



**BITS Pilani**  
Pilani Campus

# BITS Pilani presentation

K.Anantharaman  
Faculty CS Department  
[kanantharaman@wilp.bits-pilani.ac.in](mailto:kanantharaman@wilp.bits-pilani.ac.in)



# **SE ZG544 , Agile software processes**

## **Lecture No. 3- Agile Manifesto & Principles**

# Recorded Presentation: Key Points



- Agile Manifesto & Agile Principles
- Agile Project Manager:
  - Devolves control and planning to the team
  - Does not create WBS, schedule, estimates or assign tasks
  - Leader and motivator, not boss or manager
- Key Human Aspects:
  - People matter more than processes
  - Sustainable pace for work-life balance
  - Respect diversity of contributions
  - Prefer face-to-face communication
  - Customer at the center
  - Adapt to changing requirements
  - Focus on delivery over process compliance
  - Close collaboration between teams
  - Manager leads and motivates

# Three Perspectives

## (HOT Perspectives)



- The **H**uman perspective:
  - Cognitive and social aspects, and refers to learning and interpersonal (teammates, customers, management) Process.
- The **O**rganizational perspective:
  - Managerial and cultural aspects and refers to the workspace and issues that extend beyond the team.
- The **T**echnological perspective:
  - Practical and technical aspects and refers to Technical and Coding Practices.

# Agile Manifesto Principles (Year 2001)



- <https://agilemanifesto.org/>

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 Process and tools (H)
- Working software over comprehensive documentation (T)
- Customer collaboration over contract negotiation (HO)
- Responding to change over following a plan (OT)

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

- Each principle supports and supported by other principles
- Redefined roles for Developer, Manager, Customer
- No “Big Upfront” Steps
- Iterative Development
- Negotiated and limited functionality
- Focus on Quality – Achieved through testing

# Agile Principles (Not Official)



## Organizational

- 1 Put the customer at the center.
- 2 Let the team self-organize.
- 3 Work at a sustainable pace.
- 4 Develop minimal software:
  - 4.1 Produce minimal functionality.
  - 4.2 Produce only the product requested.
  - 4.3 Develop only code and tests.
- 5 Accept Change

## Technical

- 6 Develop iteratively:
  - 6.1 Produce frequent working iterations.
  - 6.2 Freeze requirements during iterations.
- 7 Treat tests as a key resource:
  - 7.1 Do not start any new development until all tests pass.
  - 7.2 Test first.
- 8 Express requirements through user stories or scenarios.

# Agile Practices



- Agile Manifesto → Agile Principles → Agile Practices
- Agile Practices → Project Outcome
- Agile Practices
  - Sprint Planning, Product Backlog, Sprint Review, Planning Game, Frequent Delivery, Retrospective
  - Definition of Done
  - Whole Team, Osmotic Communication, Daily Scrum
  - TDD, Pair Programming, Continuous Integration, 10-minutes Build
- Agile methods/methodologies
  - Scrum, XP, Kanban, Crystal

# Anti Patterns: Agile Manifesto



- The tool makes us Agile, Relentless automation
- Hierarchies
- Over-standardization
- Proxy customers (Business Analysts, Architect acting as customer)
- Considering plans and roadmaps as commitments
- Expecting too much detail
- Not engaging stakeholders



# Anti Patterns: Agile Principles



- Out of sight, out of mind - Stakeholders
- Requiring additional documentation or reporting, "We will need this later", Documentation as collaboration, Write only documentation
- One size fits all approach towards team management
- Chasing the metrics
- Ignoring the environment
- Multiple deployment environments
- Detailed story descriptions, Fixed standards or Process, Aiming for Small stories on the backlog
- Restricting who can talk to the customer
- Not considering cultural differences
- Lacking collaboration skills
- Over-complicating things/Future proof everything
- Insisting on Sign-off Process
- "Just in case" development
- Management focus on individuals
- Iterations planned in advance
- Focus on the tasks not the value

# Q&A



Q1, .....Q.9