AGILE SOFTWARE DEVELOPMENT

Introduction to Agile

Outline

- What is Agility?
- Agile Manifesto
- Agile Principles
- Traditional vs Agile Approach
- Defined vs Empirical Process
- Agile Outsourcing
- Agile Team Characteristics
- Agile Success Factors
- Benefits and Challenges
- Agile Flavors

What is “Agility”?

Agility is the ability to change the body's position efficiently, and requires the integration of isolated movement skills using a combination of balance, coordination, speed, reflexes, strength, and endurance

Source:Wikipedia

In business, agility means the **capability of rapidly and efficiently adapting to changes**. Recently agility has been applied e.g. in the context of agile software development and agile enterprise.

Source:Wikipedia

What is Agile Software Development?

Agile software development refers to a **group of software development methodologies that are based on similar principles.**

Agile methodologies generally promote:

- A project management process that encourages **frequent inspection** and **adaptation;**
- a leadership philosophy that encourages **team work, self-organization** and **accountability;**
- a set of engineering best practices that allow for **rapid delivery of high-quality software;**
- and a business approach that **aligns development with customer needs** and **company goals**.

The ability to create and respond to change in order to succeed in an uncertain and turbulent environment.

Source: AgileAlliance

History of Agile

History & Influences

Early 1900s

- Walter Shewhart: Plan-Do-Study-Act, SPC

Mid 1900s

- Edward Deming: SPC, TQM
- Toyota: Toyota Production System (TPS)
- Peter Drucker: Knowledge Worker

Late 1900s

- Womack and Jones: Lean Thinking
- Eli Goldratt: Theory of Constraints
- Tom Gilb: Evo
- The Toyota Way

Evolution

Early 1990s

- Crystal Methods
- Lean Software Development
- Dynamic Software Development Method (DSDM)

Mid 1990s

- Feature Driven Development (FDD)
- eXtreme Programming (XP)
- Adaptive Software Development

2001: Manifesto for Agile Software Development

- http://www.agilemanifesto.org

2005: Declaration of Interdependence

- http://www.pmdoi.org/

17 Software developers created Agile

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

Agile Process

The agile methodology is as much a philosophical shift as it is a process shift.

Barber Shop Example

Metaphor

"Driving is not about getting the car going in the right direction. Driving is about constantly paying attention, making a little correction this way, a little correction that way."

This is the paradigm for Agile. Stay aware. Adapt. Change.

Agile Manifesto

"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 Agile movement is not anti-methodology, in fact, many of us want to restore credibility to the word methodology. We want to restore a balance. We embrace modelling, but not in order to file some diagram in a dusty corporate repository. We embrace documentation, but not hundreds of pages of never-maintained and rarely-used tomes. We plan, but recognize the limits of planning in a turbulent environment. Those who would brand proponents of XP or SCRUM or any of the other Agile Methodologies as "hackers" are ignorant of both the methodologies and the original definition of the term hacker.”

— Jim Highsmith, History: The Agile Manifesto

12 principles of Agile

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

12 principles of Agile contd...

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

12 principles of Agile contd...

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

12 principles of Agile contd...

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

12 principles of Agile contd…

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

12 principles of Agile contd...

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

12 principles of Agile contd...

7. **Working software** is the primary measure of progress.

12 principles of Agile contd…

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

12 principles of Agile contd...

9. Continuous attention to **technical excellence** and good design enhances agility.

12 principles of Agile contd...

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

12 principles of Agile contd...

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

12 principles of Agile contd…

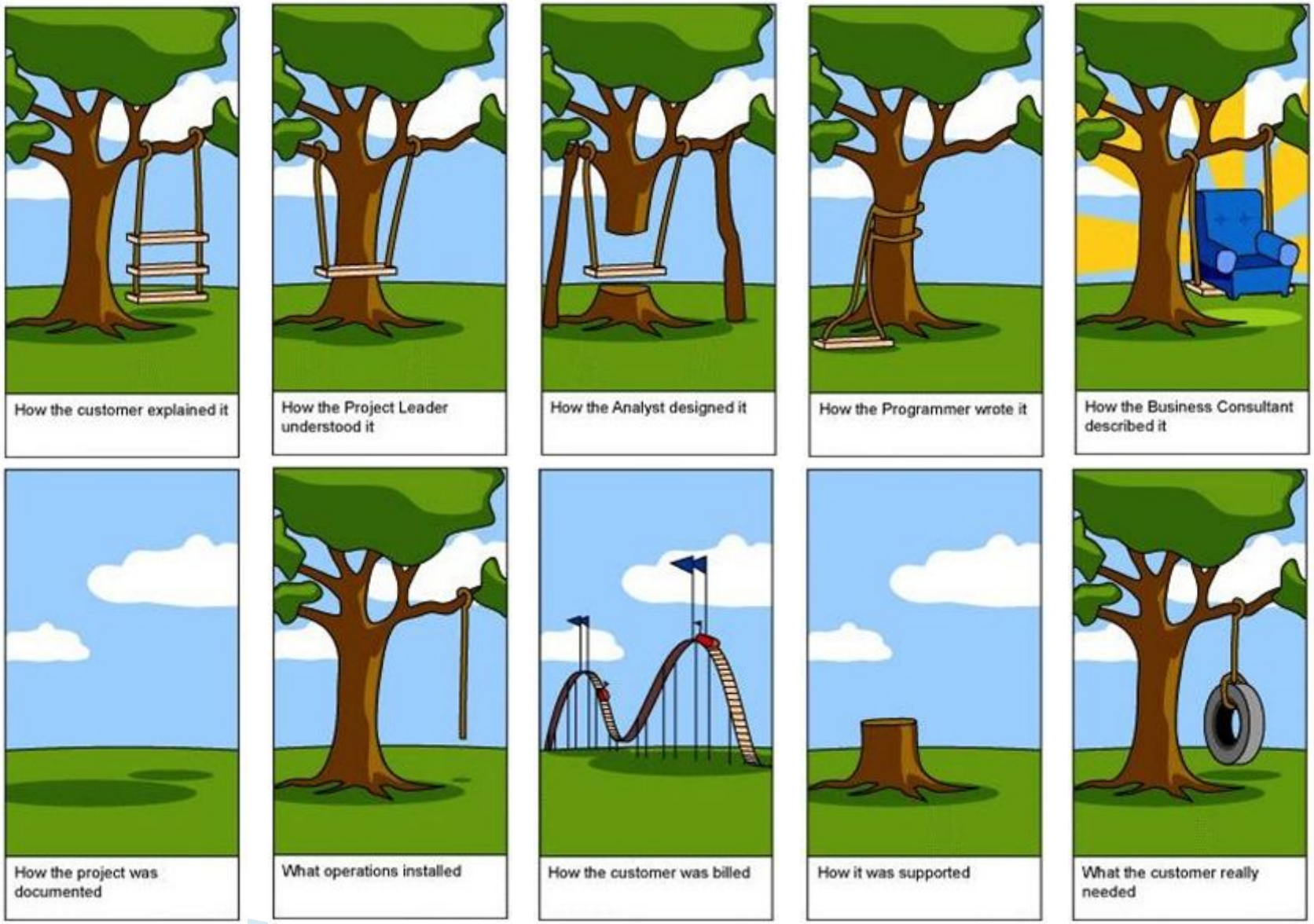
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
Inspect/Adapt

Coin Game

What is waterfall?

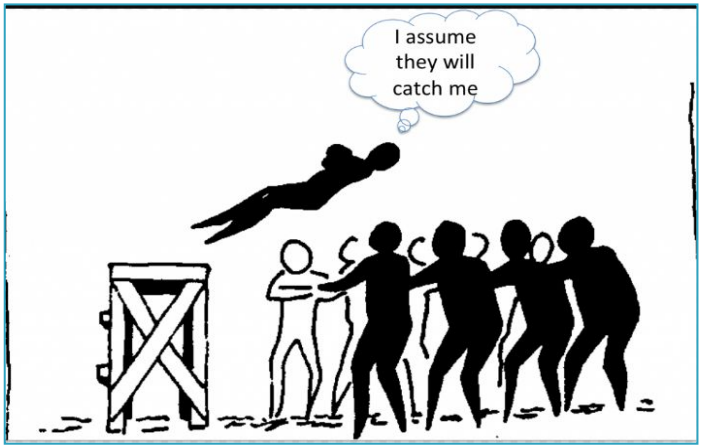
It's What I Asked for But Not What I Need

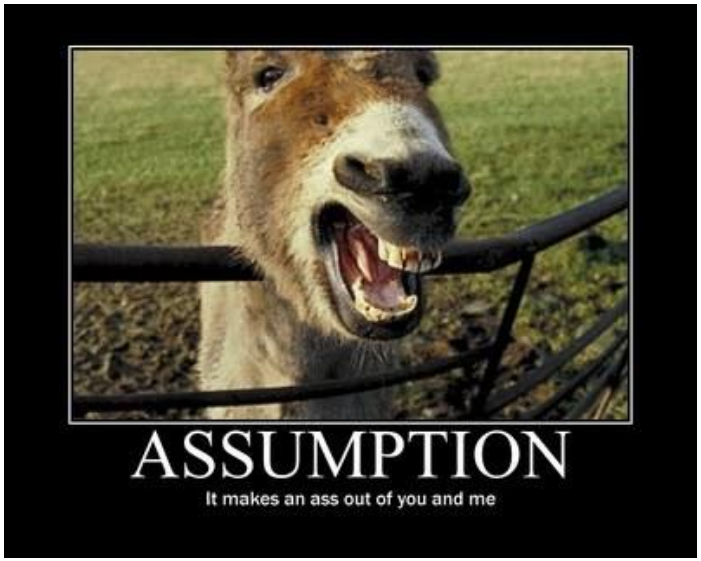
Typical software output



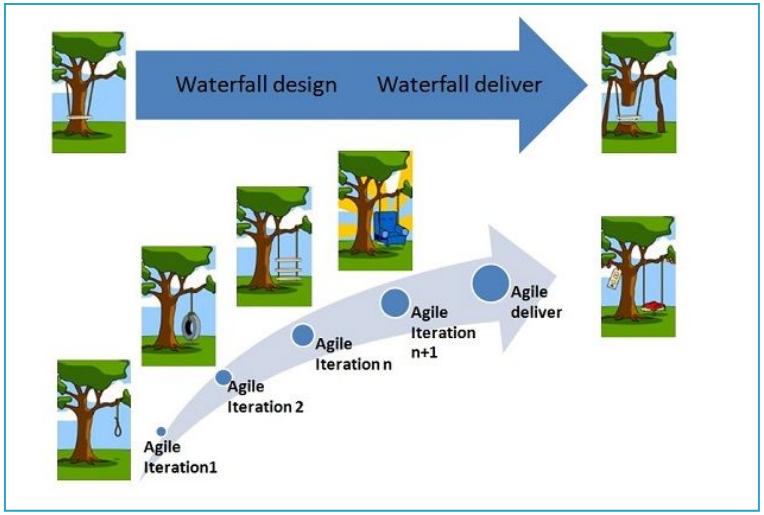
Assumptions in waterfall

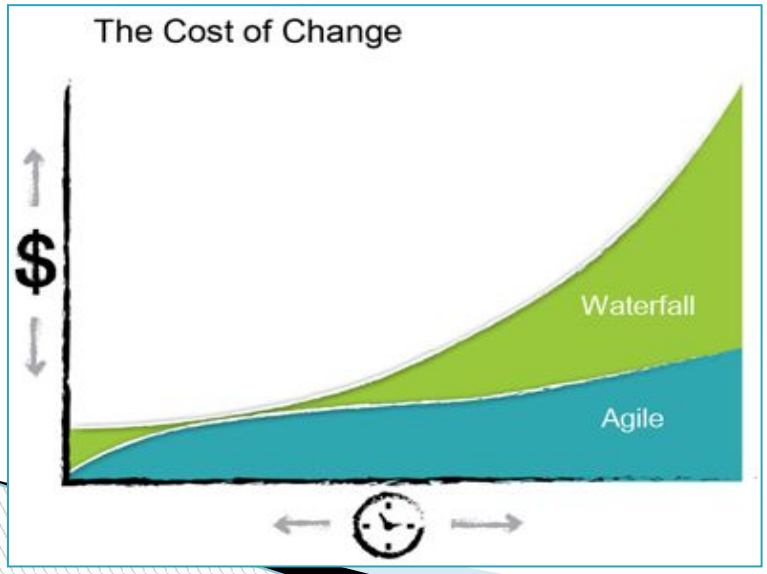
- Assumes all requirement known at start.
- Change is risky and costly.
- Hand-offs is risky and costly.
- System Integration Will Go Well.
- Anything delayed = everything delayed.
- Testing done mostly at the end.
- Working software only at the end.

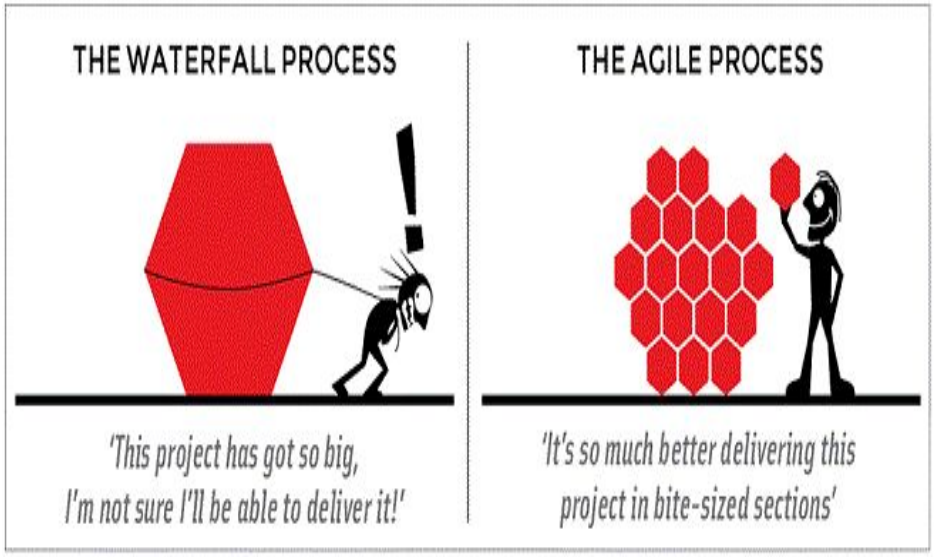




Waterfall vs Agile



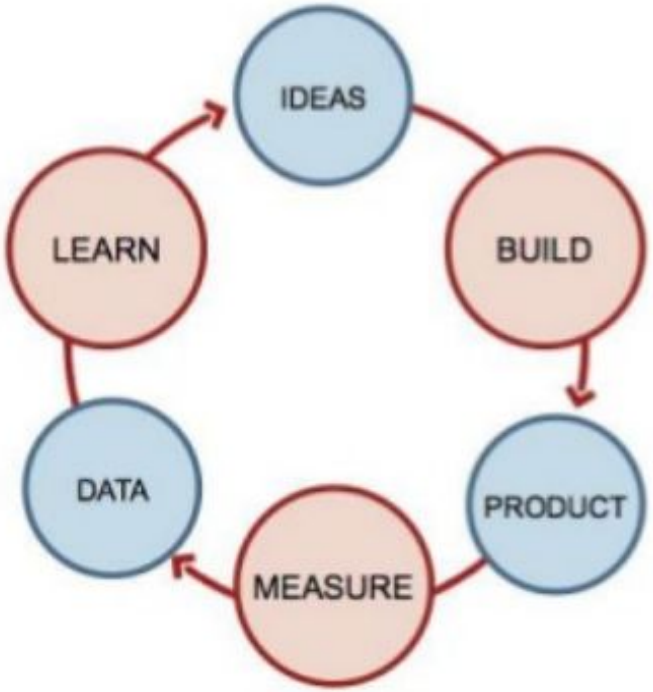




Defined vs Empirical Process

Defined Process

versus

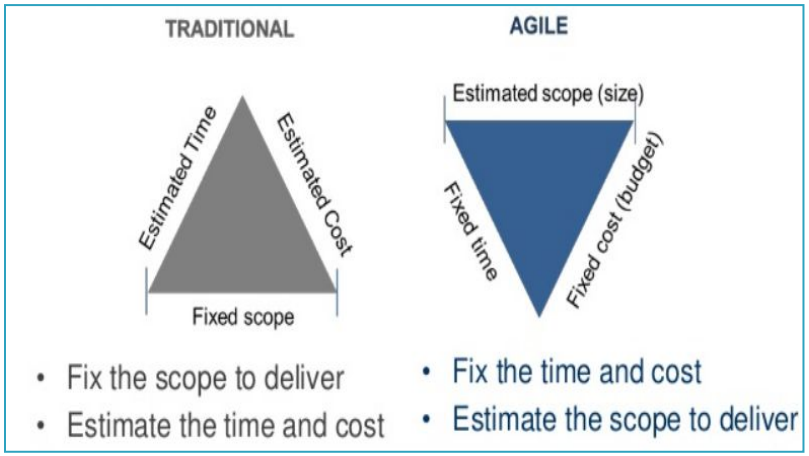


Empirical Process

A iterative incremental approach to software development that seeks to maximize business value at all times.

Manage complexity, unpredictability and change through Visibility, Inspection and Adaptation

Agile Outsourcing



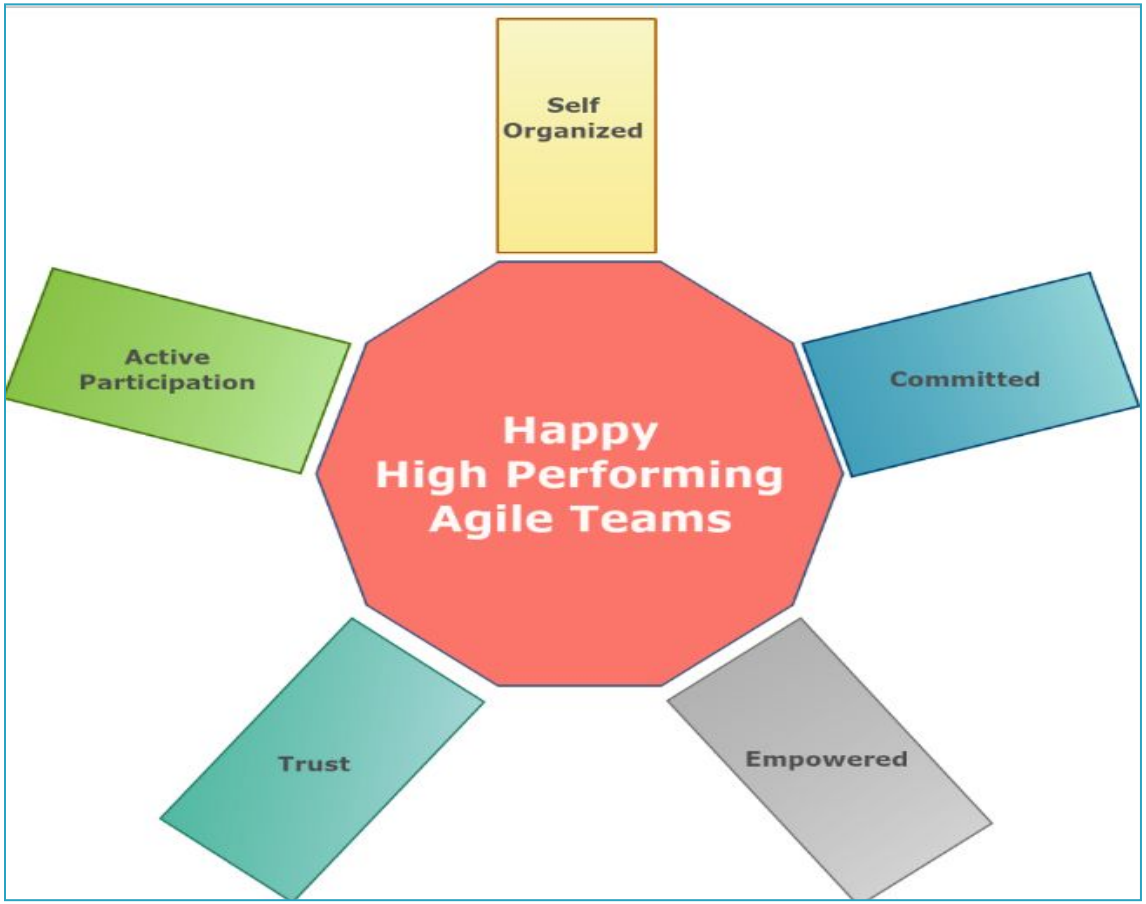
- Fix the scope to deliver
- Estimate the time and cost

- Fix the time and cost
- Estimate the scope to deliver

Common Agile Outsourcing Pitfalls

- The customer procures the "agile" project via traditional strategies
- The customer takes a Water-Scrum-Fall approach
- The customer governs the service provider via a traditional approach
- The customer really isn't agile
- The service provider really isn't agile
- Neither are agile
- Agile is based on trust, yet it behooves you to not trust the service provider

Agile Team Characteristics



Key Success Factors for Agile Software Development

- Management Support
- Agile Thinking
- Cross-Functional Teams
- Empowered Team Members
- Single Voice of the Business
- Shared Accountability
- Servant Leadership
- Collaboration
- Continuous Flow of Value
- Attention to Technical Excellence
- Rapid Risk Reduction
- Early Feedback and Adaptation
- Openness and Transparency
- Trust

Benefits of Agile

- Faster Time to Market
- Frequent Delivery
- Transparency
- Flexibility
- Increase Quality
- Increase Productivity
- Customer Satisfaction
- Predictability
- Decrease Risk

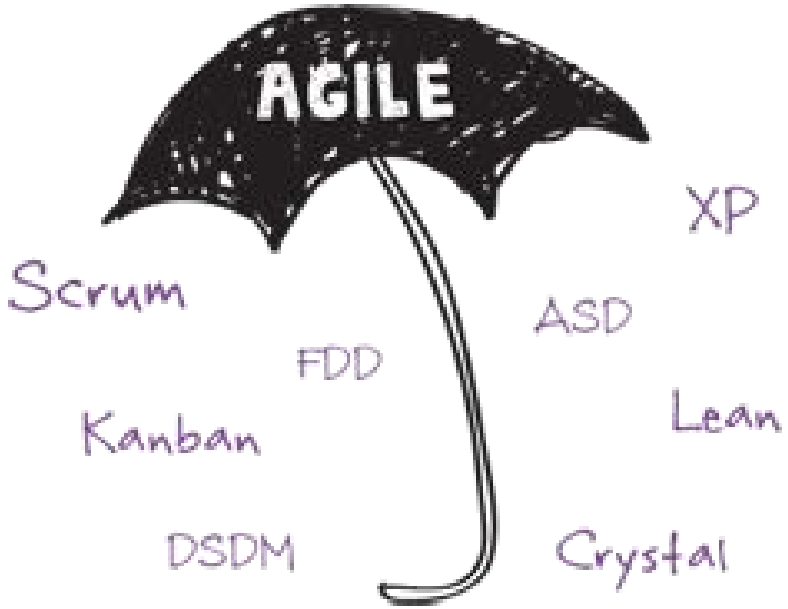
Agile Challenges

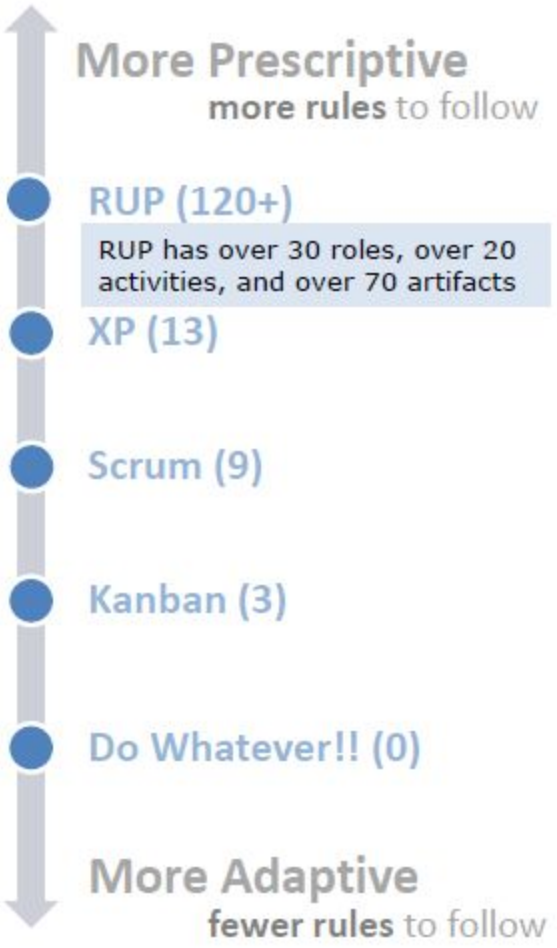
- Customer Availability and Co-Operation
- Customer with clear vision
- People working together
- Team Composition
- Project Cost and Effort Estimation
- Doing Agile over Being Agile

Flavors of Agile

Agile Flavors

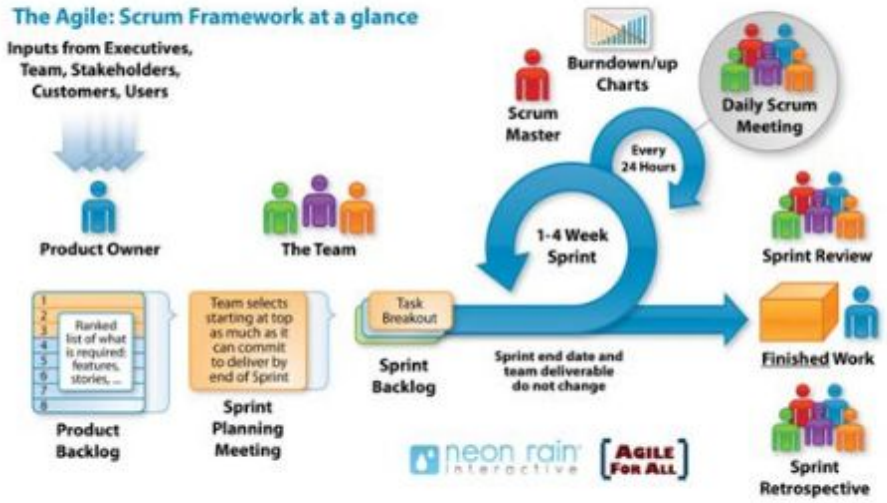
- Scrum
- XP (eXtreme Programming)
- Lean Software Development
- Kanban
- DSDM (Dynamic Systems Development Method)
- Crystal
- FDD (Feature Driven Development)
- Others





Scrum

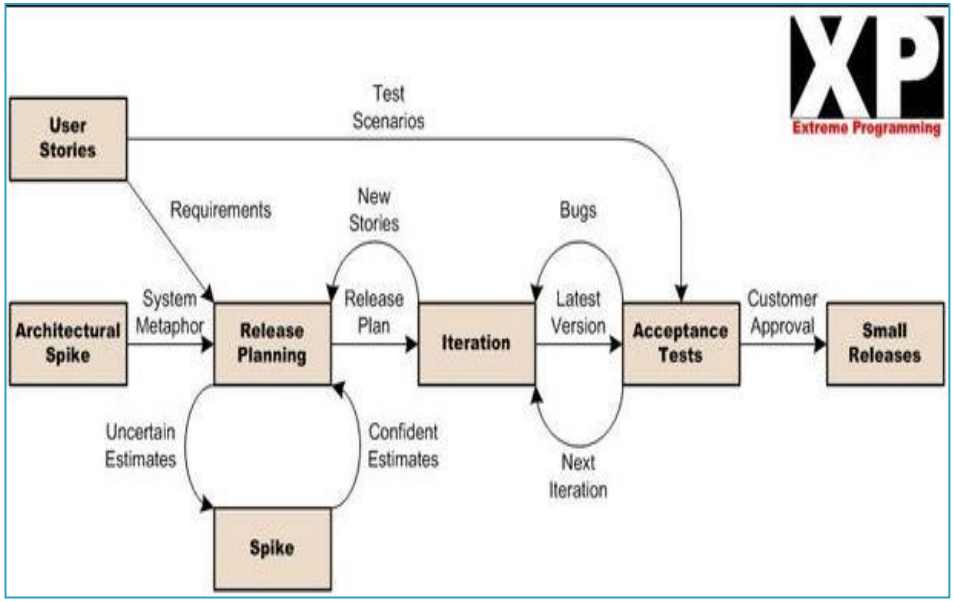
- SCRUM is one of many Agile process frameworks that allows us to focus on delivering the highest business value in the shortest time.
- It allows us to rapidly and repeatedly inspect actual working software (every two weeks to one month).
- The business sets the priorities. Teams self-organize to determine the best way to deliver the highest priority features.
- Every two weeks to a month anyone can see real working software and decide to release it as is or continue to enhance it for another sprint.



Extreme Programming (XP)

Traditional software engineering practices (daily interactions, working software, testing, etc.) taken to so-called "extreme" levels, leads to a process that is more responsive to customer needs ("agile"), while creating software of better quality

- Kent Beck
- Shorter life cycle (2 days - a week)
- On-Site customer
- Pair Programming
- TDD
- BDD
- Continuous integration

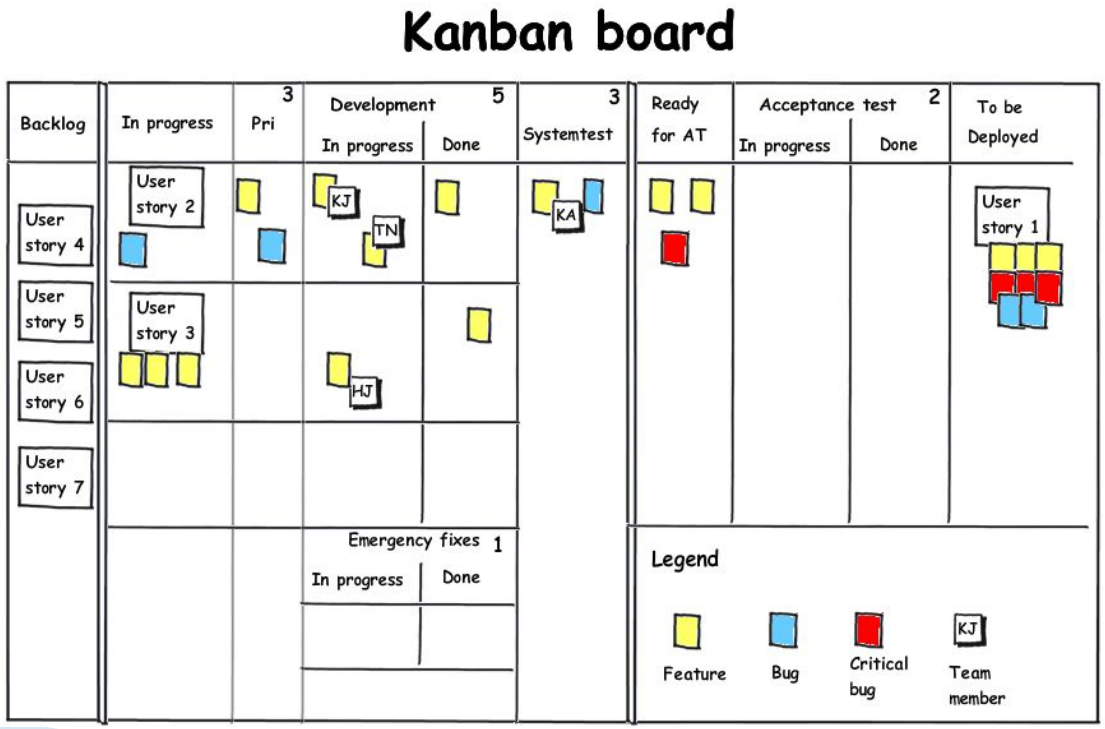


Lean Software Development (LSD)

- Adapted from Toyota Production System.
- Value stream focussed
  - Select only the truly valuable features
  - Work product comes from customer request
- Lean principles
  1. Eliminate waste (Muda)
  2. Amplify learning
  3. Decide as late as possible
  4. Deliver as fast as possible
  5. Empower the team
  6. Build integrity in
  7. See the whole

Kanban

- Emphasis is on JIT (Just In Time) delivery
- Visual process-management system that tells
  - What to produce?
  - When to produce?
  - How to produce?
- Again inspired from TPS
- Kanban in Japanese means “Signboard” or “Billboard”



Thank You!