

# CS 301 Process Models - Agile

Professor Eswaran 23 Aug 2024

#### **Agile Manifesto**



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

### **Agile Principles**

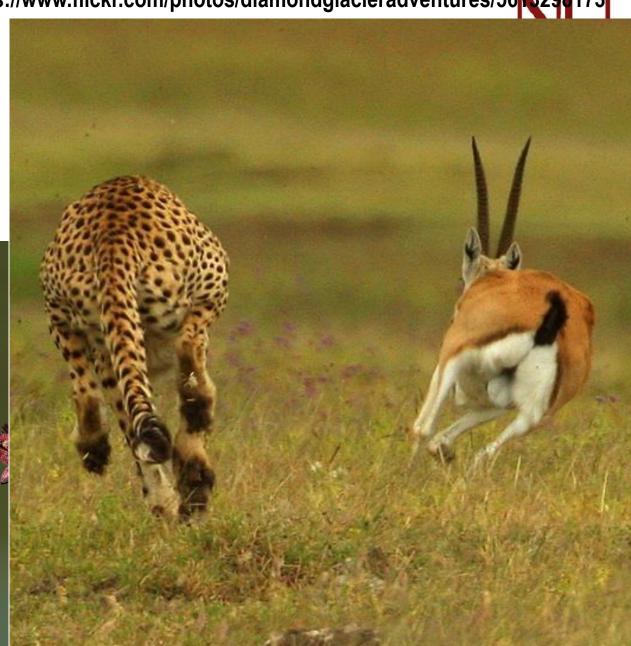


Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
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 then 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.
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.
Simplicitythe art of maximizing the amount of work not doneis 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 Methodology**

https://www.flickr.com/photos/diamondglacieradventures/5613298173





#### **Agility**

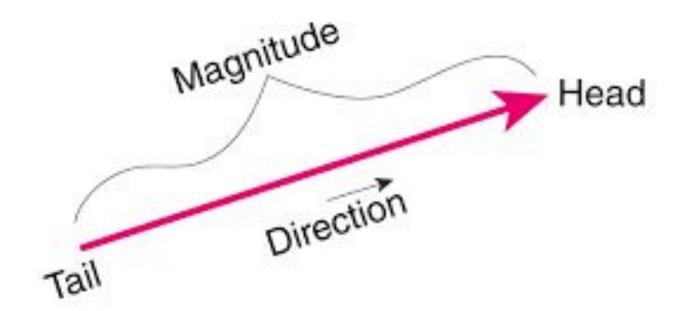


Ability to change direction and velocity at same time.

#### **Agility**

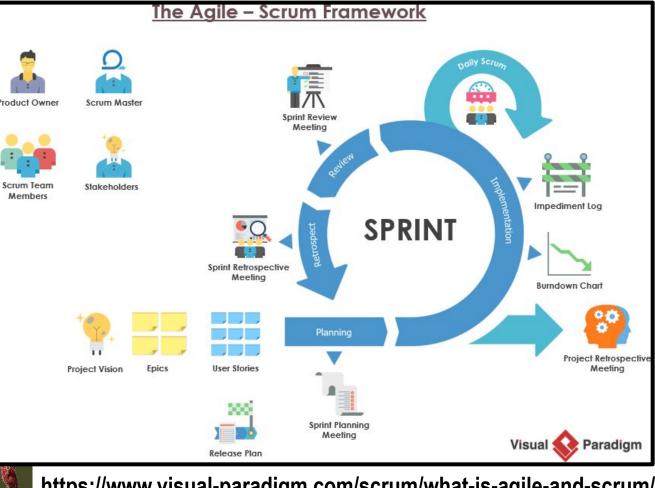


#### Ability to change direction and velocity at same time.



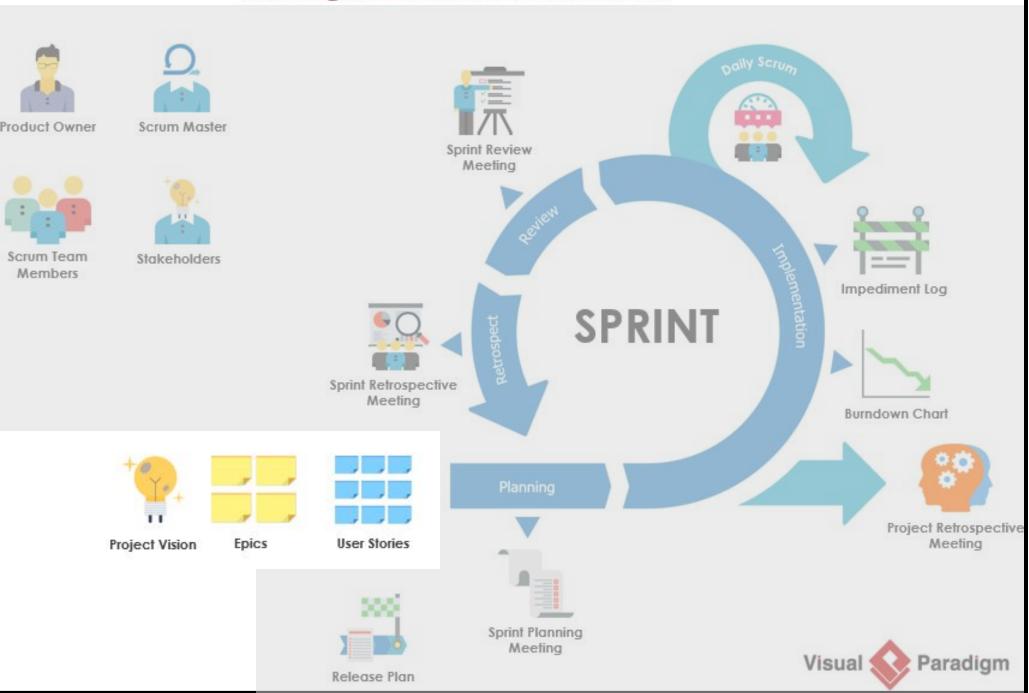
### **Agile Methodology**





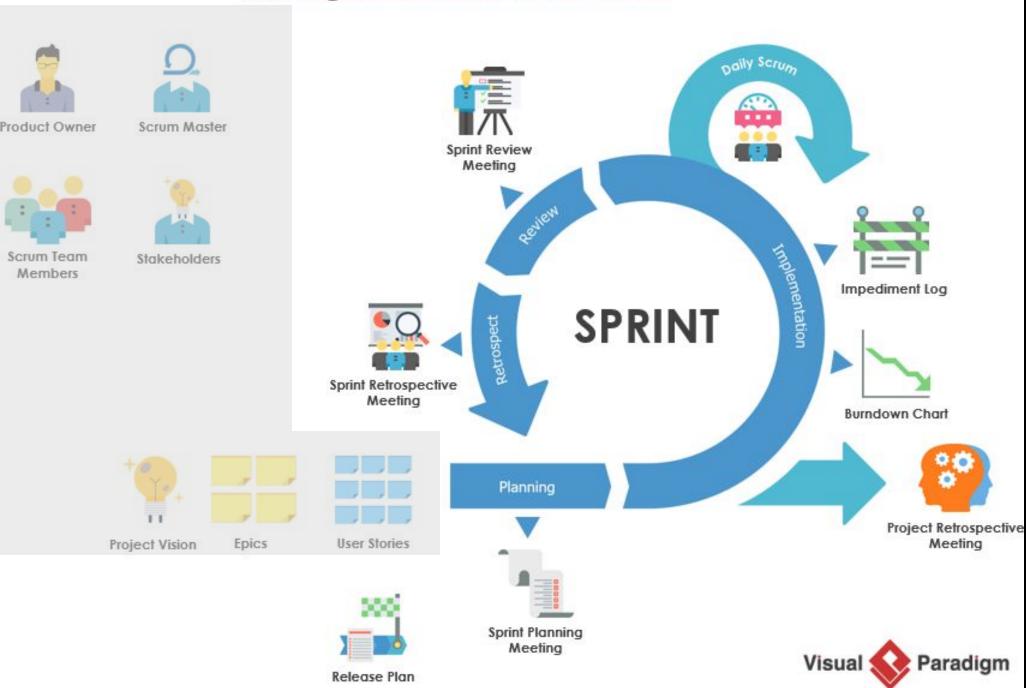
https://www.visual-paradigm.com/scrum/what-is-agile-and-scrum/

#### <u>The Agile – Scrum Framework</u>





#### <u> The Agile – Scrum Framework</u>





#### <u>The Agile – Scrum Framework</u>





Scrum Master



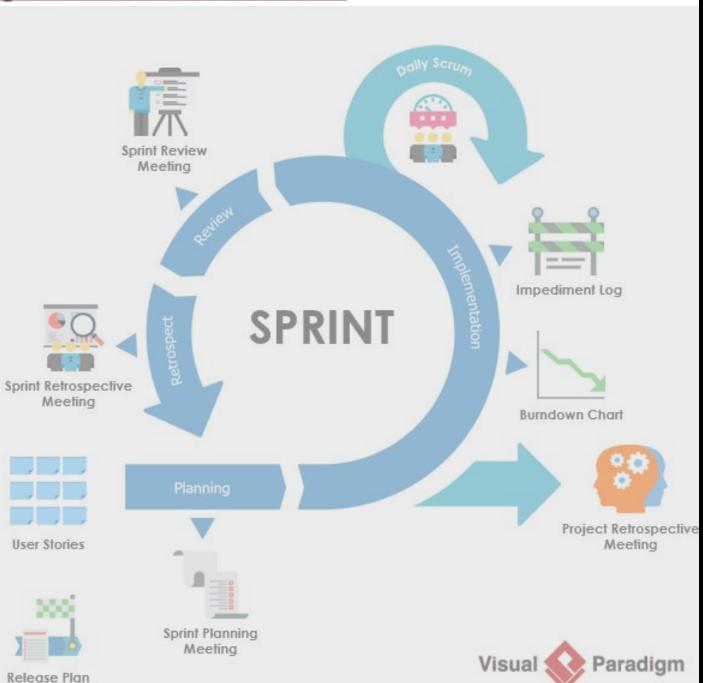
Members



Project Vision

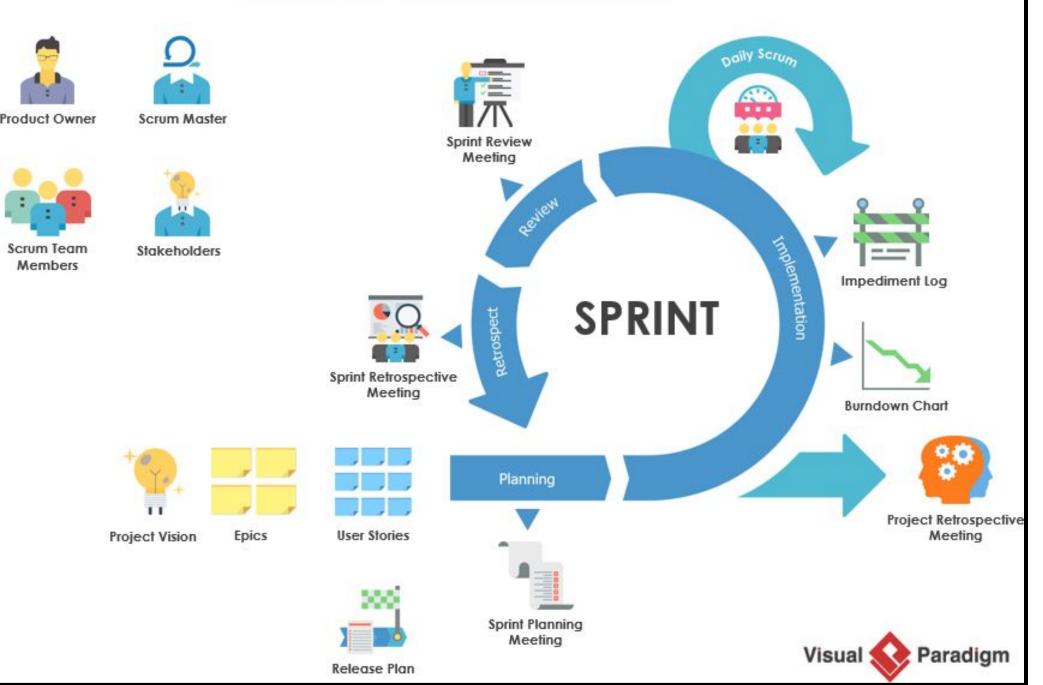
**Epics** 

Stakeholders





#### <u> The Agile – Scrum Framework</u>





### **Agile Methodology**



#### **Advantages**

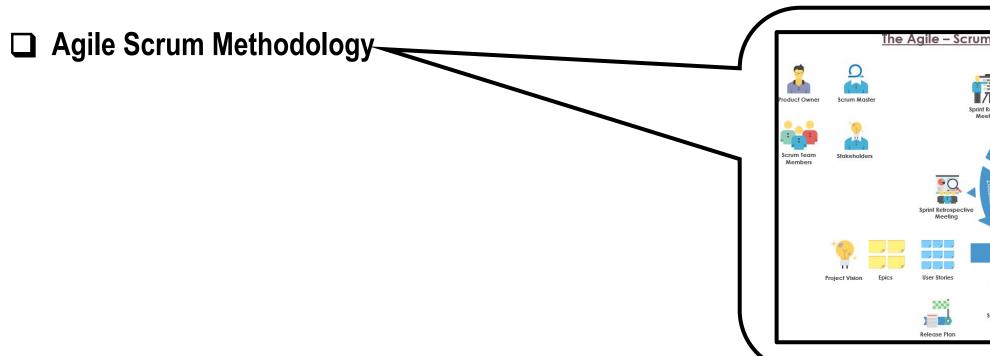
- Favours change
- **Short communication for transparency**
- Shift left philosophy

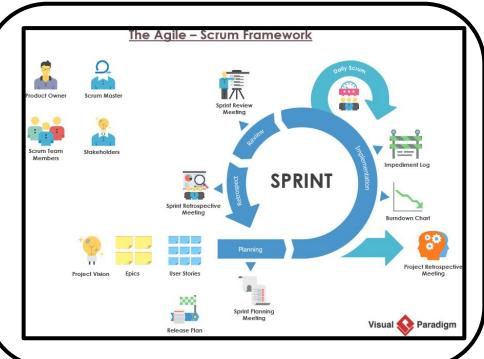
#### **Disadvantages**

- **☐** Focus on documentation is weak
- ☐ Losing the plot

https://www.visual-paradigm.com/scrum/what-is-agile-and-scrum/





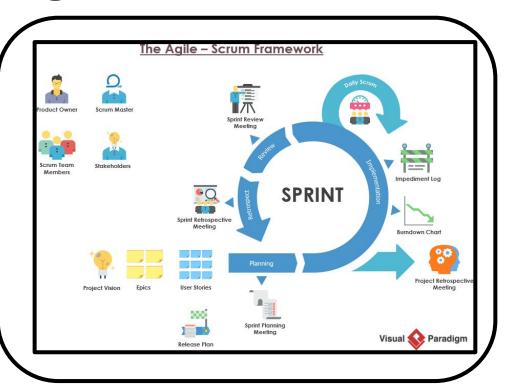




### **Agile Variations**

Agile Scrum Methodology

- Regular releases and fixed-length iterations
- **☐** Working, tested software
- Value-driven development
- Continuous planning
- □ Relative estimation
- ☐ Feature discovery
- Continuous testing

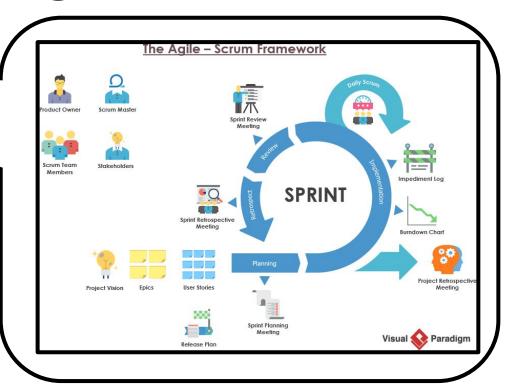




#### **Agile Variations**

Agile Scrum Methodology

- ☐ Regular releases and fixed-length iterations
- **☐** Working, tested software
- Value-driven development
- Continuous planning
- □ Relative estimation
- ☐ Feature discovery
- Continuous testing

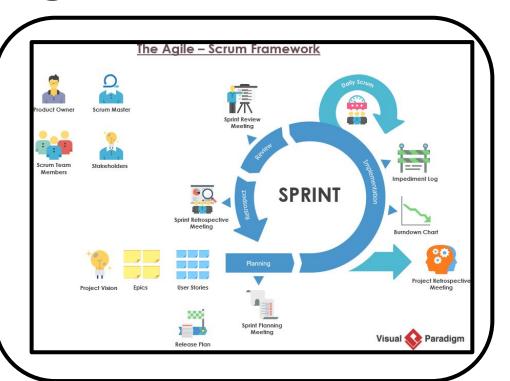




### **Agile Variations**

Agile Scrum Methodology-

- □ Regular releases and fixed-length iterations
- ☐ Working, tested software
- Value-driven development
- Continuous planning
- ☐ Relative estimation
- ☐ Feature discovery
- Continuous testing

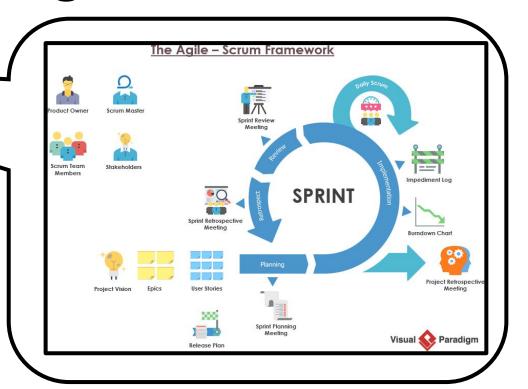




### **Agile Variations**

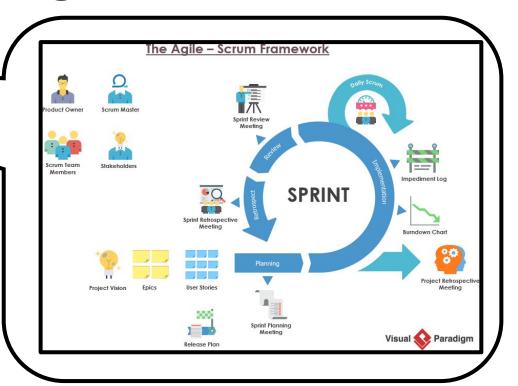
☐ Agile Scrum Methodology-

- ☐ Regular releases and fixed-length iterations
- ☐ Working, tested software
- Value-driven development
- Continuous planning
- Relative estimation
- ☐ Feature discovery
- Continuous testing



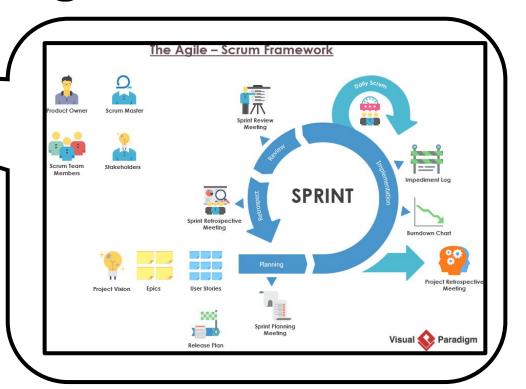


- Agile Scrum Methodology Regular releases and fixed-length iterations
- Working, tested software
- Value-driven development
- **Continuous planning**
- **Relative estimation**
- Feature discovery
- **Continuous testing**





- ☐ Agile Scrum Methodology
- ☐ Regular releases and fixed-length iterations
- ☐ Working, tested software
- Value-driven development
- Continuous planning
- ☐ Relative estimation
- ☐ Feature discovery
- Continuous testing

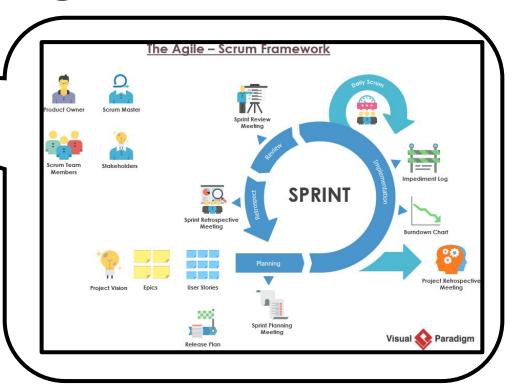




### **Agile Variations**

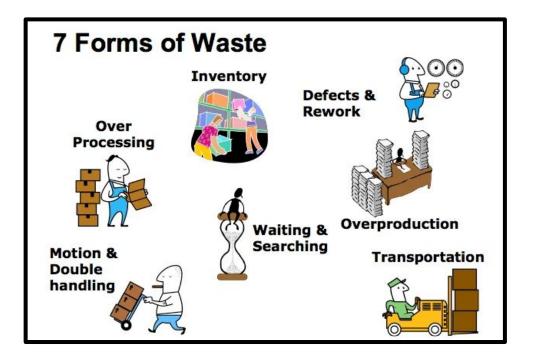
☐ Agile Scrum Methodology

- ☐ Regular releases and fixed-length iterations
- ☐ Working, tested software
- Value-driven development
- Continuous planning
- ☐ Relative estimation
- ☐ Feature discovery
- ☐ Continuous testing



अवारि अवन

- □ Agile Scrum Methodology
- ☐ Lean Software Development



- ☐ Eliminate waste
- □ Amplify learning
- □ Decide as late as possible
- □ Deliver as fast as possible
- **☐** Empower the team
- **☐** Build integrity in
- □ Optimize the whole



- → Agile Scrum Methodology
- ☐ Lean Software Development
- ☐ Feedback
- ☐ Synchronization
- **☐** Iterations

- ☐ Eliminate waste
- Amplify learning
- Decide as late as possible
- □ Deliver as fast as possible
- **☐** Empower the team
- **☐** Build integrity in
- Optimize the whole



- → Agile Scrum Methodology
- ☐ Lean Software Development

- Develop when needed
- ☐ Delay special requirements

- ☐ Eliminate waste
- Amplify learning
- ☐ Decide as late as possible
- □ Deliver as fast as possible
- **☐** Empower the team
- **□** Build integrity in
- □ Optimize the whole



- ☐ Agile Scrum Methodology
- ☐ Lean Software Development

- Develop quickly
- ☐ Reduce work in progress

- ☐ Eliminate waste
- □ Amplify learning
  - ☐ Decide as late as possible
- **□** Deliver as fast as possible
- **☐** Empower the team
- **□** Build integrity in
- □ Optimize the whole



- → Agile Scrum Methodology
- ☐ Lean Software Development
- ☐ Enable decision making at lower levels
- ☐ Reduce approval cycles

- ☐ Eliminate waste
- □ Amplify learning
- Decide as late as possible
- **□** Deliver as fast as possible
- **☐** Empower the team
- **☐** Build integrity in
- □ Optimize the whole



- ☐ Agile Scrum Methodology
- ☐ Lean Software Development
- ☐ Increase quality assurance
- ☐ Integrity through transparency

- ☐ Eliminate waste
- □ Amplify learning
- ☐ Decide as late as possible
- □ Deliver as fast as possible
- Empower the team
- **□** Build integrity in
- □ Optimize the whole



- → Agile Scrum Methodology
- ☐ Lean Software Development

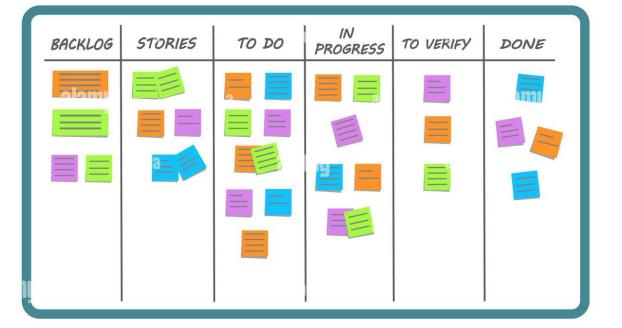
- ☐ Reduce lead and lag situations
- ☐ Delivery clarity

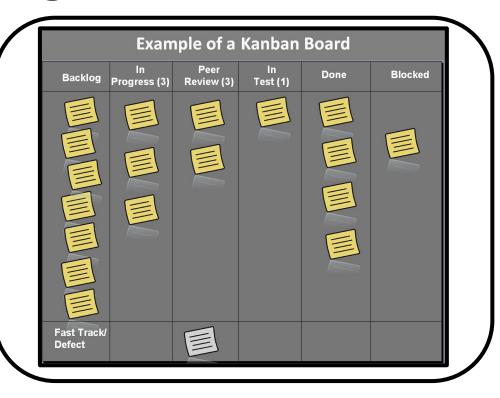
- ☐ Eliminate waste
- □ Amplify learning
- ☐ Decide as late as possible
- □ Deliver as fast as possible
- **☐** Empower the team
- **☐** Build integrity in
- Optimize the whole



#### **Agile Variations**

- ☐ Agile Scrum Methodology
- ☐ Lean Software Development
- ☐ Kanban





https://upload.wikimedia.org/wikipedia/commons/f/f5/Kanban\_board\_example.jpg



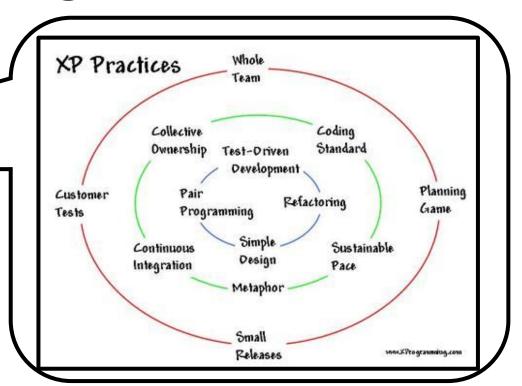
- ☐ Agile Scrum Methodology
- ☐ Lean Software Development
- Kanban

- ☐ Simple production overview
- Individual visibility and accountability
- □ Bottleneck Identification
- ☐ Improved Focus
- □ Collaborative work environment
- Minimize backlog overload
- □ Coherent flow



### **Agile Variations**

- ☐ Agile Scrum Methodology
- ☐ Lean Software Development
- ☐ Kanban
- ☐ Extreme Programming (XP)



https://www.mitchlacey.com/intro-to-agile/extreme-programming



**Agile Variations** 

- → Agile Scrum Methodology
- ☐ Lean Software Development
- ☐ Kanban
- Extreme Programming (XP)

- Pair programming
- ☐ Pride in work
- No Overtime
- Useable code at all times
  - **☐** Time investment
  - ☐ High cost
  - **□** Self Discipline
  - **☐** Version Management

https://www.mitchlacey.com/intro-to-agile/extreme-programming