Agile Transformation
Agile Values, Principles and Practices

Prof. V S Prakash Attili

Information Technology and Systems
Indian Institute of Management, Lucknow

***Note**: Class notes/presentation prepared with due credit to all the references.*

Agenda

- A Simple Story
- Challenges in IT industry / Product Delivery
- Why Agile?
- Cost of change as a function of time
- Agile Values, Principles and Practices
- Agile Approaches : Compare and Contrast
- Extreme Programming
- LEAN software development
- Questions & Answers

The Headlights and the Tunnel…

A Story ….

There was a highway department testing a new safety proposal. They asked motorists to turn on their headlights as they drove through a tunnel. However, shortly after exiting the tunnel the motorists encountered a scenic-view overlook. Many of them pulled off the road to look at the reflections of wildflowers in pristine mountain streams and snow-covered mountain peaks 50 miles away. When the motorists returned to their cars, they found that their car batteries were dead, because they had left their headlights on. So, the highway department decided to erect signs to get the drivers to turn off their head-lights.

[Based on Gause and Weinberg, 1990]

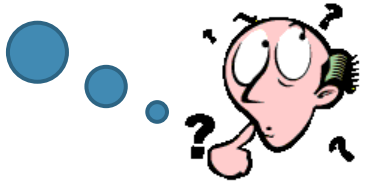
TURN YOUR LIGHTS OFF

Not everyone would heed the request to turn their headlights on, and they couldn't turn their headlights off

IF YOUR HEADLIGHTS ARE ON, TURN THEM OFF

IF IT IS DAYTIME AND YOUR HEADLIGHTS ARE ON, THEN TURN THEM OFF

would be inappropriate if it were overcast and visibility was greatly reduced

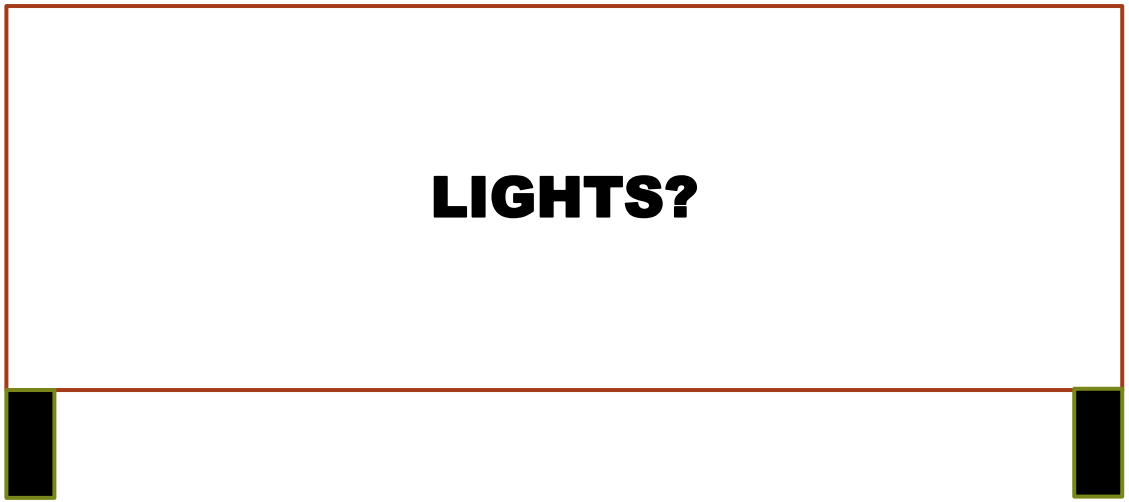


IF YOUR HEADLIGHTS ARE ON AND THEY ARE NOT REQUIRED FOR VISIBILITY, THEN TURN THEM OFF

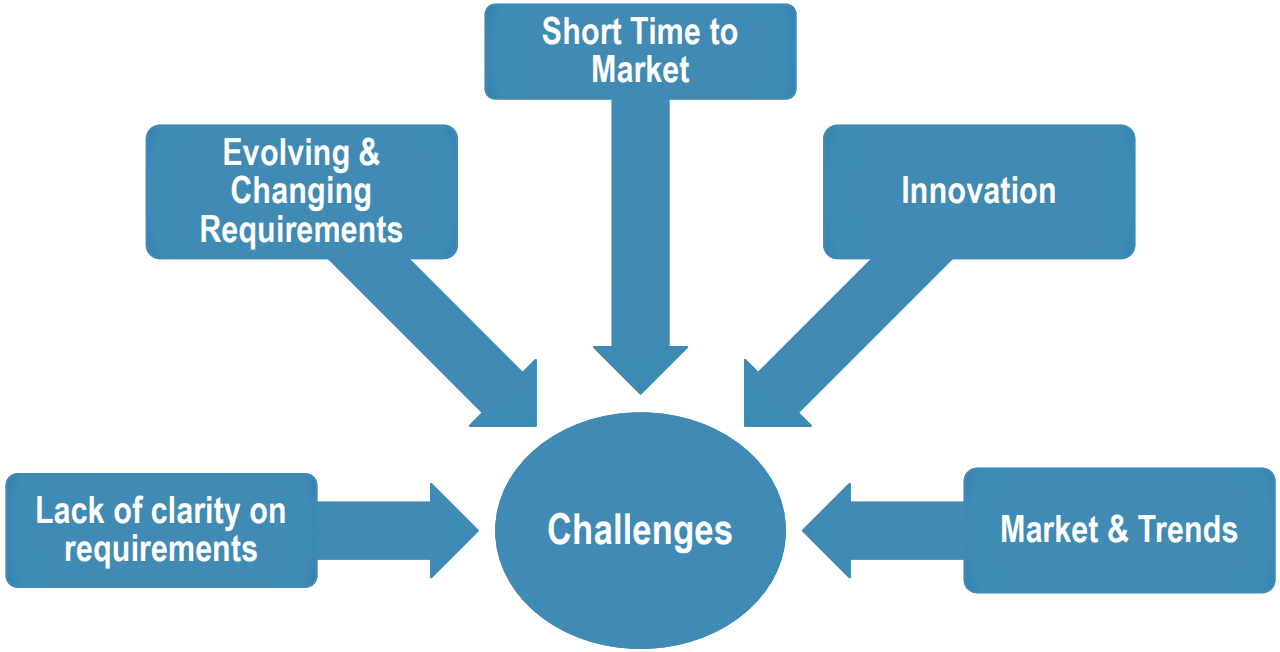
Many new cars are built so that their headlights are on whenever the motor is running, so they couldn't be turned off

ARE YOUR LIGHTS ON?

They decided to stop trying to identify applicable states – just alert the drivers and let them take appropriate action



Challenges - IT Industry / Product Delivery



Why Agile?

The Golden Circle

By Simon Sinek

WHAT

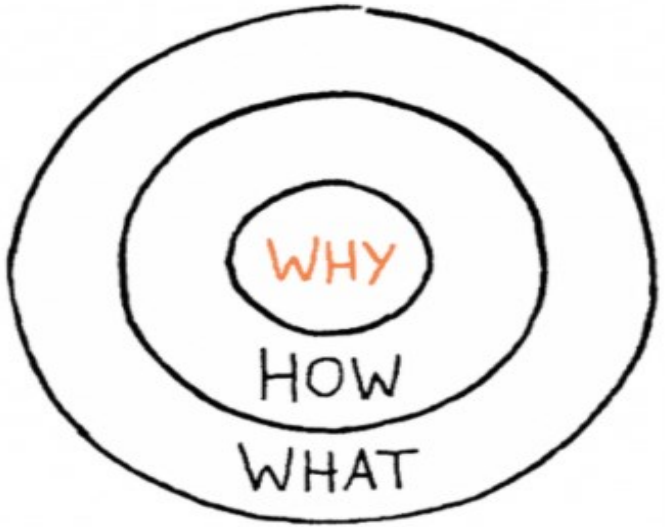
Every organization on the planet knows WHAT they do. These are products they sell or the services

HOW

Some organizations know HOW they do it. These are the things that make them special or set them apart from their competition.

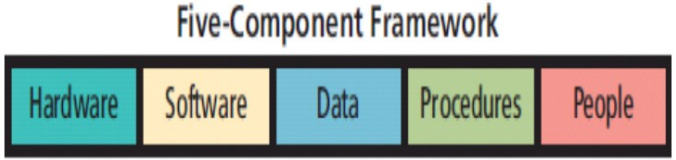
WHY

Very few organizations know WHY they do what they do. WHY is not about making money. That's a result. WHY is a purpose, cause or belief. It's the very reason your organization exists.



Management Information System

- Information Technology - What comes to mind?
  - ✓ INFRA – **Hardware** (storage/processing power)
  - ✓ **Software** (logical/programming power)
  - ✓ Peripheral equipment's that inputs/outputs **Data** (Can communicate i.e. including IoT devices)
- Information Systems

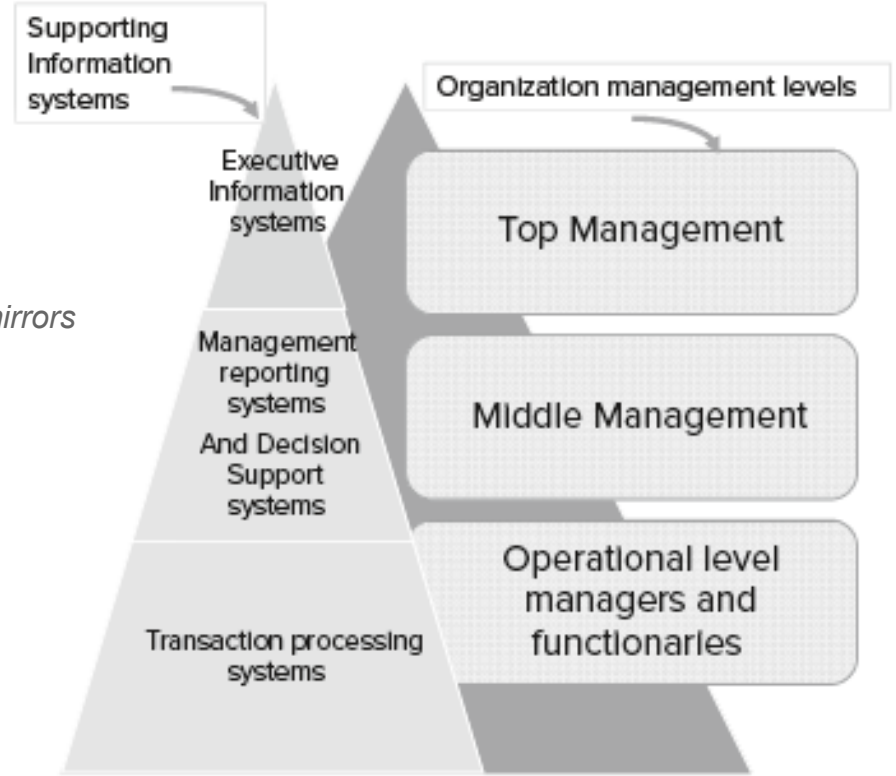


- Management Information systems – Key elements
  - ✓ Management and Use
  - ✓ Information Systems
  - ✓ Strategies

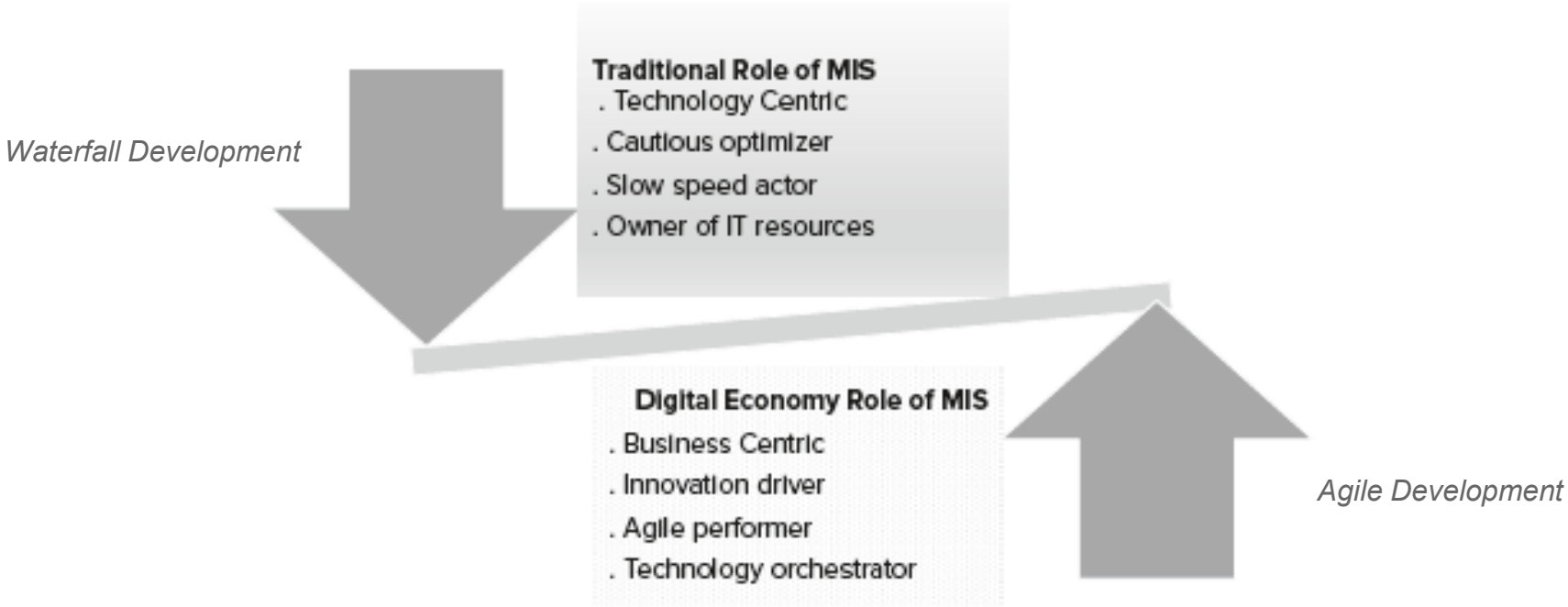
Managing IS to Achieve Business Strategy Goals

Management Information System

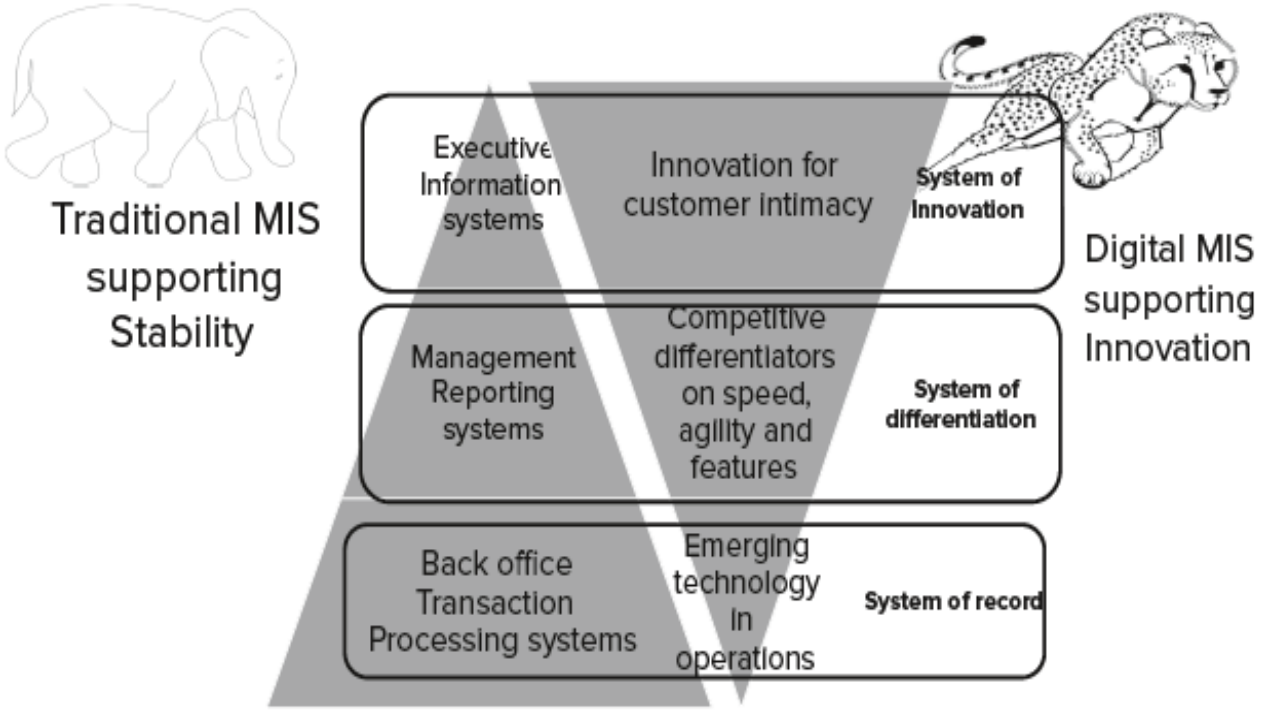
MIS in traditionally set organisations mirrors the organisation pyramid.



Changing Role of MIS in the Digital Age



MIS Development Approaches



Why Agile?

Time to Market

- Early and regular releases. Traditional projects designed to minimize and resist change wherever possible. By contrast, Distributed Scrum Development projects accept change; in fact they expect it

Revenue

- Early ROI : Iterative nature of Distributed Scrum means features are delivered incrementally, enabling some benefits to be realized early as the product continues to develop.
- The approach of fixed timescales and evolving requirements enables a fixed budget.

High Quality and Productivity

- Testing is integrated throughout the lifecycle, enabling regular inspection of the product as it develops
- Defects caught early reduce cost of quality and improve productivity

Risk Mitigation by short deliveries

- Multiple opportunities to recover from missteps
- Validation of requirements
- Confirmation of technical approach and Realistic assessment of progress

Business Engagement/Customer Satisfaction

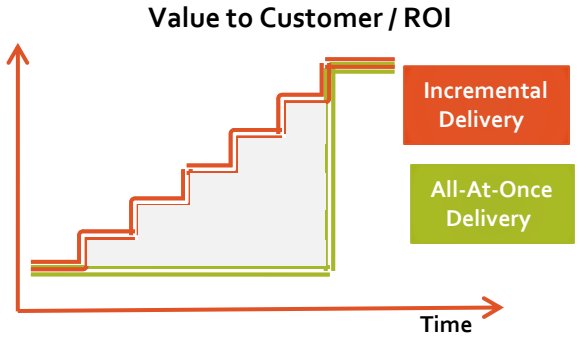
- The active involvement of a business owner, the high visibility of the product and progress, and the flexibility to change when change is needed, create much better business engagement and customer satisfaction

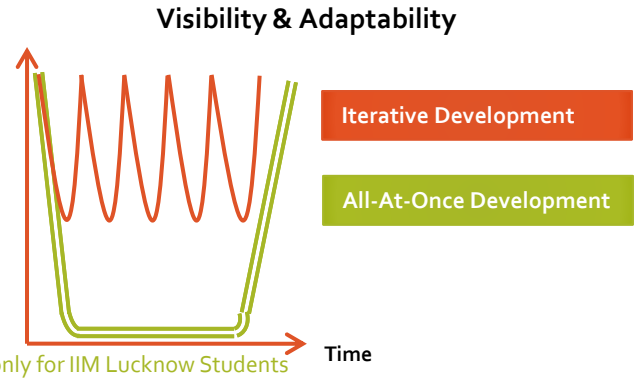
Motivated Teams

- The active involvement and collaboration make Agile development teams a much more enjoyable place for most people. The team is empowered to make decisions which in turn helps to create highly motivated and high performance teams

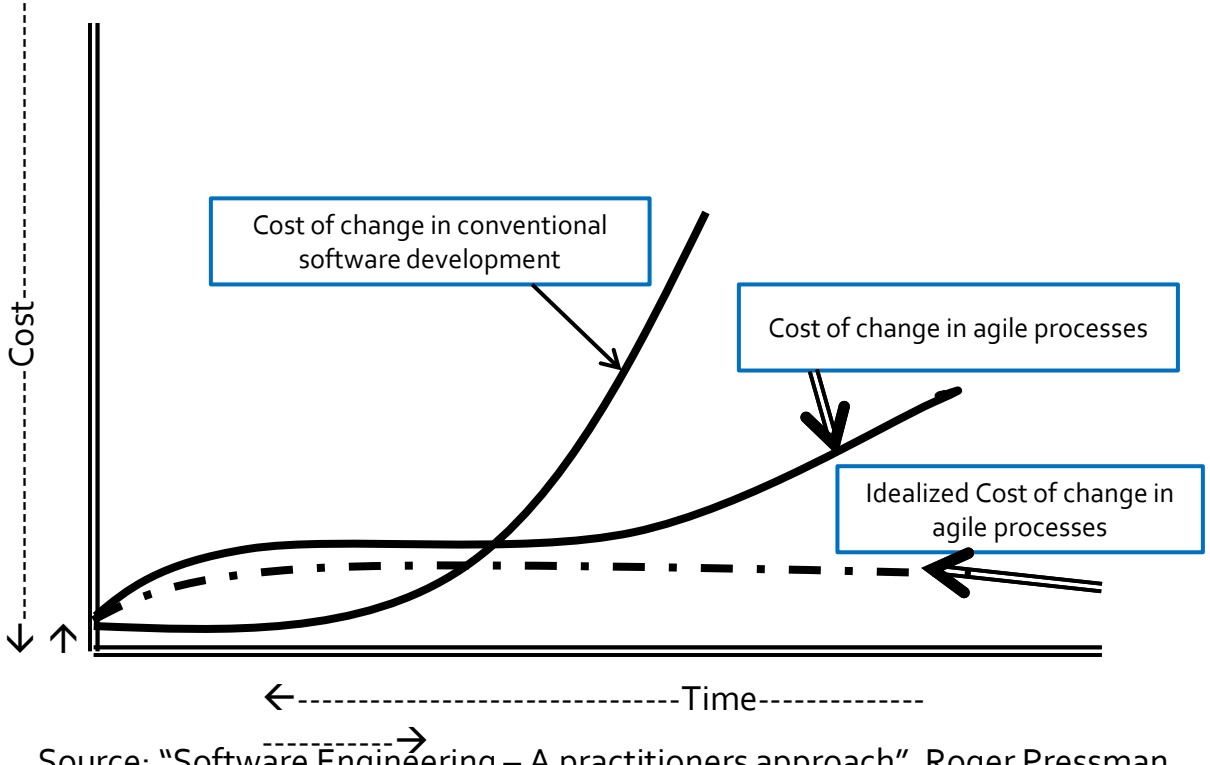
Shortcomings in Traditional Projects where Agile helps

Cycle times in months/quarters as against weeks:	All or none into production:	Decision Making and Line of sight effect	Lack of responsiveness:	Focus on documentation and signoffs:
•The plan to production cycle time spans in months/quarters and there is no means of mid path corrections and to accommodate change in the goals	•If 80% of the software is ready 100% of it is held back due to problems in the rest 20%.	•Decision making and ability to influence events is normally limited by line of sight. This leads to planning horizon mismatch and resultant impact on the efficacy of decision making	•The fixing of scope and adherence to the plan drawn impacts flexibility. Rigid change request process adds to issue	•This leads to a signoff process dependence and resultant delay. The situation is further aggravated as the teams are distributed.



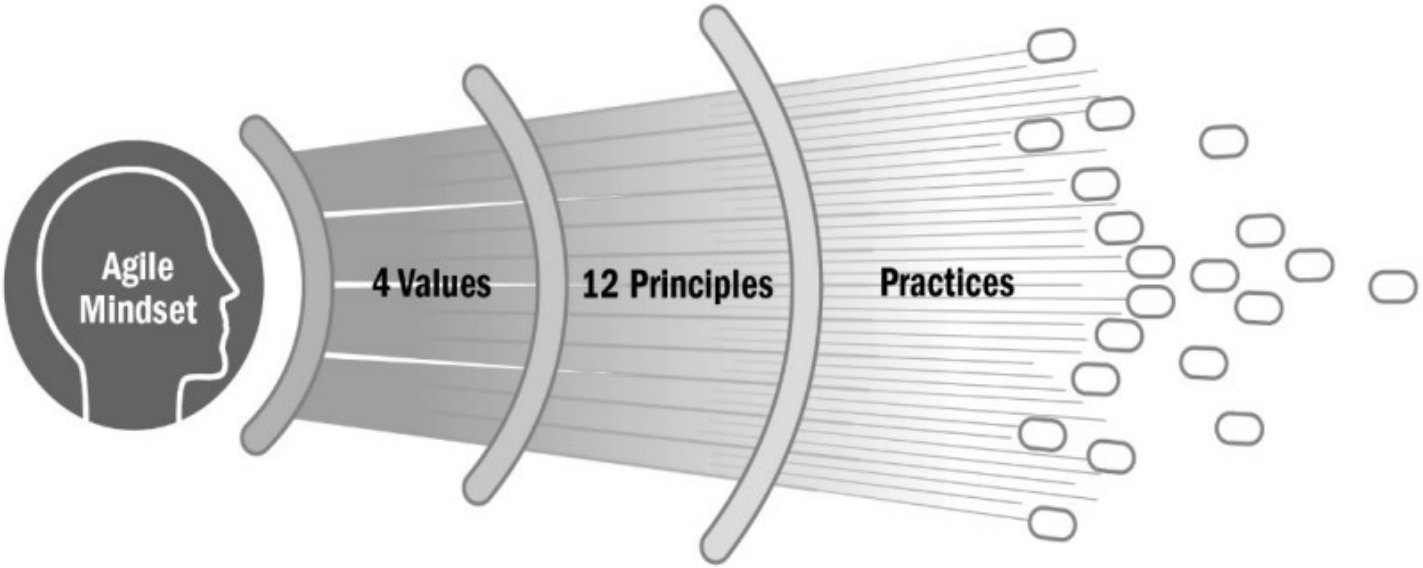


Cost of change as a function of time



Source: "Software Engineering – A practitioners approach", Roger Pressman

Agile Values, Principles and Practices



Agile is a mindset defined by values and guided by principles and manifested through many different practices. Agile practitioners select practices based on their needs.

Source: PMI Agile Practice Guide.

Agile Values

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Responding to change over following a plan

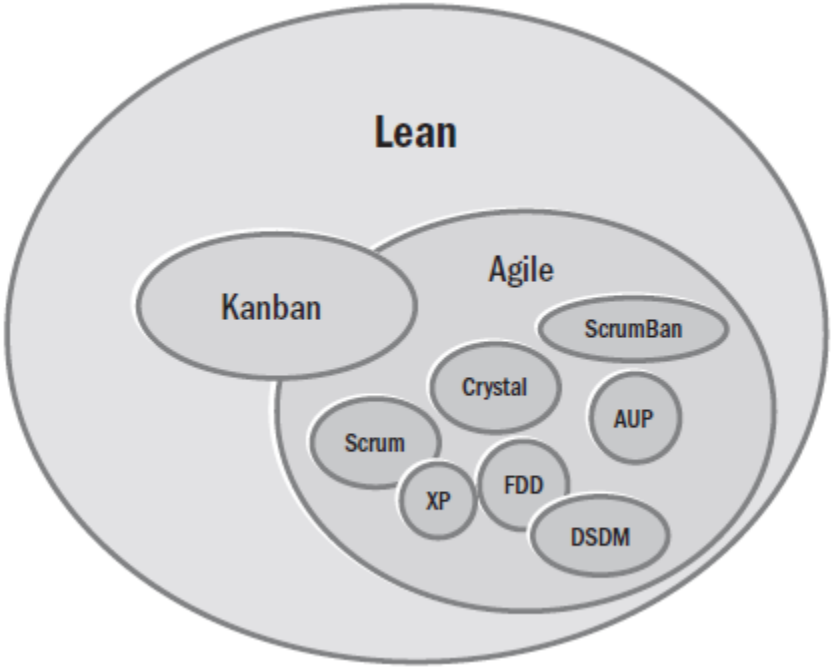
Customer collaboration over contract negotiation

Agile Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

***Source**: https://agilemanifesto.org/principles.html*

Agile is a Blanket Term for Many Approaches



***Source**: PMI PMBOK 6th Edition*

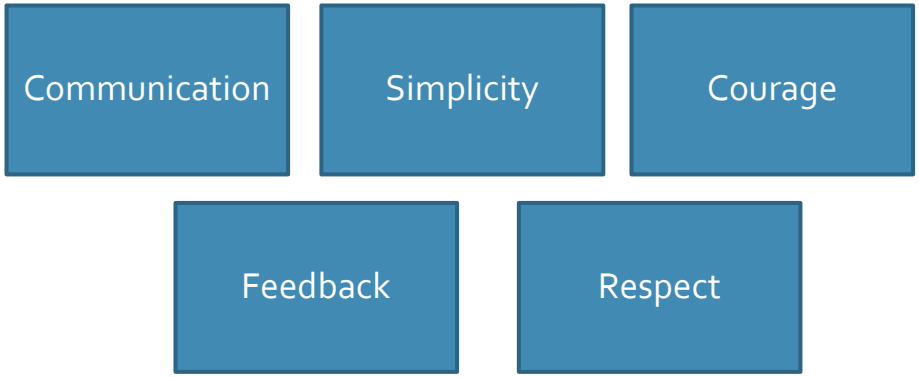
Compare and Contrast Agile Approaches

Agile methodology	Characteristics	Phases of development life cycle supported	Usage constraints
ASD	Promotes adaptive paradigm, derives principles from radical software development	Requirements, design, code, unit test, integration test, system test, acceptance test	It is more about concepts and culture rather than in-practice
AM	Agile modelling – agility and rapid development in producing sufficiently advanced models to support acute design.	None	It does not work independently but work within other methods as supplement
Crystal	Family of methods to be chosen suitable for the business needs with rules of thumb for tailoring. Flexible and configurable process	Design, coding, unit test, integration test, system test	Lack of support for mission-critical systems, distributed teams, scalability, insistence on only collocation.
DSDM	Provides control framework for rapid application development. Keeps time and resources as constant and adjust the functionality to be developed	Project inception, requirements, design, code, unit test, integration test, system test, acceptance test, system in use	Availability issue to wider audience
XP	Customer focused and close customer participation, short iterations and short release, continuous re-factoring.	Requirements, design, code, unit test, integration test, system test	Lack of attention on management practises
FDD	More emphasis on quality, frequent and tangible deliveries and accurate monitoring of project progress. Very short iterations	Requirements, design, code, unit test, integration test, system test	Focused mainly on design and implementation
ASP	The Agile Software Process model focusses on accelerated development with flexibility to include volatile requirements	Requirements, design, code, unit test, integration test, system test, acceptance test	Unable to predict effort upfront and threat of loss of focus due to changing requirements,
PP	Pragmatic programming – more of pragmatic perspective with a set of best practises	None	Does not define any concrete methodology to develop a system
SCRUM	Focusses on flexibility, adaptability, productivity, through small, self- motivated teams. Integrated project management process to overcome deficiencies in the development process	Requirements specification, integration test	Coding and all testing process not defined completely.

***Source**: Abrahamsson et al., 2010*

Extreme Programming

5 values



Roles

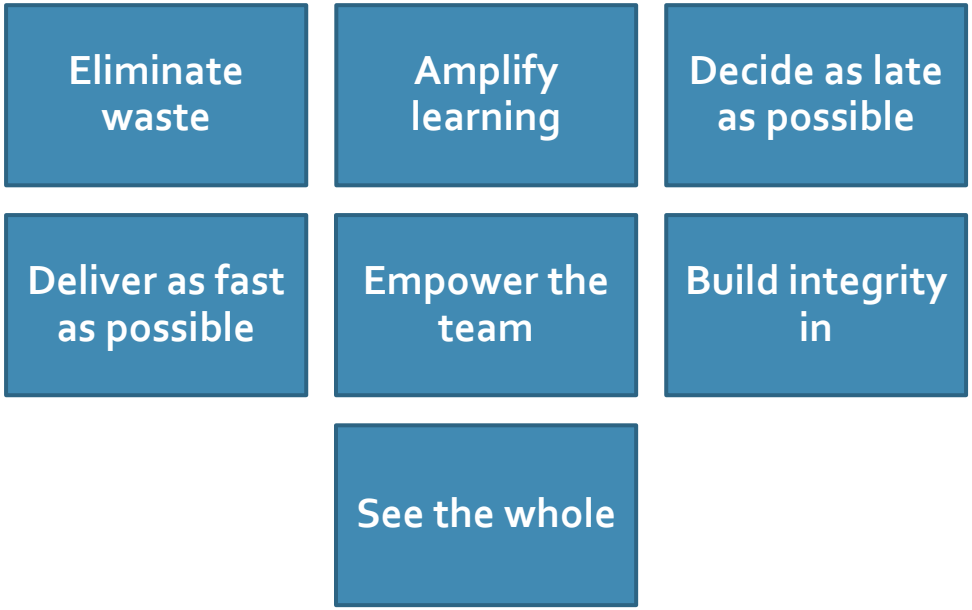
Extreme Programming

Practices

- Sit together
- Whole team
- Informative workspace
- Simple Design
- Test driven development
- Energized work
- Pair Programming
- Collective Code Ownership
- Continuous Integration
- User stories
- Frequent delivery

LEAN software development

Principles



THANK YOU