - Satisfying regulatory requirements
  - Original research
  - Strategic information

Refer to the Ref.2 pg.5-6

Organizational Value and Agile

- Organizational successes by focusing on delivering value and decreasing costs
- Agile methods set expectations early in the project
  - if your project won't be an organizational success, you'll find out early enough to cancel it before your organization has spent much money.
- Agile teams increase value by
  - including business experts and by focusing development efforts on the core value

Refer to the Ref.2 pg.5-6

Organizational Value and Agile

- Agile Projects release their most valuable features first and release new versions frequently
- When business needs change or when new information is discovered, agile teams change direction to match.
- An experienced agile team will actually seek out unexpected opportunities to improve its plans.

Refer to the Ref.2 pg.5-6

Organizational Value and Agile

- Decreasing Cost by
  - cancelling bad projects early and replacing expensive development practices with simpler ones.
  - communicating quickly and accurately, and make progress even when key individuals are unavailable
  - regularly reviewing the process and continually improving the code
  - making the software easier to maintain and enhance over time

Refer to the Ref.2 pg.5-6

Technical Success

- Ex. Extreme Programming achieving technical successes
  - XP programmers work together, which helps them keep track of the nit-picky details
  - At least two people review every piece of code.
  - Programmers continuously integrate their code
  - The whole team focuses on finishing each feature completely before starting the next
  - Create simple, ever-evolving designs that are easy to modify when plans change

Refer to the Ref.2 pg.7

Personal Success and Agile

- Agile once adopted will directly / indirectly translate these results to you and your team
  - Testers – Involvement and influence quality at every phase of s/w development
  - Developers – Increased technical quality / Greater influence on estimates and schedules / greater autonomy
  - Product / Project Manager – Greater ability to change direction as client requirement changes /team's ability to deliver / Better stakeholder satisfaction

Refer to the Ref.2 pg.7

Personal Success and Agile

- Architect / Domain – product experts – Greater ability to influence development / team's ability to deliver better results
- Executive / Senior Management – Appreciation of team's focus for higher ROI and enhancement to business and services / product.

Refer to the Ref.2 pg.7

1.2 How to use Agile

- Every project and situation is unique
- It is better to have an agile method that is customized to the situation
- Rather than making an agile method from scratch, start with an existing, proven method and iteratively refine it

Steps to follow in order To master agile development

- Decide why you want to use agile development
- Adopt as many of XP's practices as you can
- Follow the XP practices rigorously and consistently
- Start experimenting with changes
- Each time you make a change, observe what happens and make further improvements

Refer to the Ref.2 pg.12

1.3 Agile Manifesto

- Origin of Agile Manifesto
  - Authored at a ski lodge in Utah in 2001
  - The original signees of the Agile Manifesto believed were the core to good software development
  - Seventeen people met to talk, ski, relax, and try to find common ground and of course, to eat.
  - What emerged was the Agile Software Development Manifesto.

http://agilemanifesto.org/history.html

Significance of Agile Manifesto

- Not tangible artifacts such as templates, instructions, rules or procedures, but values
- Agile is not complex in it's beliefs.
- The most difficult aspect of the Agile approach is to trust in it and believe that it will work.
- It will only work if it is applied consistently and completely.

Refer to the Ref.2 pg.12

Agile manifesto

"We are uncovering better ways of developing software by doing it and helping others do it"

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

Refer to the Ref.2 pg.10

Principles behind the agile manifesto

- 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.

Refer to the Ref.2 pg.11

Principles behind the agile manifesto

- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face to-face conversation.

Refer to the Ref.2 pg.11

Principles behind the agile manifesto

- 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.

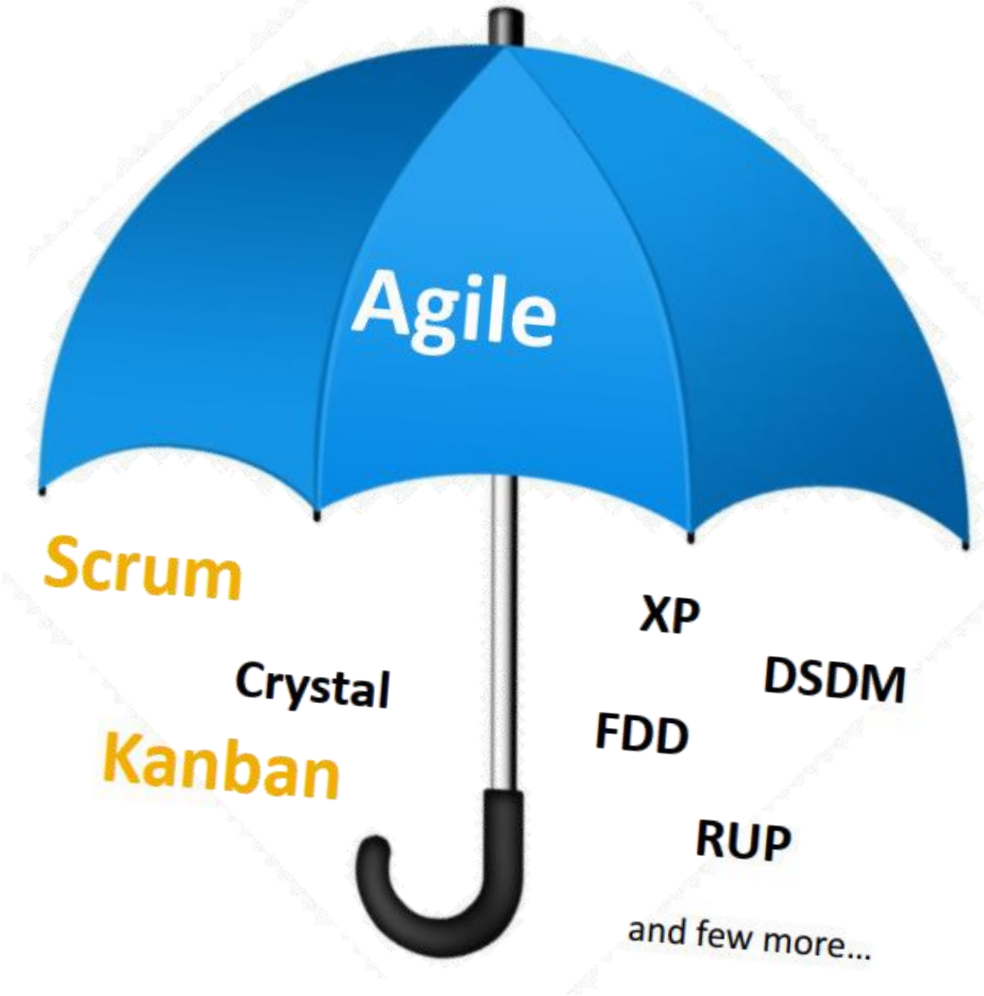
Refer to the Ref.2 pg.11

Principles behind the agile manifesto

- 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 behaviour accordingly.

Refer to the Ref.2 pg.11

1.4 Scrum, Lean, Kanban, Extreme Programming



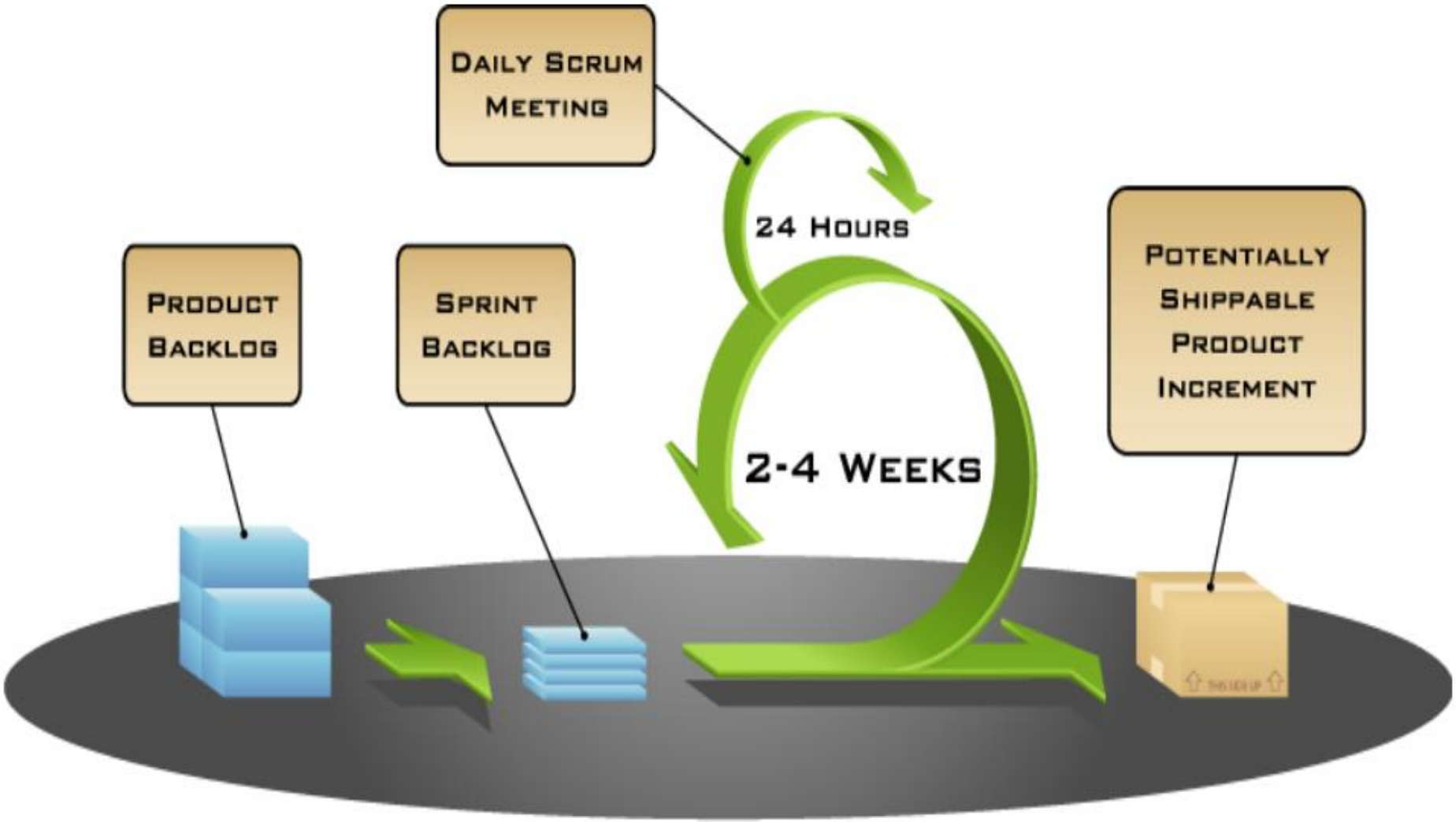
Origin of Scrum

- Jeff Sutherland and Ken Schwaber conceived the Scrum process in the early 90's.
- They codified Scrum in 1995 and published the paper "SCRUM Software Development Process".
- Inherited the name 'Scrum' from the 1986 ground-breaking paper 'The New Product Development Game' by Takeuchi and Nonaka, two acknowledged management thinkers.

Origin of Scrum

- With the term ‘Scrum’ Nonaka and Takeuchi referred to the game of rugby to stress the importance of teams and some analogies between a team sport like rugby and being successful in the game of new product development.

Scrum in a nutshell



For more information Refer to Section 3

Origin of Lean

- The term Lean Software Development was first coined in October 1992.
- Robert "Bob" Charette in 1993 suggested the concept of "Lean Software Development" as part of his work exploring better ways of managing risk in software projects.
- The term "Lean" dates to 1991, suggested by James Womack, Daniel Jones, and Daniel Roos, in their book The Machine That Changed the World: The Story of Lean Production as the English language term to describe the management approach used at Toyota.

https://msdn.microsoft.com/en-us/library/hh533841.aspx

Origin of Lean

- The idea that Lean might be applicable in software development was established after the term was used in in manufacturing processes and industrial engineering.

https://msdn.microsoft.com/en-us/library/hh533841.aspx

Lean

- Five core pillars of Lean Thinking
  - Value
  - Value Stream
  - Flow
  - Pull
  - Perfection
- If a SDLC or a project management process was observed to be aligned with the values of the Lean Software Development movement and the principles of Lean Software Development, it is lean.

Refer to URL 2

Lean

- Accept the human condition
  - Successful processes will be those that embrace and accommodate the human condition rather than those that try to deny it and assume logical, machine-like behavior.
- Accept that complexity & uncertainty are natural to knowledge work
  - The behavior of customers and markets are unpredictable. The flow of work through a process and a collection of workers is unpredictable. Defects and required rework are unpredictable.

Refer to URL 2

Lean

- Work towards a better Economic Outcome
  - Employees and workers deserve a fair rate of pay for a fair effort in performing the work.
- Enable a better Sociological Outcome
  - Creating a workplace that respects people by accepting the human condition and provides systems of work that respect the psychological and sociological nature of people is essential.
- Seek, embrace & question ideas from a wide range of disciplines
- A values-based community enhances the speed & depth of positive change

Refer to URL 2

The 7 Principles of Lean

- Eliminate Waste
- Amplify Learning
- Decide as Late as Possible
- Deliver as Fast as Possible
- Empower the Team
- Build Integrity In
- See the Whole

Refer to URL 2

Origin of Kanban

- A Kanban system is a practice adopted from Lean manufacturing.
- It uses a system of physical cards to limit the quantity of working-progress at any given stage in the workflow.
- A scheduling system for lean and just-in-time (JIT) production.
- Kanban is a system to control the logistical chain from a production point of view, and is not an inventory control system. Kanban was developed by Taiichi Ohno, at Toyota, as a system to improve and maintain a high level of production. Kanban is one method to achieve JIT.

https://msdn.microsoft.com/en-us/library/hh533841.aspx

Kanban

- Lean approach to Agile development
- Aim is to eliminate ‘waste’ wherever possible
- 3 basic principles

1. Start with what you do now

   Kanban does not prescribe a specific set of roles or process steps

Refer to URL 2

Kanban

2. Agree to pursue incremental, evolutionary change

   continuous small changes that stick vs. sweeping changes that fail due to resistance and fear in the organization

3. Respect the current process, roles, responsibilities & titles

   gain support, reduce fear/resistance to change and experience the benefits as a team

Refer to URL 2

Kanban - 5 Core Properties

1. Visualize the workflow

   Kanban literally means "signboard" or "billboard"

2. Limit Work In Process (WIP)

   use a pull system - establish and respect your ideal capacity

3. Manage Flow

   monitor, measure and report the flow of work through each state

Refer to URL 2

Kanban - 5 Core Properties

4. Make Process Policies Explicit

   describe the process accurately in order to improve it

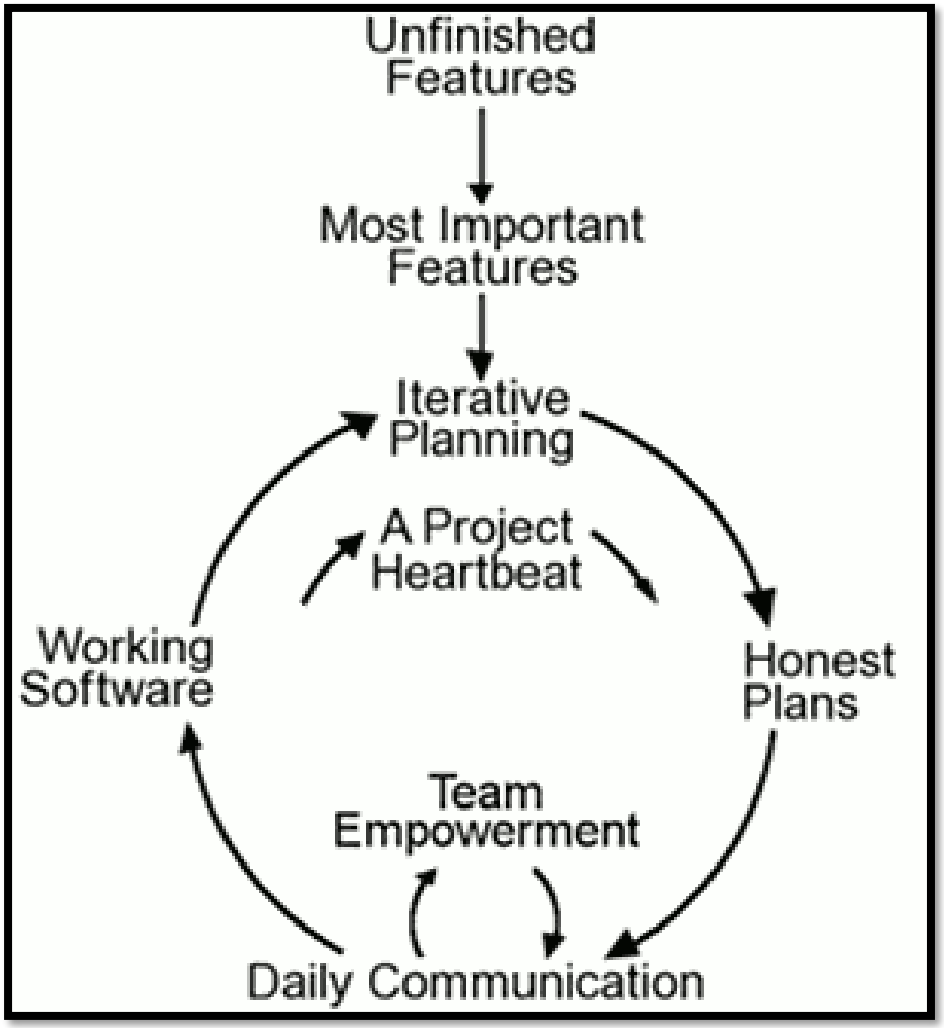
5. Improve Collaboratively

   using models & the scientific method (empirical) to implement continuous, incremental and evolutionary changes

Refer to URL 2

Extreme Programming

- Improve **software quality** and **responsiveness** to changing customer requirements
- **Frequent releases** in **short development cycles**
- **Improve productivity** and **regular checkpoints** with the customer
- **Paired programming**



http://www.extremeprogramming.org/

Refer to section 8

Summary

- Introduction to Agile.
  - Rationale
  - Principles
  - Methods
  - Success types
  - Agile manifesto
- Current agile development frameworks
  - Scrum
  - Lean
  - Kanban
  - XP

Refer to URL 2