

# Agile Project Management

## 3 day workshop

## BirlaSoft

Trainer : Hari Prasad Thapliyal

(PMP, PMI-ACP, CSM, MBA, MCA, PGDOM, PGDFM, CIC, PRINCE2-Practitioner)

[PMI-ACP : #318050]

[PMP : #1563385]

## SageTechnologies



thara@sagetech.co.in



<http://www.sagetech.co.in/>



(080) – 41750777  
Koramangala, Bangalore – 560047, India

**This presentation copy is authorized only for BirlaSoft, Bangalore, SJR Tech Park employees.**

'PMI-ACP', 'PMI', and 'ACP' are registered marks of the Project Management Institute, Inc.

# **Faculty & Participants**

## **An Introduction**

Name, Company, Current Designation, Why  
Certification, Hobby

# Workshop Ground Rules

- ✓ Please keep your mobile on the silent mode. Always take your calls outside the training room.
- ✓ No corner talk! Discussions only when group discussion is allowed
- ✓ Keep your focus on the ongoing topic. Await your turn during the questionnaire round.
- ✓ Strictly follow the workshop schedule for management of time.
- ✓ There is parking lot. Write you questions and post with your name on parking lot.
- ✓ Breaks only on agreed time
  - ✓ Tea
  - ✓ Lunch
  - ✓ Tea
- ✓ Everybody need to contribute
- ✓ Use your experience only for relating the processes and best practices. To avoid confusion keep it outside of the class. Unlearning is first and biggest learning to learn something new.
- ✓ Two Bowls

# Topics

## Day 1

- History of Agile
- Agile Certifications
- The Most General APM Certificate
- The Foundation of Agile Project Management
- Project Management in General
- Agile Project Management
- Agile : Developing Mindset
- Agile : Methodologies
- Agile : Metrics & Estimation



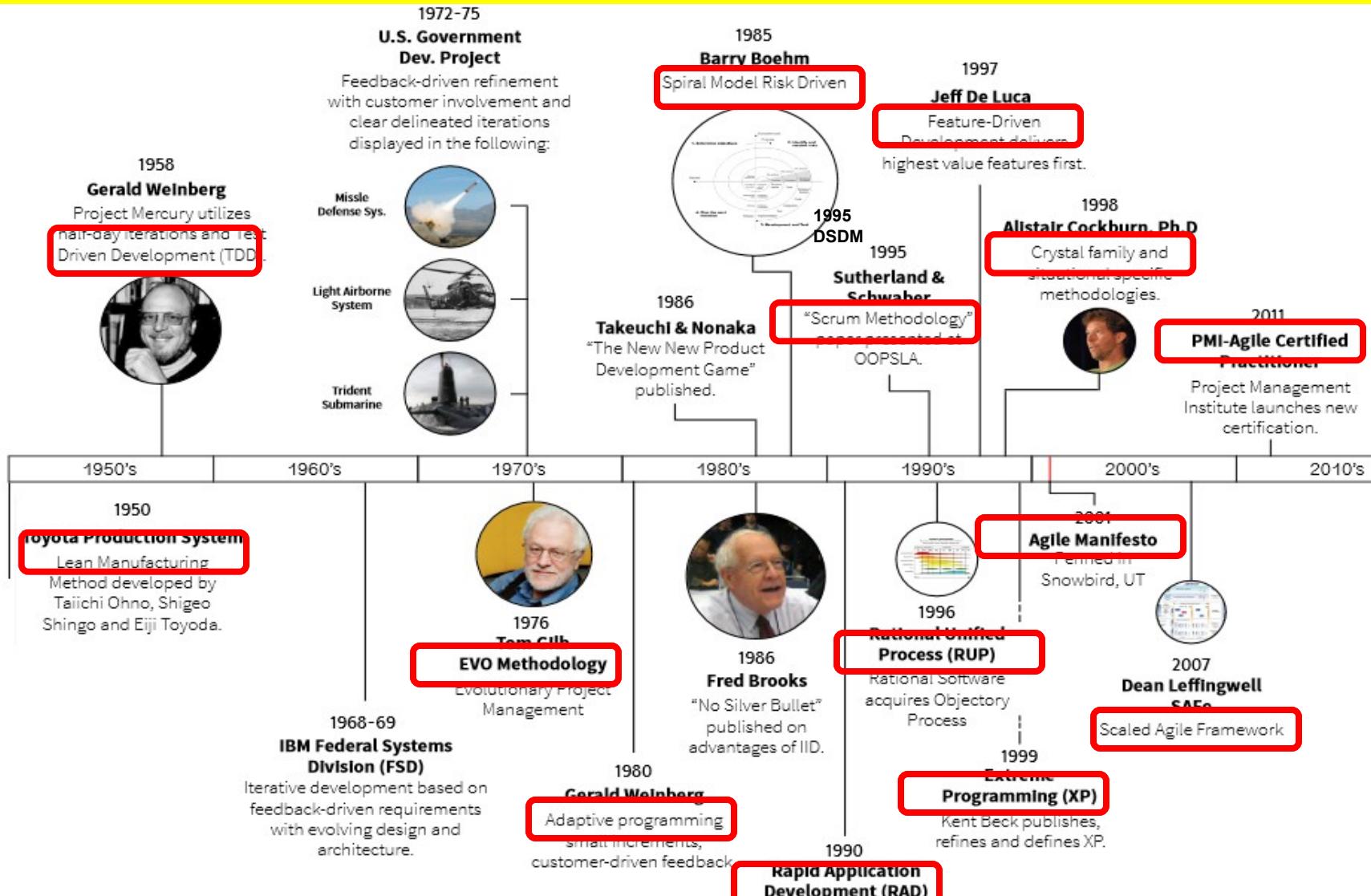
## Day 2

- Agile : Planning, Monitoring & Adapting
- Agile : Communication
- Agile : Analysis and Design
- Agile : Product Quality
- Agile : Value Prioritization

## Day 3

- Agile : Risk Management
- Soft Skills for Agile Project Manager
- Agile : Roles
- Scaling Agile
- Executing an Agile Iteration
- Professional Responsibility

# Agile History



# Agile Market of Certification

## 1- ScrumAlliance

- Certified Scrum Master (CSM)
- Certified Scrum Developer (CSD)
- Certified Scrum Product Owner (CSPO)
- Certified Scrum Professional (CSP)
- Certified Scrum Coach (CSC)
- Certified Scrum Trainer (CST)

## 2- SCRUMstudy

- Scrum Developer Certified (SDC)
- Scrum Master Certified (SMC)
- Agile Expert Certified (AEC)
- Scrum Product Owner Certified (SPOC<)
- Expert Scrum Master (ESM)
- SCRUMstudy Certified Trainer (SCT)

## 3- Scrum.org

- Professional Scrum Foundations™ (PSF)
- Professional Scrum Master™ (PSM)
- Professional Scrum Developer™ (PSD)
- Professional Scrum Product Owner™ (PSPO)

## 4- PMI

- PMI-ACP

## 5- DSDM consortium

- Agile Foundation
- Agile Advanced Practitioner
- DSDM Foundation
- DSDM Advanced Practitioner
- DSDM Trainer
- DSDM Coach

## 6- Scaled Agile Academy

- SAFe Program Consultant (SPC)
- SAFe Agilist (SA)
- SAFe Practitioner (SP)

## 7- ICAgile

- The ICAgile Certified Professional
- Enterprise Agile Coaching
- Agiler Leadership
- Business Value Analysis
- Value Management (Pro)
- Value Management (Expert)
- Project Management
- Adaptive Management (Pro)
- Agile Management (Expert)
- Agile Coaching (Pro)
- Agile Coaching (Expert)
- Agile Team Facilitation
- Agile Programming
- Agile Development
- Agile Software Design
- Agile Testing (Pro)
- Agile Testing (Expert)
- Agile Test Automation

# Topics

## Day 1

- ✓ History of Agile
- ✓ Agile Certifications
- The Most General APM Certificate
- The Foundation of Agile Project Management
- Project Management in General
- Agile Project Management
- Agile : Developing Mindset
- Agile : Methodologies
- Agile : Metrics & Estimation

## Day 2

- Agile : Planning, Monitoring & Adapting
- Agile : Communication
- Agile : Analysis and Design
- Agile : Product Quality
- Agile : Value Prioritization

## Day 3

- Agile : Risk Management
- Soft Skills for Agile Project Manager
- Agile : Compliance
- Agile : Roles
- Scaling Agile
- Executing an Agile Iteration
- Professional Responsibility

# The Most General Agile Project Management Certificate

is

PMI-ACP

# Topics for Examination

## 6 Domains for PMI-ACP Exam

1. Value Driven Delivery
2. Stakeholder Engagement
3. Boosting team performance practices
4. Adaptive planning
5. Problem detection and resolution
6. Continuous Improvement (Product, Process, People)

**mnemonics : ACP for Stakeholder Value Boosting**

# 1- Value Driven Delivery

## Define Positive Value

- To define business values
- To optimize business values and surface new information
- To shared definition of “done”
- To tailor Process

## Incremental Development

- To achieve rapid ROI and quick feedback
- To plan exposure of problems at early and minimal cost
- To release high-quality deliverables to stakeholders
- To optimize Benefit-to-Cost of the developed system
- To reduce the overall cost of incremental development

## Avoid Potential Downsides

- To manage unknown affect of risks on project outcome.
- To ensure stakeholders are aware and their expectations are managed
- To ensure functionality meets customer need

## Prioritization

- To balance stakeholder values, business value, and residual risk
- To reflect changes in the environment and stakeholder understanding
- To minimize the impact of failure

# 2- Stakeholder Engagement

## Stakeholder Needs

- To ensure that team's knowledge about an agreed, prioritized feature set reflecting all stakeholder's interests
- To unimpeded flow of value throughout the lifespan of the project

## Stakeholder Involvement

- To promote effective collaboration and participation of stakeholders on project activities
- To ensure new stakeholders on the project are appropriately engaged
- To continually assess the changes
- To maintain proper stakeholders' involvement

## Stakeholder Expectations

- To align expectations and build trust
- To help the business stakeholders make informed decisions about scope, time and cost
- To give greater assurance to stakeholders that project outcomes will help them meet their business objectives.

# 3- Boosting Team Performance Practices

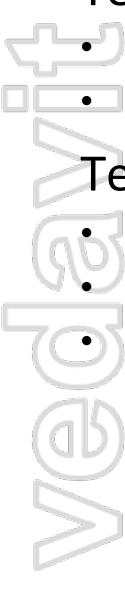
## Team Formation

- To remove fear of conflict and strengthen member's commitment to shared outcomes.
- To enable the team to deliver on their commitment.
- To maximize teamwork and reduce bottlenecks.
- To ensure cohesion between team members

## Team Empowerment

- To manage the project's complexity and produce effective solutions
- To ensure that team learn and continually improve the way they work
- To ensure that team remains motivated and productive throughout the project

## Team Collaboration

- 
- To allow team to take responsibility for outcomes and improve their effectiveness
  - To reduce the cost of miscommunication and rework

## Team Commitment

- To establish a predictable outcome and optimize the value delivered
- To ensure that the team understand how their objectives fit into the overall goals of the project
- To gain a better understanding of team's velocity and commitment

# 4- Adaptive Planning

## Levels of Planning

- To support the necessary level of understanding
- To gain increased levels of commitment
- To set and manage sponsor expectations

## Adaptation

- To adapt as per the business need
- To maximize business value delivered

## Estimation

- To encourage the team to create estimates
- To manage stakeholder expectations
- Velocity/Throughput Cycle Time
- To gauge progress and extrapolate completion
- To ensure team does not over commit

# 5- Problem Detection and Resolution

- To surface problems and impediments that are slowing the team down or preventing its ability to deliver values,
- To identify risks and create mitigation strategies,
- To create high productivity culture
- To elevate accountability and track ownership and resolution status of risks and impediments.
- To manage the expectations of the impacted stakeholders

## 6- Continuous Improvement (Product, Process, People)

- To ensure that team is effective within established organizational norms
- To improve process, individuals, and team effectiveness.
- To improve efficiency within the existing process with a goal of keeping a team together for long term
- To become more efficient
- To avoid re-occurrence of problems identified, improving the effectiveness of the organization as a whole.
- To improve overall team effectiveness and lowering risk around knowledge silos
- To identify opportunities to reduce waste.
- To discover more efficient and effective ways of working.

# 11 Books Recommended by PMI for PMI-ACP Exam

1	Agile Retrospectives: Making Good Teams Great, Author: Esther Derby, Diana Larsen, Ken Schwaber, ISBN #0977616649
2	Agile Software Development: The Cooperative Game – 2nd Edition, Author: Alistair Cockburn, ISBN #0321482751
3	The Software Project Manager's Bridge to Agility, Author: Michele Sliger, Stacia Broderick, ISBN #0321502752
4	Coaching Agile Teams, Author: Lyssa Adkins, ISBN #0321637704
5	Agile Project Management: Creating Innovative Products – 2nd Edition, Author: Jim Highsmith, ISBN #0321658396
6	Becoming Agile in an imperfect world, Author: Greg Smith, Ahmed Sidky, ISBN #1933988258
7	Agile Estimating and Planning, Author: Mike Cohn, ISBN #0131479415
8	The Art of Agile Development, Author: James Shore, ISBN #0596527675
9	User Stories Applied: For Agile Software Development, Author: Mike Cohn, ISBN #0321205685
10	Agile Project Management with Scrum, Author: Ken Schwaber, ISBN #073561993X
11	Lean-Agile Software Development: Achieving Enterprise Agility, Author: Alan Shalloway, Guy Beaver, James R. Trott, ISBN #0321532899

# Topics

## Day 1

- ✓ History of Agile
- ✓ Agile Certifications
- ✓ The Most General APM Certificate
- The Foundation of Agile Project Management
- Project Management in General
- Agile Project Management
- Agile : Developing Mindset
- Agile : Methodologies
- Agile : Metrics & Estimation

## Day 2

- Agile : Planning, Monitoring & Adapting
- Agile : Communication
- Agile : Analysis and Design
- Agile : Product Quality
- Agile : Value Prioritization

## Day 3

- Agile : Risk Management
- Soft Skills for Agile Project Manager
- Agile : Compliance
- Agile : Roles
- Scaling Agile
- Executing an Agile Iteration
- Professional Responsibility

# **Discussions !**

# The Foundation of Agile Project Management

is  
Values & Principles

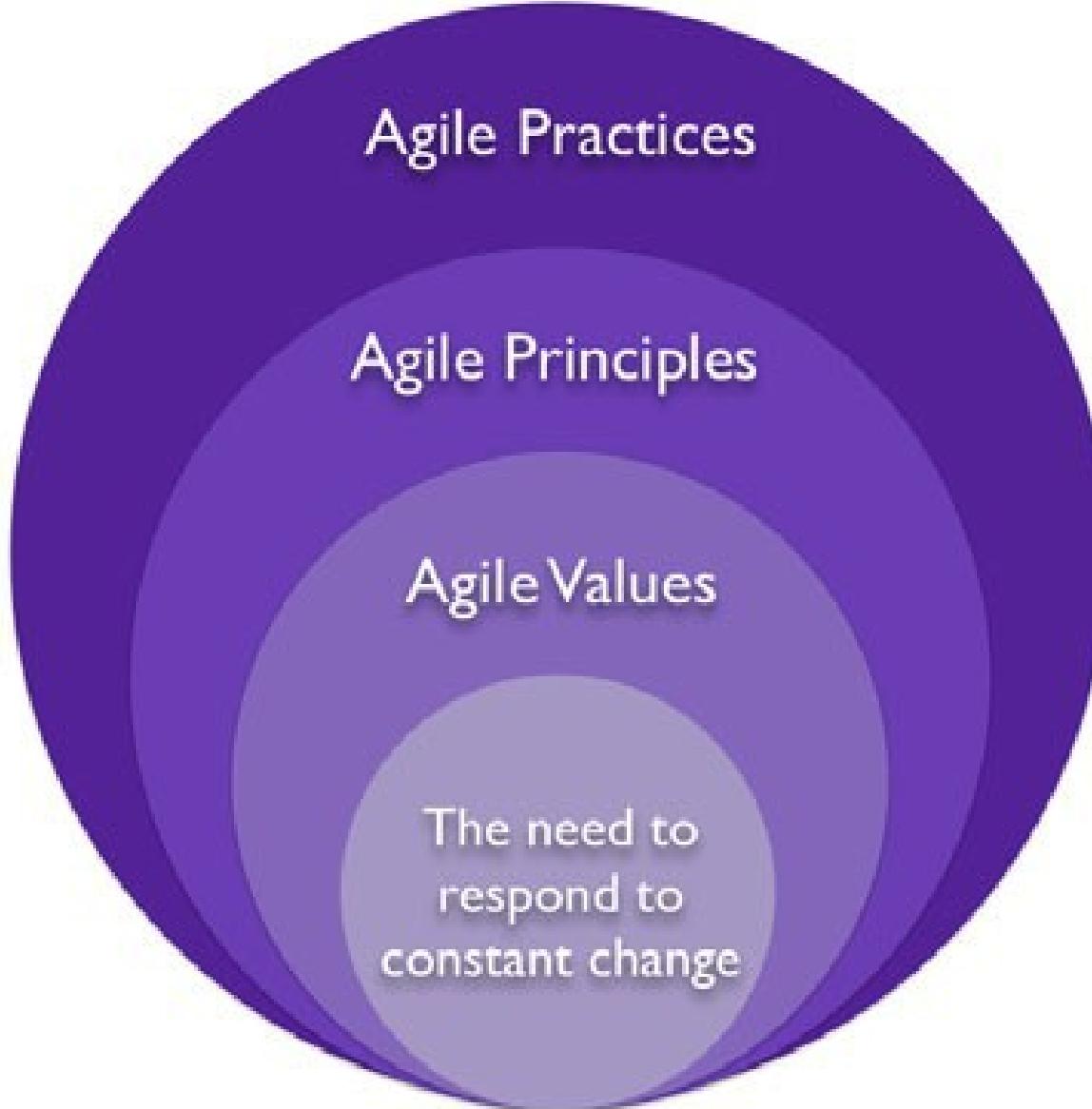
# Values, Principles & Practices

- **Values** are sets of beliefs about good and bad, right and wrong. e.g. Abortion is wrong. God is kind. Satan is bad. Parents are always our well wishers.
- **Principles** can be described as unwritten rules or laws that are universal in nature. Having a set of principles is like having a compass in the journey of life when we feel that we are lost or unable to find the right path. One must have clear cut principles about concepts such as fairness, justice, equality, truthfulness, honesty so that when situation arise one can stand tall even in dark.
- **Values** are the concepts that drive your choices. Principles are the manner in which you carry out your values.
- **Practices** are activities that are used to manifest or implement the principles and values. Going temple or church, giving donations, helping poor etc.

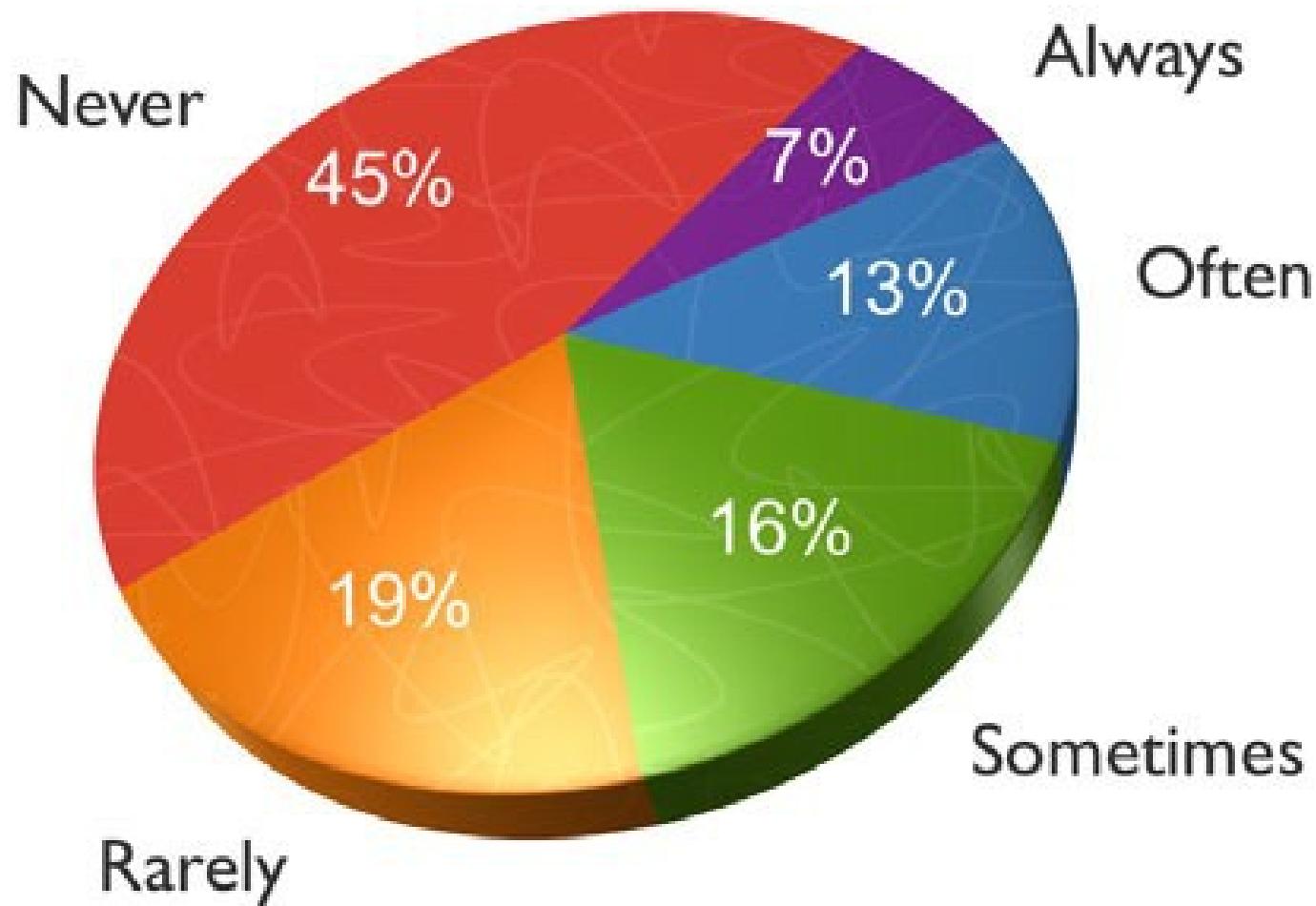
# Some Generic Terms

- **Values** : Values are **unchanging** foundation in Agile.
- **Principles** : Principles are **unchanging** foundation in Agile and they are there to live agile values.
- **Practices** : Practices are values and principle based but specific to industry.
- **Knowledge** : Which you acquire through trainings
- **Skills** : Which you acquire through practicing the knowledge on which you have conviction (values and principles)
- **Tools & Techniques** are external to the person they are not virtues of person.

# Agile : Values, Principles, Practices



# Features Usage of a Typical Application



*Standish Group Report*

# Which Methodology is Best?

- Known Agile development methodologies like Extreme Programming (XP), Scrum, Lean, and Feature Driven Development (FDD). Every methodology consist of a set of Agile practices that have a certain synergy.
- No specific set of Agile practices is defined—it's anti-Agile to say that there is a defined set of practices and that no new practices can be created. Organizations create different Agile practices or tailor existing Agile practices to address specific organizational or team needs. Teams may also need to be creative and come up with new Agile practices to achieve agility while adhering to organizational constraints.
- One methodology may have practices which are more related to project management, other may be related to technology other may be business. But no one is better than other.
- There are more than **60 widely known practices**

# Practices : Requirements

Product Vision / Vision Statement	SA
Product Backlog	SA
User Stories	SA,AM,XP
Use Cases	AM,XP
Usage Scenarios	AM
Personas	AM
Planning Poker	
Requirement Prioritization	AM

# Practices : Design

Architectural Spikes / Spike Solutions

Domain Driven Design

IXP

Emergent Design / Evolutionary Design

IXP

CRC Cards

AM,XP

Design by Contract

System Metaphor

XP

# Practices : Construction

Coding Style / Coding Guidelines / Coding Standard	IXP
Test Driven Development	XP
Behavior Driven Development	XP
Pair-Programming / Pairing	IXP
Refactoring	XP,IXP
Collective Code Ownership	XP,IXP
Daily Builds / Automated Builds / Ten-Minute Builds	
Continuous Integration	XP,IXP
Code Reviews / Peer Reviews	
Software Metrics / Code Metrics & Analysis	
Source Control / Version Control	
Issue Tracking / Bug Tracking	
Configuration Management	
Frequent Delivery / Frequent Releases	XP,IXP

# Practices : Testing

Unit Testing	XP
Smoke Testing / Build Verification Test	
Integration Testing	
System Testing	
Exploratory Testing	
Test Automation	SA
Storytesting / Acceptance Criteria / Acceptance Testing	AM,XP,IXP

# Practices : Planning

Timeboxing / Fixed Sprints / Fixed Iteration Length	XP
Release Planning	XP
Iteration Planning / Planning Game / Sprint Planning	SA,XP,IX P
Sprint Backlog	SA
Task Board	SA
Definition of Done / Done Done	SA
Daily Stand-up Meeting / Daily Scrum	SA,XP
Velocity	XP
Value Stream Mapping	

Vedavit

# Practices : Sharing & Adapting

Root Cause Analysis / 5 Whys

Burn Down Charts / Burn Up Charts

SA

Big Visible Charts / Information Radiators

Retrospective / Reflection Workshop

SA, IXP

Sprint Review / Iteration Demo

SA

# Practices : Organization

Small Team	IXP
Cross-Functional Team	
Self-Organizing Team / Scrum Team	
Colocated Team / Sitting Together / Common Workspace	SA, IXP
On-Site Customer / Product Owner	SA
Scrum Master	SA
Sustainable Pace	IXP
Move People Around	XP
Scrum of Scrums	SA

# Signatories of Agile Manifesto

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

# **Discussions !**

# Topics

## Day 1

- ✓ History of Agile
- ✓ Agile Certifications
- ✓ The Most General APM Certificate
- ✓ The Foundation of Agile Project Management
- Project Management in General
- Agile Project Management
- Agile : Developing Mindset
- Agile : Methodologies
- Agile : Metrics & Estimation

## Day 2

- Agile : Planning, Monitoring & Adapting
- Agile : Communication
- Agile : Analysis and Design
- Agile : Product Quality
- Agile : Value Prioritization

## Day 3

- Agile : Risk Management
- Soft Skills for Agile Project Manager
- Agile : Compliance
- Agile : Roles
- Scaling Agile
- Executing an Agile Iteration
- Professional Responsibility

Lets us Understand

# Project Management

in  
General

# Topics

1. Project
2. Project Management
3. Program Management
4. Portfolio Management
5. Program Management Office
6. Stakeholders
7. Organization Types
8. Responsibilities of a Project Manager

# What is Project???

Introduction / Project Management

Vedavit

# What is Project?

**Project – A temporary endeavor undertaken to create a unique product, service or result**

How Temporary?

- Has a definite *beginning and end*, not an on-going effort
- Ceases when objectives have been attained
- Team is *disbanded* upon project completion

Unique?

- The product or service is *different* in some way from other product or services
- Product characteristics are *progressively elaborated*

# Project .....

- Is goal oriented (verifiable and measurable)
- Finite duration with a beginning and end
- Uniqueness to a great extent and related uncertainties
- Coordinated undertaking of interrelated activities
- Performing the activities involve resources
- Resources cost money

# “Projects” different from “operations”?

## Projects

- Permanent Project Charter
- Catalyst for change
- Unique product or service
- Heterogeneous teams
- Start and end date
- Progressive elaboration

## Operations

- Semi-permanent charter
- Maintains status quo
- Standard product or service
- Homogeneous teams
- Ongoing
- Predefined product

# Project Constraints



# Project Management?

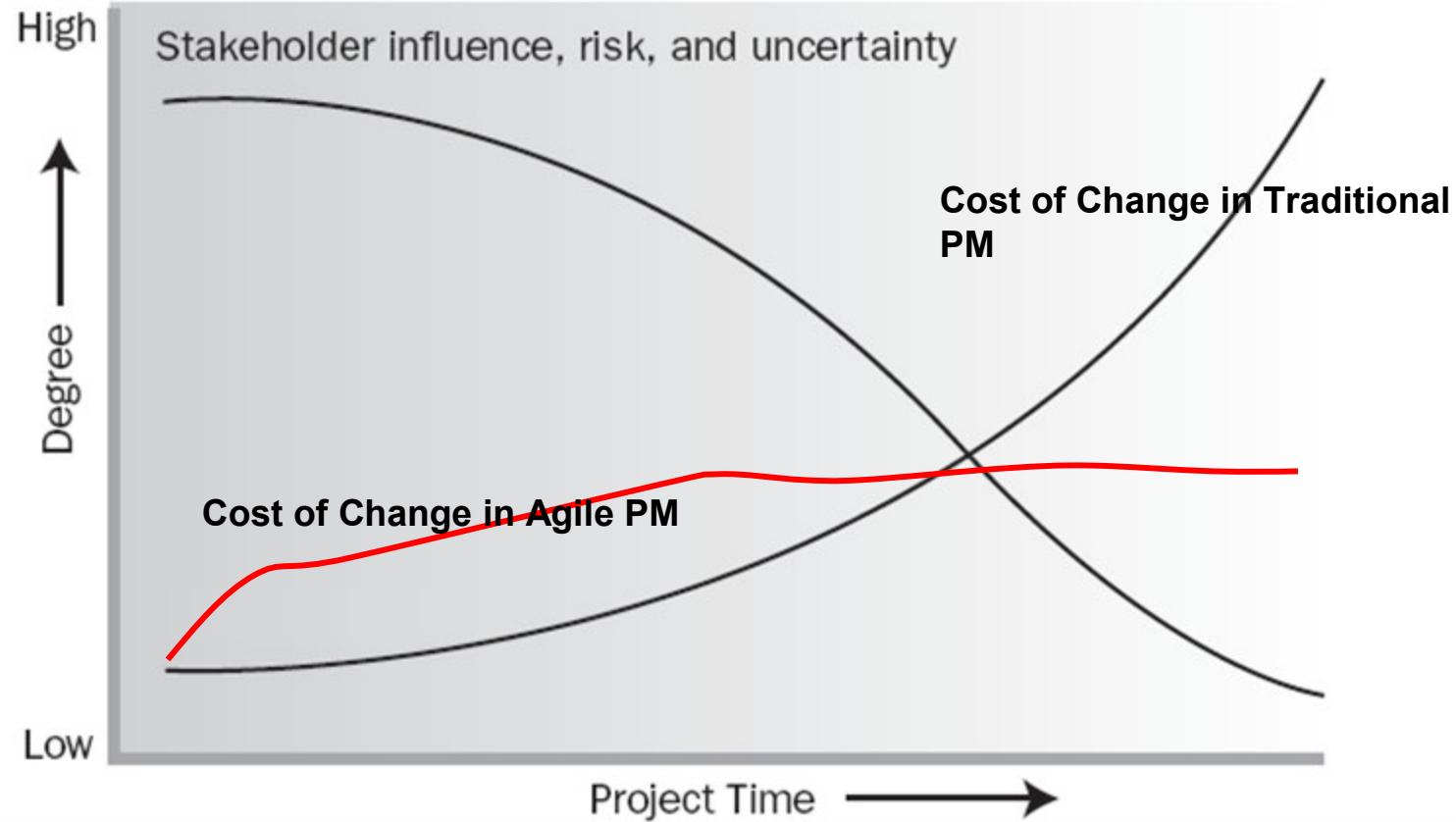
The application of **knowledge, skills, tools and techniques** to project activities in order to **meet the project requirements**

**Requirements are put by**

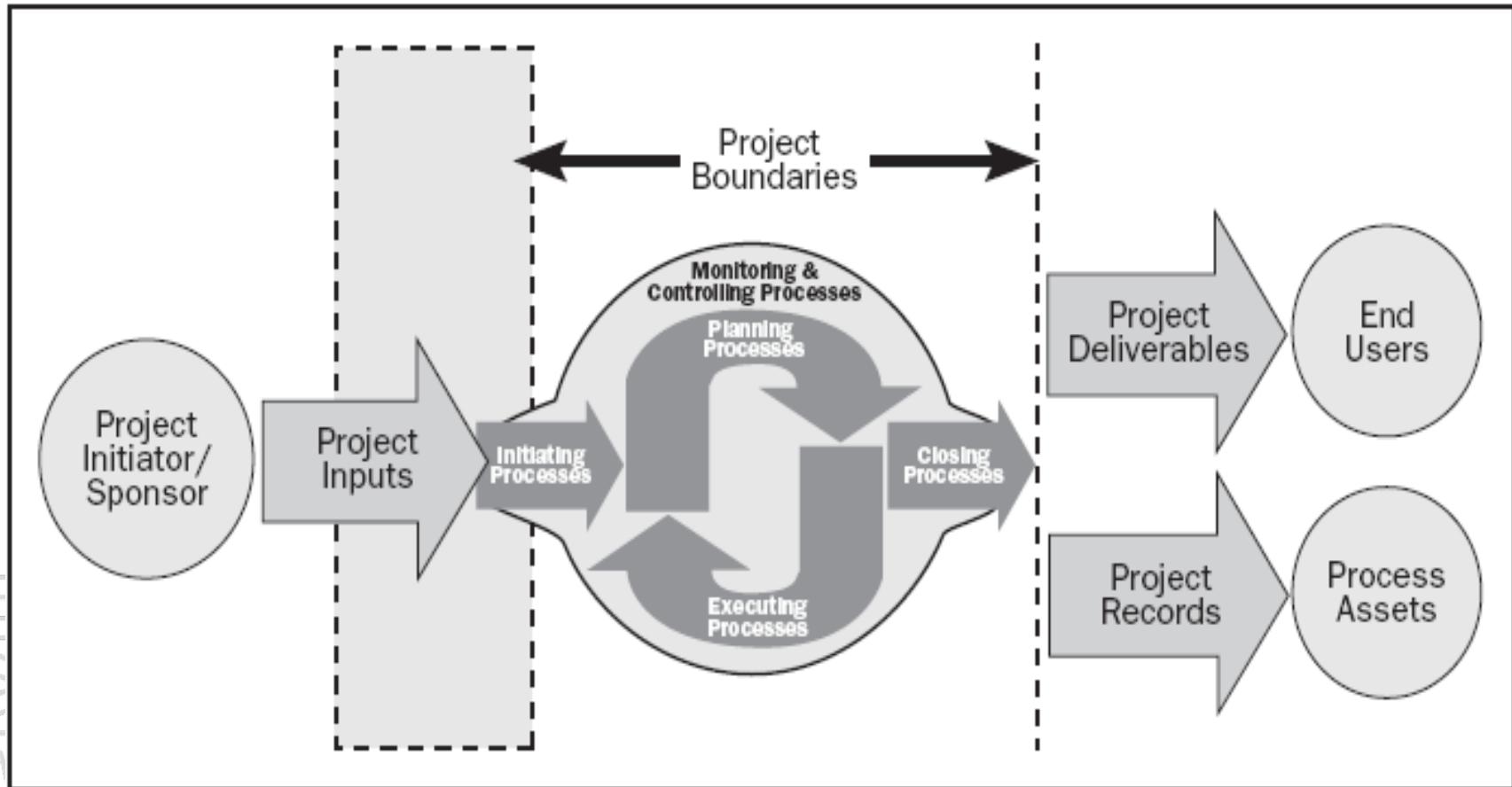


**Stakeholders**

# Impact of Variables based on Time



# Project Boundaries



# What is Program?

Group of related projects managed in a coordinated way to obtain benefits and control not available from managing them individually

# Project Portfolio Management

Collection of projects or programs and other work that are grouped together to facilitate effective management of that work to meet strategic business objectives

Portfolio manager is responsible for prioritizing project or programs, seek funding for his portfolio project

# What is PMO?

- ✓ Managing shared resources across all project administered by PMO
- ✓ Identifying and developing project management methodologies, best practices and standards
- ✓ Coaching, mentoring, training and oversight
- ✓ Monitoring compliance with project management standards, policies, procedures and templates via project audits
- ✓ Developing and managing project policies, procedures, templates and other shared documentation (organizational process assets), and
- ✓ Coordinating communication across projects

# Who are Stakeholders?

Persons or organizations who are actively *involved* in the project or whose interests maybe positively or negatively *affected* by the performance or completion of the project



# Who are Stakeholders?

Introduction / Project Management



Source PMBOK Guide Version 5.0

# **Discussions !**

# Topics

## Day 1

- ✓ History of Agile
- ✓ Agile Certifications
- ✓ The Most General APM Certificate
- ✓ The Foundation of Agile Project Management
- ✓ Project Management in General
- Agile Project Management
- Agile : Developing Mindset
- Agile : Methodologies
- Agile : Metrics & Estimation



## Day 2

- Agile : Planning, Monitoring & Adapting
- Agile : Communication
- Agile : Analysis and Design
- Agile : Product Quality
- Agile : Value Prioritization

## Day 3

- Agile : Risk Management
- Soft Skills for Agile Project Manager
- Agile : Compliance
- Agile : Roles
- Scaling Agile
- Executing an Agile Iteration
- Professional Responsibility

Let us understand

# Agile Project Management

# Topics

1. Modern Project Management Challenges
2. How to address modern day PM challenges?
3. Challenges in Starting Agile Project Management
4. What is Agility?
5. Which Project Management Methodology is best?
6. Type of Systems
7. Complex Adaptive Systems (CAS)
8. Defined vs Empirical Processes
9. Values in Agile Manifesto
10. Agile Principles
11. When Does Agile Methodologies Works Best?
12. Value Delivery & Project Life Cycle

# Before we go ahead...

Lets see one case study on agile project management

# Modern Project Management Challenges

- Uncounted uncertainties
- Very tough to negotiate with stakeholders a change in baselined plan
- Product value realization at the end of project lifecycle
- Huge difference between expectations of end user, customer and sponsor
- Typically development team is isolated from business scenario and it becomes very difficult to implement change request
- Technology and project environment changes during project execution
- Customer does not want to hear about new timelines even after requirement changes. Because project is baselined!
- Customer end up paying more because critical dates missed, non-usuable product features, less-value product
- Not enough decentralized power stations to make decisions during project execution
- Execution team and project management teams are different. Execution team does not have power and execution does not happen as per initial plan
- Work product delivered at the end of every phase is not usable work product. Typical it is some paper prototype, document or some other thing.

# How to address modern day PM challenges?

- Involve end user
- Engage relevant stakeholders
- Produce in increment
- Deliver high value feature first
- Take frequent feedback and allow customer to change original requirements
- Allow customer to prioritize
- Get commitment from team for valuable product not of activities
- Employ the power of level of planning
- Involve team in risk identification and responding to risk
- Transparency in project management
- Continuous improvement
- Learn quickly

# Challenges in Starting Agile Project Management

- Team
  - Getting experienced team member
  - Getting 100% committed team
  - Getting collocated team in early agile adoption
  - Team's mindset shifting from action to delivery
  - A cross disciplined team with generic skills
- Environment
  - Trust building
  - Open communication with customer
  - Keeping politics out
- Infrastructure & Support
  - Active risk management framework
  - Robust, flexible and adaptable configuration and data management systems
  - You need a variety of **communication, collaboration, management** and **development tools**.  
Therefore a culture is required to support and facilitate this endeavors.
  - Management need a framework from where they get to know what is happening in the project.
  - Upfront training
  - Supporting in early period when velocity is less

# What is Agility?

- Agility is about quickly responding to changes
- Learning quickly from mistakes and incorporate lessons learned
- Being proactive
- It helps in all aspects of success- Personal, Technical and Organizational

# What is Agility?

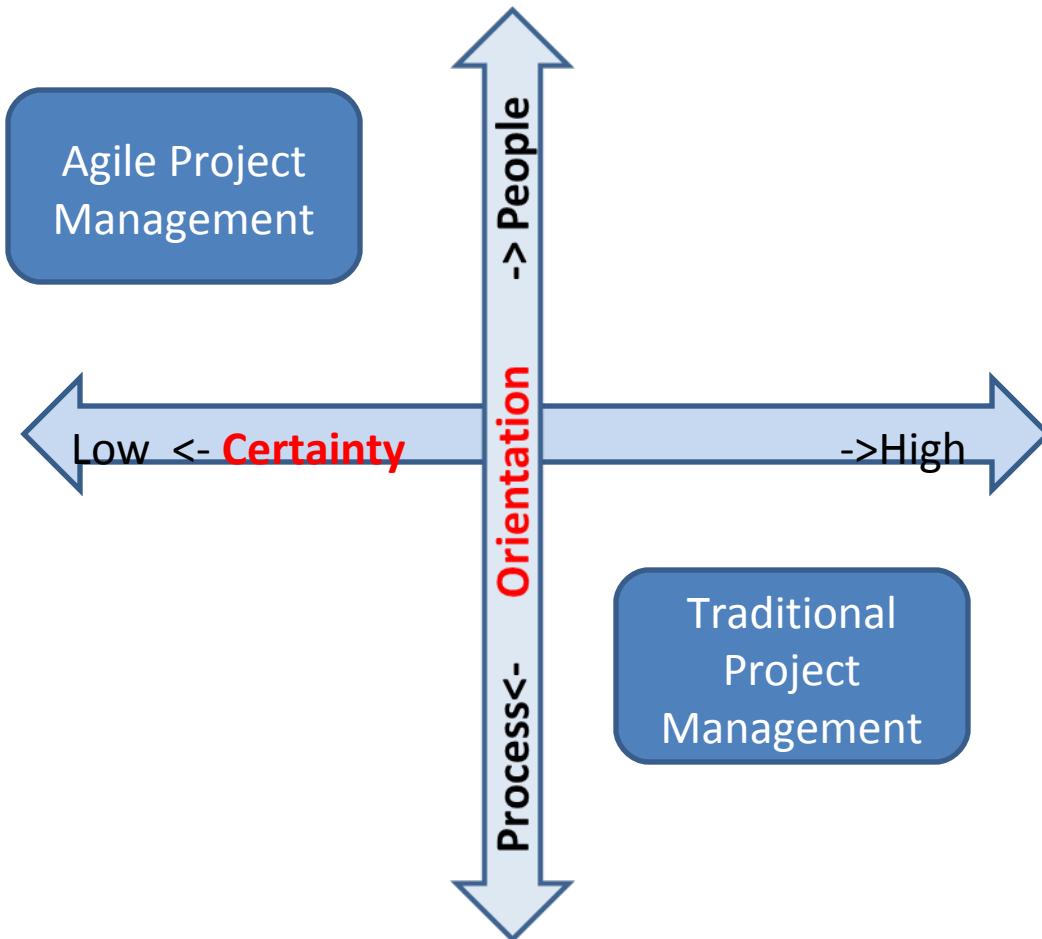
- How agility is possible with complex, bureaucratic, gigantic processes.
- Traditional processes are reactive in nature, they learn very late, respond very slow

# What is Agility?

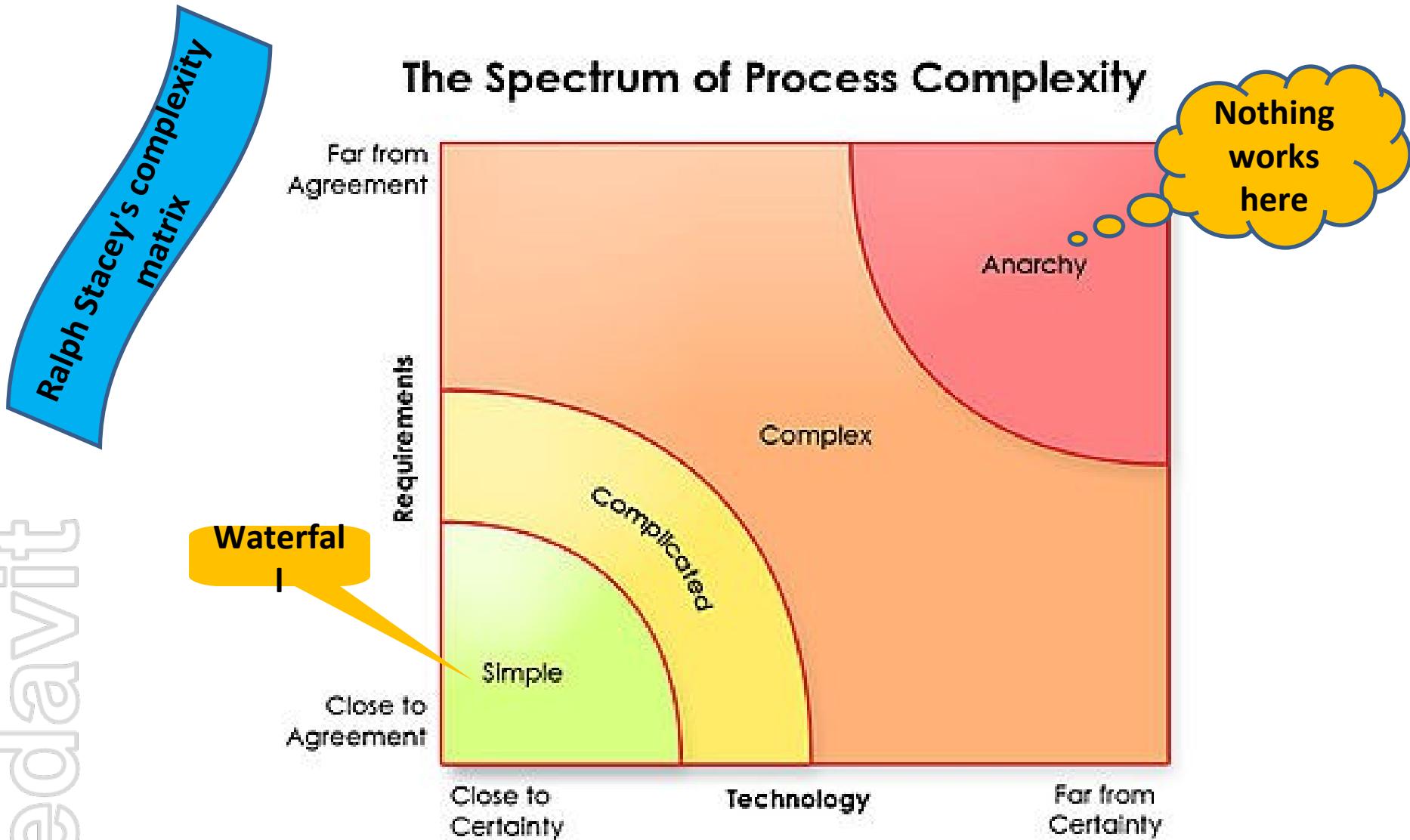
- This refinery along with township in Jamnagar, Gujarat, India was completed in 1999 in 36 months.
- There was the time when over 100,000 people working on the project on any particular day and sponsor was paying rental for a crane Rs 10 million every day.



# Which PM Methodology is best?



# Which PM Methodology is best?



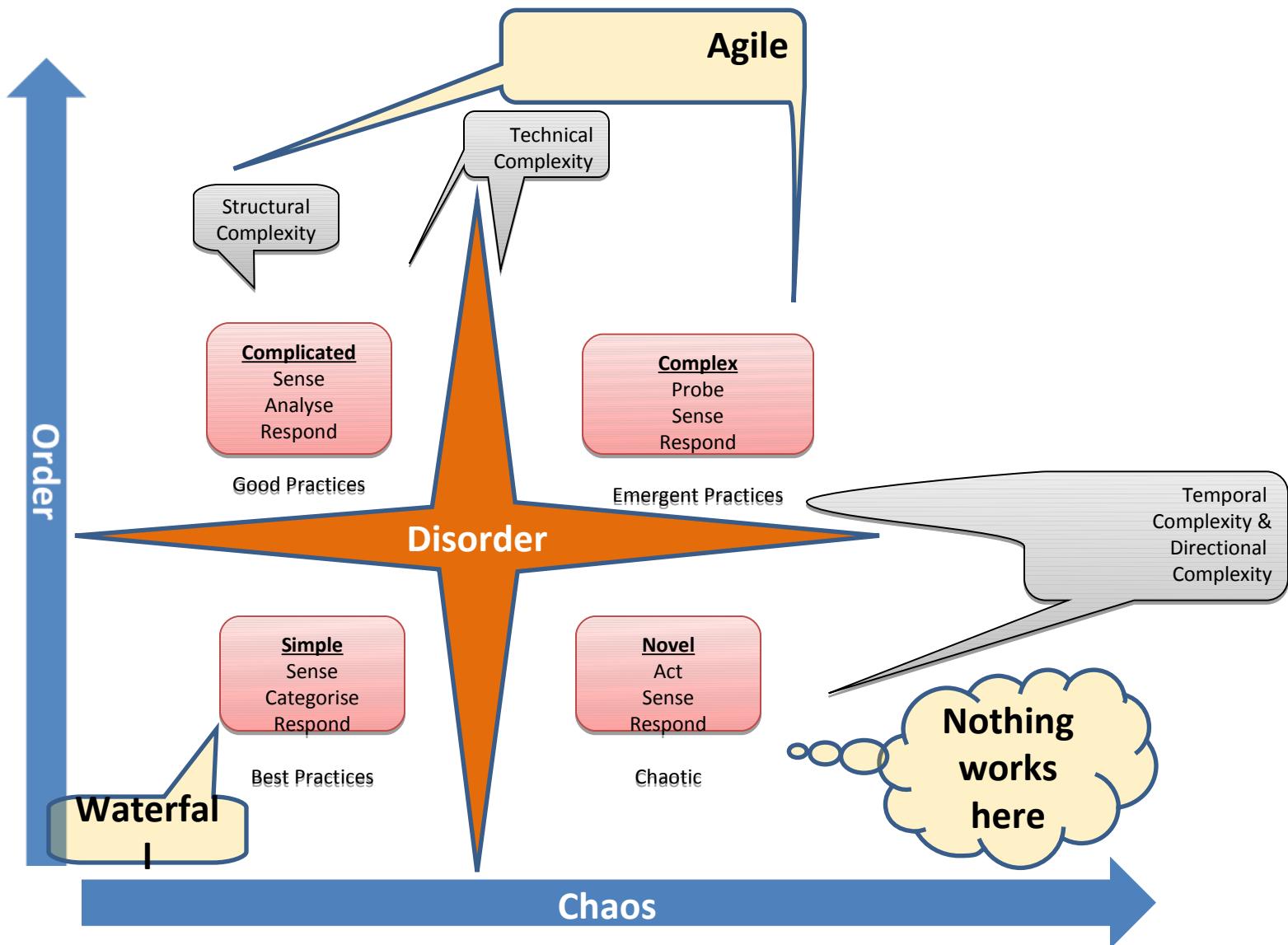
# Types of Systems

Act	Probe	Sense	Analyze	Categorize	Respond	
✓		✓			✓	Chaotic
	✓	✓			✓	Complex
		✓	✓		✓	Complicated
		✓		✓	✓	Simple

# Types of Systems

- In your kitchen you need to make tea but you realized there is no water, milk, tea bag, sugar! First you need to sense. Then categorise to solve.
  - Best practices will help here. No change is required in practices. **Simple.**
- Your school going boy is not interested in going school but you want him to go. First you need to sense the problem.
  - Good practices learned from your elders and friends will help you here. Little tailoring may be required. **Complicated**
- As a head of department you want to remove corruption from your department. First you need to probe.
  - Practices keep emerging. **Complex.**
- Bus, Scooter, Cycle, Tram, Car, Pedestrian using the same road to commute is chaos. First you need to act.
  - No practice helps here. **Chaotic.**

# Types of Systems



# Complex Adaptive Systems

- Think about how newly born child is going to learn about talking, walking, learning, eating etc.
- What process you use to climb mountain?
- In the last couple of decades scientists and managers have articulated a profound shift in their view about how organisms and organizations evolve, respond to change, and manage their growth.
- Complex Adaptive Systems theory is one of the root threads of agile development. The concepts about how biological systems evolve and adapt have relevance, if only metaphorically, to organizations and how they evolve and adapt.
- Creativity and innovation are the emergent results of well-functioning agile teams.
- In complex system things get done because people adapt not because they blindly follow
- Former Visa International CEO **Dee Hock** coined the word “**chaordic**” to describe both the world where Order exists in Chaos.

# Defined vs Empirical Processes

## Defined Processes

- Assumes that every piece of work is completely understood
- Input is well-defined
- A set of well-defined input produces same output every time within known variance limit
- Has tightly coupled steps
- No checkpoint and feedback steps

# Defined vs Empirical Processes

## Empirical Processes

- Relies on frequent inspections and adaption
- Applies to those process which are loosely defined because of their complexity
- Understand that output of a process can be unpredictable and unrepeatable

# Value in 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.

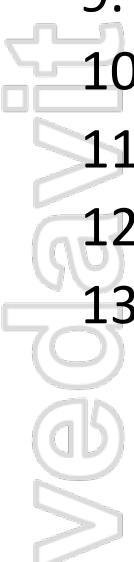
[www.agilemanifesto.org](http://www.agilemanifesto.org)

# Agile Principles

1. Satisfy customer by continuous delivery of valuable product
2. Welcome Change even at late stage
3. Deliver working software Frequently
4. Business and Developer work together
5. Build around Motivated Individuals- Give team opportunity, trust them
6. Face-to-face communication
7. Working software is the primary measure of progress.
8. Sustainable Development
9. Continuous Technical Excellence
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. Self Organizing Team
12. Team reflection at regular intervals

[www.agilemanifesto.org](http://www.agilemanifesto.org)

# Agile Core Practices

- 
1. Active Stakeholder Participation
  2. Apply the Right Artifacts
  3. Collective Ownership
  4. Create Several Models in Parallel
  5. Create Simple Content
  6. Depict Models Simply
  7. Display Models Publicly
  8. Iterate to Another Artifact
  9. Model in Small Increments
  10. Model with Others
  11. Prove it with Code
  12. Single Source Information
  13. User the Simplest Tools

# Agile Characteristics

- Adaptability not Predictability
- Accepting that outcomes are not predictable and process are not repeatable
- Values and Principles of Collaboration
- The conventions which we agree we define those
- Processes are in manuals; practices are in field.

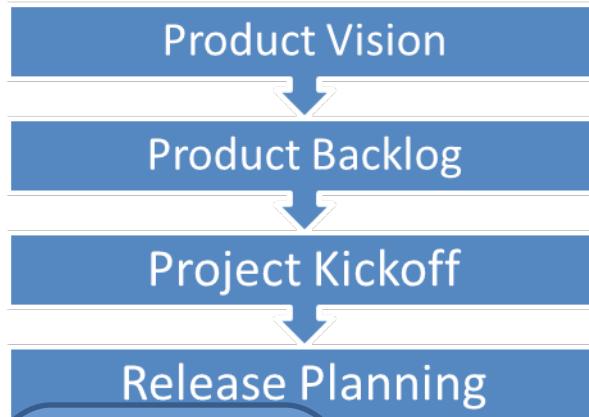
# When does Agile Methodologies Works Best?

- When problems you are solving have following characteristics
  - Going to **change** while solving
  - **Speed** of development cannot be determined
  - **Turbulence** in environment
  - Customer **doesn't know** how exactly it will look like

# Agile Project Life-cycle

Introduction / Agile Project Management

Vedavit



## Inside Story

### The Agile: Scrum Framework at a glance

Inputs from Executives,  
Team, Stakeholders,  
Customers, Users



Product Owner

1	2	3	4	5	6	7	8
Ranked list of what is required: features, stories, ...							

Product Backlog

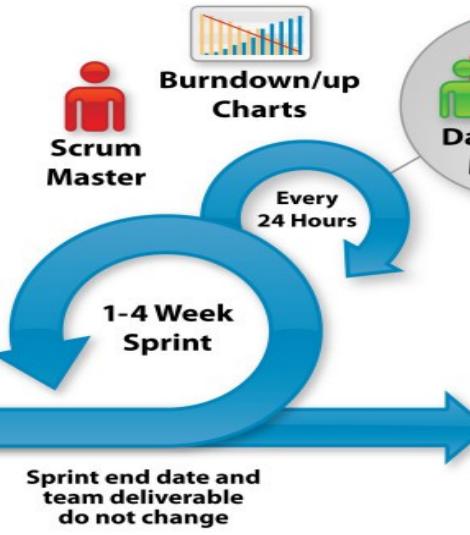


The Team

Team selects starting at top as much as it can commit to deliver by end of Sprint

Sprint Planning Meeting

Task Breakout  
Sprint Backlog



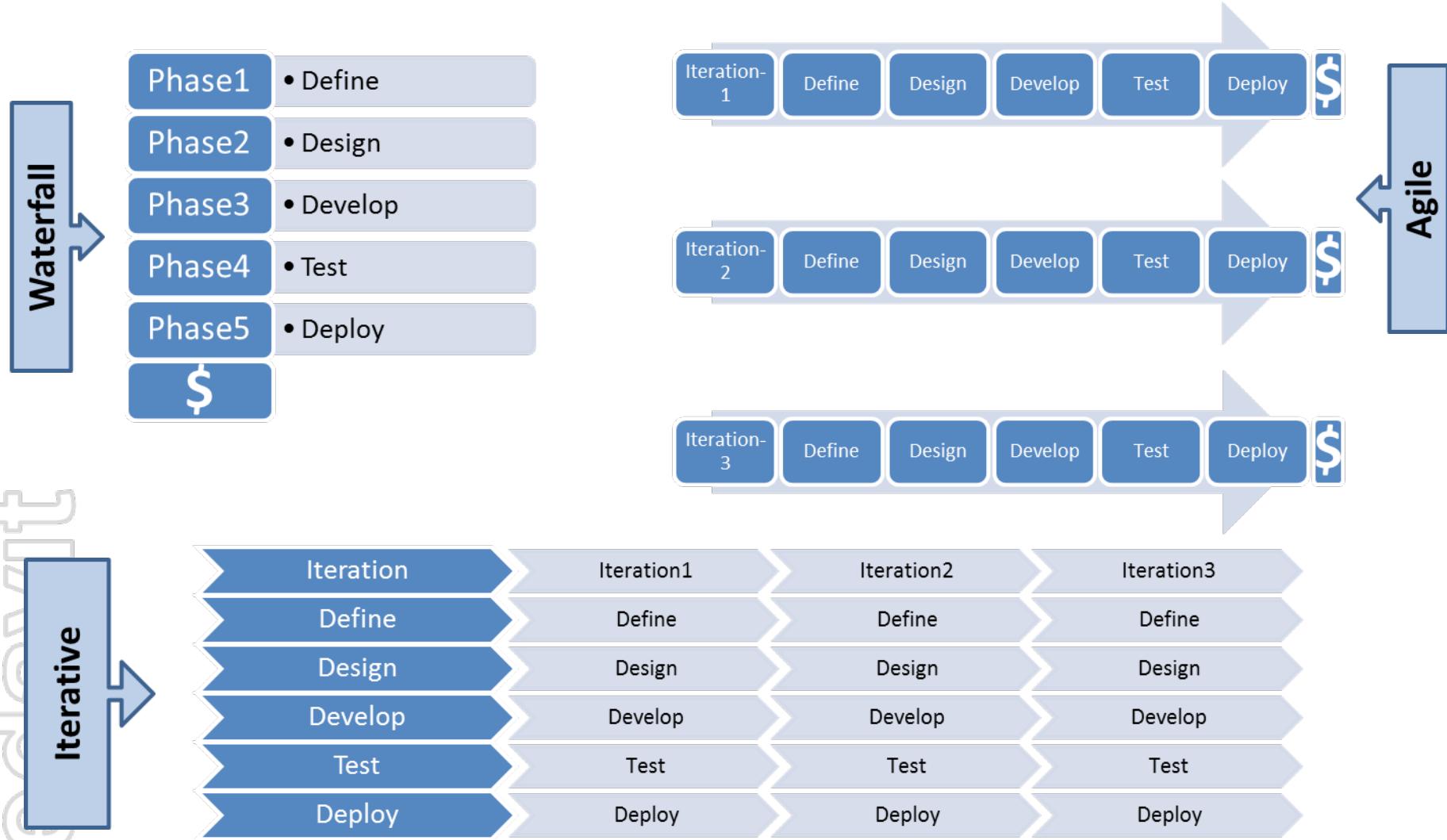
neon rain®  
interactive

[AGILE  
FOR ALL]

# Value Delivery & Project Life Cycle

Introduction / Agile Project Management

## Lifecycles: Waterfall- Iterative- Agile



# Recap

1. Modern Project Management Challenges
2. How to address modern day PM challenges?
3. Challenges in Starting Agile Project Management
4. What is Agility?
5. Which Project Management Methodology is best?
6. Type of Systems
7. Complex Adaptive Systems (CAS)
8. Defined vs Empirical Processes
9. Values in Agile Manifesto
10. Agile Principles
11. When Does Agile Methodologies Works Best?
12. Value Delivery & Project Life Cycle

# **Discussions !**

# Topics

## Day 1

- ✓ History of Agile
- ✓ Agile Certifications
- ✓ The Most General APM Certificate
- ✓ The Foundation of Agile Project Management
- ✓ Project Management in General
- ✓ Agile Project Management
- Agile : Developing Mindset
- Agile : Methodologies
- Agile : Metrics & Estimation



## Day 2

- Agile : Planning, Monitoring & Adapting
- Agile : Communication
- Agile : Analysis and Design
- Agile : Product Quality
- Agile : Value Prioritization

## Day 3

- Agile : Risk Management
- Soft Skills for Agile Project Manager
- Agile : Compliance
- Agile : Roles
- Scaling Agile
- Executing an Agile Iteration
- Professional Responsibility

# Agile : Developing Mindset

# Topics

1. Iteration
2. Incremental Delivery
3. Travel Light
4. Agile Product Building
5. Agile Documentation
6. Agile Architecture
7. Agile Testing
8. Last Responsible Moment
9. Refactoring
10. Technical Debt
11. Mindfulness
12. Energized Team
13. Poka Yoka
14. Agile Health Checkup

# Iteration / Sprint

- Agile project management relies on planning, developing and delivering product features in iterations
- Iteration or sprint are time-boxed and duration can not changed from one iteration to another
- Iteration is fixed time period in which team need to deliver some valuable product to customer
- Prescription for duration length varies from one method to another in agile. In XP iteration length varies 1-3 week (typically one week), scrum and other methods it varies between 2-4 weeks (typically 4 weeks)
- Iteration is also known as Sprint in Scrum Methodology

# Incremental Delivery

Delivering the complete product of a project in iterations. It helps in

- Learning from previous iteration
- Delivering high business value earlier
- Adapt to change
- Delivering more business value
- Removing the waste by not doing those things which are not needed

# Travel Light

- You need far less than what you think.  
Therefore carry on only those things which are most important and urgent.
- YAGNI (You aren't gonna need it)
- TAGRI (They aren't gonna Read It)

This is different type of mountaineering. You may need to come back, you may need to change your path, trust that you will get stuff you need on the way etc factor exists here.

# Agile Product Building

- Build complete product, all the time
- One button should produce needed documentation, build the product executables, create installation materials, produce test results and tested components
- Build should also work from command line
- Everyone in team should use the same build process

# Agile Documentation

- Maximize stakeholder investment. Produce document only when
  - It is needed by a stakeholder
  - Needed to define contract model
  - You need to think something in many iterations involving multiple groups
  - It is needed for external communication
- Document only those things which are least likely to change
  - First identify the specific customer of the document
  - The document facilitates in estimating
  - Sufficiently index, details, accurate and consistent

# Agile Documentation

## Strategies for reducing documentation CRUFT

- C- How correct is document?
- R- Will document be read?
- U- Will document be understood?
- F- Will document be followed?
- T- Will document be trusted?

# Agile Architecture

- Do the simplest thing possible which makes future changes/additions easier
- No up-front high-level system architecture
- No up-front high-level component architecture
- No up-front high-level data model
- Determine the details of technology when building functionality

# Agile Testing

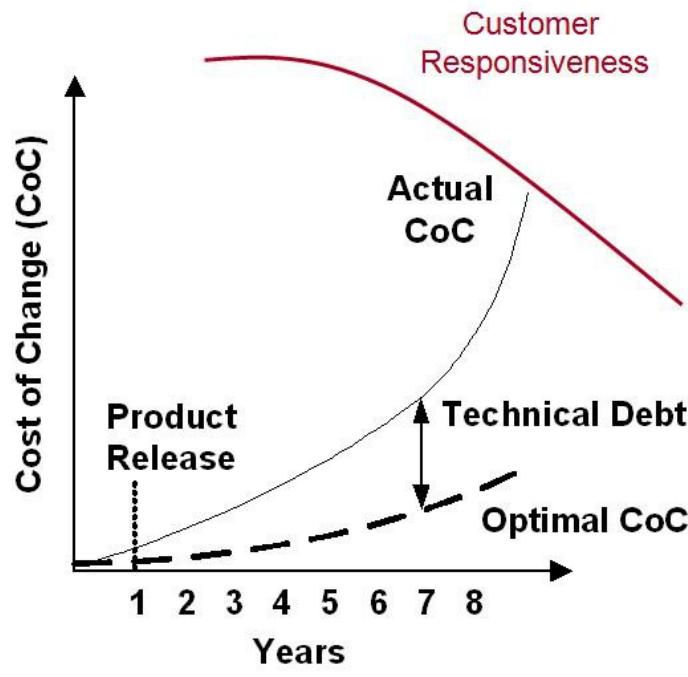
- All code must have test cases, ideally they should be created earlier
- Unit tests should be executed during automated build
- A build should be performed many times a time, ideally whenever anything is checked-in or committed to configuration server
- If anybody's code causing crash he should be informed immediately and that person should fix that problem first
- All unit test must pass before code can be released

# Last Responsible Moments

- Delay the decisions till the moment you have option to exercise because after this point one important options may be eliminated
- It helps you in delivering more value in less time by doing only those things which are of high priority
- It reduces inventory carrying cost
- Getting more information and making more informed decision at last moment

# Refactoring

- Agile programmer writes simple and bare minimum code they do not complicate the code
- Down the line structure the code without changing its behavior. It helps in improving the quality (maintainability, readability) of code



- Once on far right of curve, all choices are hard
- If nothing is done, it just gets worse
- In applications with high technical debt, estimating is nearly impossible
- Only 3 strategies
  - Do nothing, it gets worse
  - Replace, high cost/risk
  - Incremental refactoring, commitment to invest

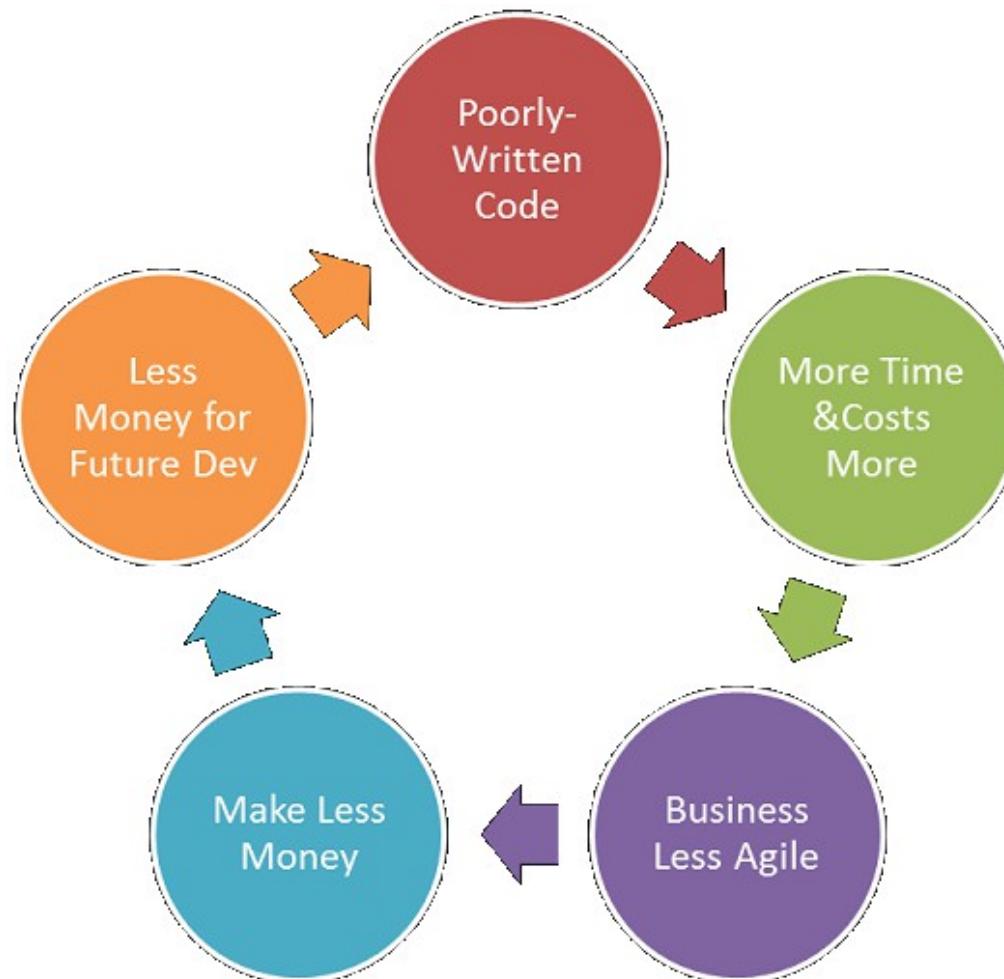
©2008 Information Architects, Inc.

# Technical Debt

- While writing code agile programmer do not pay attention to structure, duplication etc but to functionality and make sure that code is passing all unit test cases
- In this process if they do not clean the code by structuring (refactoring) it will become unreadable and un-maintainable and over a period of time this cause increases response time to fix the problems, adding new feature and deteriorate the quality.
- The concept of unclean code is called technical debt. Technical debt keep increasing over the period of time therefore Agile team need to pay this debt back by putting efforts in refactoring.

# Technical Debt

## Vicious Cycle of Technical Debt



# Mindfulness

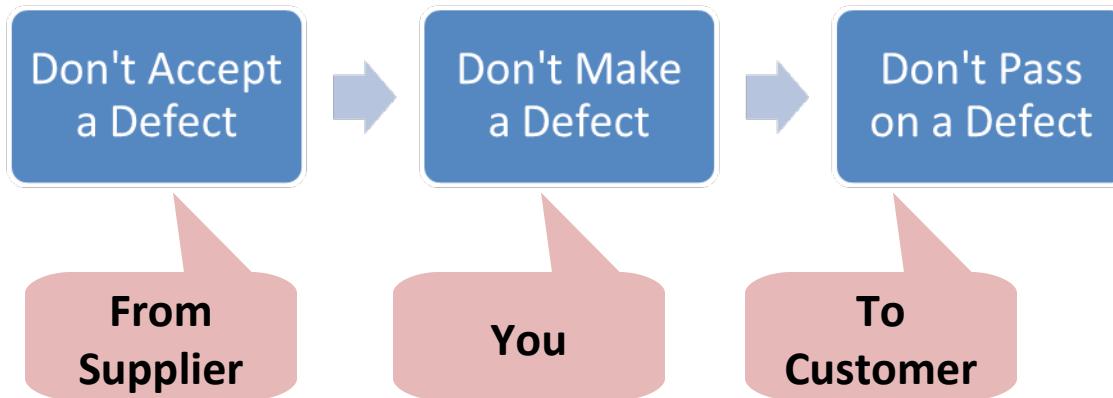
- Agile team need to be conscious all the time, they need to keep observing patterns and proactively resolving the issues at root.
- **For example** if they know that now it is taking more time to add even simple thing in existing code then it means time has come and they need to pay technical debt by refactoring

# Energized Work

- Love your work: When you feel that I enjoy programming. I enjoy solving problems, writing good code, watching tests pass, and especially removing code while refactoring. I love to program in my spare time and sometimes even think about work in the shower. You love your work
- Do not work overtime, enjoy balanced life, remain healthy, energetic and excited about work
- When at work completely cut off from interruption like phone, email, IM etc. Pay 100% attention to work at hand
  - If you are making more mistake than progress then that is the time for break.

# Poka Yoka

## Mistake (Error) Proofing



Ideally, design the product so that it can't be assembled incorrectly

Error is deviation from process. Defect is in product who is deviating from specification or don't meet customer expectations. Defect are introduced by errors

# Agile Health Checkup

To know the agile maturity in your project you can perform following Agile Health Checkup. Discuss in group and ask team to rate these parameters on the scale of 1-5

- 1. Frequent Delivery**
- 2. Reflective Improvement**
- 3. Close Communication** (does it take less than 1 min to get your question answered by a person who know the answer?)
- 4. Focus** (everybody understand the goal and desired outcome of the delivered software?)
- 5. Personal Safety** (can you give bad news to your boss?)
- 6. Easy Access to Outside Experts**
- 7. Strong Technical Environment** (SVN used? Test Automation?)
- 8. Sunny Day Visibility** (Does everyone on the team understand the rate of progress being made on the product?)
- 9. Regular Cadence or Rhythm** (Is heartbeat of the system on?)

# Recap

1. Iteration
2. Incremental Delivery
3. Travel Light
4. Agile Product Building
5. Agile Documentation
6. Agile Architecture
7. Agile Testing
8. Last Responsible Moment
9. Refactoring
10. Technical Debt
11. Mindfulness
12. Energized Team
13. Poka Yoka
14. Agile Health Checkup

# **Discussions !**

# Topics

## Day 1

- ✓ History of Agile
- ✓ Agile Certifications
- ✓ The Most General APM Certificate
- ✓ The Foundation of Agile Project Management
- ✓ Project Management in General
- ✓ Agile Project Management
- ✓ Agile : Developing Mindset
- Agile : Methodologies
- Agile : Metrics & Estimation



## Day 2

- Agile : Planning, Monitoring & Adapting
- Agile : Communication
- Agile : Analysis and Design
- Agile : Product Quality
- Agile : Value Prioritization

## Day 3

- Agile : Risk Management
- Soft Skills for Agile Project Manager
- Agile : Compliance
- Agile : Roles
- Scaling Agile
- Executing an Agile Iteration
- Professional Responsibility

Let us understand various

# Agile : Methodologies

# Agile Methodologies

1. Scrum
2. Extreme Programming (XP)
3. Lean Software Development (LSD)
4. Feature Driven Development (FDD)
5. Dynamic System Development Methods (DSDM)
6. Kanban
7. Scrum-Ban
8. Crystal Methods

# Agile Methodologies- Scrum

Introduction / Agile Project Management/ Agile Methodologies

- Scrum is widely accepted agile project management methodology among dozen of methodologies.
- Scrum is **lightweight management framework** with broad applicability for managing iterative and incremental projects of all type of projects
- Typical iteration (also called “**Sprint**”) length varies between 2-4 weeks.

## Scrum Values

1. Commitment
2. Focus
3. Openness
4. Respect
5. Courage

# Agile Methodologies- Scrum

Introduction / Agile Project Management/ Agile Methodologies

## Scrum Practices

- Backlog Grooming
- Sprint Planning
- Daily Standup
- Sprint Review
- Sprint Retrospective

# Agile Methodologies-XP

Introduction / Agile Project Management/ Agile Methodologies

Originally devised by Kent Beck as an agile method focused on engineering practices. Typical iteration length varies between 1-3 weeks.

## XP Values

1. Communication
2. Simplicity
3. Feedback
4. Respect
5. Courage

# Agile Methodologies- XP

Introduction / Agile Project Management/ Agile Methodologies

## XP Practices

### Fine scale feedback

1. Test Driven Development via Programmer Tests and Customer Tests (Unit Tests & Acceptance Tests)
2. Planning Game
3. Whole Team (including on-site customer)
4. Pair Programming (2 people sitting on one work station, one writing test case on notes and another writing code)

### Continuous process rather than batch

5. Continuous Integration
6. Design Improvement / Refactoring
7. Small Releases

# Agile Methodologies- XP

Introduction / Agile Project Management/ Agile Methodologies

## XP Practices

### Shared understanding

8. Simple Design (Do Simple Things, You Aren't Gonna Need It (**YAGNI**), Once And Only Once (**DRY**), Simplify Vigorously)
9. System Metaphor
10. Collective Code Ownership
11. Coding Standard or Coding Conventions

### Programmer welfare

12. Sustainable Pace

# Agile Methodologies- LSD

Introduction / Agile Project Management/ Agile Methodologies

Much of its principles and practices are from Lean Enterprise Movement and companies like Toyota

## Lean Software Development (LSD) 7-Principles

1. Eliminate Waste (Just in Time –JIT, Kanban)
2. Amplify Learning
3. Decide as late as possible (Make a decision when not making it means leaving some important option)
4. Deliver as fast as possible
5. Empower the team
6. Build Integrity In (overall experience of system quality, product quality should be dependable)
7. See the whole (see big picture)

# Agile Methodologies- FDD

Introduction / Agile Project Management/ Agile Methodologies

- FDD is a model-driven, short-iteration method
- First step is to establish an overall model shape
- Followed by 2 week “design by feature, build by feature” iterations.

# Agile Methodologies- FDD

Introduction / Agile Project Management/ Agile Methodologies

## Eight practices of FDD

1. Domain Object Modeling
2. Develop by Feature
3. Component/class ownership
4. Feature Teams
5. Inspections
6. Configuration Management
7. Regular Builds
8. Visibility of Progress and Results

# Agile Methodologies- DSDM

- Earlier in 1994 it was known as RAD method
- Previous version of DSDM is 4.2, that was released in 2003. Latest version of DSDM released in 2007 is also called **DSDM Atern**
- DSDM Atern used MoSCoW principle to prioritize requirements
- It is compatible with ISO9000 and PRINCE2
- “Fitness for business purpose” is primary criteria for delivery and acceptance of system
- It focuses that 80% of the system can be developed in 20% of the time

# Agile Methodologies- DSDM

Introduction / Agile Project Management/ Agile Methodologies

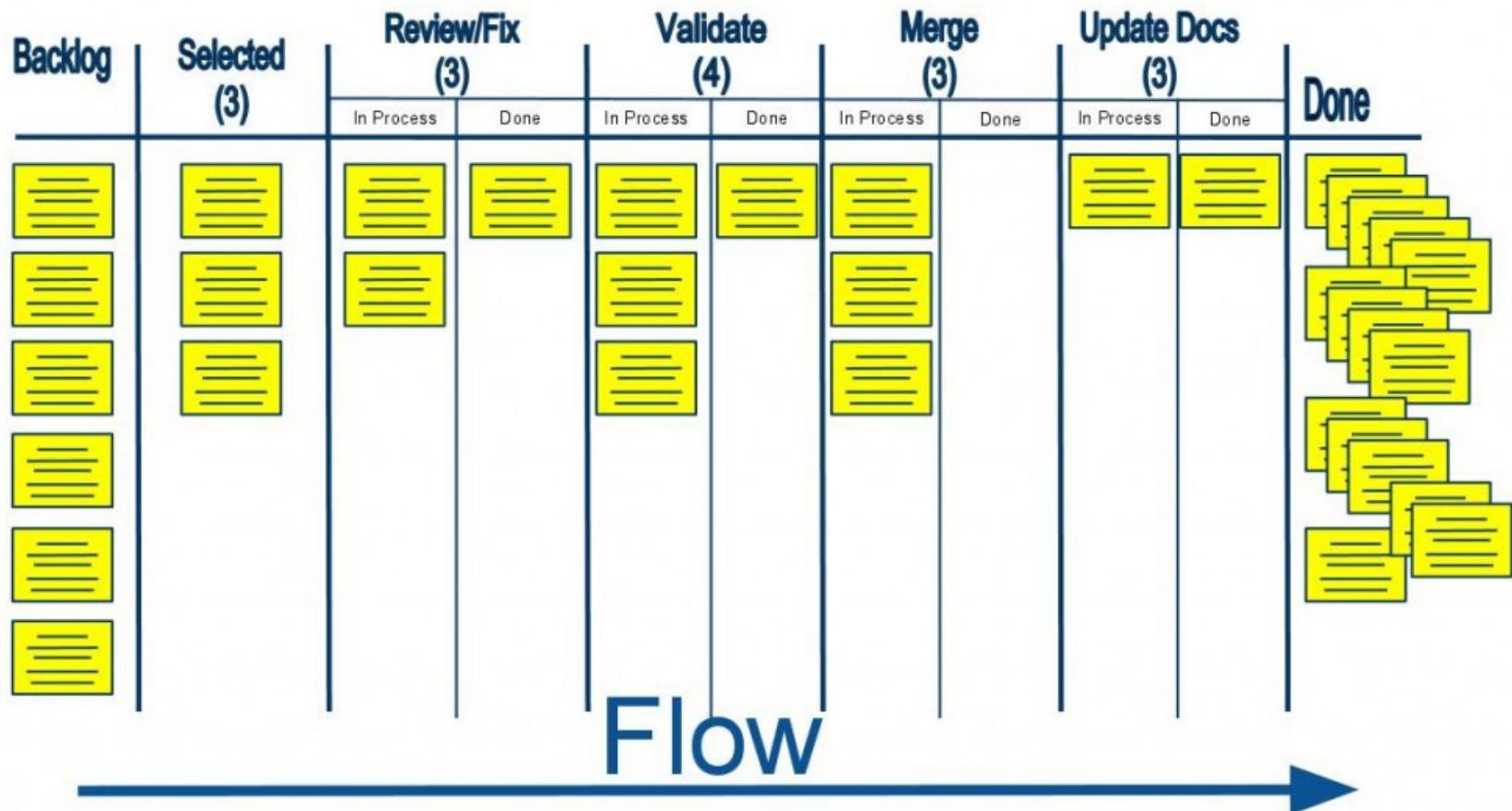
## 8 Principles of DSDM

1. Focus on the business need
2. Deliver on time
3. Collaborate
4. Never compromise quality
5. Build incrementally from firm foundations
6. Develop iteratively
7. Communicate continuously and clearly
8. Demonstrate control

# Agile Methodologies- Kanban

- In Japanese Kanban means Sign-Board
- Kanban was developed by Taiichi Ohno, at Toyota, to find a system to improve and maintain a high level of production.
- Kanban is scheduling system for Lean and JIT production.
- Kanban is one method through which JIT is achieved.
- Kanban is pull based planning and execution method. Work is not planned and pushed into work queue of team. Team signals when they are ready to take more work and pulls into their queue. Uses cards to signal the need of items
- Under Kanban or Lean approach queues or inventory of work in any state is seen as waste.
- **WIP limits:** helps team in optimal flow of work items, minimizing any associated waste
- Allow team to achieve process optimization while respecting and maintaining sustainable pace.

# Kanban Board



# Agile Methodologies- Scrum-ban

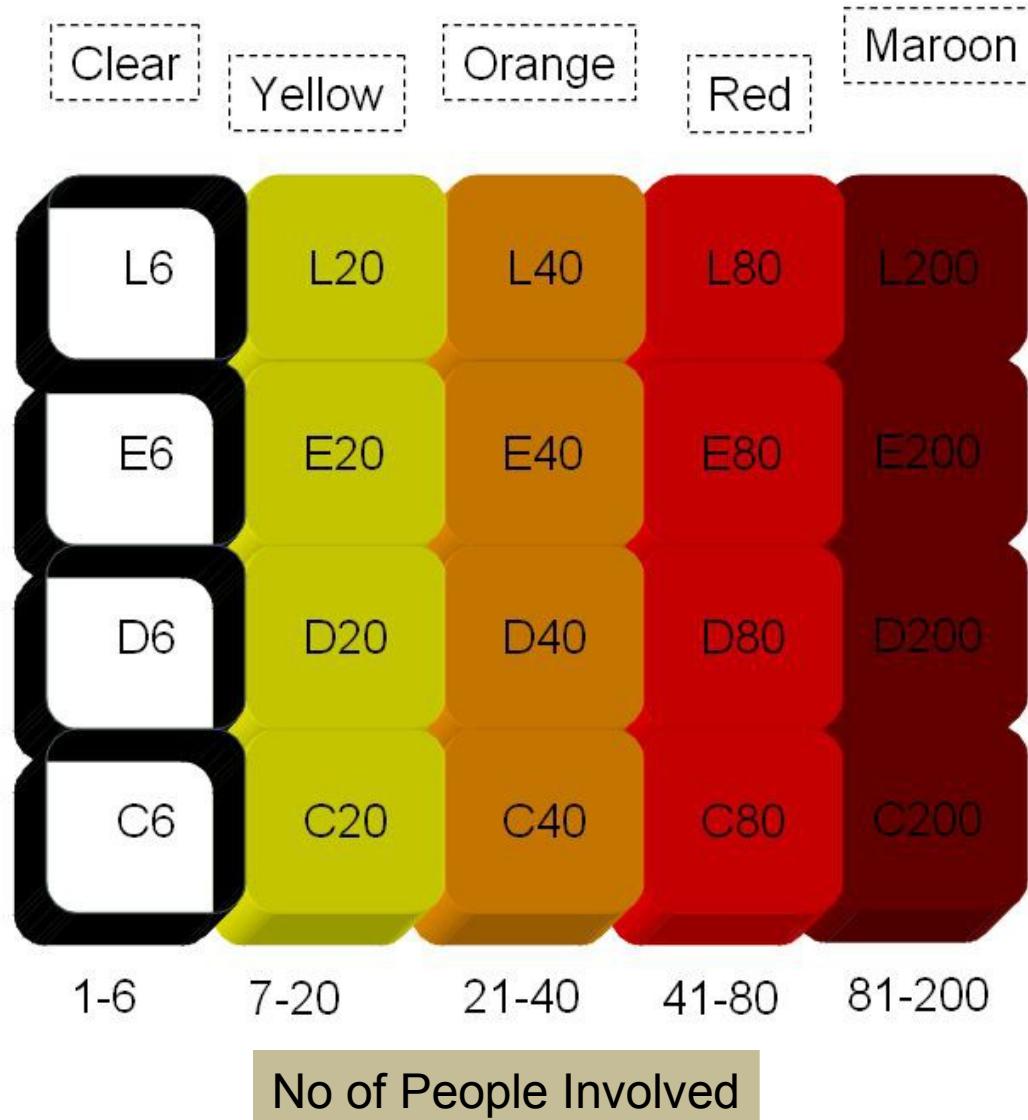
- Scrum-ban is a software production model based on Scrum and Kanban
- Scrum-ban is especially suited for maintenance projects or (system) projects with frequent and unexpected user stories or programming errors. Thus time-limited sprints of the Scrum model are of no appreciable use.
- Scrum's daily meetings and other practices are highly useful
- Visualization of the work stages<sup>(1)</sup> and limitations for simultaneous unfinished user stories<sup>(2)</sup> and defects<sup>(3)</sup> are part of the Kanban model.
- The team's workflow is directed in a way that allows for minimum completion time for each user story or programming error, and on the other hand ensures each team member is constantly engaged.

# Agile Methodologies- Crystal

- Crystal is one of the most light-weighted software development methodology
- Crystal is a family of methodology. Family members are Crystal Clear, Crystal Yellow, Crystal Orange etc.
- Which type of crystal methodology should be used for a particular project depends upon **criticality of the project and number of people involved**
- This methodology relies on the fact the processes can be tailored based on unique characteristics of the project

# Agile Methodologies- Crystal

Criticality, Defect cause ...



# **Discussions !**

# Topics

## Day 1

- ✓ History of Agile
- ✓ Agile Certifications
- ✓ The Most General APM Certificate
- ✓ The Foundation of Agile Project Management
- ✓ Project Management in General
- ✓ Agile Project Management
- ✓ Agile : Developing Mindset
- ✓ Agile : Methodologies
- Agile : Metrics & Estimation



## Day 2

- Agile : Planning, Monitoring & Adapting
- Agile : Communication
- Agile : Analysis and Design
- Agile : Product Quality
- Agile : Value Prioritization

## Day 3

- Agile : Risk Management
- Soft Skills for Agile Project Manager
- Agile : Compliance
- Agile : Roles
- Scaling Agile
- Executing an Agile Iteration
- Professional Responsibility

# Quotes Relevant to the Topic

10 Different estimator gives 10 different estimates of the same piece of work, even one estimator gives 10 different estimates of the same work at 10 different times. Then what is the use of estimation?

Unknown

# Agile Metrics & Estimation

# Metrics

1. Velocity
2. Lead Time
3. Cycle Time
4. Ideal Time
5. Escaped Defects
6. Project Scheduled
7. Features & Value Delivered
8. Delivery Performance
9. Product and Technical Quality Assessment

## Velocity

- Velocity is number of story a team can do in an iteration.
- When you are starting a new project that point of time best way to know the productivity is guess based on previous experience. But never benchmark velocity of team based on previous project
- It takes 3-4 iteration to stabilize velocity of team
- Team members should not be added or removed from development team otherwise it destabilized the velocity
- **Methods to improve velocity**
  - Periodically pay down technical debt
  - Improve customer involvement
  - Support energized work environment
  - Provided needed resources (people, equipment, software etc)
  - Offload admin duties of programmer
  - Add experienced programmers

## Lead Time (LT)

- Lead time is the time required to deliver a given amount of work.
  - eg. Lead time to make tea is 15 min. From the time order is placed till the time tea is on table.

## Cycle Time (CT) & Throughput

- **Cycle Time** is actually a measure of **Throughput**
- **Cycle Time = 1/ Throughput**
- **Throughput** is measured as **units /period of time**
- **Cycle Time** is measured as **amount of time/unit.**

Example : At a tea stall for preparing a tea

- Cycle Time = 2 minutes/ cup of tea
- Throughput = 30 cups / hours

## Relationship Between CT, LT and WIP

- Lead Time = Cycle Time \* WIP or  
Lead Time = WIP/Throughput

# Metrics

- Kanban does not have concept of iteration and thus it does not talk about bunch of user stories in that iteration.
- Kanban deal with individual user stories (it may be bug user story) and tracks the time to complete that user story. Thus the concept of Cycle Time and Throughput and WIP is important here.

## Ideal Time

### Case Study:

Suppose a project where 5 team members are working starts on 1-Jan (Monday) and Finishes on 31-Jan. Company working times are from 8.00 am to 12.00pm and 12.00pm to 17.00pm. Company working days are Mon to Fri. In this period, there are two state holidays falling on organization working days.

- How much is elapsed time of this project?
- How much is duration of this project?
- How much is effort required (actual effort) to complete the project?
- How much is Ideal time in the project?
- How many Ideal Engineering Hours (IEH) in the project?

## Ideal Time

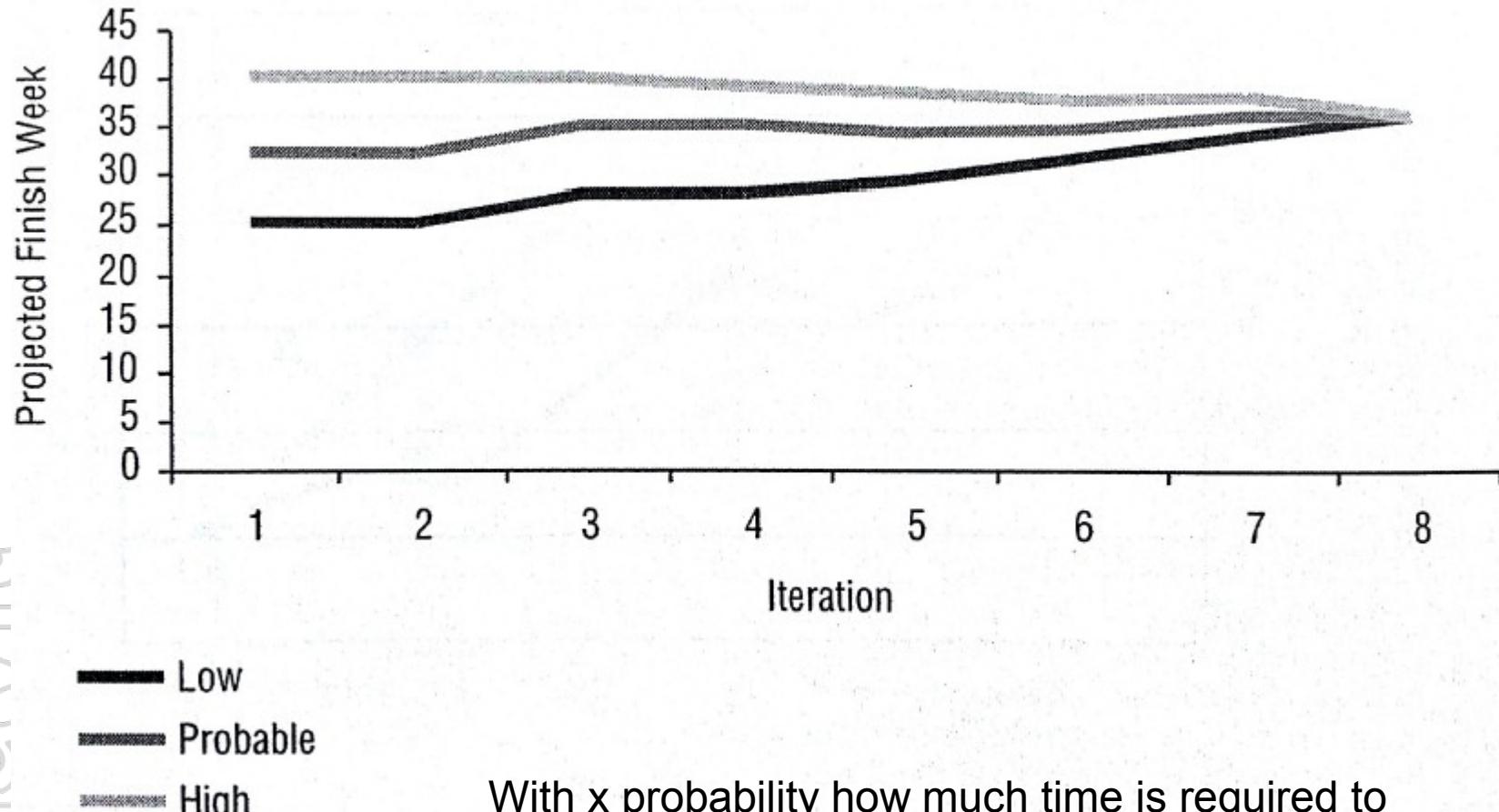
- How much is elapsed time of this project?
  - 31 Days
- How much is duration of this project?
  - $31 - 8$  (4 week ends) – 2 (holidays) = 21 Days = 168 Hours
- How much is effort required (planned effort) to complete the project?
  - 21 days X 5 people X 8 hours = 840 Hours
- How much is Ideal time in the project?
  - Ideal time excludes the time of meetings, emails, peer review, phone calls, training, being pulled off to another work, task switching, tea breaks, lunch breaks, smoke break, biological break etc.
  - Excluding lunch and tea breaks a company typically has 8 working hours will not have more than 80% ideal time. That is ideal time on any day is 6.4 hours (max)
  - Thus ideal time for above project is 21 days X 6.4 hours = 135 hours
- Ideal Engineering Hours (IEH)
  - If all 5 people on project are working individually (i.e. no peer review etc) then IEH = Ideal time \* No of People = 675 hours

## Escaped Defects

- In Agile escaped defects mean those defects which are leaked to next iteration or sprint.
- **Escaped Defect Rate** = Total number of weighted defects for a sprint / number of total stories delivered in that sprint
- Tracking the trend of this metrics helps you understanding the quality of product and maturity of testing process

# Metrics

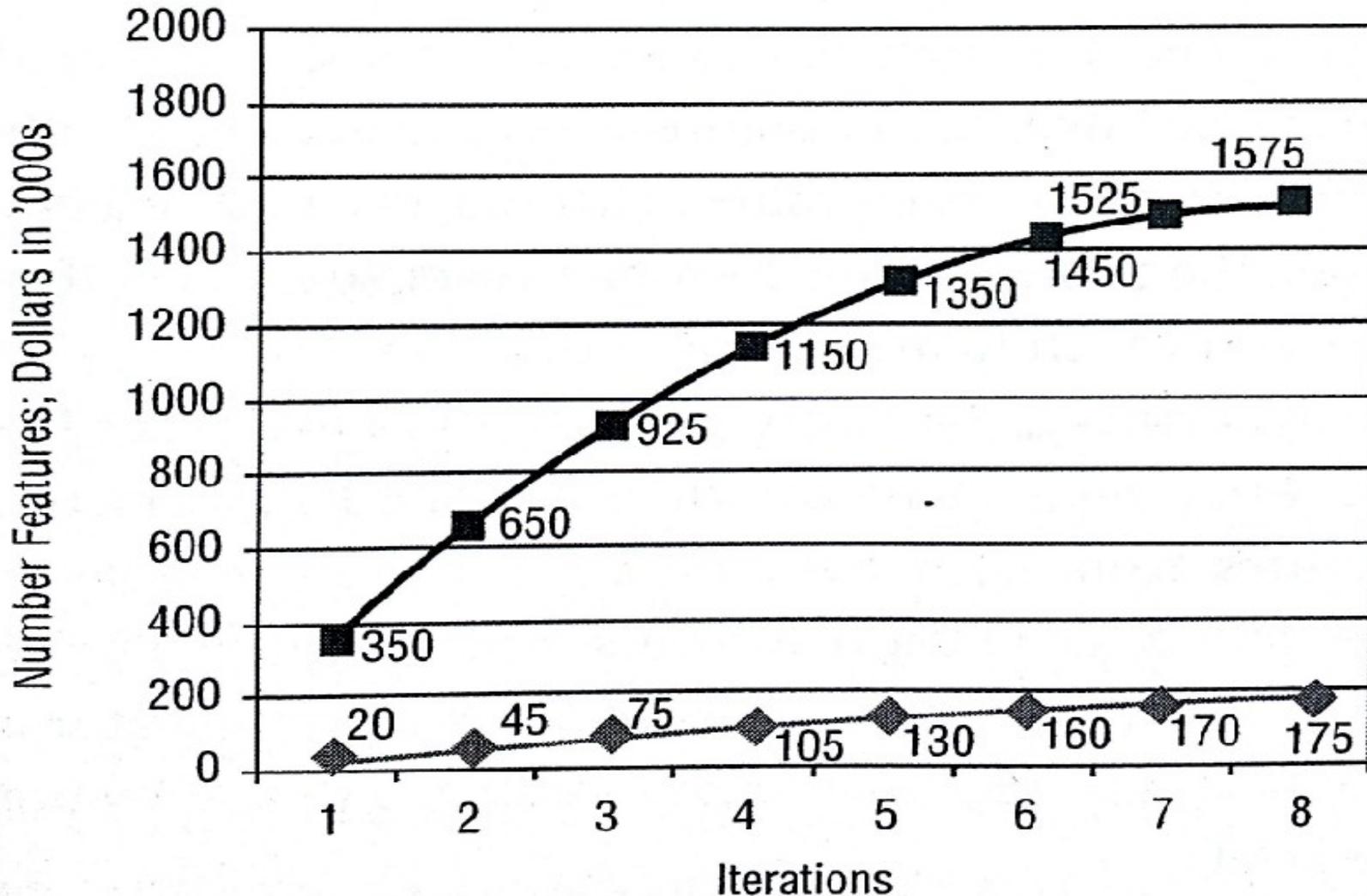
## Projected Scheduled



With x probability how much time is required to complete the project. Track this information for every iteration

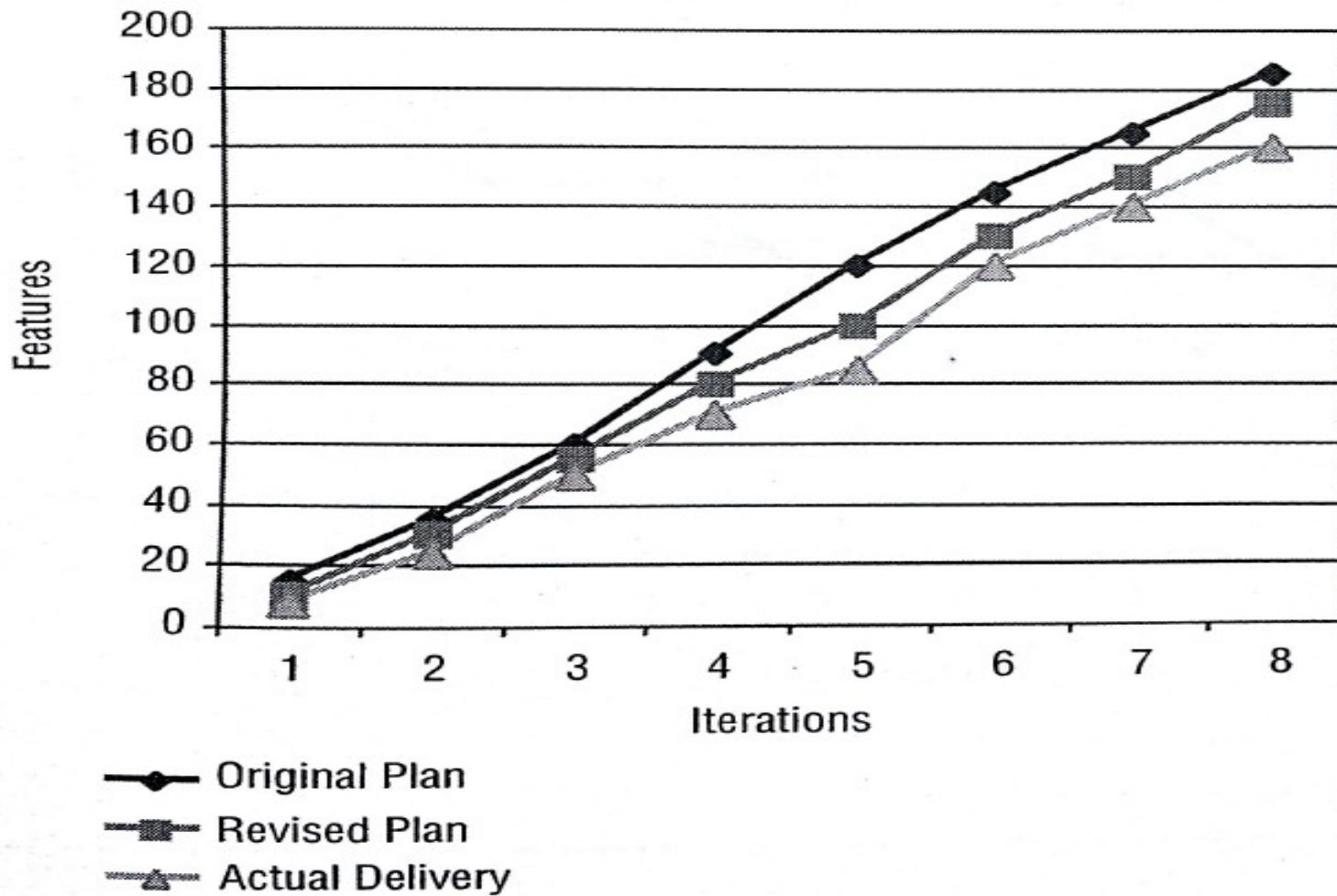
# Metrics

## Features & Value Delivered



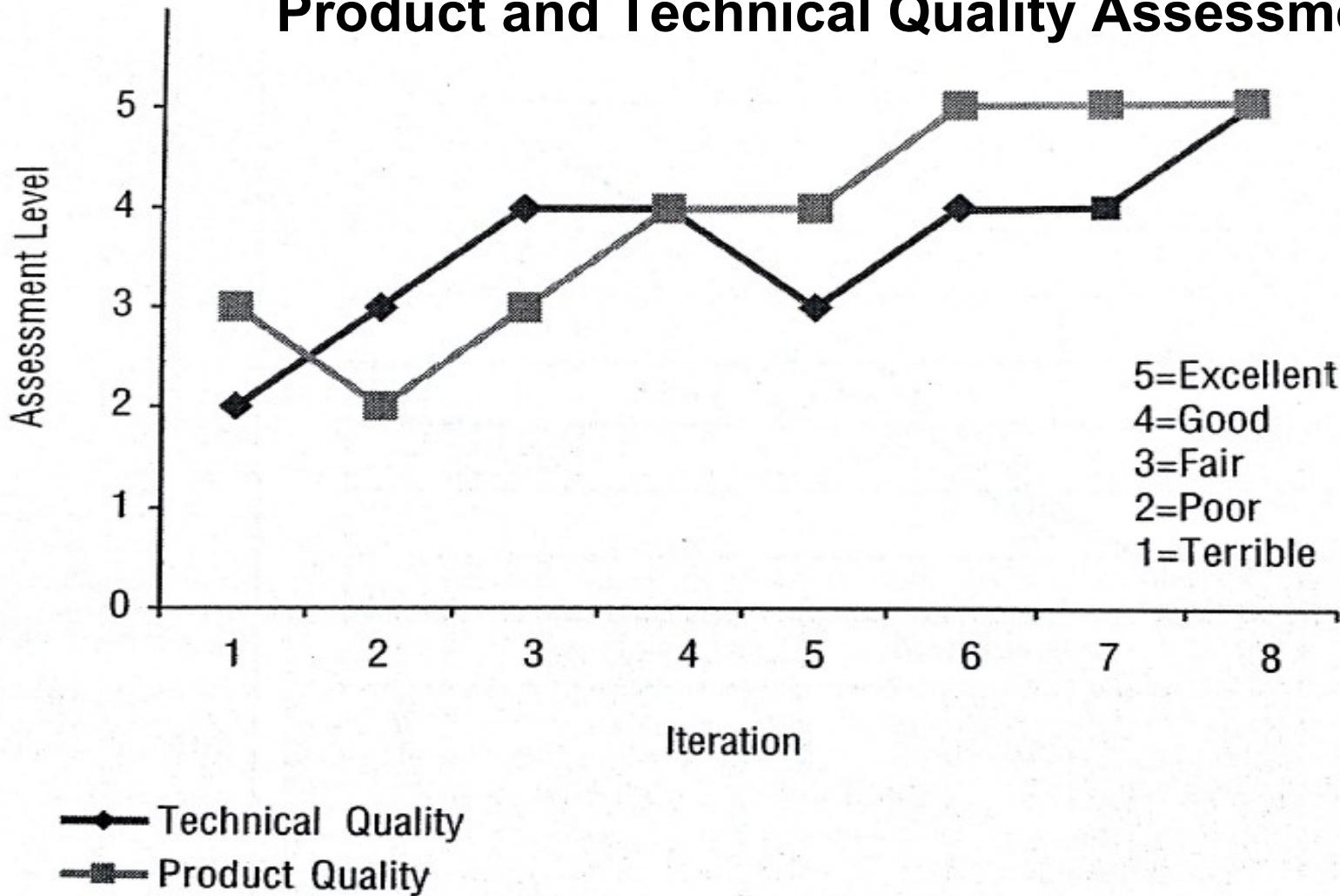
# Metrics

## Delivery Performance



# Metrics

## Product and Technical Quality Assessment



# Cost Estimate

- Cost estimate for any release or iteration implies cost of all the resources used delivering user stories of the release or iteration.
- Cost of resource includes cost of service used from vendor, cost of human resources, cost of software/hardware used, cost of training, cost of expenses like travelling, communication etc.
- Opportunity cost is economic cost.

# Cost Estimate

- Methods of estimations
  - Analogous Estimating
  - Parametric Estimating
  - Bottom-up Estimating
  - Vendor-bid Analysis
  - Reserve Analysis
  - Cost of Quality
- Type of cost estimates (range names)
  - Order of magnitude (-25% to +75%)
  - Budgetary (-10% to +25%)
  - Definitive (-5% to +10%)

# Basis of Estimation

- Estimates are never absolute. There are certain assumptions related to productivity, dependency, complexity, resource availability, resource rates etc which dictates the efforts, cost, duration and quality of the product of project
- Basis of estimation helps you determining the range

# Earn Value Management

- Earned value in Agile can be calculated using story point, efforts, cost. Being a time-boxed methodology time does not change
- Lets us assume that product owner has planned one product
  - Release with three iteration and each iteration is of 2 weeks.
  - IEH in first, second and third iteration is 60, 60, 80 (adding resources) respectably (total 200 IEH).
  - Story points of user stories to be completed in this release is 30 SP.

**Note:** Although this approach is detrimental to agile because agile project management does not allow product owner to decide IEH to complete the stories he selected to complete in the release. But lets us assume that after some iterations of working team has stabilized the velocity and this velocity factor helps them determining IEH required to complete the release. Based on the velocity team feels confident to complete 30SP in 200 IEH. Lets assume iteration is complete and we are calculating EVM

\*IEH is Ideal Engineering Hours

# Earn Value Management

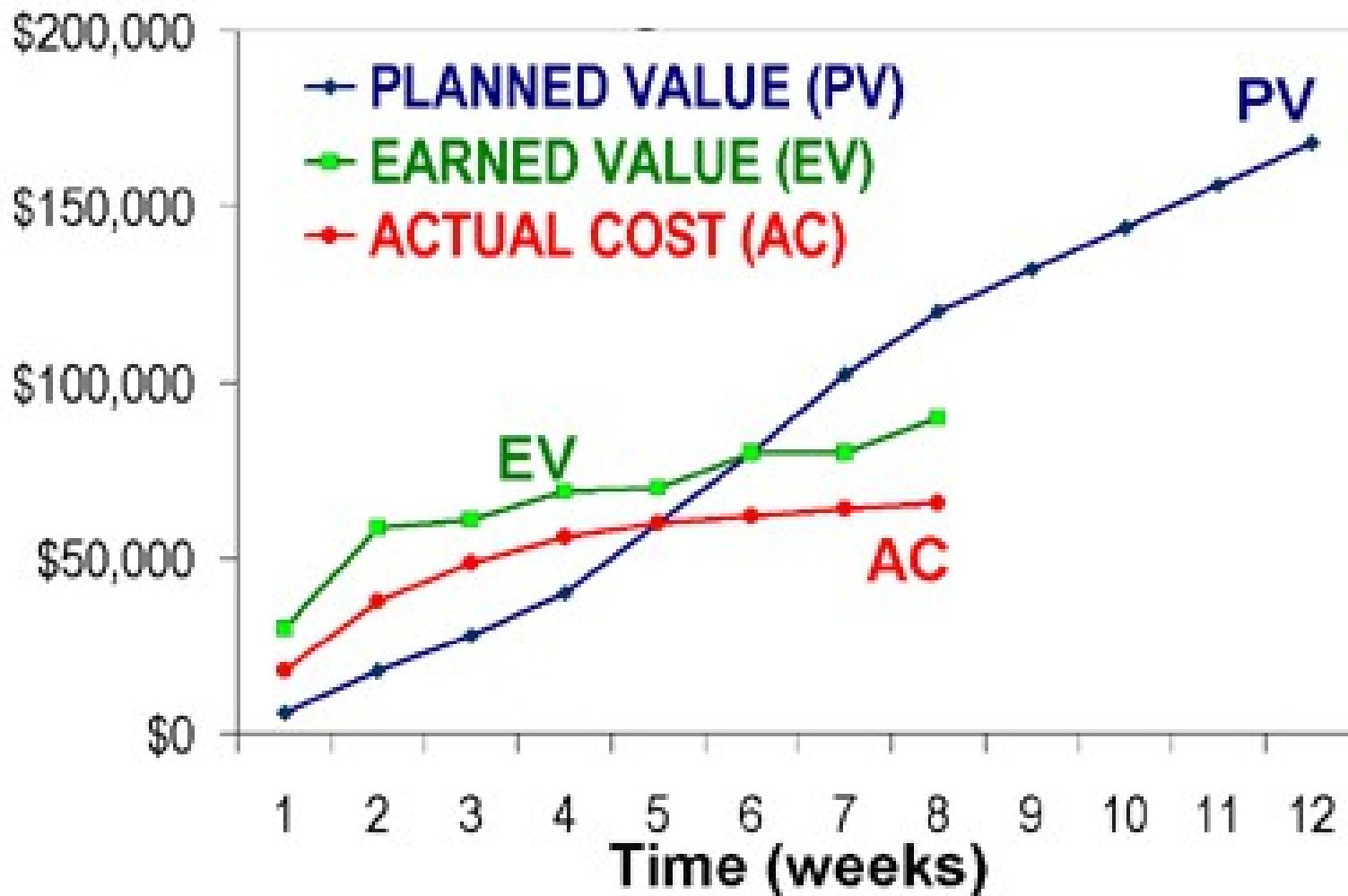
- Case 1 –Story Point based EVM
  - BAC- Total Story Points in current release = 30
  - PV- Planned story points to be completed at the end of 1<sup>st</sup> Iteration = 8
  - AC- Completed story point =7
  - ETC- More story to be completed in future iterations = 24
  - **EV = BAC – ETC =6**
- Case2- Efforts based EVM
  - BAC- Planned efforts for current release = 200 IEH
  - PV- Planned efforts for the of 1<sup>st</sup> iteration = 60 IEH
  - AC- Actual efforts at the end of 1<sup>st</sup> iteration = 60 IEH (Agile does not encourage over time)
  - ETC- Efforts required to complete remaining story points = 120 IEH
  - EV- Budgeted efforts for the completed user stories = BAC – ETC = 80 IEH
- If you know the cost of resources you can calculate EVM based on cost also

# Earn Value Management

Tools & Techniques / Agile Estimating & Metrics

- $CV = EV - AC$
- $CPI = EV / AC$
- $SV = EV - PV$
- $SPI = EV / PV$
- $EV = BAC - ETC$
- $EAC = AC + ETC$
- $EAC = BAC / ETC$
- $EAC = BAC / (SPI * CPI)$
- $VAC = BAC - EAC$
- $TCPI = (BAC - EV) / (EAC - AC)$

# S Curve



# Estimation Techniques

1. Relative Sizing
2. Story Points
3. Wideband Delphi
4. Planning Poker
5. Affinity Estimation

# Relative Sizing

- We know at the start of project not enough detail is available to estimate the size of work. But high level requirements are available in the form of epic or module
- Best way to estimate in this situation is relative sizing method
- Based on current understanding of system and modules team picks up smallest size epic and then start comparing that with other modules.
- Down the line at the time of release planning features or user stories are identified in epic and finer estimation is possible but again at this level preferred method of estimation is Relative Sizing
- Hours, duration, cost based quantitative estimation makes sense only when you know the task to be done. But even for user story level estimation in release planning neither you have time nor it is worth to identify task to do quantitative estimation

# Story Points

- User story is the functionality of the system which has value in the eyes of customer
- Count of user stories is called story points this is used to size user stories.
- Because all stories are not same in efforts, complexity and risk therefore different methods are used to normalize this factor
- Size is function of relative efforts, complexity and risk not of duration or IEH etc. Therefore when estimating size take away duration, number of people etc from your mind.
- Some story point estimation methods are
  - T-shirt size – Story Point Estimation
  - Fibonacci Series – Story Point Estimation
  - Fruit size – Story Point Estimation

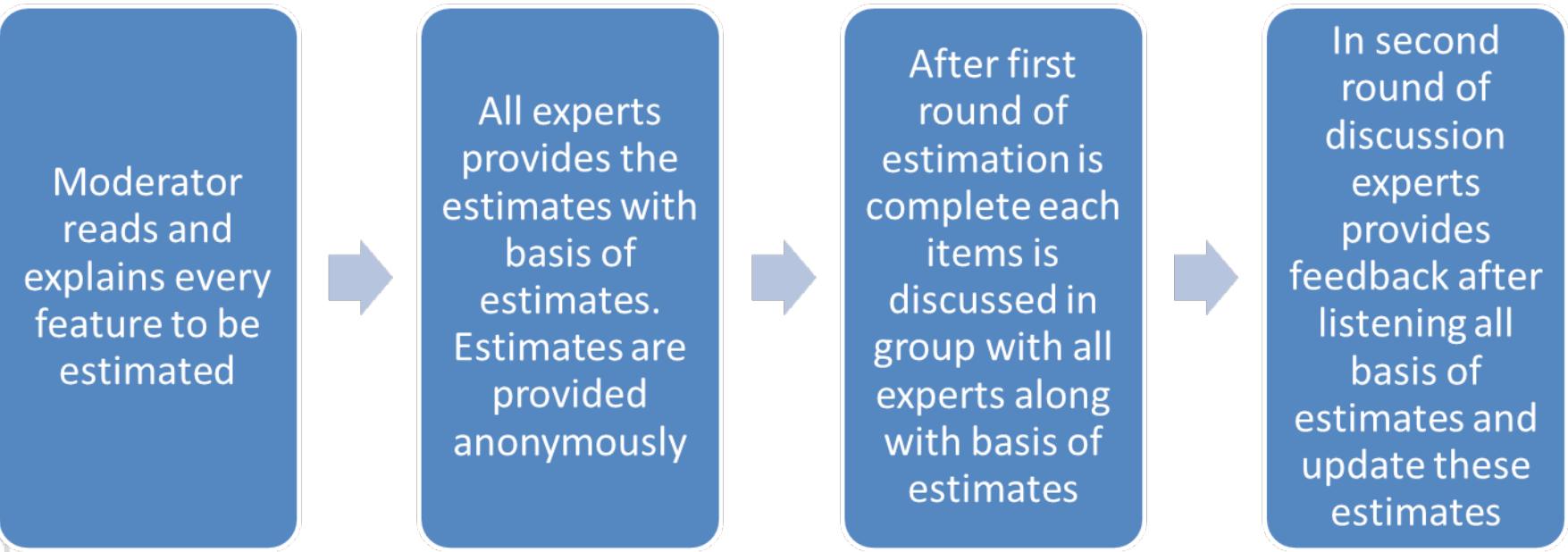
# Story Points

- Let us say our **product back log** has
  - 8 x bananas (5 points each) = 40 story points
  - 10 x pineapple (20 points each) = 200 story points
  - 6 x oranges (8 points each) = 48 story points
  - 30 x bunches of grapes (2 points each) = 60 story points
- **Total story points** in product backlog is  $40+200+48+60 = 348$
- Let us assume that based on guess or some previous release teams knows that their **velocity** is 30 story points
- Total iterations required =  $348 / 30 = \mathbf{11 \ iterations}$
- If iteration is of 2 weeks then **22 weeks** required to finish the work in product backlog

# Wideband Delphi

Tools & Techniques / Agile Estimating & Metrics/ Size Estimation Techniques

- Wideband Delphi is consensus based estimation method. A moderator facilitates the session.
- Delphi technique keep names anonymous and helps avoiding biases and politics



# Planning Poker

- Planning poker is widely used estimation technique in agile projects. This is a variation of Wideband Delphi.
- Logistic Required for Planning Poker Estimation
  - As many set of deck cards as many estimators in the room
  - All the team members with a deck of cards inside the room
  - A coordinator with laptop
  - A facilitator
  - Product owner
- Planner Poker Process
  1. Facilitator projects the product backlog on wall using a projector
  2. He reads loudly a user story at a time so that everybody in the room understand what they need to estimate
  3. Estimator estimates it silently and shows the card only to the coordinator
  4. Coordinator documents all the estimates provided by all estimator
  5. If there is difference between estimates then estimators who provided two extreme estimates need to talk the basis of estimates.
  6. After a short discussion. Next round of estimation starts the same way as before and coordinator document the estimate provided by estimator
  7. Keep repeating steps 3,4,5,6 till estimates provided by estimators are not converged
  8. Read next user story

# Affinity Estimation

- If product backlog has more than 20 items then implementing planning poker method of estimation is too much time consuming. In that case affinity estimation technique is most suitable
- This helps in providing coarse estimates and good enough for release planning but for iteration planning you should use Planning Poker.
- Logistics for Affinity Estimation
  - A set of cards with printed user story. A story on a card.
  - A wide magnetic board with magnetic balls
  - All the team members inside the room
  - A coordinator with laptop
  - A facilitator
  - Product owner

# Affinity Estimation

Tools & Techniques / Agile Estimating & Metrics/ Size Estimation Techniques

Facilitator writes “smallest” at extreme left end of magnetic board and “largest” at extreme right of magnetic board



Set the ground rule of estimation session. Like it is mute session so we should not talk at all. All cards which are put on table need to be arranged on wall from left to right in terms of their relative size. You can shift position of cards from left to right or vice versa put by other member but without talking, convincing anybody, arguing or any kind of body language.



Estimators starts reading card and arranging them on the wall. If required they can discuss in private with product owner



If they are not able to understand any card ever after discussion with product owner they can move the card into Questionable column



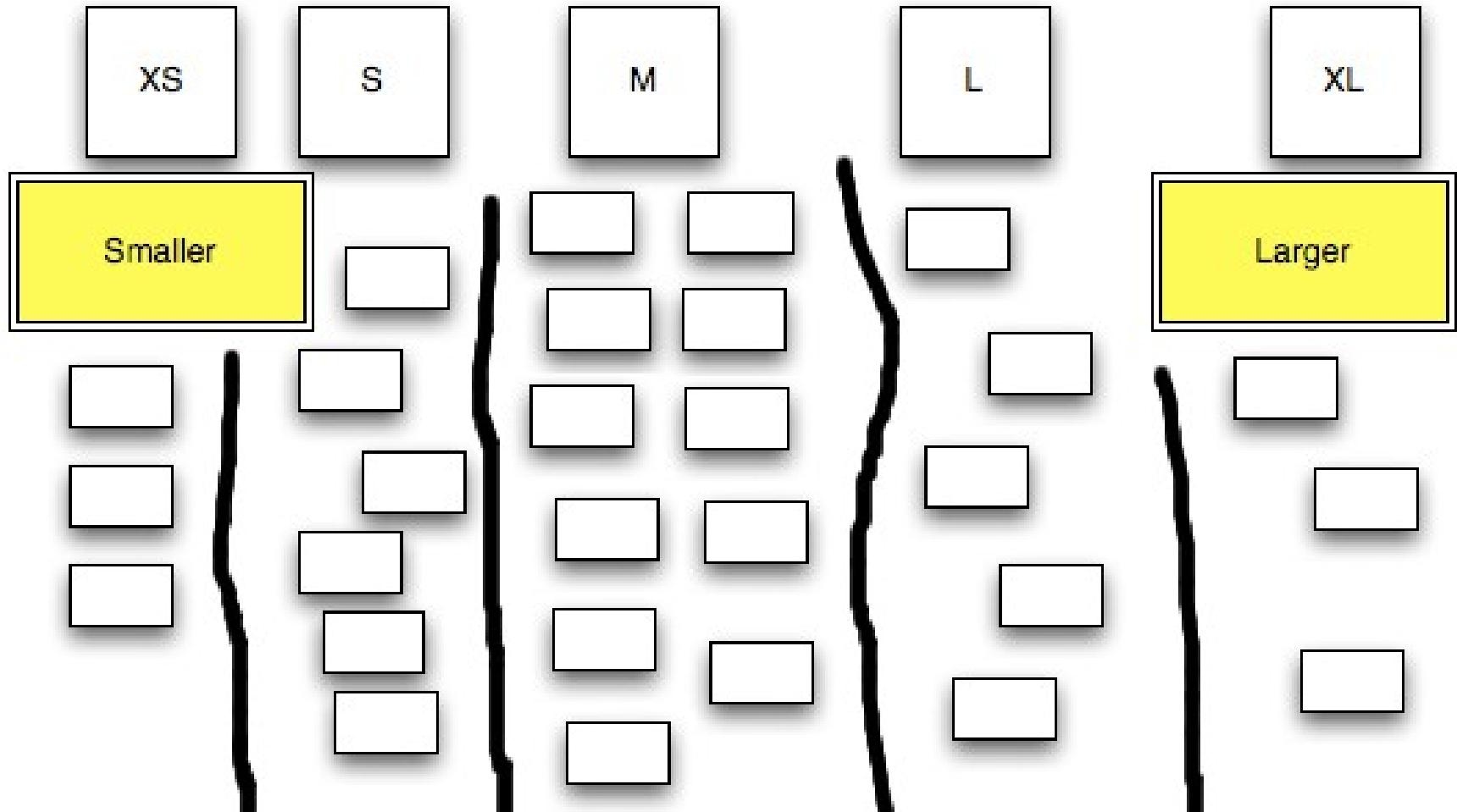
After the session is over they need to group these cards in 5 or more groups like smallest, minor, medium, larger, largest



Coordinator need to type all these estimates into computer and share with project manager



# Affinity Estimation



# Recap

Tools & Techniques / Agile Estimating & Metrics

## Estimation

1. Velocity
2. Lead Time
3. Cycle Time
4. Ideal Time
5. Escaped Defects
6. Project Scheduled
7. Features & Value Delivered
8. Delivery Performance
9. Product and Technical Quality Assessment

## Metrics

1. Relative Sizing
2. Story Points
3. Wideband Delphi
4. Planning Poker
5. Affinity Estimation

# **Discussions !**

# Topics

## Day 1

- ✓ History of Agile
- ✓ Agile Certifications
- ✓ The Most General APM Certificate
- ✓ The Foundation of Agile Project Management
- ✓ Project Management in General
- ✓ Agile Project Management
- ✓ Agile : Developing Mindset
- ✓ Agile : Methodologies
- ✓ Agile : Metrics & Estimation

vedavit

## Day 2

- Agile : Planning, Monitoring & Adapting
- Agile : Communication
- Agile : Analysis and Design
- Agile : Product Quality
- Agile : Value Prioritization

## Day 3

- Agile : Risk Management
- Soft Skills for Agile Project Manager
- Agile : Compliance
- Agile : Roles
- Scaling Agile
- Executing an Agile Iteration
- Professional Responsibility

# Agile : Planning, Monitoring & Adapting

# Quotes Relevant to the Topic

- Failing to Plan is Planning to Fail
- If you do not have your own plan then you are fitting in somebody else's plan

# Topics

Tools & Techniques / Planning, Monitoring & Adapting

1. Progressive Elaboration
2. Adaptive Planning
3. User Story
4. Types of User Stories
5. 3Cs of User Story
6. INVEST Model of User Story
7. Epic, Feature, Story, Task
8. Agile Planning
9. Time-boxing
10. Minimum Marketable Features (MMF)
11. Process Tailoring
12. Agile Contracting Methods
13. Agile Project Tracking
14. Innovation Game

# Progressive Elaboration

Tools & Techniques / Planning, Monitoring & Adapting

- Requirements in a project may not be known fully at the start of project. Because
  - Future requirements depends upon the results of earlier stages
  - It will be decided based on the reaction of market
  - Business conditions are too volatile so nothing can be predicted right now
- **Progressive elaboration** relies on planning and executing only that work which is clearly known, has high priority and must be done in current period. Planning for remaining work can be done in future as and when detail requirements are known

## School Management Application

Teacher Registration  
Module  
(1 Release of 3  
Iteration; estimation  
possible because end  
user knows what  
exactly needed)

Student Registration  
Module  
(Complete  
requirements not  
available)

Some other N  
modules

Rolling wave  
planning means  
planning of one  
iteration at a time

# Adaptive Planning

- Adaptive planning is about incorporating lessons learned. Adjusting your future pace, processes, resources etc based on previous cycle of delivery.
- Adaptive planning is required because what we want to develop may not be completely known or other factors like processes, tools, technologies, skills availability, business environment, market condition etc are not fully known or reliable and they all affect your planning.
- It is complete opposite of predictive planning where everything is known at the time of planning.
- In modern time it is extremely difficult to know about all the factors, which are affecting project success, in advance
- Adaptive planning and empirical processes are the truth of 21<sup>st</sup> century! Days of definitive processes and predictive planning are not many.

# User Story

- A user story concept is kernel of Agile Project Management
- A user story is work which a user want system to accomplish because it meets some of his objectives.
- A user story is not functional specification document. It is a promise of product owner to the team that he will explain the requirements in details when the team is working on this
- User story template
  - “As a user I want to accomplish something so that business value”

*“There have been great societies that did not use the wheel, but there have been no societies that did not tell stories.” Unknown*

# Types of User Stories

- Business user story
  - “As a class teacher I want to mark attendance of student so that we can issue them certificate”
- Bug user story
  - “An error message is displayed whenever I try to save file in pdf format”
- Technical user / Technical Spikes story
  - “Research a search component in .NET3.5 which is fit for our application”
- Non-functional user story
  - “The Student Affairs Information System is up and running 99.9% during the registration time period defined in the Academic Calendar.”
- Documentation user story
  - “Develop a user manual for teachers to use teacher module”

# 3Cs of User Story

Tools & Techniques / Planning, Monitoring & Adapting

- Card
- Conversation
- Confirmation

# INVEST Model of User Story

Tools & Techniques / Planning, Monitoring & Adapting

INVEST model defines following characteristics of a user story

- Independent
- Negotiable
- Verifiable
- Estimatable
- Sized Appropriately
- Testable

# Epic, Feature, Story, Task

Tools & Techniques / Planning, Monitoring & Adapting

- Epic is a collection of features. An epic is typically 1-3 months in duration
- Feature is collection of stories. A feature is typically 2-4 weeks in duration
- User-story is smallest unit of requirement created from features. A user-story is typically less than a week in duration
- Task are smallest unit of executable items which team members assign to themselves to complete a user story. A task is typically of 8 hours in duration

# Agile Planning

- Agile project management does not rely on big bang planning rather it believes in level of planning.
- Three level of planning in agile are
  - Release Planning
  - Iteration Planning
  - Daily Planning

# Agile Planning

## Release Planning

- Creating a release plan is responsibility of product owner
- If some user story cannot be estimated due to technical complexity then technical spikes are created
- Typical length of a release varies between 3-6 months. For every month of work you can spend max one day for release planning. Thus 3 month release you can spend max 3 days.
- Release planning should not be done without knowing velocity
- It takes 3-4 uninterrupted iterations to benchmark velocity for a project team
- Release planning depends upon
  - Deadline from competitor
  - Supporting the contract
  - To meet predetermined schedule
  - Supporting financial deadline
  - When there is enough value
  - To test the product
- Two types of release plan
  - Scope-boxed release plan
  - Time-boxed release plan

# Agile Planning

Tools & Techniques / Planning, Monitoring & Adapting

## Release Planning



Product owner is ready with prioritized product backlog (prioritization is done based on business value)

Product owner defined release goals

During release planning product owner picks up only those user stories and features which helps him achieving release goals

All the selected user stories are pushed into release backlog

User stories in release backlog are size estimated using agile estimation techniques like story points

User stories in release backlog are again prioritized. It helps in creating number of iterations and iteration plan.

# Agile Planning

Tools & Techniques / Planning, Monitoring & Adapting

## Iteration/Sprint Planning

- In iteration planning team identifies the task to be performed for each user story.
- Tasks are pushed into iteration backlog
- Iteration planning is held after retrospective meetings of last iteration
- Typical iteration length varies from 1-4 weeks.
- Iteration planning duration (1 hour for every week of iteration)
- Iteration planning identifies iteration backlog items, assumptions, risks, actions, dependencies
- Some teams start counting iteration from zero so the initial iteration is called **Sprint 0** or Iteration 0. In this iteration they take all technical spikes which will help them in estimating size of complex user stories. Preparing initial architecture, solving infrastructure setup and configuration issues. Following Sprint 0 they start release planning
- Tasks in iteration backlog are not assigned to any team member. Agile teams are self-organized and they pickup the task on their base.
- Number of hours available in any iteration are calculated as Ideal engineering hours.

# Agile Planning

Tools & Techniques / Planning, Monitoring & Adapting

## Iteration Planning

Product owner defines Iteration Goal

User stories to be completed in an iteration is prioritized and selected by product owner. Agile team helps him in prioritizing technically related user stories.

Team picks selected user story and identifies task to be done in order to complete selected user story. All the identified tasks are pushed into Iteration Backlog. Mostly these will be programmer-centric tasks. Consider definition of "done".

Effort estimate of each task is completed by team member using their experience

During planning team stop pushing tasks into iteration backlog when number of hours available for the iteration are exhausted

# Agile Planning

Tools & Techniques / Planning, Monitoring & Adapting

## Daily Planning

- Immediately after or before daily standup meeting sessions and the team updates the kanban board.
- This reflects the progress and planning of the day based on
  - The work progress of previous day
  - Any internal or external dependency not met
  - Something critical comes up
  - Client want to drop some user story

# Time-boxing

- Iteration length is time-boxed.
- Following sequence of activities takes place in any time-boxed iteration
  - Grooming product backlog (done by product owner, in parallel to iteration work)
  - Iteration Planning (1 hour for every week of iteration)
  - Daily stand-up (15 min, max 2 min for one person, typical team size of agile team is  $7\pm2$ )
  - Regular iteration work
  - Iteration Review (1 hour for every week of iteration)
  - Iteration Retrospective (1 hours for every week of iteration)

# Minimum Marketable Features (MMF)

Tools & Techniques / Planning, Monitoring & Adapting

- MMF is basic minimum set feature of the product so that people can start using it
- MMF can be made of a single or combined of multiple user stories
- MMF has business value to the end user

# Process Tailoring

- Tailoring is the process in which Agile Project Team customizes organization level defined process for their agile project.
- One of the major input for process tailoring is output of retrospective meetings
- Tailoring is done based on the understanding of project environment including customer need, skills etc.
- Some example of Process Tailoring
  - Define the “Definition of doneness”
  - Define the format of Retrospective meetings
  - Define Kanban board
  - Define columns of Product Backlog
  - Define some report format
  - Define Iteration Duration
  - Define process of automatic testing

# Process Tailoring

Tools & Techniques / Planning, Monitoring & Adapting

## Tailoring Mistake

- **Unfamiliarity:** Early adaptor implements whatever works for them temporarily not what is right and then fail down the line
- **Top-down Thinking :** Agile is successful when it is adopted from bottom. So it should be bottom-up
- **Culture change:** Undermining cultural changes
- **Incomplete implementation:** Thinking that it is waste of time same activity every time so not giving due importance to practices
- **Silver Bullet Syndrome:** To deliver a successful project you need good tools, experienced people, high team spirit, SMEs etc. Process is just one but important aspects of all these.

# Agile Contracting Methods

Tools & Techniques / Planning, Monitoring & Adapting

- Time & Material Contract
  - Most suitable contract type for agile project
  - Execution is driven by customer and they decide when a team member is required
  - Mutual trust is required in terms of skills provided to do the work and money will be provided for resource whether work is given by customer or not.
- Fixed Price (FP) Contract
  - Most unsuitable for agile project which expects you to understand the full scope of work before you bid for the project
  - During execution customer need to help in prioritization while vendor may end up wasting efforts for nothing
  - FP can work in agile projects only
    - If customer should pay for every sprint
    - If provision to convert FP into T&M at any point of time is there
    - Provision to terminate by any party at any point of time is there

# Agile Contracting Methods

Tools & Techniques / Planning, Monitoring & Adapting

- Time & Material with Cap Contract
  - Customer can mention maximum number of people required in project and he can ask for those people at any point of time
  - Vendor need to ensure those resources are available on demand and have updated knowledge and skills to perform the work
- Fixed Price (FP) Per Story Point
  - From vendor's point of view this is better than fixed price
  - A method to calculate story point need to determined and agreed in advance because story point rate is agreed earlier than story point calculation
- Fixed Price per Release
  - With respect to customer and vendor both this is better than FP and FP Per Story Point
  - Number of iteration in release should be defined upfront
  - Duration of each iteration need to defined earlier

# Agile Project Tracking

1. Information Radiators
2. Self-assessment
3. Kanban Board
4. Cumulative Flow Diagram (CFD)
5. WIP Limit
6. Burn Charts
7. Retrospectives

# Kanban Board

- After iteration planning is complete and before agile team starts working on task of the iteration, agile team need to pull the task from sprint backlog and put on kanban board
- Kanban board has several columns. You can customize these columns based on your project characteristics. Typical columns are “User Story”, “To be done”, “Doing”, “Done”, “Accepted”
- You keep moving tasks from of any user story from left to right based on the progress. Finally when all the work is done all the cards will move under “Done” and user story will move user “Accepted” column

# Kanban Board

Tools & Techniques / Planning, Monitoring & Adapting/ Agile Project Tracking

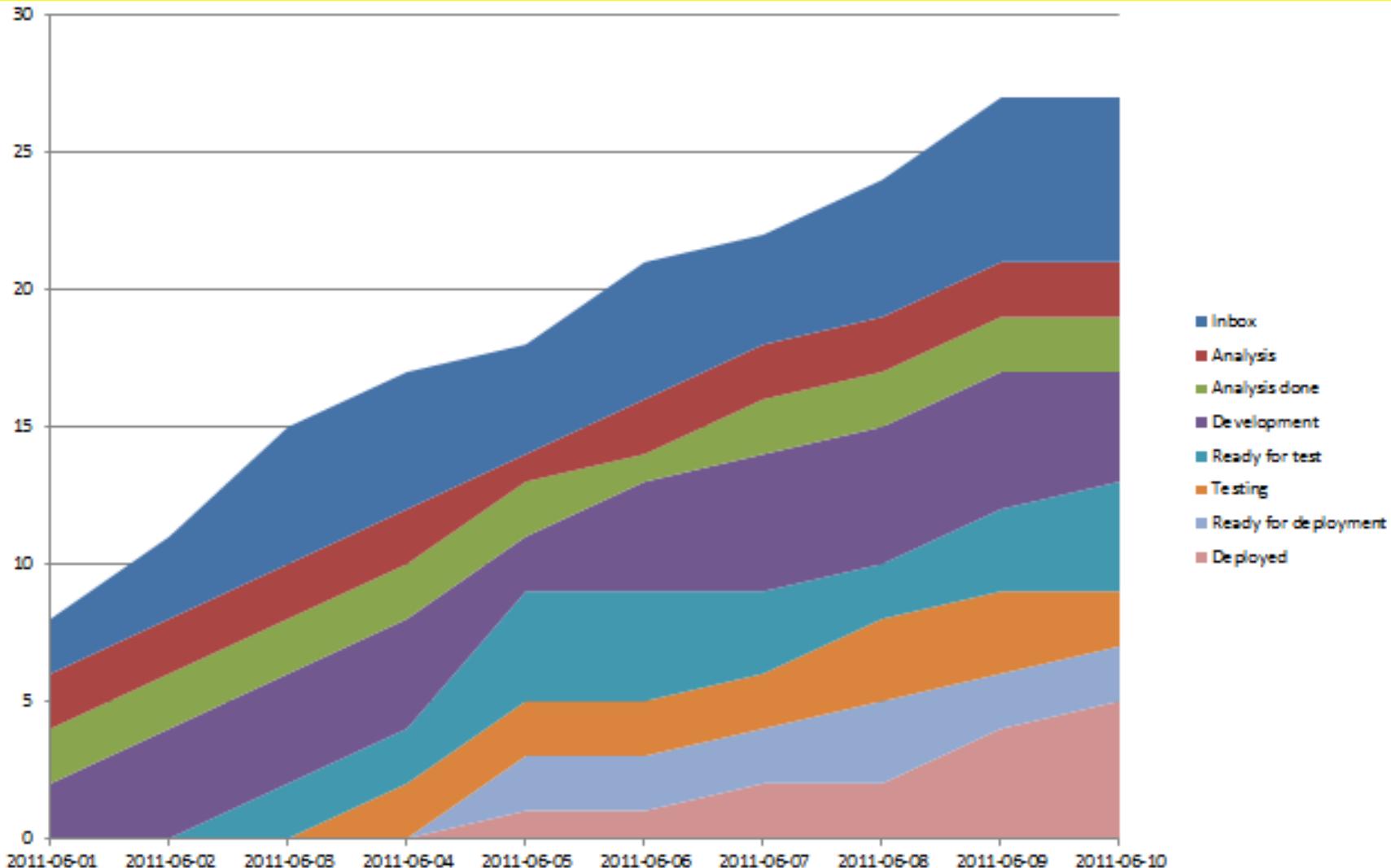
# Story Kanban Board

# Task Kanban Board

TODO	ANALYSIS 4	DESIGN 2	Proposed 4	READY for Dev 4	Active	5	READY for Test	TEST 1	READY for Validation
					Task	DEV			

# Cumulative Flow Diagram (CFD)

Tools & Techniques / Planning, Monitoring & Adapting/ Agile Project Tracking



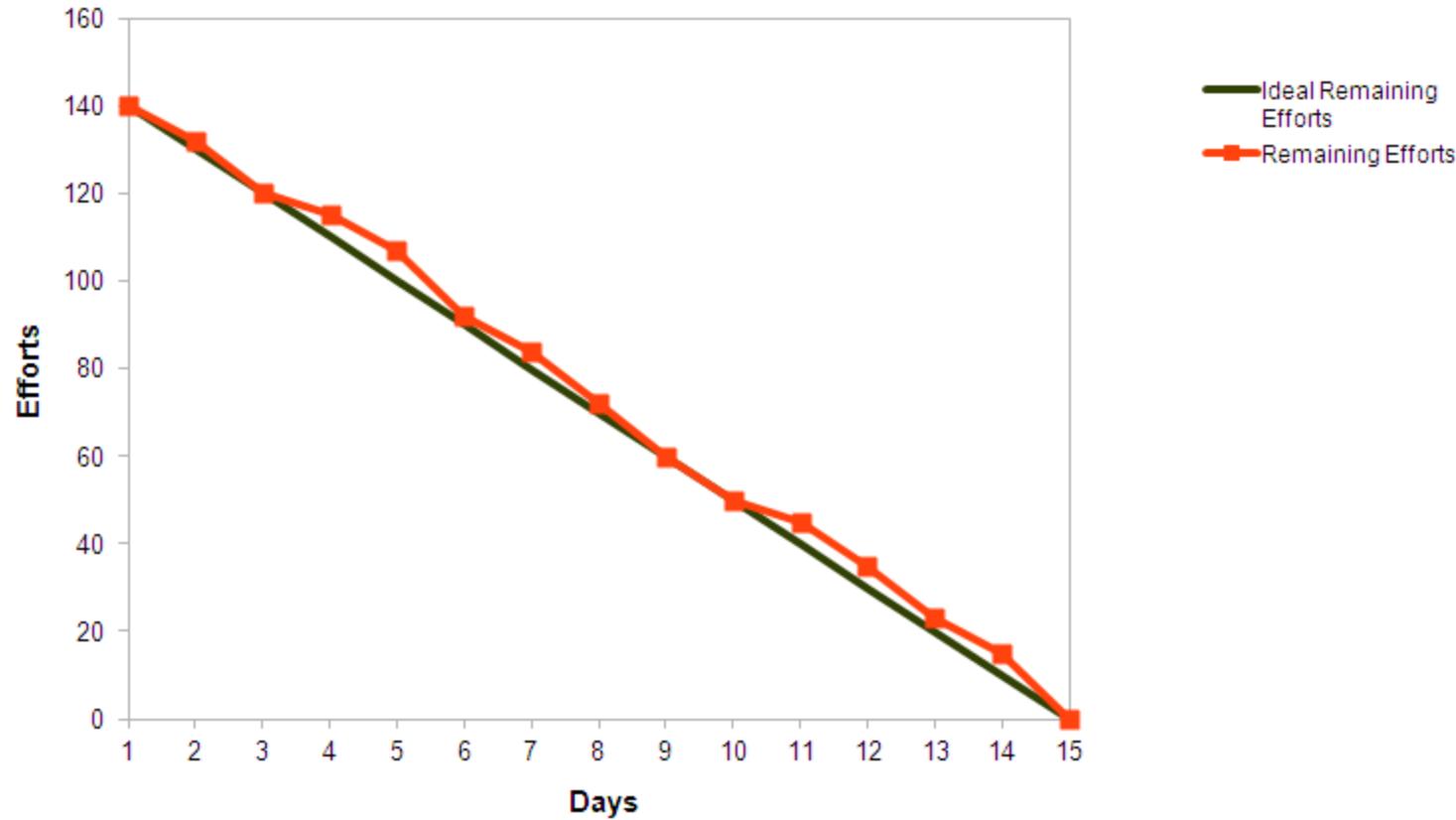
# WIP Limit

- WIP limit is maximum number of work in progress items
- In Kanban concept agile team should ensure that during execution at no point of time items in “in-progress” column should cross a defined WIP limit
- “In-progress” work is treated as of no-value thus it is treated as waste. Therefore to maximize the value focus should be to convert WIP items into completed
- Just-in-time (JIT) concept should be utilized to keep WIP limit smallest possible
- Low WIP limit is also a challenge as it creates wastage of machine and workforce time

# Burn Chart

## Burn Down Chart

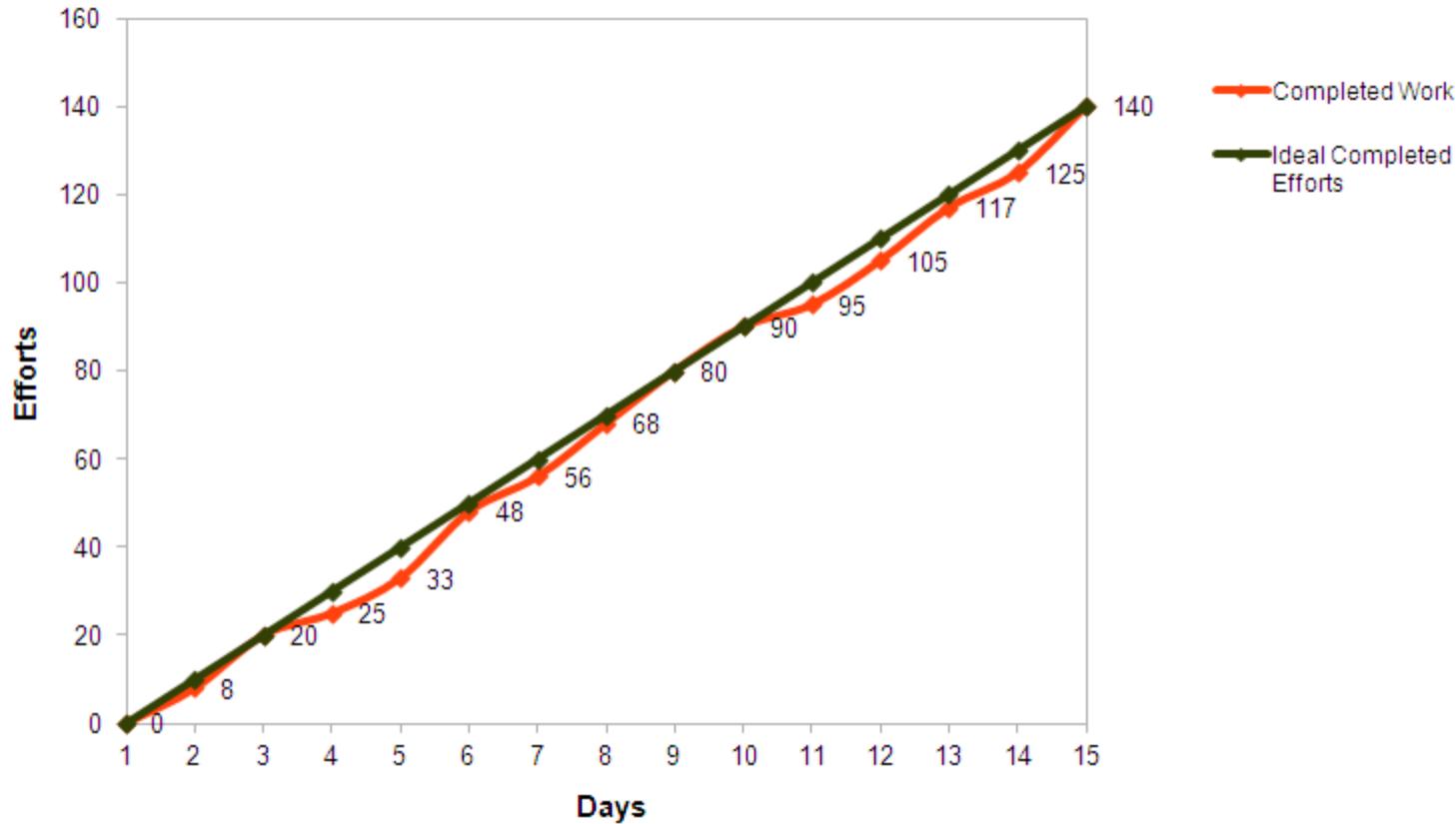
### Burndown Chart for Sprint 1



# Burn Chart

## Burn Up Chart

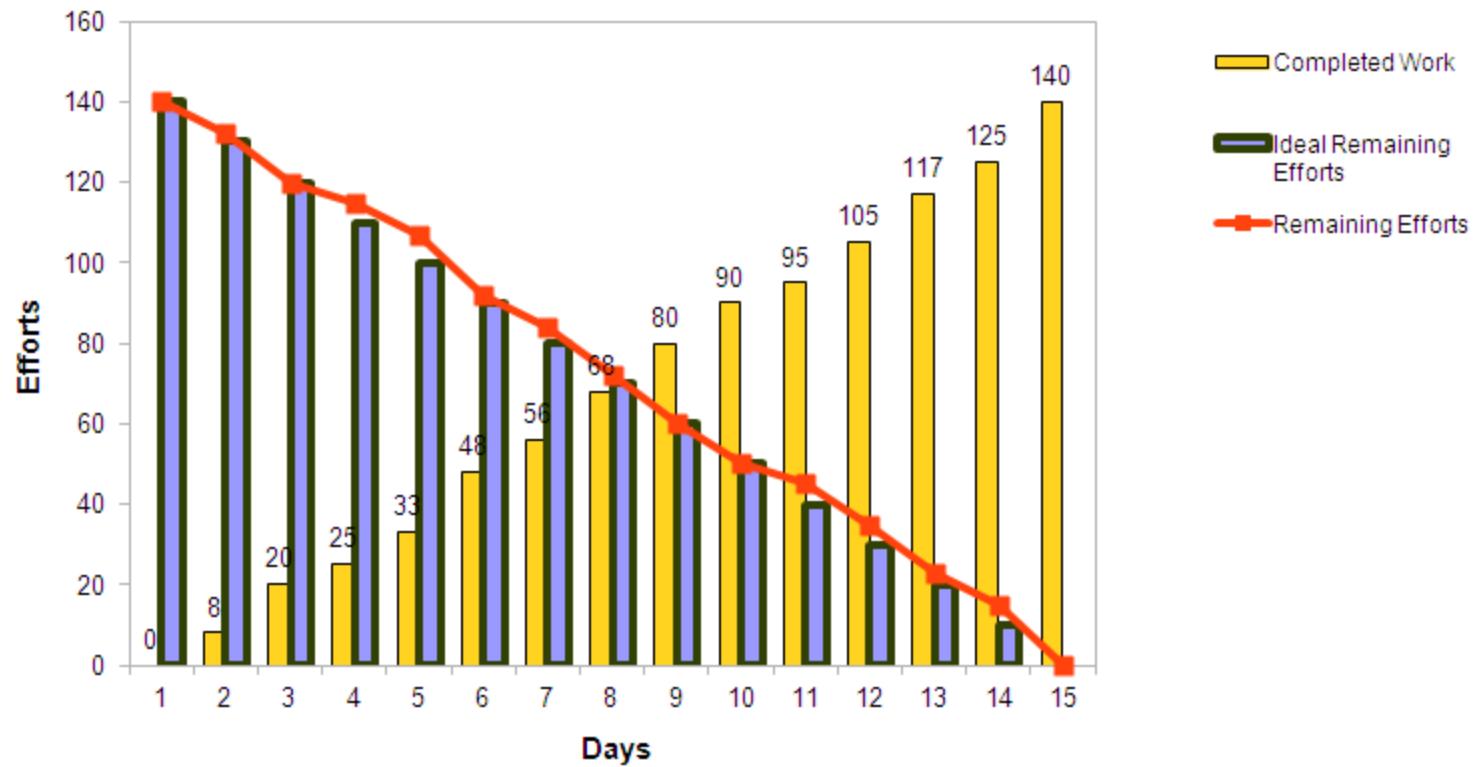
### Burnup Chart for Sprint 1



# Burn chart

## Burn-down & Burn-up combine chart

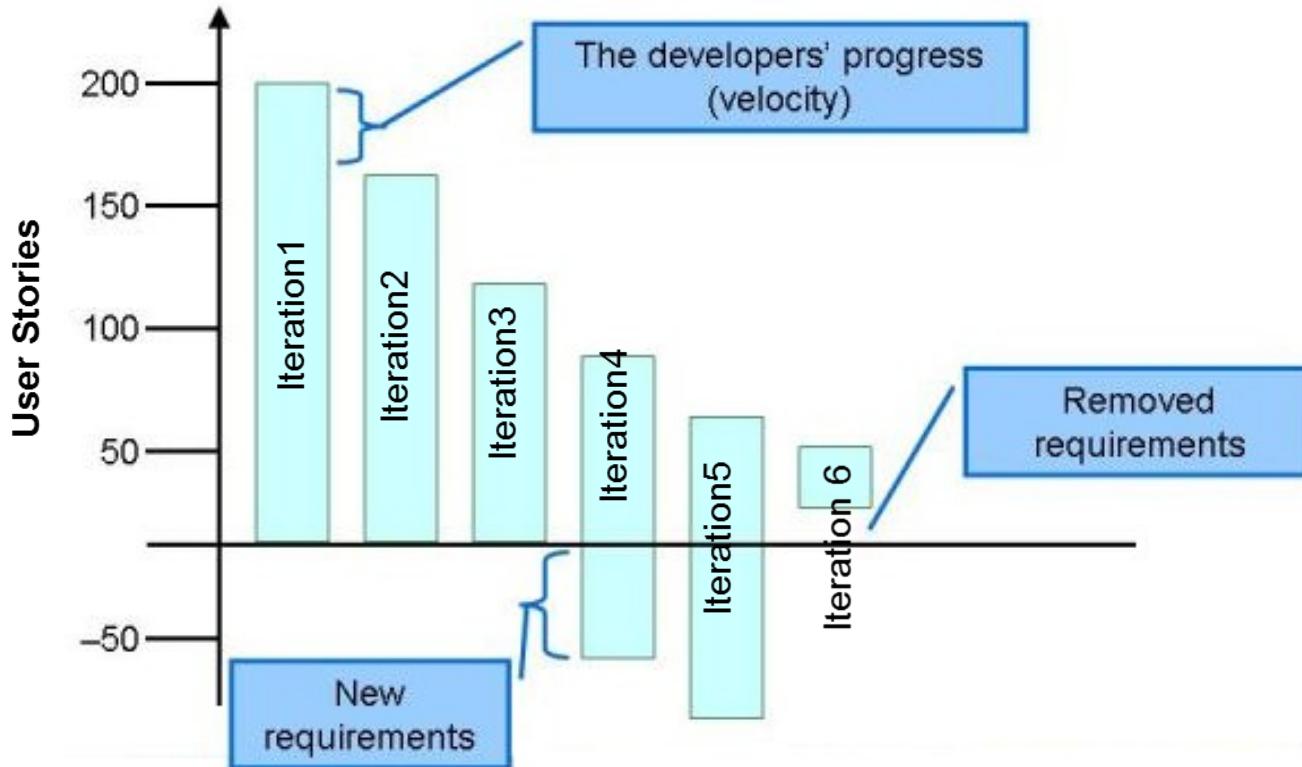
Burndown & Burnup Chart for Sprint 1



# Burn Chart

## Burn Down Bar Chart

### Release burndown bar



# Retrospectives

Tools & Techniques / Planning, Monitoring & Adapting/ Agile Project Tracking

- Retrospective is conducted by agile team at the end of each sprint and each release
- Retrospective sessions are self-reflection sessions, to understand how we are doing and how can we improve it further. Kaizen is at the heart of every retrospective session.
- Retrospective leads to adaptive planning
- Depending upon the need, confidence of team, time of retrospective, nature of team members structure of retrospective can be adopted

# Retrospectives

Tools & Techniques / Planning, Monitoring & Adapting/ Agile Project Tracking

## Structure of retrospective

- Set the stage (set ground rule/ working agreement)
- Gather Data (Share relevant data)
- Generate Insights (ask why it happened and how it happened)
- Decide what to do (Action plan)
- Close Retrospective (Appreciate participation)

# Retrospectives

Tools & Techniques / Planning, Monitoring & Adapting/ Agile Project Tracking

## Methods to generate insights

1. Brainstorming
2. Force Field Analysis
3. Five Whys?
4. Fish bone Diagram
5. Pattern and shifts
6. Prioritize with dots
7. Report out with synthesis
8. Identify Themes
9. Learning Ideas

# Retrospectives

Tools & Techniques / Planning, Monitoring & Adapting/ Agile Project Tracking

## Types of Retrospectives

- **Pillars Of Agile Spiderweb Retrospective** : A retrospective in which team rates its abilities in each of the categories, displayed the different ratings on a spider graph, and then discussed the result. To talk about what abilities are important to an Agile team and how agile team rates against them. **Dur: 60**
- **Appreciative Retrospective** : Uses Appreciative Inquiry to identify what went so well. There is no blame or negativity, and builds on the Prime Directive, that everyone in the team did the best job they could possibly do. It reminds everyone that what good job they are doing rather than focusing on negatives every time you run a retrospective. **Dur: 60**
- **Top 5** : Participants choose top 5 issues and bring them along for group to discuss and resolve. Expose the most pressing issues in an initially anonymous manner and determine the most effective actions to resolve them. **Dur: 45**

# Retrospectives

Tools & Techniques / Planning, Monitoring & Adapting/ Agile Project Tracking

## Types of Retrospectives

- **Plan of Action** : A Retrospective Technique for short term actions from long term goals. Really good for forcing achievable actions from a retrospective. **Dur: 40**
- **Start, Stop, Continue** : The facilitator captures team open-ended feedback using a wheel that encourages team members to assess an iteration or milestone. What should be start new, what should we stop, what should be continue from last iterations. **Dur: 10 to 25**
- **6 Thinking Hats Retrospective** : Uses De Bono's 6 Thinking Hats to investigate process improvement. General use, but also a good alternative to shuffling card type retrospectives. **Dur: 60**

# Retrospectives

## Types of Retrospectives

- **Each One Meets All :** The method ensures that each participant meets and interacts with every other participant. This retrospective makes sense when retrospective participants do not know each other well. **Dur : Variable!**
- **The Complexity Retrospective :** Use various tools such as a complexity radar to discover and find out how to deal with the complexity in your project. Many projects go awry due to excessive complexity; use this plan to evaluate whether your team is approaching things in the simplest way that can work; especially when the deadlines begin to loom. **Dur : 40-60**
- **Force Field Analysis :** A plan designed around the force field analysis technique. A retrospective for your whole company/department or to analyse a particular topic. **Dur : 60**

# Retrospectives

## Types of Retrospectives

- **Pomodoro Retrospective** : Focused and time-constrained by using the Pomodoro technique. Useful for determining a single action to improve the work of a small team. **Dur : 25**
- **Retrospective Surgery** : A retrospective for retrospectives. To learn how to improve the effectiveness of your retrospectives. **Dur : 60+**
- **Questions Retrospective** : Iteration retrospective. To get different perspectives on the same events. **Dur : 60**

# Retrospectives

## Types of Retrospectives

- **Four L's Retrospective** : Liked – Learned – Lacked – Longed For. Iteration and project retrospectives as well as for retrospection of training and conference events.. **Dur : 90 - 120**
- **Sailboat** : What anchors slow the team down, what wind propels it forward?. Good for the "gather data" and "generate insights" portions of a retrospective. **Dur : 90 - 120**
- **Weekly Retrospective Simple + delta** : Review, Plus-Delta, Vote, Actions, Owners. A weekly retrospective for your project. **Dur : 60**
- **Jeopardy Retrospective** : Use the answers as base to get all the good things and bad things that happened. A different way to "gather data" and to get all different opinions on a subject. **Dur : 60**

# Retrospectives

Tools & Techniques / Planning, Monitoring & Adapting/ Agile Project Tracking

## Temperature Reading

This retrospective can be conducted, to know the feeling, sentiments of the team. This is conducted by a facilitator and it follows following structure. Adequate time should be give to each part.

- Specific Appreciation
- New Information
- Puzzles
- Complaints with Recommendations
- Goals, Hopes and Wishes

# Retrospectives

Tools & Techniques / Planning, Monitoring & Adapting/ Agile Project Tracking

## ROTI (Return on Time Invested)

- Feedback and Adaptation are part of Agile Project Management. Feedback starts with meeting and specifically retrospective meetings.
- ROTI (Return on Time Investment) is a quick and easy method to gauge the time spent on meetings or workshops, and to improve their effectiveness.
- Take 5-15 minutes at the end of the meeting to ask participants to rate their return on time invested, using the **Fist of Five technique**
- If most of number you are getting is 1 or 2 then it means something is serious and you are wasting time and something different need to be done
- ROTI is quick, easy, sometimes funny, and works very well, even with top management.

# Innovation Game

Tools & Techniques / Planning, Monitoring & Adapting

- Product development is innovative game therefore everybody need to contribute and play is role in the game with the spirit of game. It's a game of cooperation and collaboration.
- Things becomes boring in game when
  - You are doing same thing again and again without innovating some other method of doing work
  - People do not take action and they talk a lot
  - Only few people are participating and others are not getting involved
- Therefore
  - Innovate the newness in the game
  - Innovate the new way of engagement
  - Do what you say
- Must read book on Innovation Game: A book by Alistair Cockburn : Agile Software Development : The Cooperative Game

# Recap

Tools & Techniques / Planning, Monitoring & Adapting

1. Progressive Elaboration
2. Adaptive Planning
3. User Story
4. Types of User Stories
5. 3Cs of User Story
6. INVEST Model of User Story
7. Epic, Feature, Story, Task
8. Agile Planning
9. Time-boxing
10. Minimum Marketable Features (MMF)
11. Process Tailoring
12. Agile Contracting Methods
13. Agile Project Tracking
  - Information Radiators
  - Self-assessment
  - Kanban Board
  - Cumulative Flow Diagram (CFD)
  - WIP Limit
  - Burn Charts
  - Retrospectives
14. Innovation Game

# **Discussions !**

# Topics

## Day 1

- ✓ History of Agile
- ✓ Agile Certifications
- ✓ The Most General APM Certificate
- ✓ The Foundation of Agile Project Management
- ✓ Project Management in General
- ✓ Agile Project Management
- ✓ Agile : Developing Mindset
- ✓ Agile : Methodologies
- ✓ Agile : Metrics & Estimation

vedavit

## Day 2

- ✓ Agile : Planning, Monitoring & Adapting
- Agile : Communication
- Agile : Analysis and Design
- Agile : Product Quality
- Agile : Value Prioritization



## Day 3

- Agile : Risk Management
- Soft Skills for Agile Project Manager
- Agile : Compliance
- Agile : Roles
- Scaling Agile
- Executing an Agile Iteration
- Professional Responsibility

# Agile : Communication

# Quotes Relevant to the Topic

- Most people do not listen with the intent to understand; they listen with the intent to reply.
- “Seek first to understand, then to be understood.” – Steven R Covey

# Topics

Tools & Techniques / Communication

1. Communication Management Plan
2. Team Space
3. Information Radiator
4. Osmotic Communication
5. Daily Stand-ups
6. Dimensions of Communications
7. Level of listening
8. Communication Skills
9. Communication Barriers
10. Communication Methods
11. Communication Model
12. Communication Technologies & Effectiveness
13. Collocated vs Distributed Team
14. Skills to Manage Stakeholders
15. Define Positive Values

# Communication Management Plan

Tools & Techniques / Communication

- Identify who need to involved in communication
- Identify what information they need, when and at what frequency
- Identify how they want to be communicated
- Understand why a particular information is required by a stakeholder, at a particular frequency, in a particular format
- Understand the impact of not giving a particular information in desired format or frequency to a particular stakeholder

## Important Note

- Expand efforts in communicating only when providing information helps your project or not providing may cause some issue
- Use appropriate technology for each communication need
- **Cost of communication** is cost of communication technology, cost of time in making, sending, receiving, analyzing, responding queries etc.
- Optimize your report format, frequency and communication cost by knowing number of communication channels. Channels=  $N \times (N-1)/2$
- Document the path of escalation for effective communication

# Team Space



vedavit

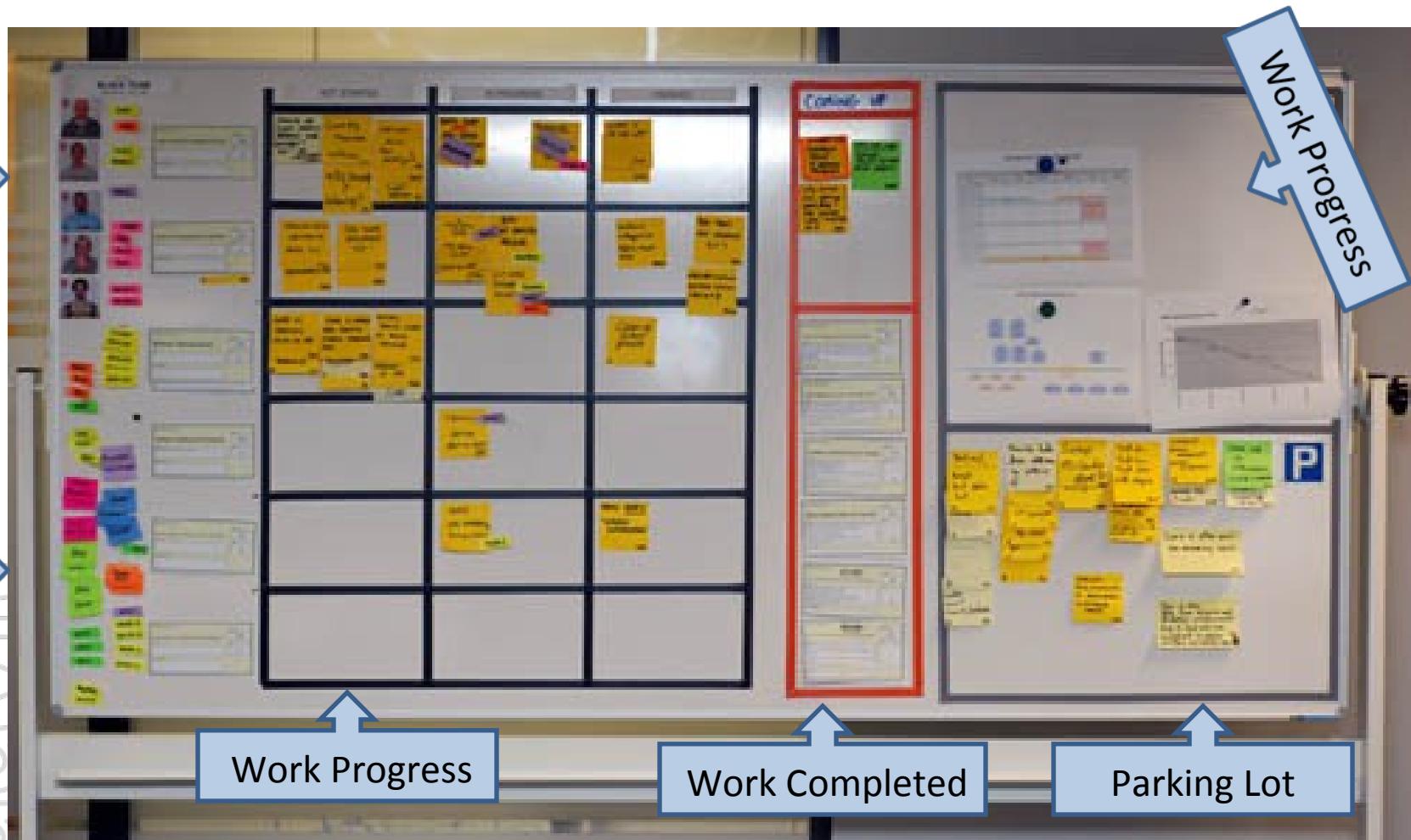
Team space is working area of agile project team. Only those people who are contributing to project directly should be allowed to sit, stay and talk in that area

This area should be

- **Informative**
- **Spacious**
- **Comfortable to work**
- **Shielded from external noise**
- **Easy to collaborate**

Having expert in the same room where rest of the team is sitting is called “Expert in earshot”

# Information Radiator



# Osmotic Communication

- Agile project relies on collocation, minimum documentation, least reporting, maximum constructive engagement. This can be achieved if frequently sought important information is available in published form in team space
- Whoever need the information can go and get the information without wasting time in requesting, making, sending, receiving the information
- Benefits
  - Least cost, effort and time waste in communication
  - Updated information is always available without making new reports
  - People can get whatever particular information they are looking for
  - Happens at the same time
  - Feedback loop is quick
  - Those people who are left in regular reporting also get benefitted
  - No junk, old, repetitive information, but fresh and useful.
- Dis-benefits
  - Some people get extra information which they do not need
  - It is left to individual's interpretation

# Daily Stand-ups

- Daily stand-up is heart beat of agile project management
- Team meets daily (typically in working area, war room) at fixed time (time should not be changed) preferable first thing in morning
- This is not reporting session but information sharing among team members
- Only “Pigs” allowed to speak, “chicken” should listen (they are not allowed to interfere or ask while stand up meeting is in progress)
- It is 15-20 min meeting, conducted while everybody is standing (showing the sense of urgency). A person should not take more than 2 min to update this work status.
- Any one in the team can facilitate this meeting
- Project manager notes the impediments and start working on those immediately after the meeting is over.

# Daily Stand-ups

Tools & Techniques / Communication

3 Questions of standup meeting which every team member must address are

- 1.What did they do yesterday?
- 2.What are they planning to do today?
- 3.Any impediments on the way today?



# Dimensions of Communications

- Internal vs External
- Formal vs Informal
- Vertical vs Horizontal
- Official vs Unofficial
- Written vs Oral
- Verbal vs Non-verbal

# Level of Listening

- Level 1: Internal Listening (interpret in own language)
- Level 2: Focused Listening (Responding without interpreting in own language)
- Level 3: Focused but understand global environment (team, organization, customer etc)

# Communication Skills

- Active Listening
- Probing Ideas for Better Understanding
- Fact Finding
- Expectation Management
- Conflict Resolution
- Next Step Identification
- Making Commitment
- Contribute Effectively

# Communication Barriers

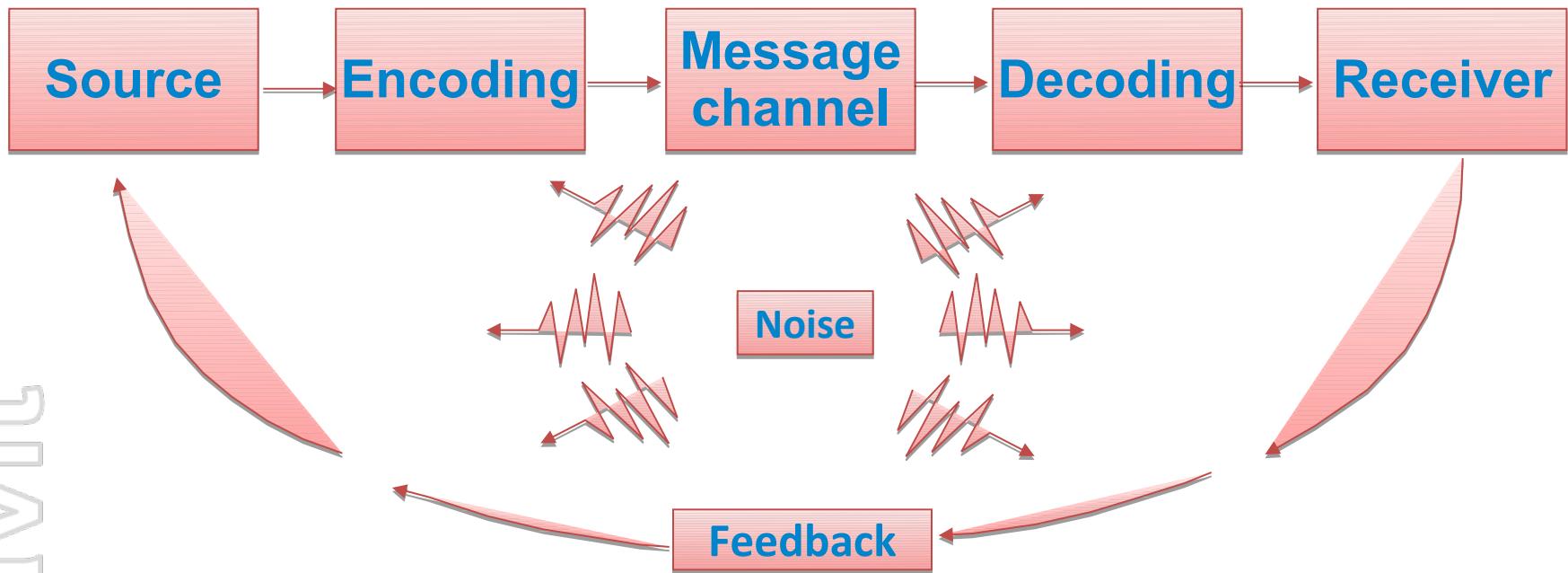
- Physical Distance => Physical Proximity
- Working in Different Shifts => Temporal Proximity
- Human Language
- Technical Knowledge
- Detrimental Attitude
- Difference in Culture
- Difference in Project Environment
- Attitude, willingness => Amicability
- Communication Channels

# Communication Methods

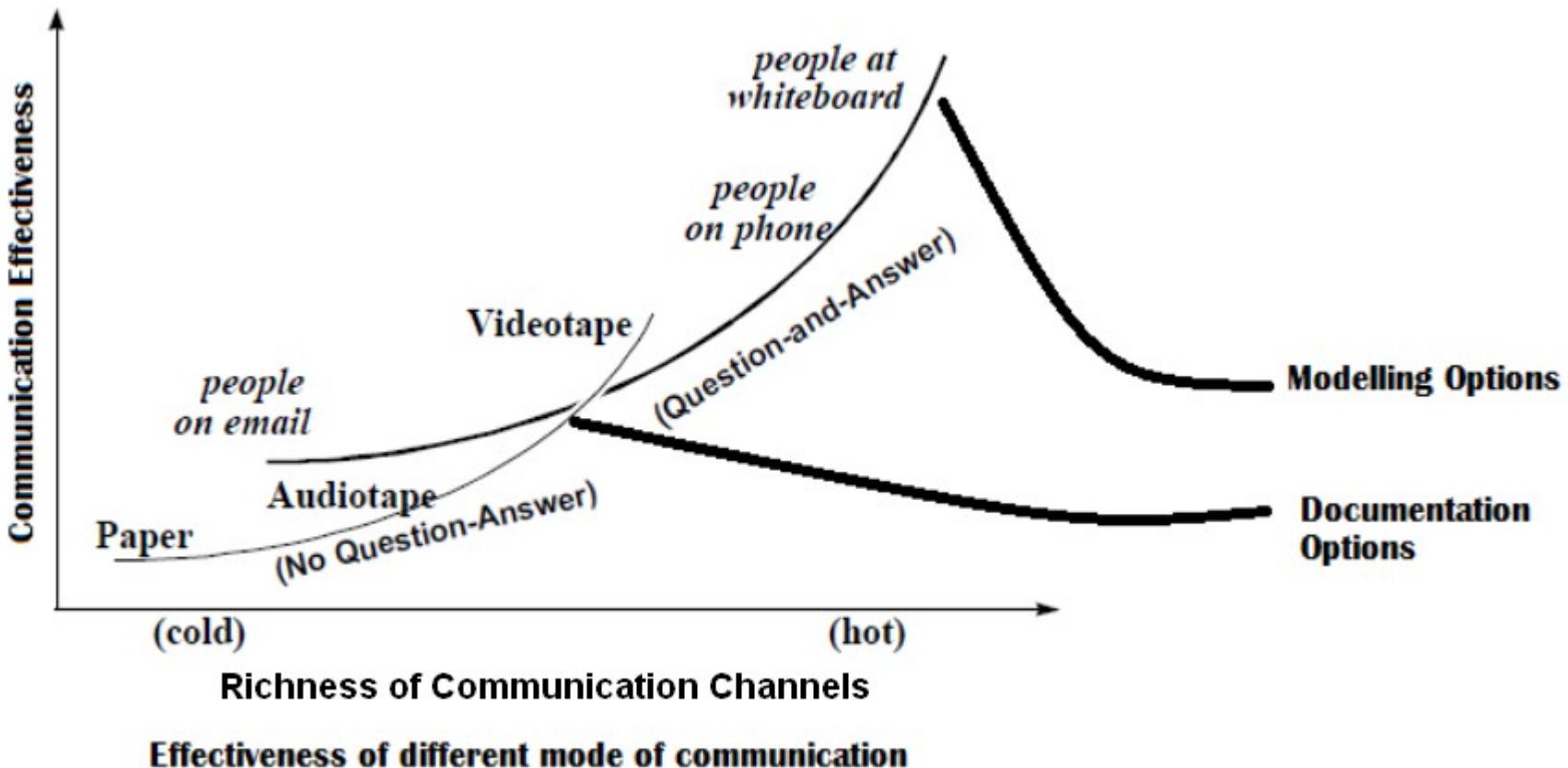
Tools & Techniques / Communication

- Push Communication
- Pull Communication
- Interactive Communication

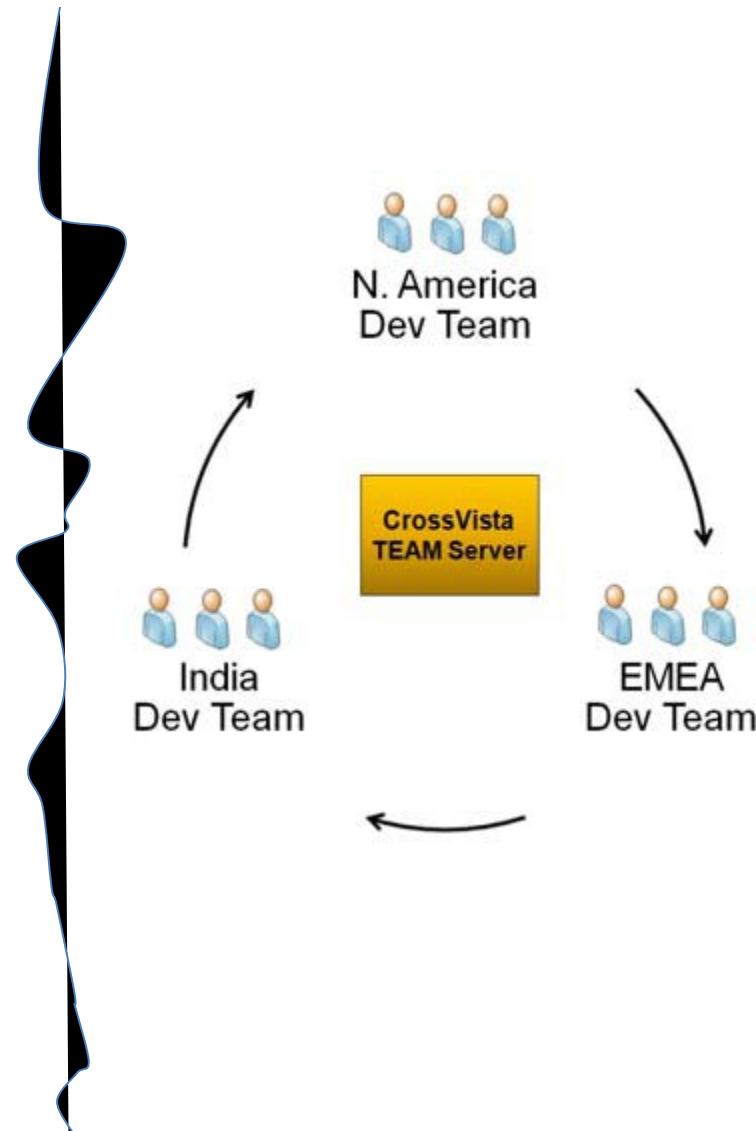
# Communication Model



# Communication Technologies & Effectiveness



# Collocated vs Distributed Team



# Skills used to Manage Stakeholders

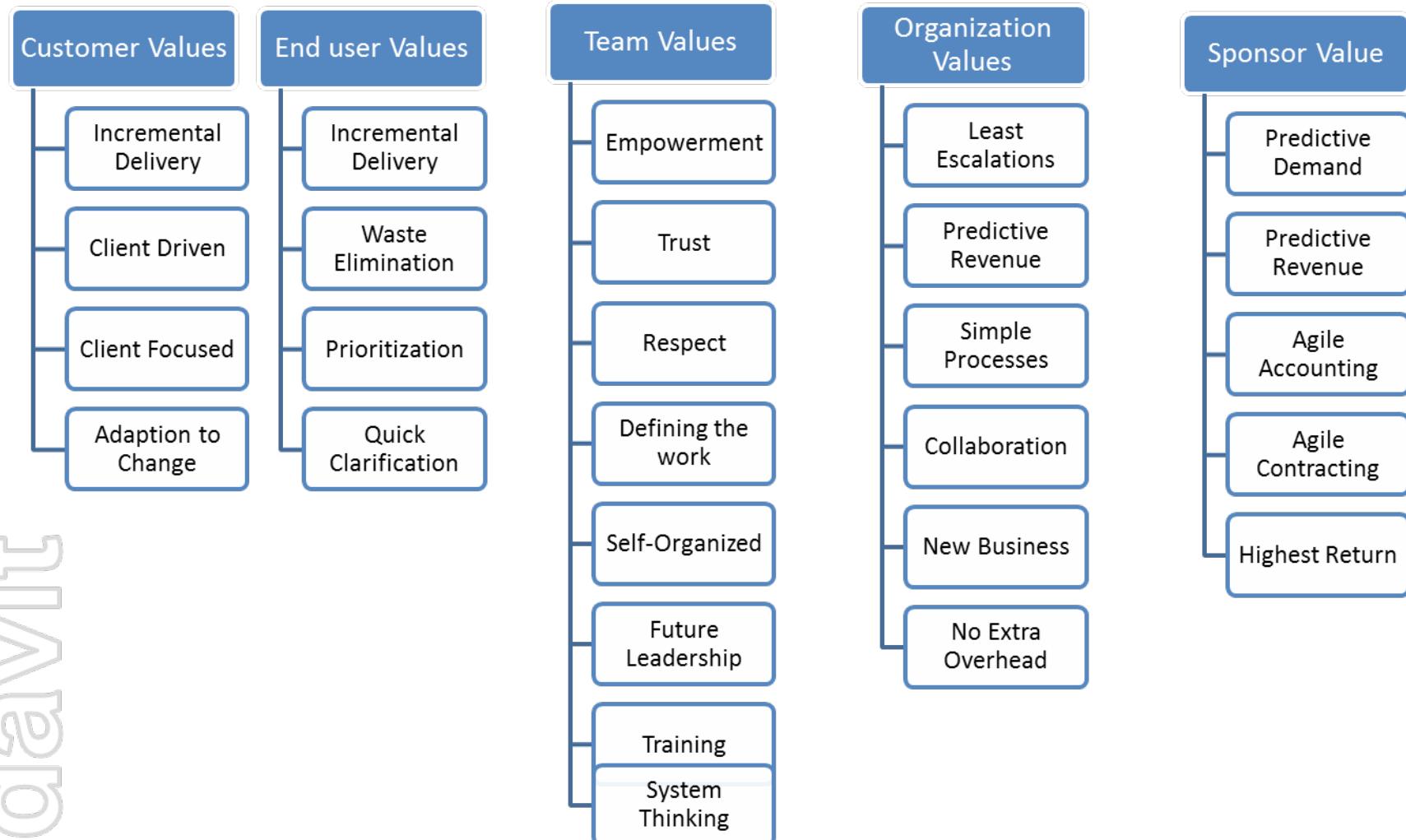
Tools & Techniques / Communication

- Interpersonal Skills
  - Conflict Resolution
  - Active Listening
  - Trust Building
  - Influencing
- Managerial Skills
  - Presentation
  - Public Speaking
  - Writing
  - Negotiation

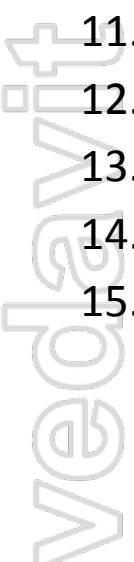
# Define Positive Values

- For effective communication we must understand values of various stakeholders. This is one of the major shift of management from Traditional to Agile Project Management
- Understand and define values of
  - Customer
  - End user
  - Team
  - Organization
  - Sponsor

# Define Positive Values



# Recap

- 
1. Communication Management Plan
  2. Team Space
  3. Information Radiator
  4. Osmotic Communication
  5. Daily Stand-ups
  6. Dimensions of Communications
  7. Level of listening
  8. Communication Skills
  9. Communication Barriers
  10. Communication Methods
  11. Communication Model
  12. Communication Technologies & Effectiveness
  13. Collocated vs Distributed Team
  14. Skills to Manage Stakeholders
  15. Define Positive Values

# **Discussions !**

# Topics

## Day 1

- ✓ History of Agile
- ✓ Agile Certifications
- ✓ The Most General APM Certificate
- ✓ The Foundation of Agile Project Management
- ✓ Project Management in General
- ✓ Agile Project Management
- ✓ Agile : Developing Mindset
- ✓ Agile : Methodologies
- ✓ Agile : Metrics & Estimation

vedavit

## Day 2

- ✓ Agile : Planning, Monitoring & Adapting
- ✓ Agile : Communication
- Agile : Analysis and Design
- Agile : Product Quality
- Agile : Value Prioritization



## Day 3

- Agile : Risk Management
- Soft Skills for Agile Project Manager
- Agile : Compliance
- Agile : Roles
- Scaling Agile
- Executing an Agile Iteration
- Professional Responsibility

# Agile : Analysis & Design

# Quotes Relevant to the Topic

Never ever be intellectual slave of any one, even of your parents, gurus, scripture and God. Develop sense of discrimination and use your analytical, logical capabilities.

Sri Adi Sankarcharya

# Topics

Tools & Techniques / Agile Analysis & Design

1. Agile Chartering
2. Product Roadmap
3. Backlogs
  - Product
  - Release
  - Sprint
4. Story Maps
5. Requirement Collection and Validation
6. Agile Modeling
  - Prototype
  - Wireframes
  - Personas
7. Agile Design Principles

# Agile Chartering

Agile chartering is done before Release Planning. The main purpose of the project chartering is to establish a vision, goal and high level requirements. Agile charter like traditional project charters establish

- The purpose of project (why we are doing it)
- What we want to achieve
- Success criteria of project
- Role of senior people engaging in the project

# Product Roadmap

- A product roadmap is a plan where the features are planned for release on a timescale
- It talks about business value addition to the customer by each release
- Date, features and objective of each release
- Overall product features (product backlog and business story)
- Technical backlog (non-functional requirements, technical story)

# Product Roadmap

Tools & Techniques / Agile Analysis & Design

CONFIDENTIAL

## Company Roadmap subtitle

Thursday, October 27, 2011

ALL OK    LOW RISK    MEDIUM RISK    HIGH RISK

### COMPANY ACTIVITY



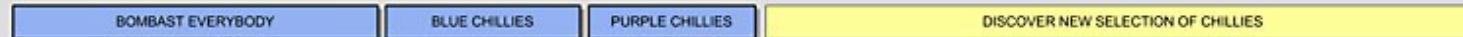
HR



FINANCE



SALES



BOARD OF DIRECTORS



### COMPANY GOALS / INITIATIVES



Page 1

# Backlog

## Product Backlog

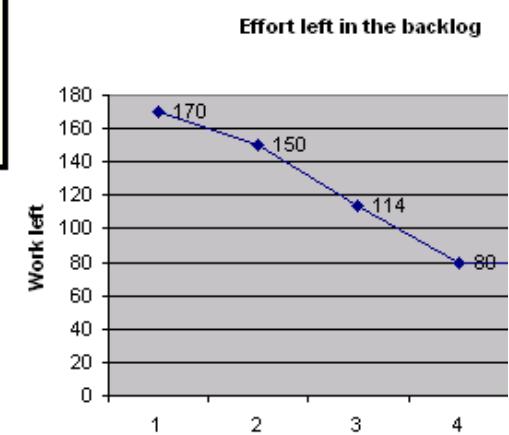
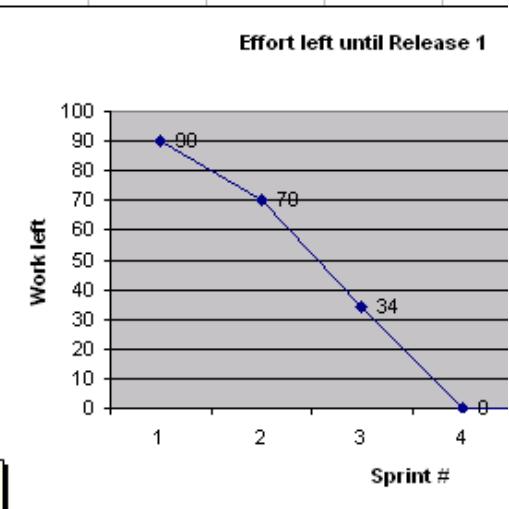
Priority	Estimate	Sprint	User Type	Story	Story Type
1	1	1	Customer	I can see when the next show will begin for the show page I am on	Story
2	2	1	Editor	I can select what I want to display for each "section" within the editorial content section of the page. My options include last episode, next episode, selected forum posts, selected editorial articles (tvg generated), no selection and free form text	Story
3	2	1	Editor	I can select what picture (if any) I want to display for the corresponding content section	Story
4	5	1	Editor	I can select the default tab for the user to see upon visit to the page, for each show	Story
5	5	1	Customer	I can roll over the fields in the media player and see the various tabs change	Story
6	13	2	Editor	I can modify the existing headline for any show page	Story
7	1	2	Customer	I can select another show page in the drop down list next to the countdown clock	Story
8	1	2	Customer	I can click "remote record" and have the show for the show page I am on record on my tivo device	Story
9	1	2	Customer	I can click "join the discussion" button (or link) on the show page which takes me to the appropriate forum page for that show	Story
10	1	2	Customer	I can see how many recent posts have been posted in the forum for the show page I am on	Story
11	3	2	Customer	I can see how many recent replies have been posted in the forum for the show page I am on	Story
12	5	2	Customer	I can blog about the show for the show page I am on (I need to be signed in to see this)	Story
13	13	3	Customer	If I am not signed in, I can see a link to sign in	Story
14	13	3	Customer	If I am logged in, I can click "favorites" and have the show page added to my favorites menu on the site	Story
15	13	4	Customer	If I have not contributed to the poll, I can see the poll questions and submit to the poll	Epic
16	20	5	Editor	I can create a new poll for a specific show	Epic
17	20	6	Editor	I can close an existing poll for a specific show	Epic

Product backlog grooming is continuous activity. PO is responsible for this.

# Backlog

## Release Backlog

ID	Description	Sprint #	1	2	3	4	5	6
			Effort needed for Release 1 as in the beginning of the sprint	90	70	34	0	0
1	Set up continuous integration system		5	0	0	0	0	0
2	Create compilable application skeleton		5	0	0	0	0	0
3	Display current temperature in a simplest possible way		13	0	0	0	0	0
4	Set up the web server for serving weather data		3	0	0	0	0	0
5	Implement stubby WeatherML support on the server side		13	0	0	0	0	0
Sprint 1	Make sample data go from server to device							
6	Graphics support on the client side		20	0	0	0	0	0
16	Make the graphics library draw some icon and sample temperature text		-	13	0	0	0	0
17	Draw the real weather screen		-	8	0	0	0	0
7	Implement support for several days		8	8	0	0	0	0
8	Implement support for rain, snow, etc. icons		2	2	0	0	0	0
9	City changing support		-	5	0	0	0	0
Sprint 2	Minimal working version							
10	Fetch one day temperature data from the weather provider system		+ ?					
11	Fetch rain, snow, etc details from the provider							
12	Fetch several days data from the provider							
13	Auto-refresh feature							
Sprint 3	Plug in the real weather data							
Release 1	Sellable version							
14	Inject simulated ads from the test server							
15	Plug real ads in							
18	Change current city automatically according to the cell info							
Sprint 4	Advertisements support							
Release 2	Ad-supported version							
	Effort in the whole backlog	170	150	114	80	80	80	
	Backlog state taken after the end of sprint 3 = after release 1							



Release backlog is created during release planning meetings. PO is responsible for this

# Backlog

## Sprint Backlog

- Unlike product backlog and release backlog which is comprised by user stories, sprint backlog is comprised of task
- Sprint Backlog can have following types of tasks
  - Design Tasks
  - Coding Tasks
  - Testing Task
  - Documentation Tasks
- No one is allowed to add user story or task in Sprint backlog but team can delete task if they feel out of time
- Sprint backlog is developed at the time of sprint planning

# Backlog

## Sprint Backlog

- Generally every task in sprint backlog should be related to some user story
- To identify task for user story programmers need to do the detail discussion about user story with PO
- Each task in sprint back log should have following information in the form of Task card
  - Task description
  - Estimated hours
  - Owner name (if taken by any team member)
  - Exit criteria
  - Verification method (test or inspection)
  - Who will perform verification

# Backlog

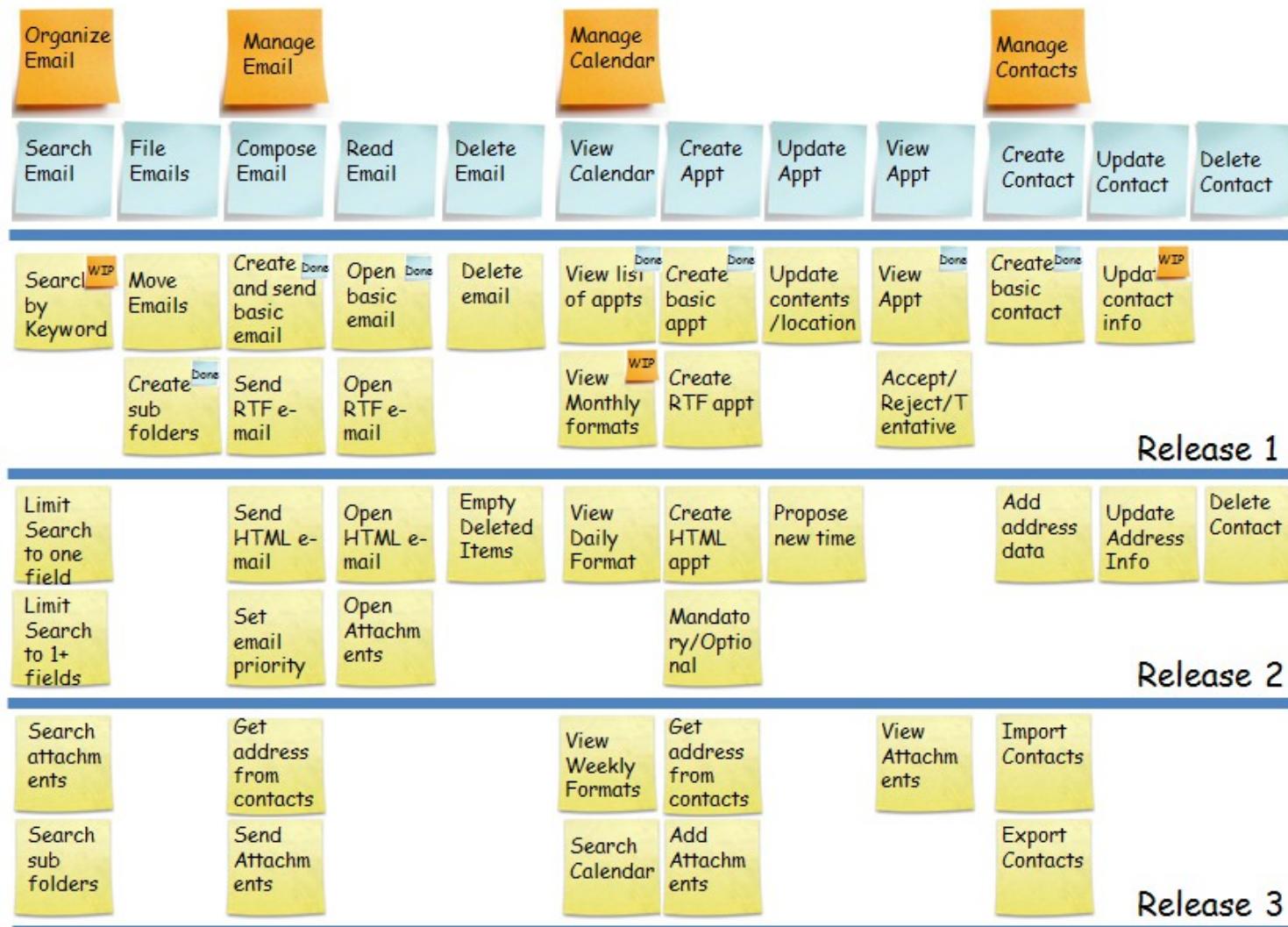
## Sprint Backlog

Story	To Do	In Process	To Verify	Done
As a user, I... 8 points	Code the... 9 Code the... 2 Test the... 8	Test the... 8 Code the... 8 Test the... 4	Code the... DC 4 Test the... SC 6	Code the... D Test the... SC 8 Test the... SC Test the... SC Test the... SC 6
As a user, I... 5 points	Code the... 8 Code the... 4	Test the... 8 Code the... 6	Code the... DC 8	Test the... SC Test the... SC Test the... SC 6

# Story-maps

- The main purpose of story mapping is to understand the end-to-end user requirements and stitch them in a logical thread so that users get maximum return on the business value
- Benefits of Story-maps
  - Overall workflow of the system help in understanding value chain.
  - Relationship between large stories and child stories get established
  - Completeness of backlog can be validated
  - A basis of prioritization
  - Helps in release planning

# Storymaps



# Requirement Collection and Validation

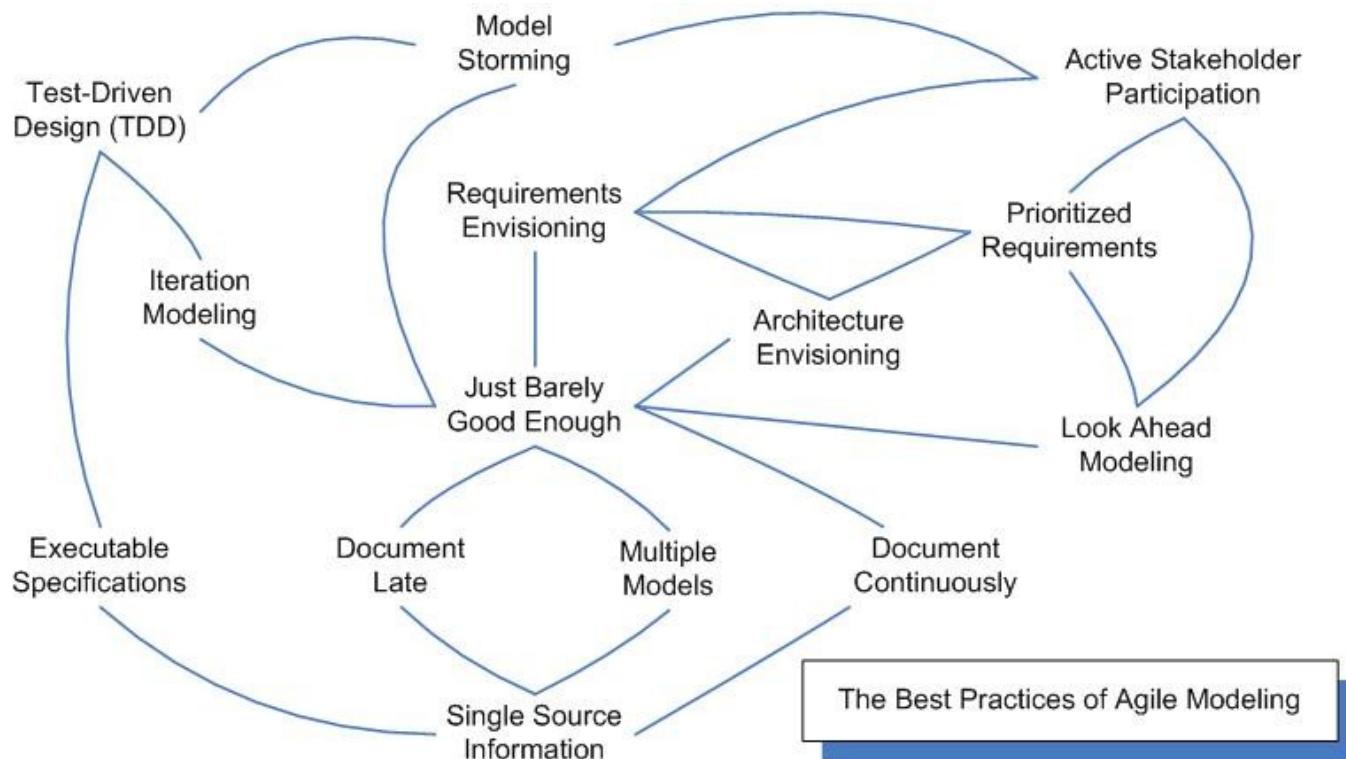
Tools & Techniques / Agile Analysis & Design

- Interviews
- Focus Group
- Facilitated Workshop

# Agile Modeling

## Tools & Techniques / Agile Analysis & Design

- Agile Modeling (AM) is a practice-based methodology for **effective modeling** and **documentation** of software-based systems.
- At a high level AM is a collection of best practices, depicted in the pattern language map below.
- At a more detailed level AM is a collection of values, principles, and practices for modeling software that can be applied on a software development project in an effective and light-weight manner.



Copyright 2005-2011 Scott W. Ambler

# Agile Modeling

## Agile requirement modeling techniques

- Active Stakeholder Participation
- Electronic Interviews
- Face-to-Face Interviews
- Focus Groups
- Joint Application Design (JAD)
- Legacy Code Analysis
- Observation
- On-Site Customer
- Reading

# Agile Modeling

## Artifacts of Analysis Modeling

- Activity Diagram
- Class Diagram
- Constraint definition
- CRC model
- Data flow diagram (DFD)
- Entity/Relationship (E/R) diagram (data diagram)
- Flow chart
- Robustness diagrams
- Sequence diagram
- State chart diagram
- System use case
- UI prototype
- Usage scenario
- Use case diagram

# Agile Modeling

**List of important Modeling which you may need to perform in Agile. Either you use some software for these or use plain word/excel/powerpoint or just draw them on white board.**

1. Acceptance Test
2. Business Rule
3. Change Case
4. Class Responsibility Collaborator (CRC) model
5. Constraint
6. Contract model
7. Data Flow Diagram (DFD)
8. Domain Model
9. Essential/Abstract Use Case
10. Essential/Abstract User Interface Prototype
11. Feature

# Agile Modeling

12. Free-Form Diagrams
13. Flow Chart
14. Glossary
15. Logical Data Model (LDM)
16. Network Diagram
17. Object Role Model (ORM) Diagram
18. Personas
19. Physical Data Model (PDM)
20. Robustness Diagram
21. Security Threat Model
22. System Use Case
23. Technical Requirement
24. Usage Scenario
25. User Interface Flow Diagram (Storyboard)

# Agile Modeling

26. User Interface Prototype
27. User Story
28. Value Stream Map
29. UML 2 Activity Diagram
30. UML 2 Class Diagram
31. UML 2 Communication/Collaboration Diagram
32. UML 2 Component Diagram
33. UML 2 Composite Structure Diagram
34. UML 2 Deployment Diagram
35. UML 2 Interaction Overview Diagram
36. UML 2 Object Diagram
37. UML 2 Package Diagram
38. UML 2 Sequence Diagram
39. UML 2 State Machine Diagram
40. UML 2 Timing Diagram
41. UML 2 Use Case Diagram

# Agile Design Principles

There are 5 principles of Agile design named as SOLID.

- **Single Responsibility Principle:** A class should have only one reason to change
- **Open Closed Principle :** Classes, modules and functions should be open for extension but closed for modification
- **Liskov Substitution Principle:** Derived types must be completely substitutable for their base types.
- **Interface Segregation Principle:** Client should not be forced to depend upon interfaces that they don't use
- **Dependency Inversion Principle:** High level modules should not depend on low level modules. Both should depend on abstractions. Abstractions should not depend on details. Details should depend on abstraction.

# Recap

Tools & Techniques / Agile Analysis & Design

1. Agile Chartering
2. Product Roadmap
3. Backlogs
  - Product
  - Release
  - Sprint
4. Story Maps
5. Requirement Collection and Validation
6. Agile Modeling
  - Prototype
  - Wireframes
  - Personas
7. Agile Design Principles

vedavit

# **Discussions !**

# Topics

## Day 1

- ✓ History of Agile
- ✓ Agile Certifications
- ✓ The Most General APM Certificate
- ✓ The Foundation of Agile Project Management
- ✓ Project Management in General
- ✓ Agile Project Management
- ✓ Agile : Developing Mindset
- ✓ Agile : Methodologies
- ✓ Agile : Metrics & Estimation

vedavit

## Day 2

- ✓ Agile : Planning, Monitoring & Adapting
- ✓ Agile : Communication
- ✓ Agile : Analysis and Design
- Agile : Product Quality
- Agile : Value Prioritization



## Day 3

- Agile : Risk Management
- Soft Skills for Agile Project Manager
- Agile : Compliance
- Agile : Roles
- Scaling Agile
- Executing an Agile Iteration
- Professional Responsibility

# Agile : Product Quality

# Quotes Relevant to the Topic

- Quality means doing it right when no one is looking

Henry Ford

- Quality is never an accident; it is always the result of high intention, sincere effort, intelligent direction and skillful execution; it represents the wise choice of many alternatives.

Unknown

# Topics

Tools & Techniques / Product Quality

1. Cost of Quality
2. Definition of Done
3. Feedback Techniques of Agile Project Management
4. Verification and Validation
5. Test Driven Development (TDD)
6. Acceptance Test Driven Development (ATDD)
7. Exploratory Testing
8. Continuous Integration
9. Continuous Process Improvement
10. 7 Wastes
11. House Keeping Standard (5S)
12. Version Control
13. Agile Tooling Mistake
14. Typical Software Tools in Agile Projects
15. Quality Tools

# Cost of Quality

## Cost of conformance

- Appraisal Costs
  - checking & Testing Purchased Goods
  - In-process and Final Inspection Testing
  - Field Testing
  - Product, Process and Service Audits
  - Calibration of Measuring and Test Equipment
- Prevention Costs
  - Capability Evaluation
  - Error Proofing
  - Quality Planning
  - New Product Review
  - Quality Improvement Project
  - Quality Education and Training
  - Supplier Evaluation

## Cost of nonconformance

- Internal Failure Costs
  - Re-design
  - Downtime
  - Re-testing
  - Shortages
  - Delays
  - Rework
  - Lack of flexibility
- External Failure Costs
  - Environmental Costs
  - Loss due to Sales Reductions
  - Warranties
  - Repairing Goods
  - Complaints

# Definition of Done (DoD)

- The DoD changes over time. Organizational support and the team's ability to remove impediments may enable the inclusion of additional activities into the DoD for features or sprints.
- Continuous Integration (CI) helps you validating the “Doneness”
- There are 3 level of DoD
  - Story DoD
  - Iteration DoD
  - Release DoD

# Definition of Done (DoD)

Tools & Techniques / Product Quality

## Story “Done”

- Unit test should provide 60-70% test coverage
- Story is either written in pair or reviewed by peer
- All code checked in
- All unit code passed
- All acceptance test case passed
- Story accepted by owner

# Definition of Done (DoD)

## Iteration “Done”

- Iteration should have defined Iteration Goal
- All acceptance test cases should run for all user stories in Iteration
- All stories completed must be accepted by the product owner
- Defects identified are fixed or planned for future
- Code performance is tested and accepted
- If database is involved then database script should be available, automated and tested
- Backup of iteration work product is taken

# Definition of Done (DoD)

## Release “Done”

- Release should have defined Release Goal
- Product has formal release date
- Product is deployed on staging area
- Stress testing done and results accepted
- All non-functional requirements are tested and results accepted
- Required documentation is available
- Release should not have any known bug

# Feedback Techniques of Agile Project Management

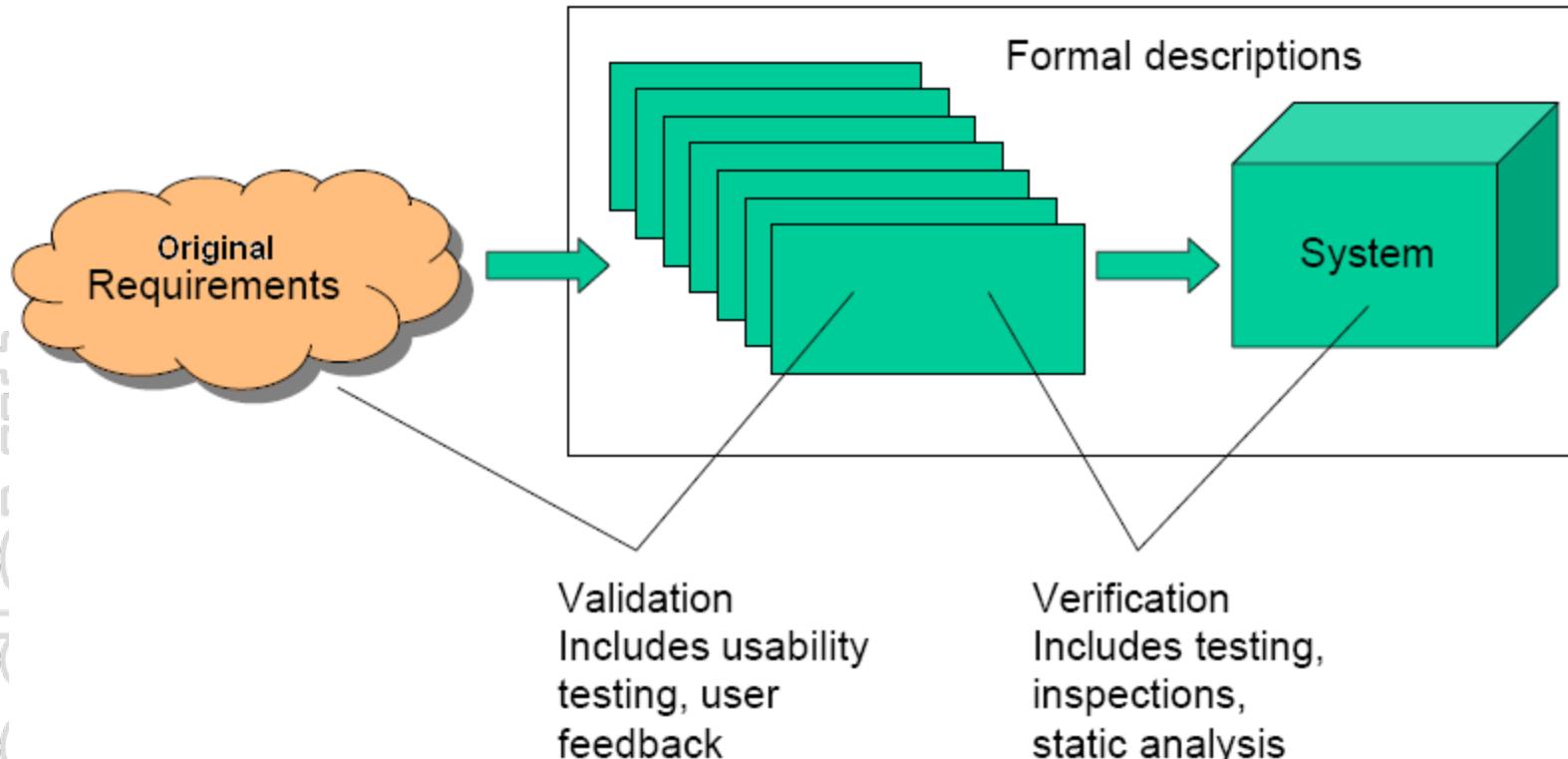
Tools & Techniques / Product Quality

**Timely & regular feedback is heart beat for producing a good quality product and managing risks. Following techniques are used to get quick feedback.**

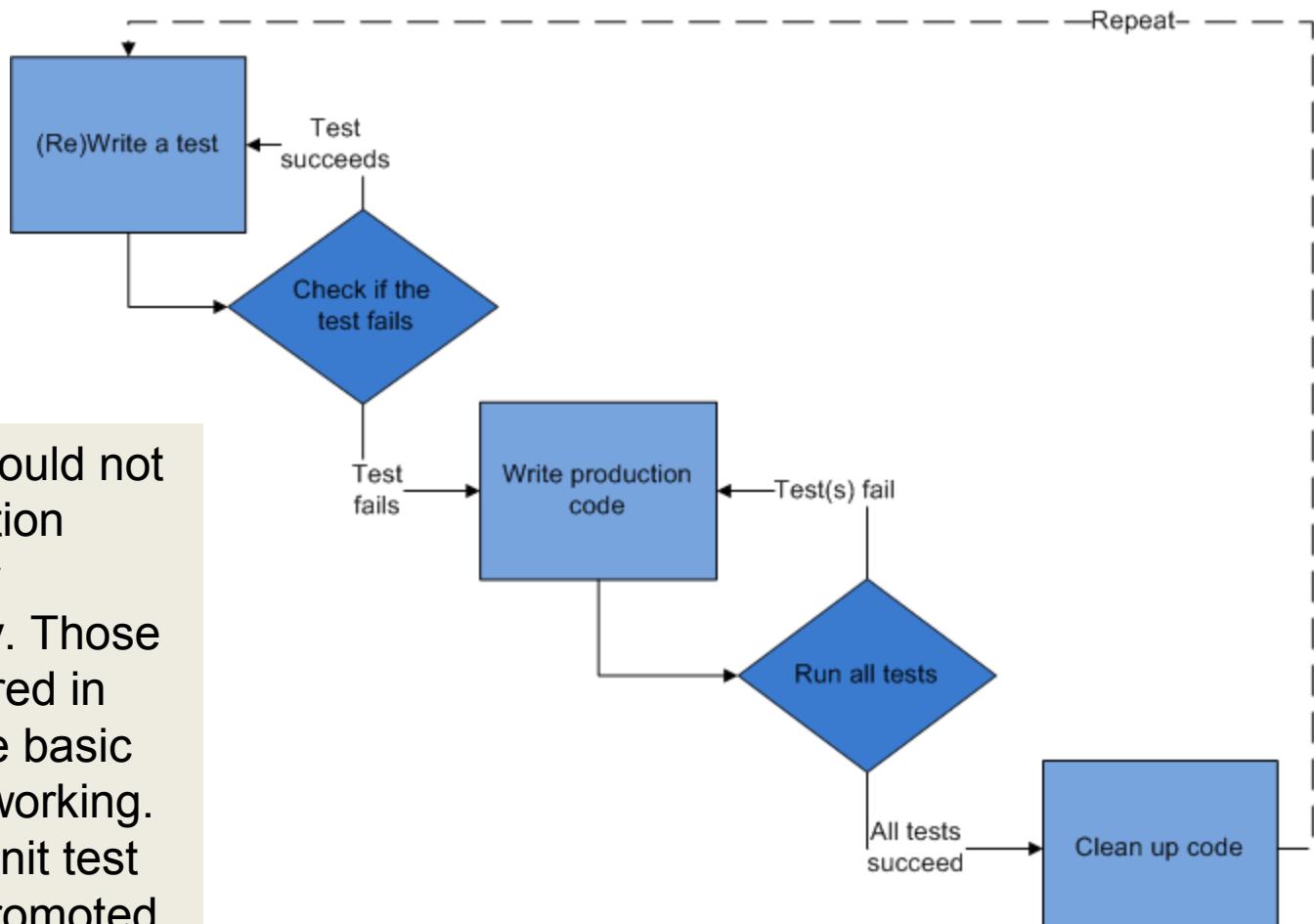
- Release planning
- Sprint planning
- Daily stand-up
- Sprint review
- Sprint retrospective
- Pair programming
- Peer review
- Customer involvement throughout PLC

# Verification and Validation

- Validation:
  - How close is **Product** with respect to original requirement
  - Building the right things
- Verification:
  - How close is product built **Process** with respect to defined and agreed
  - Building the things right



# Test Driven Development (TDD)

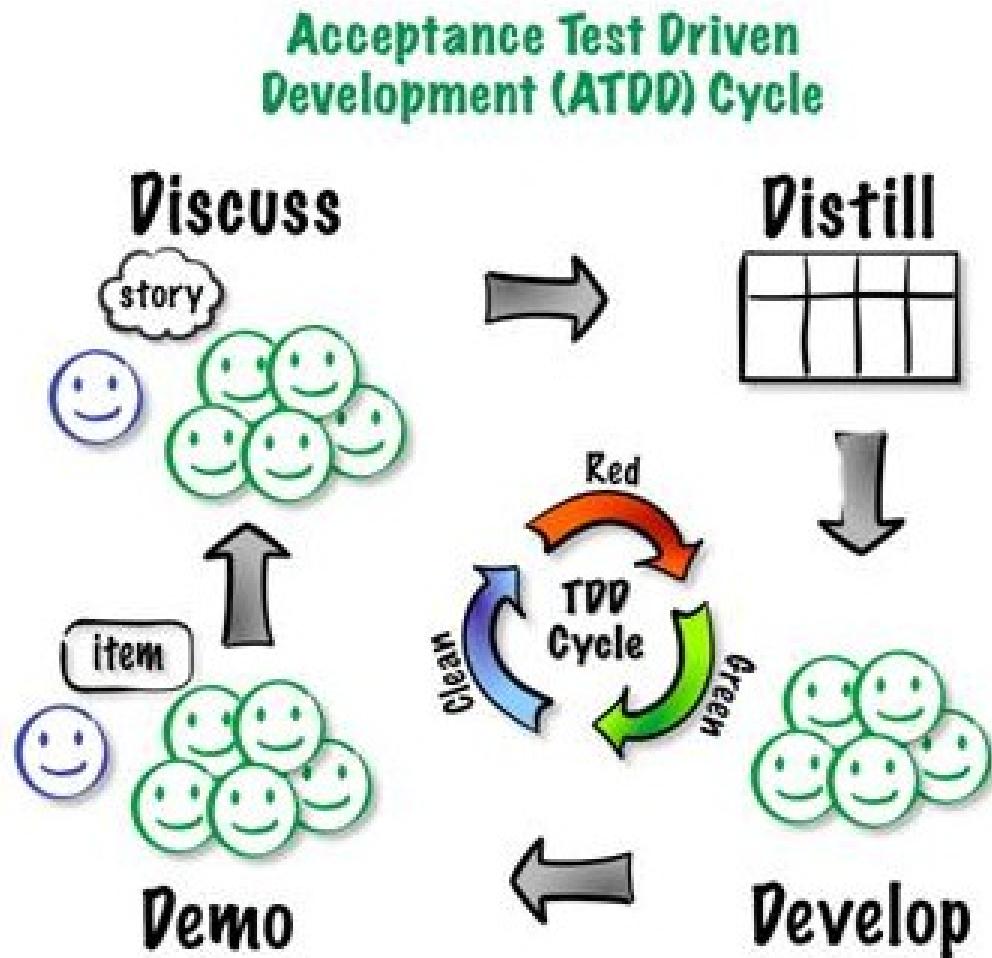


- Programmer should not consider exception cases and error handling initially. Those can be considered in next cycle, once basic functionality is working.
- If built passes unit test then only it is promoted to the next step as a candidate for acceptance test build

# Acceptance Test Driven Development (ATDD)

Tools & Techniques / Product Quality

- Major difference between ATDD and TDD is in ATDD user story or **functionality is tested** not the task like in TDD.
- Using ATDD practice you are not allowed to add more functionality until earlier user stories are successful.
- Collection of acceptance test cases forms regression test and it should run automatically on built server



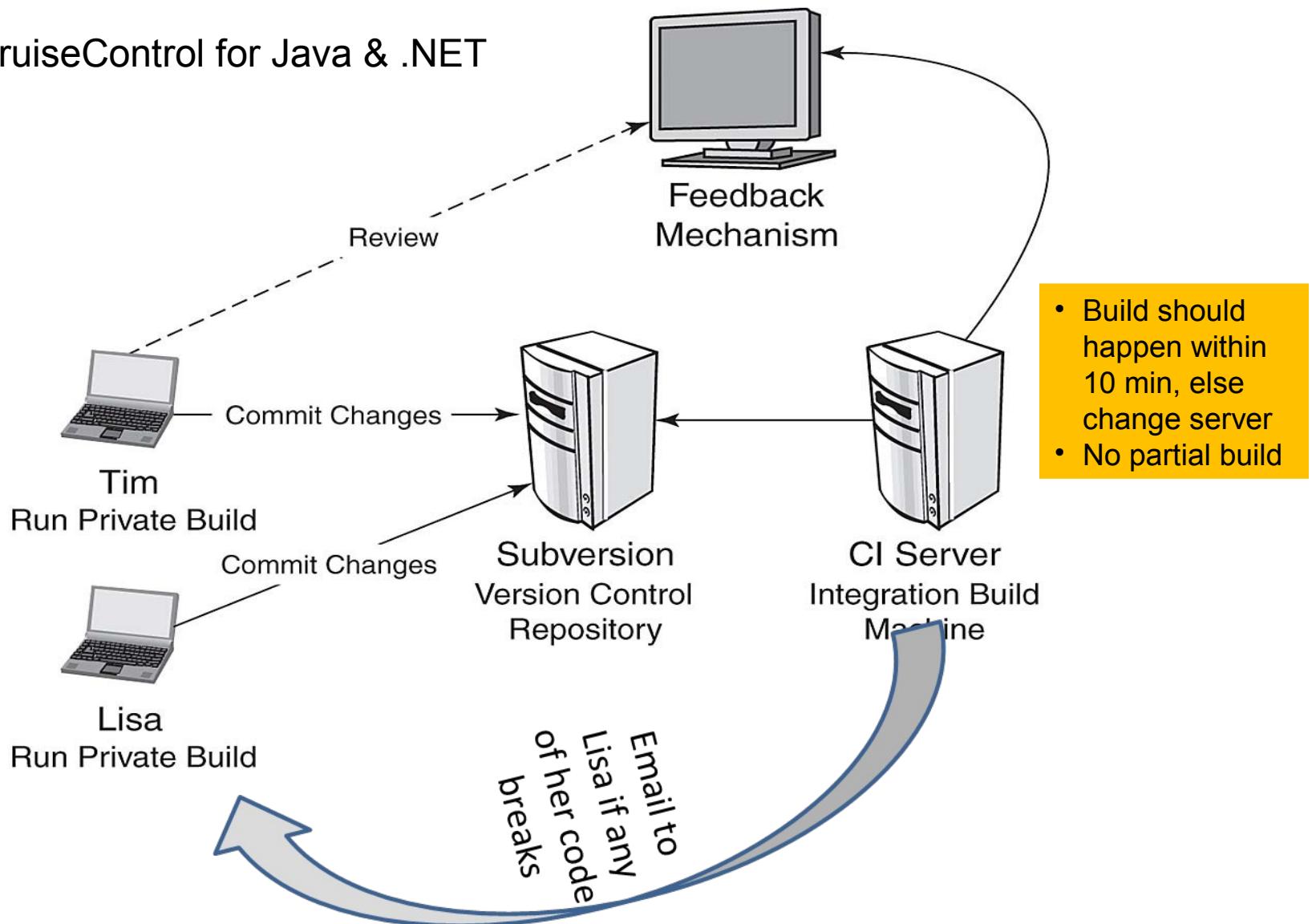
# Exploratory Testing

- After acceptance test cases are passed then Agile tester may choose to perform exploratory testing
- In exploratory testing agile tester tests those scenario which system user is not supposed to perform but if he perform in that case system should give proper message.

# Continuous Integration

Tools & Techniques / Product Quality

Tool: CruiseControl for Java & .NET



# Continuous Process Improvement

Tools & Techniques / Product Quality

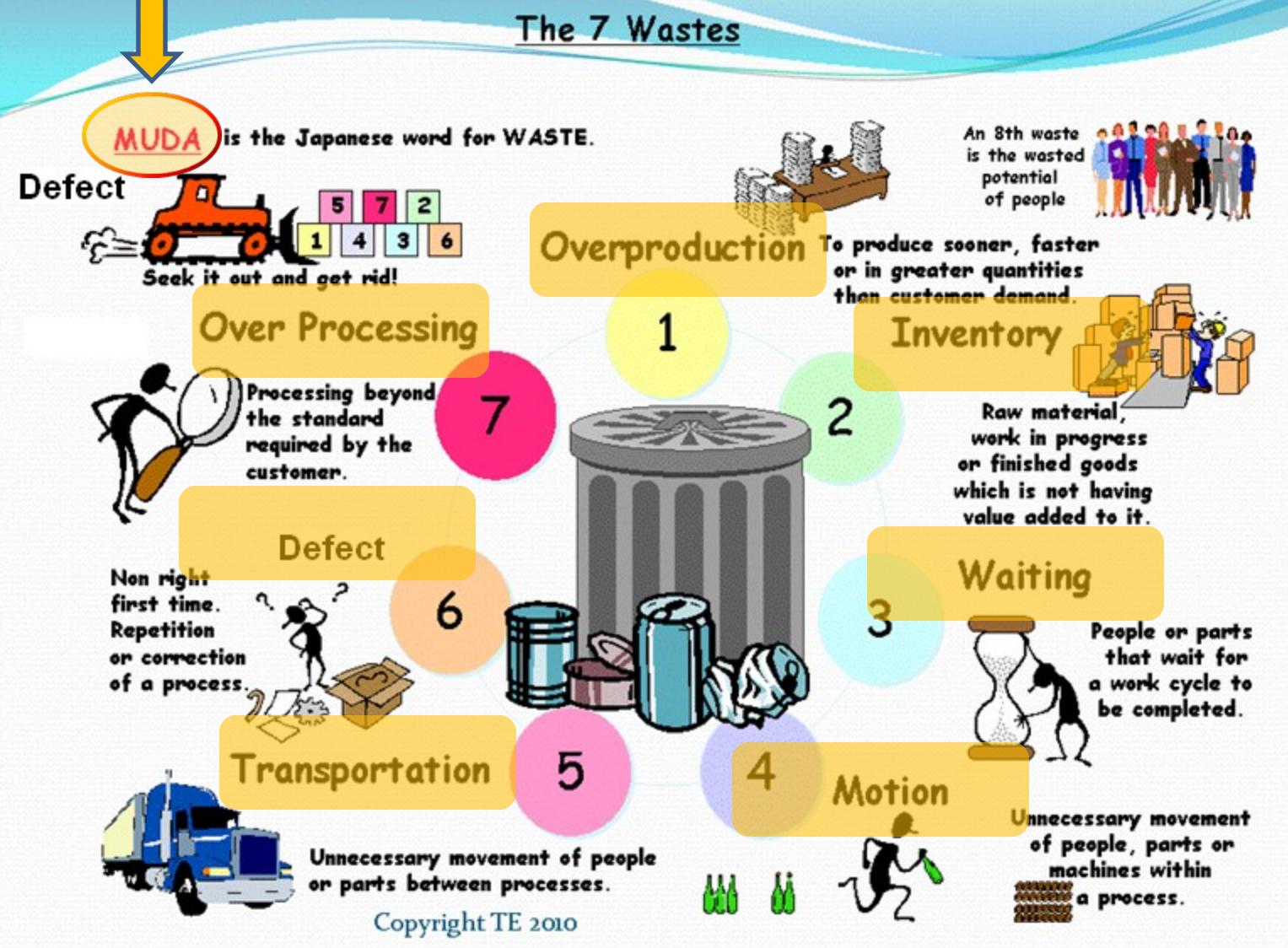
## Models for Continuous Process Improvement

- PDCA Cycle (by Juran - “Fitness for Use”)
- Score Card
  - Escaped defects, Overtime, Velocity, Cost, Time
- Balance Score Card (BSC)
  - Four aspects of BSC are Financial, Customer, Internal Business Proceeds, Learning and Growth
- Six Sigma (DMAIC)
  - Motorola in 1985
  - Used widely by Jack Welch @ GE in 1995
- Kaizen
- Eliminating the waste
- Kanban

# 7 Wastes

Tools & Techniques / Product Quality

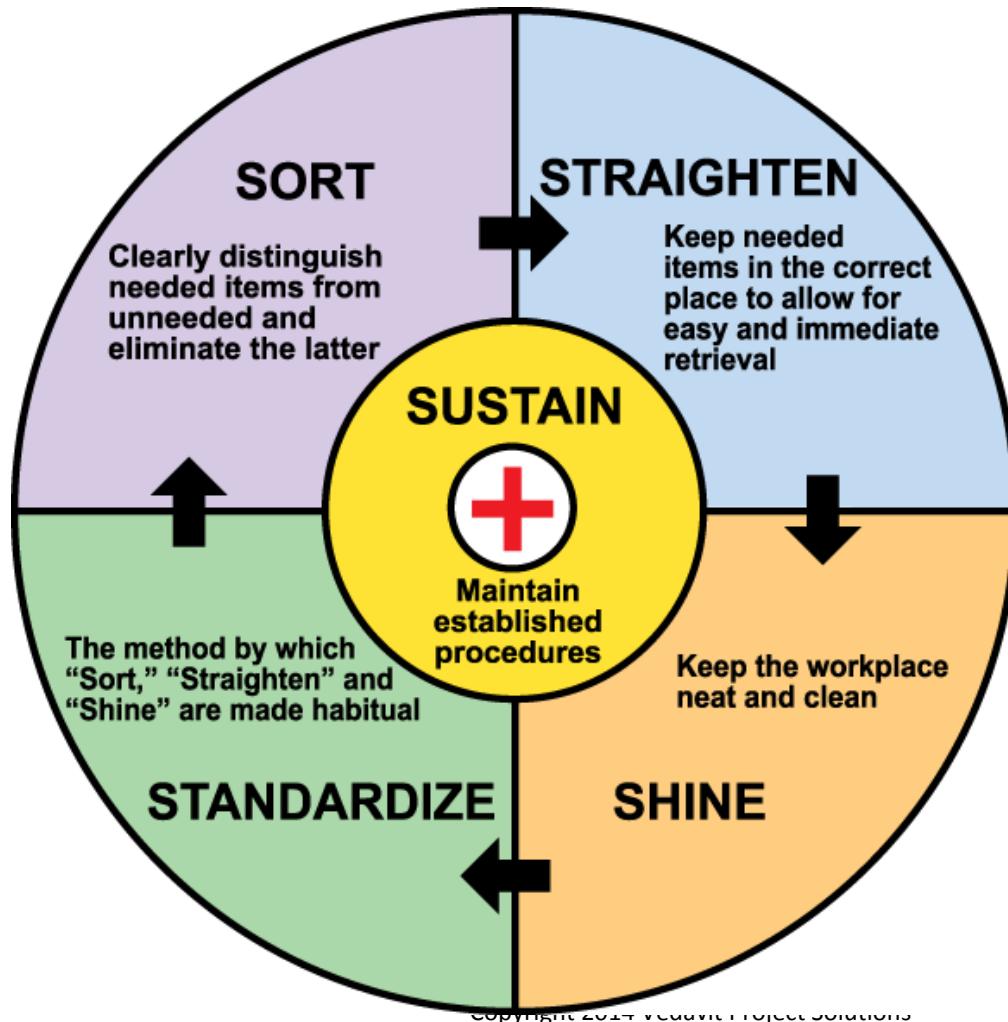
TIM WOOD



# House Keeping Standard (5S)

Tools & Techniques / Product Quality

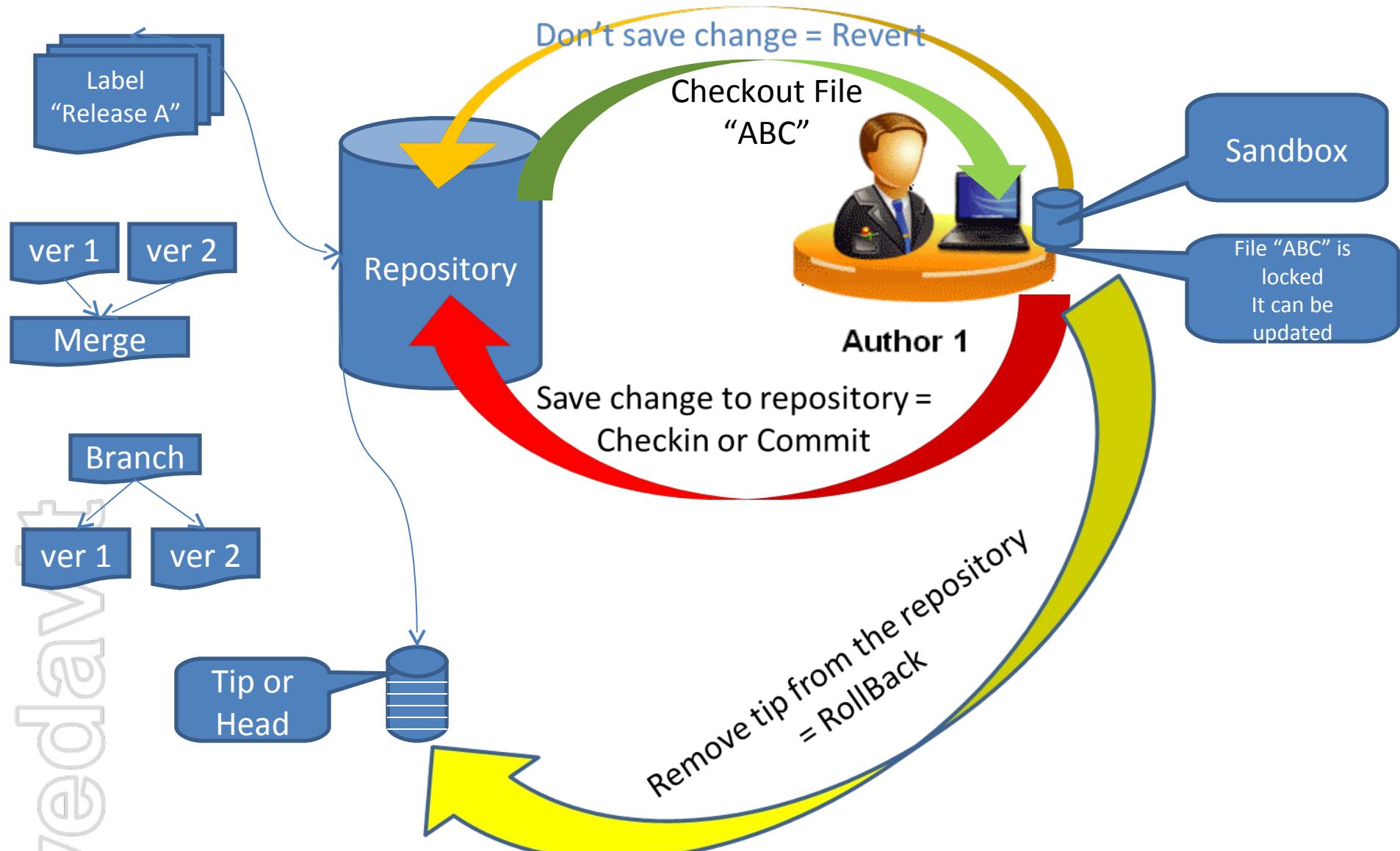
- In Lean 5S is foundation tool for continuous improvement.
- 5S is also called house keeping standard because it lists down the steps to organize a work space for efficiency and effectiveness by identifying and storing the items used, maintaining the area and items, and sustaining the new order.



# Version Control

- A **version control system** is a repository of files and source code, with monitored access. Every change made to the source is tracked, along with who made the change, why they made it, and references to problems fixed, or enhancements introduced, by the change. This is very important tool to ensure quality product.
- **Collective code ownership** means whole team takes responsibility of the quality of the code therefore anybody can modify the code, if needed. (Encouraged in XP Agile Method)
- **Strong code ownership** means each person owns his module. Only he can make changes to his module.
- **Weak Code Ownership** means one person write the module but another person can make changes as long as owner helps them or they coordinate with owner
- When two users can update the same code at same time is called **Concurrent editing**. Conflict is resolved using merging method.

# Version Control



# Agile Tooling Mistake

Tools & Techniques / Product Quality

For producing a good quality product agile project management relies heavily on automation. Therefore not investing in tools or using legacy tools for agile project management affects the moral of the team and quality of the product

Generally following mistakes are made by organization in agile tooling

- Using old version of tool which were working for waterfall
- Not investing on automation tool still continuing manual testing, documentation etc
- Working on age old slow hardware, while team is wasting time
- Not investing time on continuous-integration tool
- Using traditional waterfall metrics to manage the project
- Performing continuous integration only on mainline not on branch so branch code becomes buggy and unreliable.

# Typical Software Tools in Agile Projects

Tools & Techniques / Product Quality

- Agile project management systems (APMS) for effective communication and providing bird's eye view of the project
  - Rally, VersionOne, Jira
- Testing Tools to achieve 100% code coverage by auto unit testing.
  - xUnit
- Regression Testing Tool
  - CruiseControl, TFS
- Version Control to mange versions of code, product, test-case, tested product
  - SVN, SourceSafe, TFS

# Recap

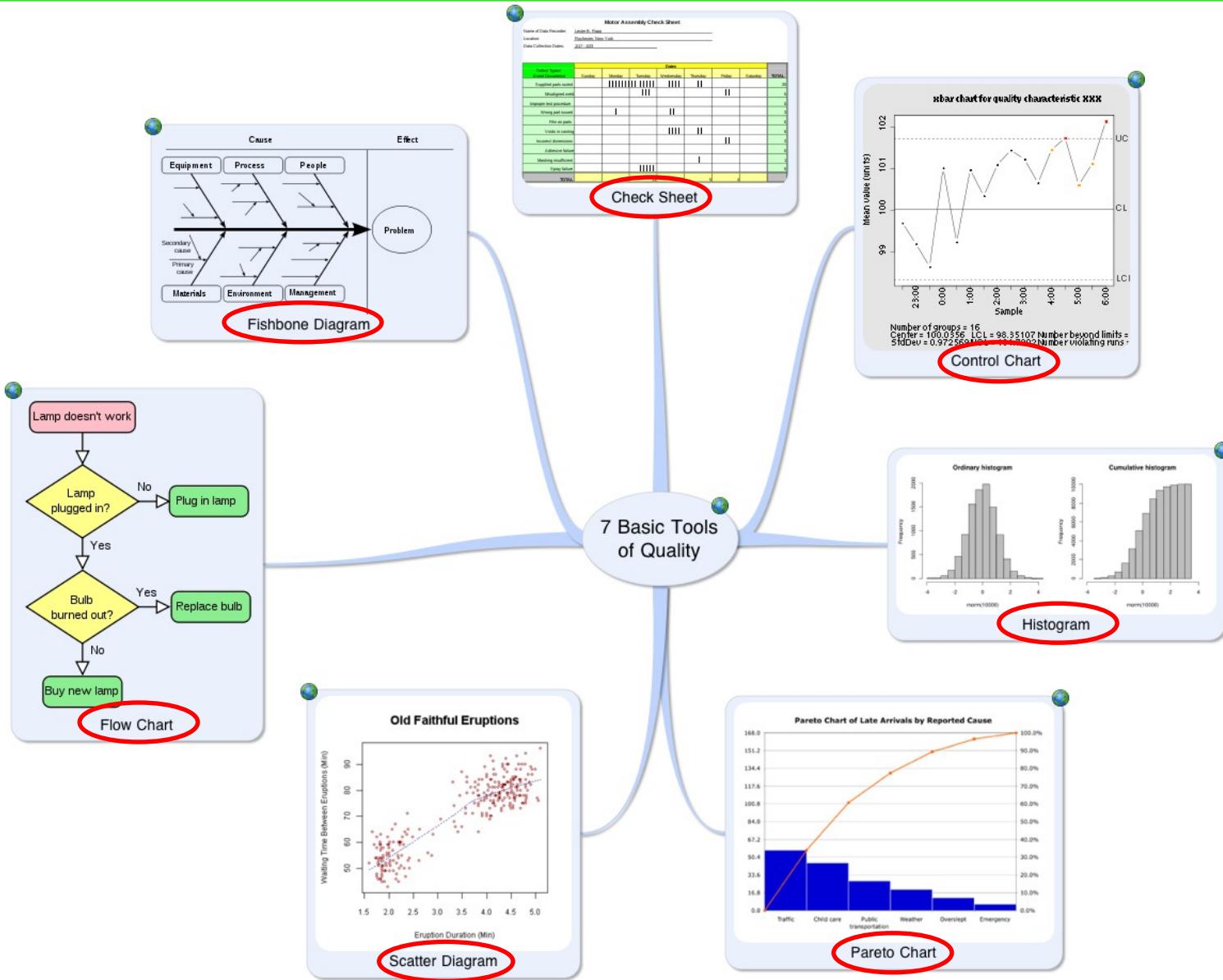
1. Cost of Quality
2. Definition of Done
3. Feedback Techniques of Agile Project Management
4. Verification and Validation
5. Test Driven Development (TDD)
6. Acceptance Test Driven Development (ATDD)
7. Continuous Integration
8. Continuous Process Improvement
9. 7 Wastes
10. House Keeping Standard (5S)
11. Version Control

# Quality Tools

These tools are useful in quality planning, quality assurance, quality control and continuous improvement

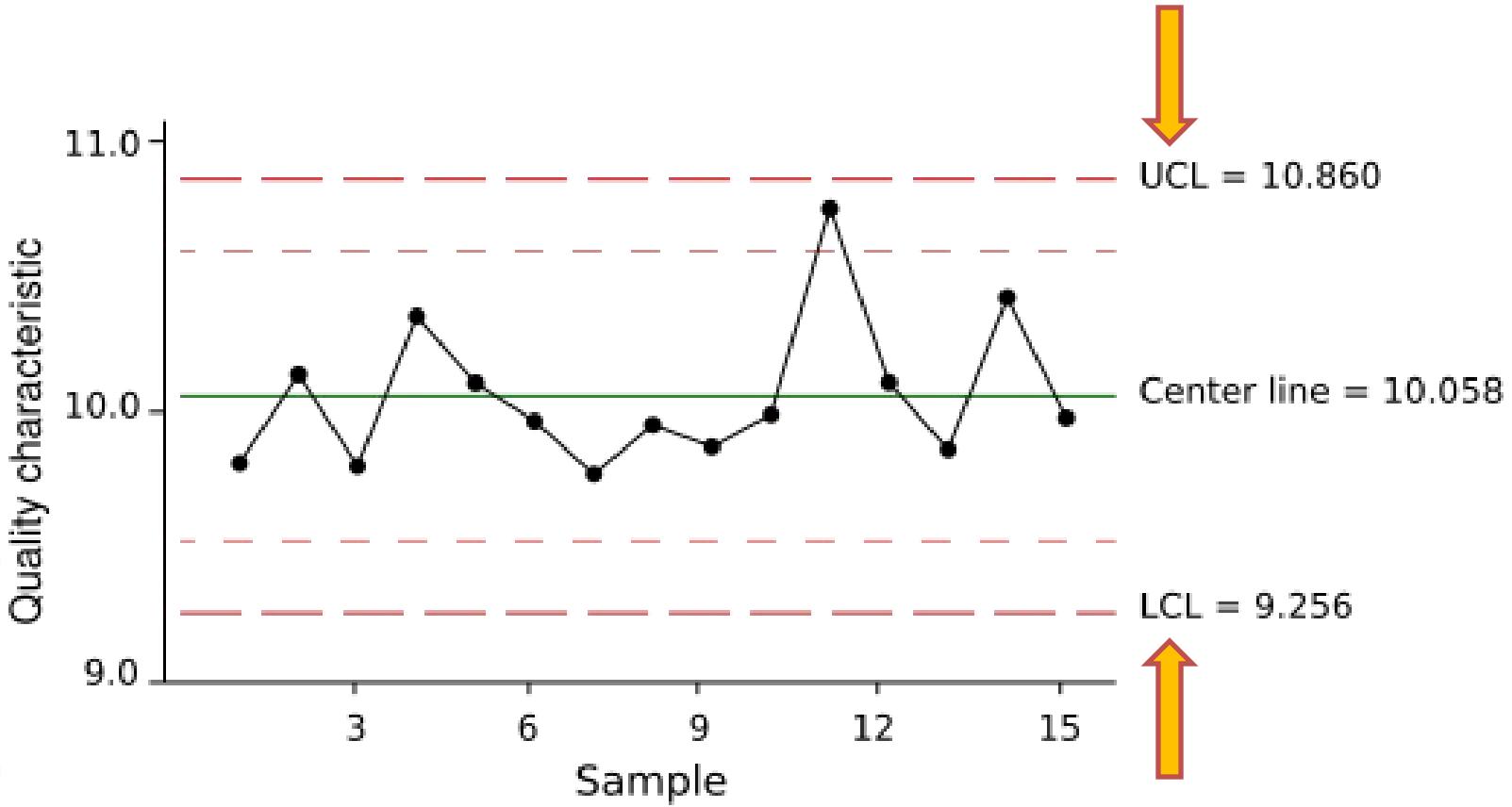
- 1.Benchmarking
- 2.Control Charts & Run of 7
- 3.Run Charts
- 4.Affinity Diagram
- 5.Force Field Analysis
- 6.Quality Checklists
- 7.Cause and effect diagram
- 8.Pareto Charts
- 9.Histograms
- 10.Influence Diagram or Decision Network
- 11.SWOT Analysis
- 12.Design of Experiments (DOE)

# 7 Basic Tools of Quality



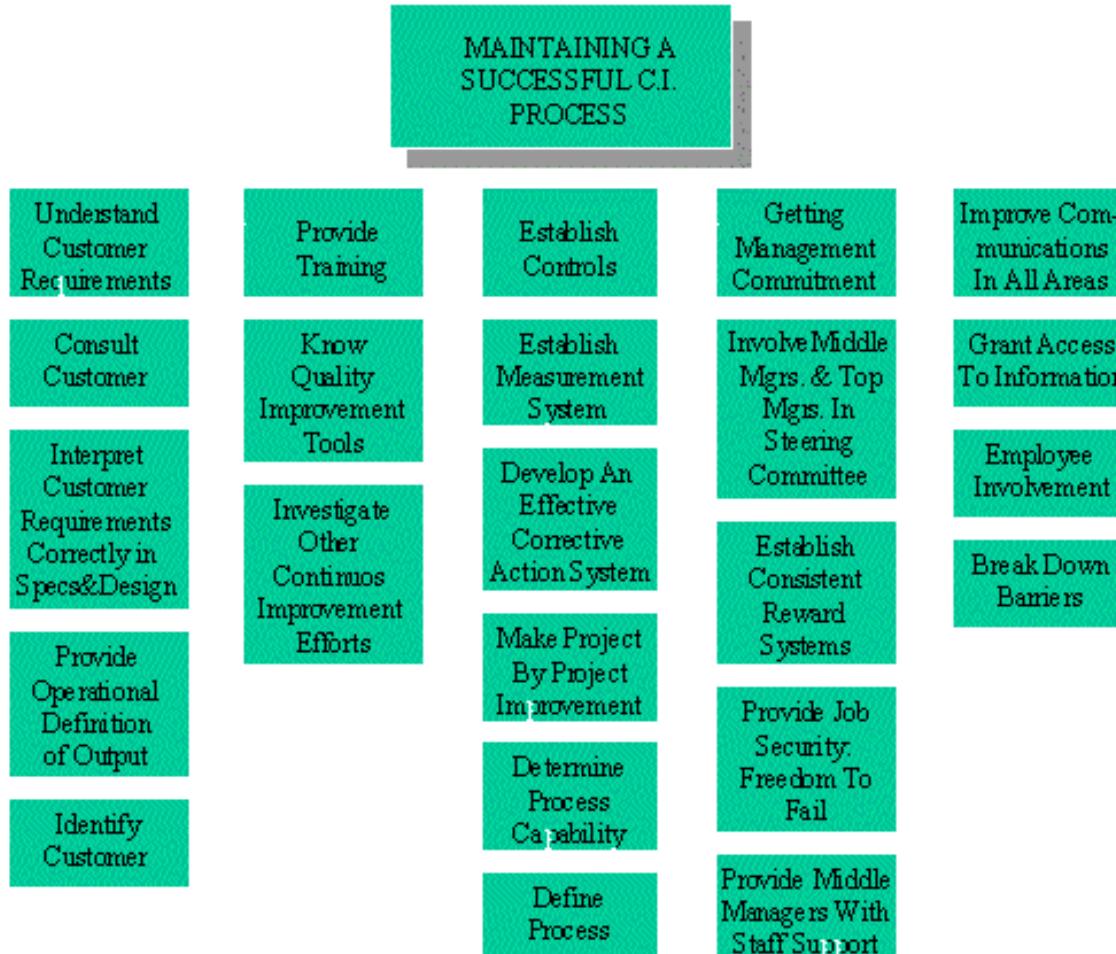
# Quality Tools

## Control Chart & Run Chart



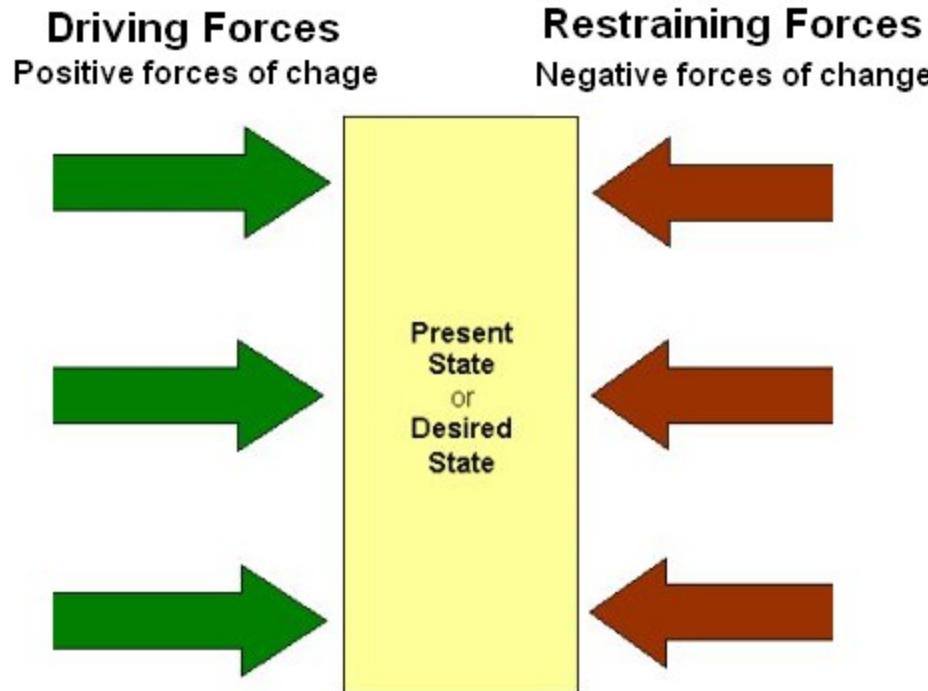
# Quality Tools

## Affinity Diagram



# Quality Tools

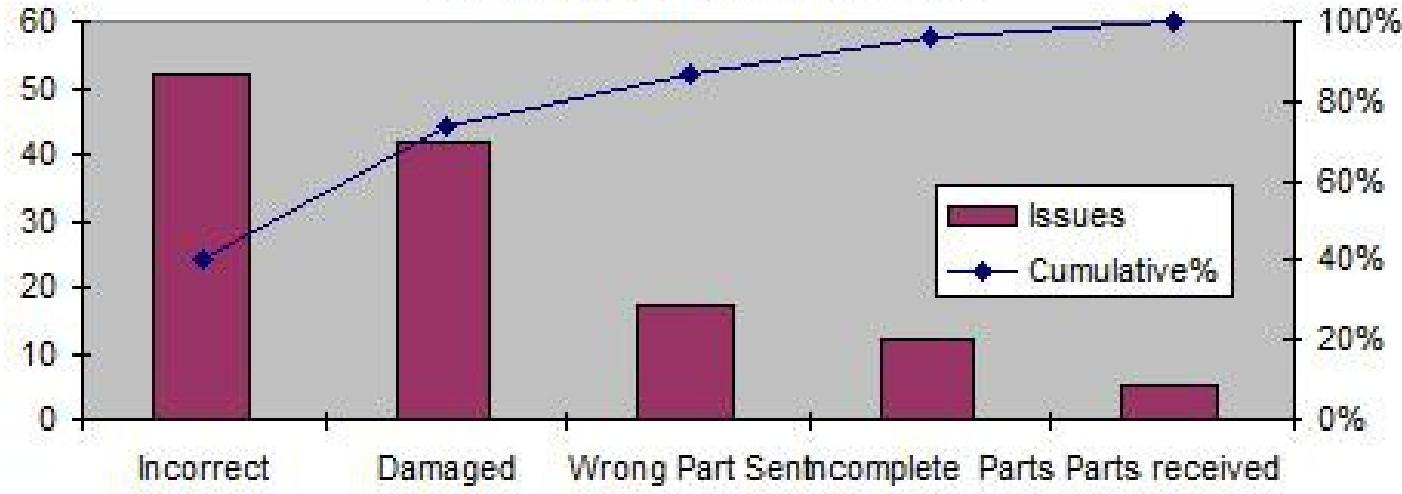
## Force Field Analysis



# Quality Tools

## Pareto Chart

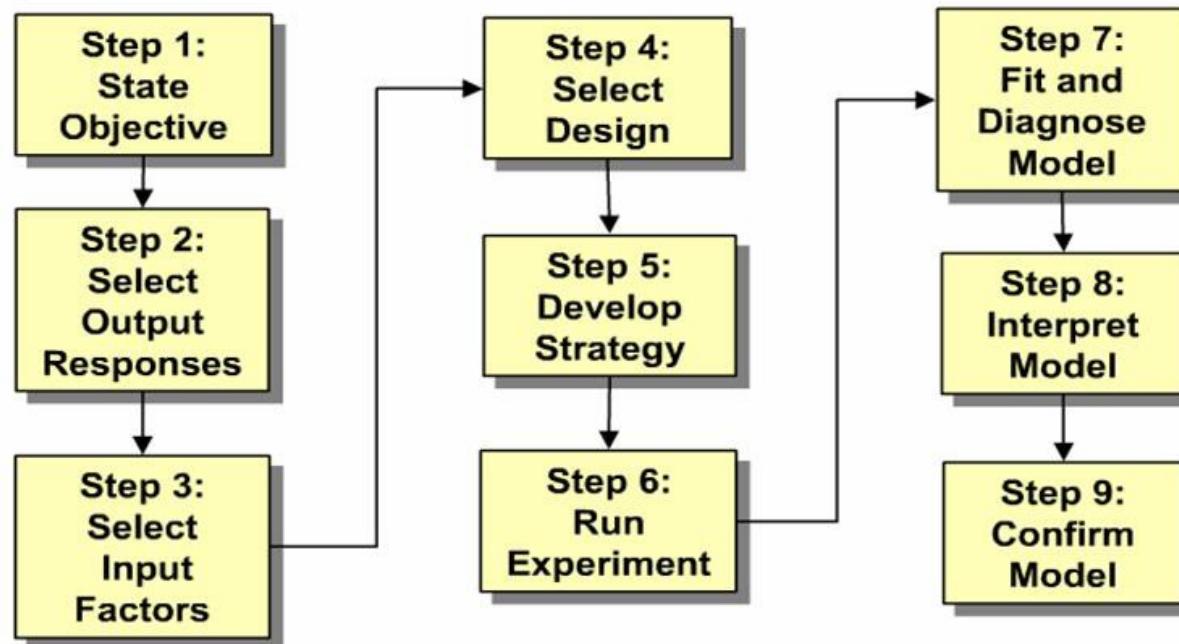
Pareto Analysis Chart showing investigations into Non Supply to customers



	Incorrect Documents	Damaged Packaging	Wrong Part Sent	Incomplete Parts	Parts received early
Issues	52	42	17	12	5
Cumulative %	41%	73%	87%	96%	100%

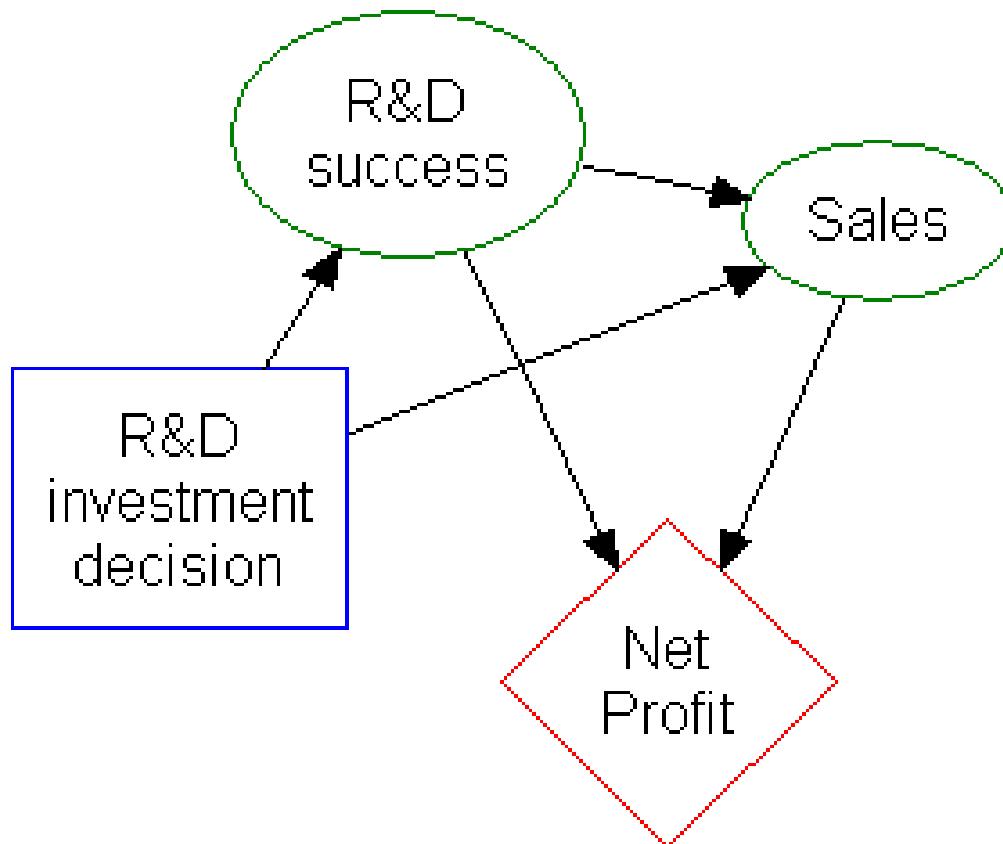
# Quality Tools

## Design of Experiments



# Quality Tools

## Influence Diagram



# Quality Tools

## Benchmarking

# Quality Checklist

# Quality Tools

## SWOT Analysis

# Recap : Quality Tools

- Benchmarking
- Control Charts
- Affinity Diagram
- Force Field Analysis
- Quality Checklists
- Cause and effect diagram
- Pareto Charts
- Histograms
- Run Charts
- Influence Diagram or Decision Network
- SWOT Analysis
- Design of Experiments (DOE)

# **Discussions !**

# Topics

## Day 1

- ✓ History of Agile
- ✓ Agile Certifications
- ✓ The Most General APM Certificate
- ✓ The Foundation of Agile Project Management
- ✓ Project Management in General
- ✓ Agile Project Management
- ✓ Agile : Developing Mindset
- ✓ Agile : Methodologies
- ✓ Agile : Metrics & Estimation

vedavit

## Day 2

- ✓ Agile : Planning, Monitoring & Adapting
- ✓ Agile : Communication
- ✓ Agile : Analysis and Design
- ✓ Agile : Product Quality
- Agile : Value Prioritization



## Day 3

- Agile : Risk Management
- Soft Skills for Agile Project Manager
- Agile : Compliance
- Agile : Roles
- Scaling Agile
- Executing an Agile Iteration
- Professional Responsibility

# Agile : Value Prioritization and Value Streaming

# Quotes Relevant to the Topic

- Nowadays people know price of everything and value of nothing. Unknown
- The 'telephone' has too many shortcomings to be seriously considered as means of communication. The device is inherently of no value to us. 1876, Internal Memo of Western Union

# Topics

Tools & Techniques / Value Based Prioritization & Value Streaming

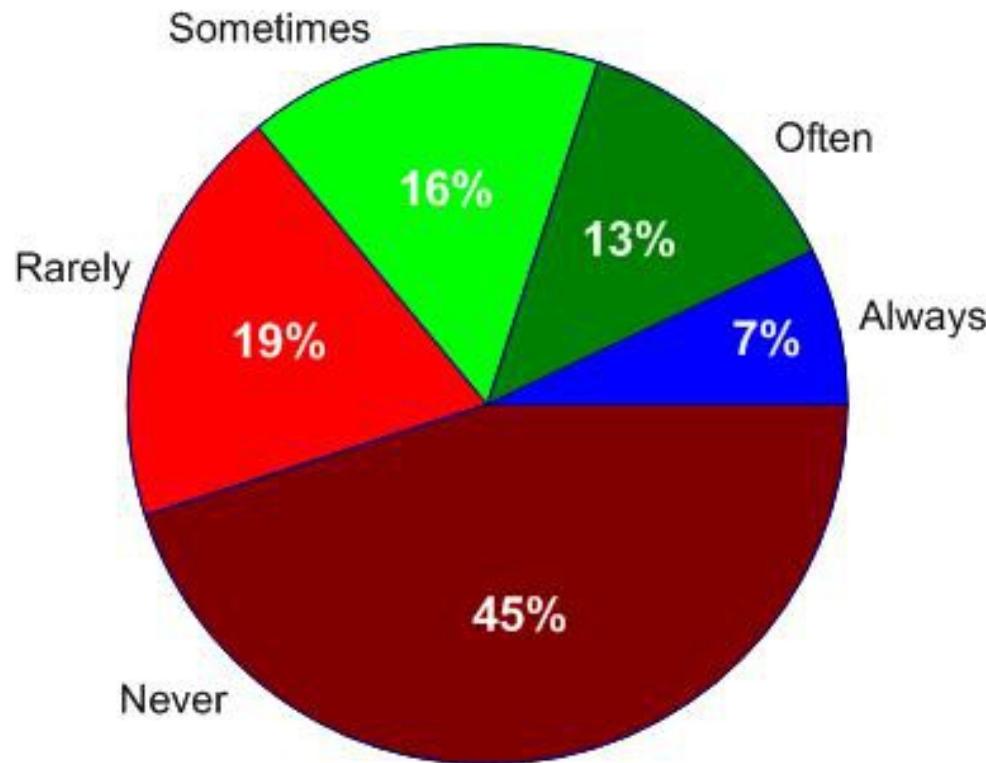
1. What is value?
2. JIT in value delivery
3. Sources of revenue
4. Value Evaluation Techniques
5. Value-based Analysis
6. Minimum Marketable Features (MMF)
7. Non Function Requirements
8. Relative Prioritization and Ranking Techniques
9. Value-based Prioritization
10. Value Stream Mapping
11. System Thinking

# What is Value?

- Worthiness of something in these eyes of customer.
- Quality, Timely Delivery, Comfort Realized, Pain Reduced are various dimension of Value

# Value of Delivered and non-used Features

Average percentage of delivered functionality actually used when a serial approach to requirements elicitation and documentation is taken on a “successful” information technology project.



Source: Chaos Report v3, Standish Group.

# JIT in Value Delivery

Your customer asks you to develop an application with 50 features, out of these 10 are business critical. Application will be ready in 12 months. You are in 3<sup>rd</sup> week in the project and ask these questions to yourself

- ✓ What is the value of detail plan of this whole application?
- ✓ What is the value of hiring those resources/expertise which you need after 6 month (that too not sure)?
- ✓ What is the value of detail design, SRS, Functional specification (especially when you know they are going to change)?
- ✓ What is the value of writing test cases for complete application?
- ✓ What is the value of hardware, software, infrastructure which you do not need today?

JIT (Just In Time) principle helps controlling our greed and maximizing value.

Companies like Toyota, Dell start using this concept early in the industry.

# Sources of Revenue

- Incremental Revenue
  - Comes from existing customer
- New Revenue
  - Comes from new customer
- Retained Revenue
  - Comes by enhancing the existing feature. But if you do not enhance features then customer will buy other product.
- Operational Efficiency Revenue
  - Comes by improving operational process

# Value Evaluation Techniques

## These techniques are also called a Project Appraisal Methods

- Present Value (PV)
  - Present value of money which business will realize in future
  - $PV = FV/(1 + r)^n$ ; n is number of years, r is discounted rate, FV is future value.
- Net Present Value Method (NPV)
  - In the NPV method, the revenues and costs of a project are estimated and then are discounted and compared with the initial investment.
  - The preferred option is that with the highest positive net present value.
  - Projects with negative NPV values should be rejected because the present value of the stream of benefits is insufficient to recover the cost of the project.

# Value Evaluation Techniques

- Discount rate
  - The discount rate is a concept related to the NPV method. The discount rate is used to convert costs and benefits to present values to reflect the principle of time preference.
- *Benefit cost Ratio (BCR) or Return on Investment (ROI)*
  - $BCR = Benefit / Cost$  ratio
  - The BCR is the discounted net revenues divided by the initial investment. The preferred option is that with the ratio greatest in excess of 1

# Value Evaluation Techniques

- Internal Rate of Return (IRR)
  - The IRR is the discount rate which, when applied to net revenues of a project sets them equal to the initial investment.
  - The preferred option is that with the IRR greatest in excess of a specified rate of return.
- Payback Period (PB)
  - Amount of time in which original investment will be realized through business returns
  - The preferred option is that which the lowest PB

# Value Evaluation Techniques

		Year 1	Year 2	Year 3	Year 4	Year 5					
	Outflow	Inflow	Inflow	Inflow	Inflow	Inflow	IRR	PBP (Years)	PV		NPV
Project 1	995	500	300	200	200	150	14.00%	3.0	1000.8		5.7524
Project 2	800	700	100	10	50	100	11.00%	2.0	803.3		3.3434
Project 3	690	400	200	200	100	200	17.00%	2.5	699.9		9.9276

Highest IRR

Lowest Period

Highest PV

Maximum Positive Value

# Value Based Analysis (VBA)

- **Value Analysis (VA)**
  - Identify the product and its end users
  - Identify the basic functions for which customer is planned to be billed (primary function)
  - How the above function to be achieved (secondary functions)
  - Relatively prioritize primary and secondary functions
  - Build high value functions first
- **Value Stream Analysis (VSA)**
  - Focus of this techniques is to eliminate the waste and achieve same functionality at less cost with better quality and performance
- **Value Proposition (VP)**
  - VP is generally a clear statement of unique value creating features, utility and tangible results perceived by the customer. VP is the reason due to which customer considers vendor A compare to vendor B
- **McKinsey Shared Value Model**
  - It helps to achieve shared values through 7S
  - Strategy, Structure , Style, Skills, Staff, System, Shared Values

# Minimum Marketable Features (MMF)

Tools & Techniques / Value Based Prioritization & Value Streaming

Product is saleable or usable or viable only when a set of minimum number of features is built there in the product. Generally all these features are part of a single release. This is called MMF.

Epic: Personal Information	Epic: Login/Logoff	Epic: User Management	Epic: Multi-Order Shipping	Epic: Basic Shipping	Epic: Priority Shipping	Epic: Shipping	Epic: Shopping	Epic: Payment Methods	Epic: Order Management
Validate Customer Contact/Shipping	Log-in to Secured Website	View or Change your One-Click	Calculate Split Shipping	Data fields for Shipping information	Overnight Shipping	Epic: Multi-Order Shipping	Shop for Items	Payment - Supported Credit Cards	Combine Orders
Customizing Product List	Log-off Website	View By Order	Update Shipping System for Online	Ship Single-Site Order	2-3 Day Shipping	Epic: Basic Shipping	Purchase Your Items	Payment - Promotional Codes	Order Modification
Change Billing Address		Epic: Personal Information	Ship Multi-Site Orders			Epic: Priority Shipping	Persistent Shopping Cart	Payment - Gift Certificates	
		Epic: Login/Logoff					Recent Purchases View		
		Find My Orders							

MMF

# 13 Non-Functional Requirements (NFR) with example

Tools & Techniques / Value Based Prioritization & Value Streaming/ Non Functional Requirements

1. **Performance:** *100 transactions per minute*
2. **Interface:** *capable of importing data with EDI format*
3. **Operational:** *must not require more than 1 megabyte of main memory*
4. **Resource:** *will use wireless encryption algorithm that is “better” than WEP*
5. **Verification:** *all data updates must be traceable*
6. **Acceptance:** *must pass a user defined system test bucket*
7. **Documentation:** *user manual is needed for novice users only*
8. **Security:** *user request to access any data must be authorized first*
9. **Portability:** *the system must operate with “any” relational db systems*
10. **Quality:** *the system must install with zero defect*
11. **Reliability:** *the system must be accessible 99.9 % of the time*
12. **Maintainability:** *the system must be modifiable (e.g. designed with exits)*
13. **Safety:** *the system must not perform “chemical material discard” functions without “explicit” user authorization.*

# Non-functional requirements and Prioritization

Tools & Techniques / Value Based Prioritization & Value Streaming/ Non Functional Requirements

- Non functional requirements of a product and Requirement of infrastructure required to manage the project are considered as Technical user story.
- Infrastructure related user stories are completed earlier
- Other non-functional requirements are implemented later in release

# Relative Prioritization & Ranking Methods

1. Prioritization
2. Customer-valued Prioritization
3. Forced Ranking Technique
4. Analytical Hierarchical Process
5. Paired Comparison Techniques
6. Weight-based Prioritization Techniques
7. Kano Analysis
8. 100 Point Method

# Relative Prioritization & Ranking Methods

## Prioritization

Agile project management is based on principle of **incremental delivery** and **deliver fast**. Thus we must know the prioritization methods so that we can deliver high business values items earlier.

Possible Factors of prioritization are as below. Each factor may have different weightage (business value has highest weightage) when you prioritize your product backlog items.

- Business Value
- Cost of Implementation
- Prioritization Scenario
- Implementation Time
- Frequency of usage
- Safety
- Implementation Difficulty
- Stability
- Re-usability

# Relative Prioritization & Ranking Methods

## 1 Customer Value Prioritization

- Customer based on input from internal stakeholders assigns some priority to each requirement

## 2 Forced Ranking Techniques

- Force customer to assign **unique priority number** to each requirement (highest number highest priority)

Requirements	Stakeholder1	Stakeholder2	Stakheolder3	Total	Ranking
Req#1	1	5	4	10	2
Req#2	2	4	5	11	1
Req#3	3	3	3	9	3
Req#4	4	1	2	7	5
Req#5	5	2	1	8	4



# Relative Prioritization & Ranking Methods

Tools & Techniques / Value Based Prioritization & Value Streaming /Relative Prioritization & Ranking Methods

## 3 Analytical Hierarchical Process (AHP)

- Relative prioritization comparing feature pair wise

## 4 Paired Comparison Techniques

- Like AHP but priority can only 0 or 1

Requirements	Req#1	Req#2	Req#3	Req#4	Req#5	Total	Ranking
Req#1	0	1	1	1	1	4	1
Req#2	0	0	1	1	1	3	2
Req#3	0	0	0	1	1	2	3
Req#4	0	0	0	0	1	1	4
Req#5	0	0	0	0	0	0	5



# Relative Prioritization & Ranking Methods

Tools & Techniques / Value Based Prioritization & Value Streaming /Relative Prioritization & Ranking Methods

## 5 Weight-based Prioritization Techniques

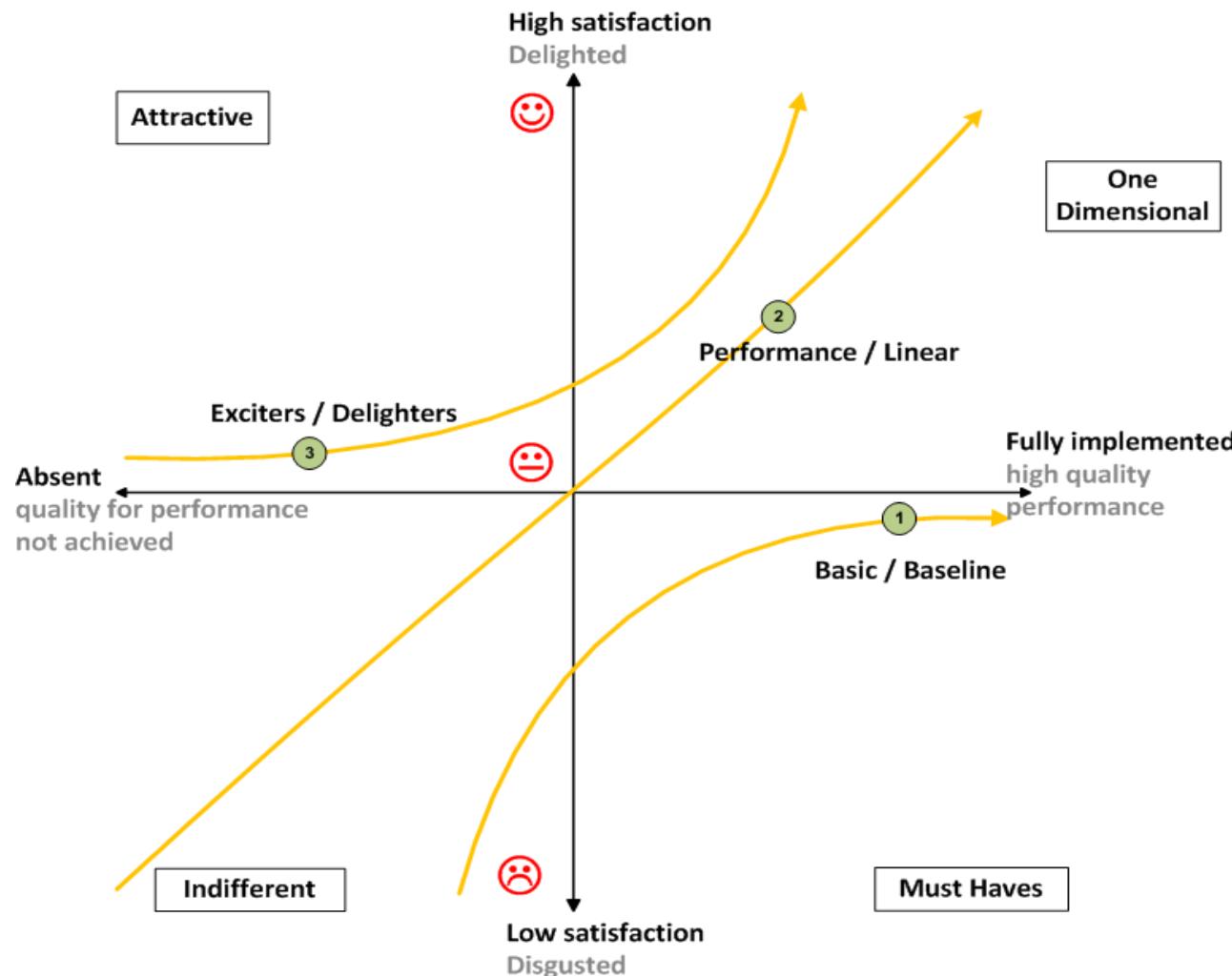
Requirements	Raw Scoring	Scaled Scoring	Ranking
Req#1	4	.22	2
Req#2	4	.22	2
Req#3	5	.27	1
Req#4	2	.11	4
Req#5	3	.16	3
	18		



# Relative Prioritization & Ranking Methods

Tools & Techniques / Value Based Prioritization & Value Streaming /Relative Prioritization & Ranking Methods

## 6 Kano Analysis



# Relative Prioritization & Ranking Methods

## 7 100 Point Method

Each stakeholder is given 100 points and they can use these points to buy the requirements they want

Requirements	Stakeholder1	Stakeholder2	Stakheolder3	Total	Ranking
Req#1	0	20	30	50	3
Req#2	30	10	40	80	2
Req#3	50	20	20	90	1
Req#4	20		10	30	4
Req#5	0	50	0	50	3

# Multiple Parameter Based Prioritization

Tools & Techniques / Value Based Prioritization & Value Streaming /Value-based Prioritization

<i>Relative Weight</i>	2	1			1		.5		
<i>Feature</i>	<i>Relative Benefit</i>	<i>Relative Penalty</i>	<i>Total Value</i>	<i>Value %</i>	<i>Relative Cost</i>	<i>Cost %</i>	<i>Relative Risk</i>	<i>Risk %</i>	<i>Priority</i>
1. Query status of a vendor order	5	3	13	8.4	2	4.8	1	3	1.345
2. Generate a Chemical Stockroom inventory report	9	7	25	16.2	5	11.9	3	9.1	0.987
3. See history of a specific chemical container	5	5	15	9.7	3	7.1	2	6.1	0.957
4. Print a chemical safety datasheet	2	1	5	3.2	1	2.4	1	3	0.833
5. Maintain a list of hazardous chemicals	4	9	17	11	4	9.5	4	12.1	0.708
6. Modify a pending chemical request	4	3	11	7.1	3	7.1	2	6.1	0.702
7. Generate an individual laboratory inventory report	6	2	14	9.1	4	9.5	3	9.1	0.646
8. Search vendor catalogs for a specific chemical	9	8	26	16.9	7	16.7	8	24.2	0.586
9. Check training database for hazardous chemical training record	3	4	10	6.5	4	9.5	2	6.1	0.517
10. Import chemical structures from structure drawing tools	7	4	18	11.7	9	21.4	7	21.2	0.365
<b>Totals</b>	<b>54</b>	<b>46</b>	<b>154</b>	<b>100</b>	<b>42</b>	<b>100</b>	<b>33</b>	<b>100</b>	--

V  
e  
d  
a  
v  
i  
t  
P  
r  
o  
j  
e  
c  
t  
S  
o  
l  
u  
t  
i  
o  
n  
s

Done by Customer

Done by Customer

Done by Developer

value percentage / (cost percentage \* cost weight) + (Risk percentage \* Risk weight)

Potential negative impact of not doing the feature

# Multiple Parameter Based Prioritization

Tools & Techniques / Value Based Prioritization & Value Streaming /Value-based Prioritization

<u>Product Catalog</u>	<u>Hours</u>	<u>Priority</u>
Product Catalog	100	Must
Product Categories	160	Must
Browse Products by Age	120	Should
Search	100	Must
Product Catalog Admin	80	Could
Product Category Admin	200	Won't
Product Sale Pricing	80	Could
<hr/>		
<u>Payment</u>		
Credit Card Payment	160	Must
Paypal	100	Should
CVV2 - Security Code	20	Should
SSL	80	Must
<hr/>		
<u>Account</u>		
Save Credit Card	100	Could
Multiple Address Shipments	350	Won't
Order History	120	Could
Wish List Feature	300	Won't
<hr/>		
<u>Shopping Cart</u>		
Shopping Cart	100	Must
Gift Wrapping	120	Could
Coupons	120	Could
Tax Calculation	120	Must
Cart Progress Bar	40	Could
<hr/>		
<u>Order Maintenance</u>		
Backend System Integration	1000	Won't
Order Data Export	160	Must
Reports and Statistics	1000	Won't
<hr/>		
<u>Shipping</u>		
Shipping Rate Lookup	120	Must
Shipping Confirmation E-mail	80	Should
<hr/>		
<u>Site Framework</u>		
Page Framework	80	Must
About, Home, Terms, Privacy	80	Must
Estimated Total Project Hours	5000	

	<u>Hours</u>	<u>% of Total Hours</u>
Must Have	1200	56%
Should Have	320	14%
Could Have	660	29%
Total Hours/Time box	2240	
<hr/>		
Must have hours/Time Box	57%	

Mo = Must Have

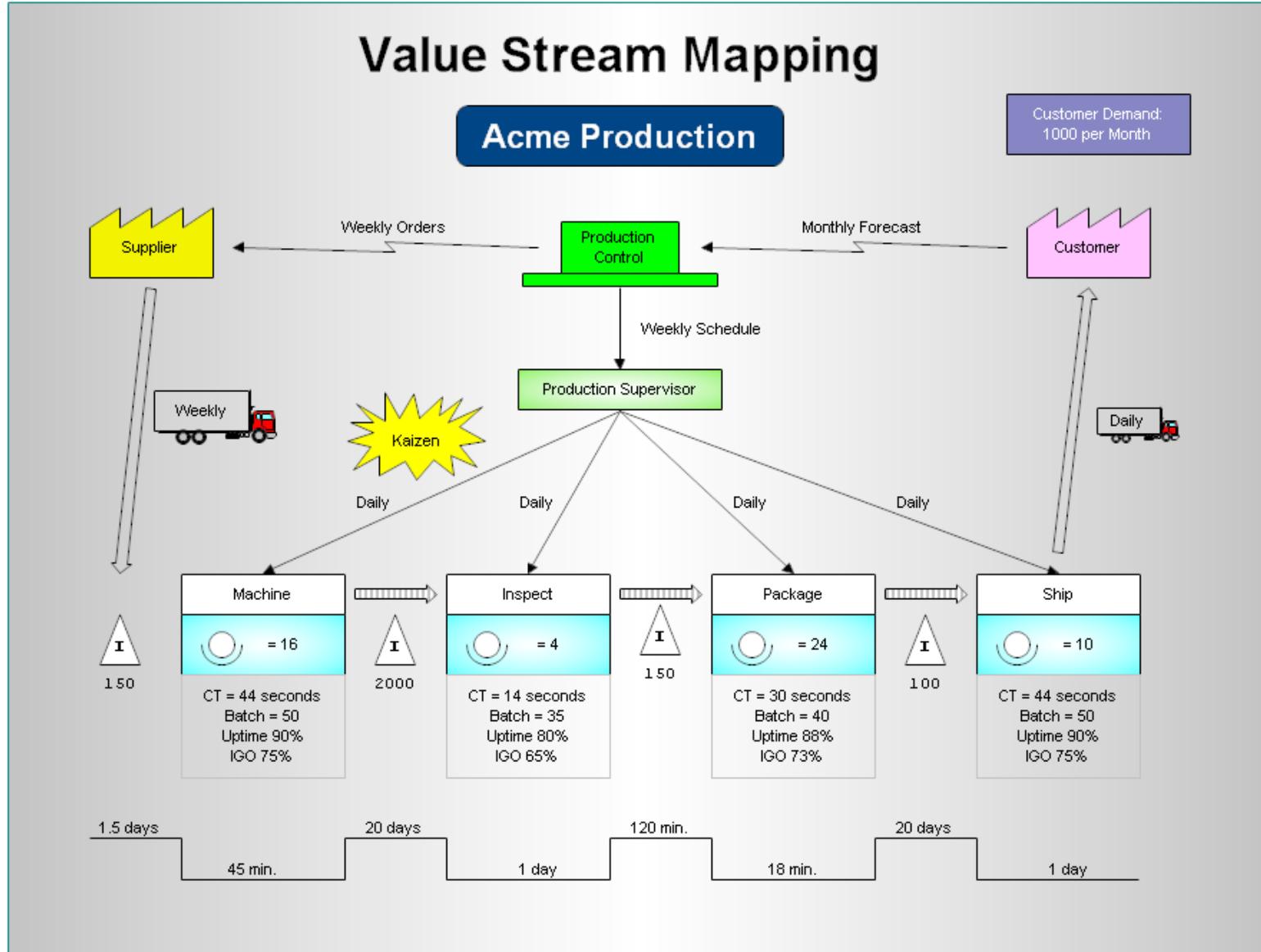
S = Should Have

Co = Could Have

W = Wouldn't have now

# Value Stream Mapping

Tools & Techniques / Value Based Prioritization & Value Streaming/ Value Stream Mapping



# Value Stream Mapping Tools

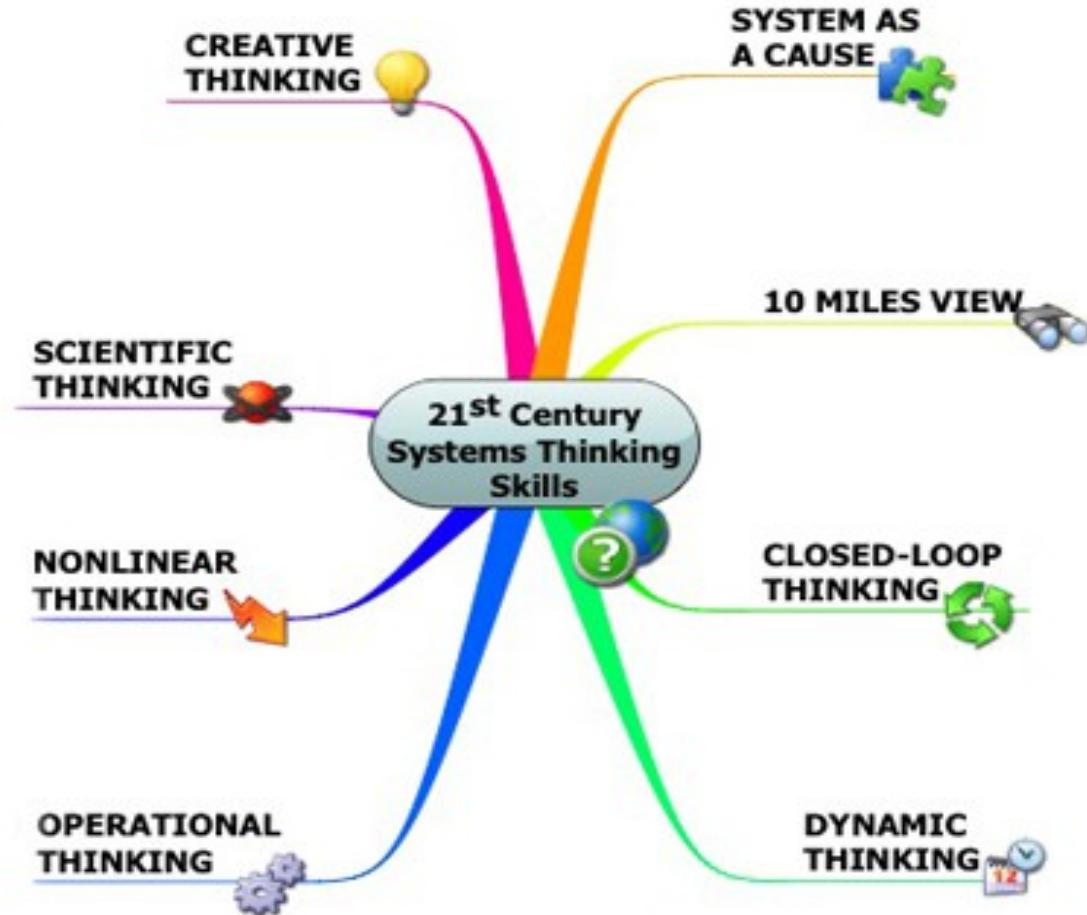
Tools & Techniques / Value Based Prioritization & Value Streaming/ Value Stream Mapping

- Process Activity Mapping
- Production Variety Funnel
- Supply chain response matrix
- Demand amplification mapping
- Physical Structure Mapping
- Quality Filter Mapping
- Decision Point Analysis
- Value Adding Time Profile
- Process Costing

# System Thinking

Tools & Techniques / Value Based Prioritization & Value Streaming

System thinking is the discipline for seeing the big picture rather than concentrating only on a particular feature or aspect of product or business



# Recap

Tools & Techniques / Value Based Prioritization & Value Streaming

1. What is value?
2. JIT in value delivery
3. Sources of revenue
4. Value Evaluation Techniques
5. Value-based Analysis
6. Minimum Marketable Features (MMF)
7. Non Function Requirements
8. Relative Prioritization and Ranking Techniques
9. Value-based Prioritization
10. Value Stream Mapping
11. System Thinking

# **Discussions !**

# Topics

## Day 1

- ✓ History of Agile
- ✓ Agile Certifications
- ✓ The Most General APM Certificate
- ✓ The Foundation of Agile Project Management
- ✓ Project Management in General
- ✓ Agile Project Management
- ✓ Agile : Developing Mindset
- ✓ Agile : Methodologies
- ✓ Agile : Metrics & Estimation

vedavit

## Day 2

- ✓ Agile : Planning, Monitoring & Adapting
- ✓ Agile : Communication
- ✓ Agile : Analysis and Design
- ✓ Agile : Product Quality
- ✓ Agile : Value Prioritization

## Day 3

- Agile : Risk Management
- Soft Skills for Agile Project Manager
- Agile : Compliance
- Agile : Roles
- Scaling Agile
- Executing an Agile Iteration
- Professional Responsibility



# Tools & Techniques Agile Project Risk Management

# Quotes Relevant to the Topic

Take Risk in Your Life!,  
if you win, you can lead.  
if you Loose you Guide!

# Topics

1. What is Risk?
2. Risk Attitude
3. Risk Management Terminologies
4. Plan Risk Management
5. Project Risk Profile
6. Risk Adjusted Backlog
7. Risk Burn-down Graph
8. Identify Risks
9. Probability and Impact Matrix
10. Qualitative Risk Analysis
11. Risk Categorization
12. Decision Tree
13. Sensitivity Analysis
14. Risk-based Spikes
15. Risk Response Plan
16. Monitoring & Controlling Risks
17. Understanding Reserves

# What is Risk?

- Risk is an event which has probability of happening in future and it can have negative or positive impact on the objective of the project
- Risk itself is not negative or positive. Risk is perceived as negative or positive based on its impact on the project's objectives.

# Risk Attitude

- Risk Seeker
  - They look for opportunities to get more return
- Risk Neutral
  - They do not look for opportunity but if something comes on the way and they are getting corresponding returns for this then they take risk
- Risk Averse
  - They are very conscious about the risk and they do not want to take risk.

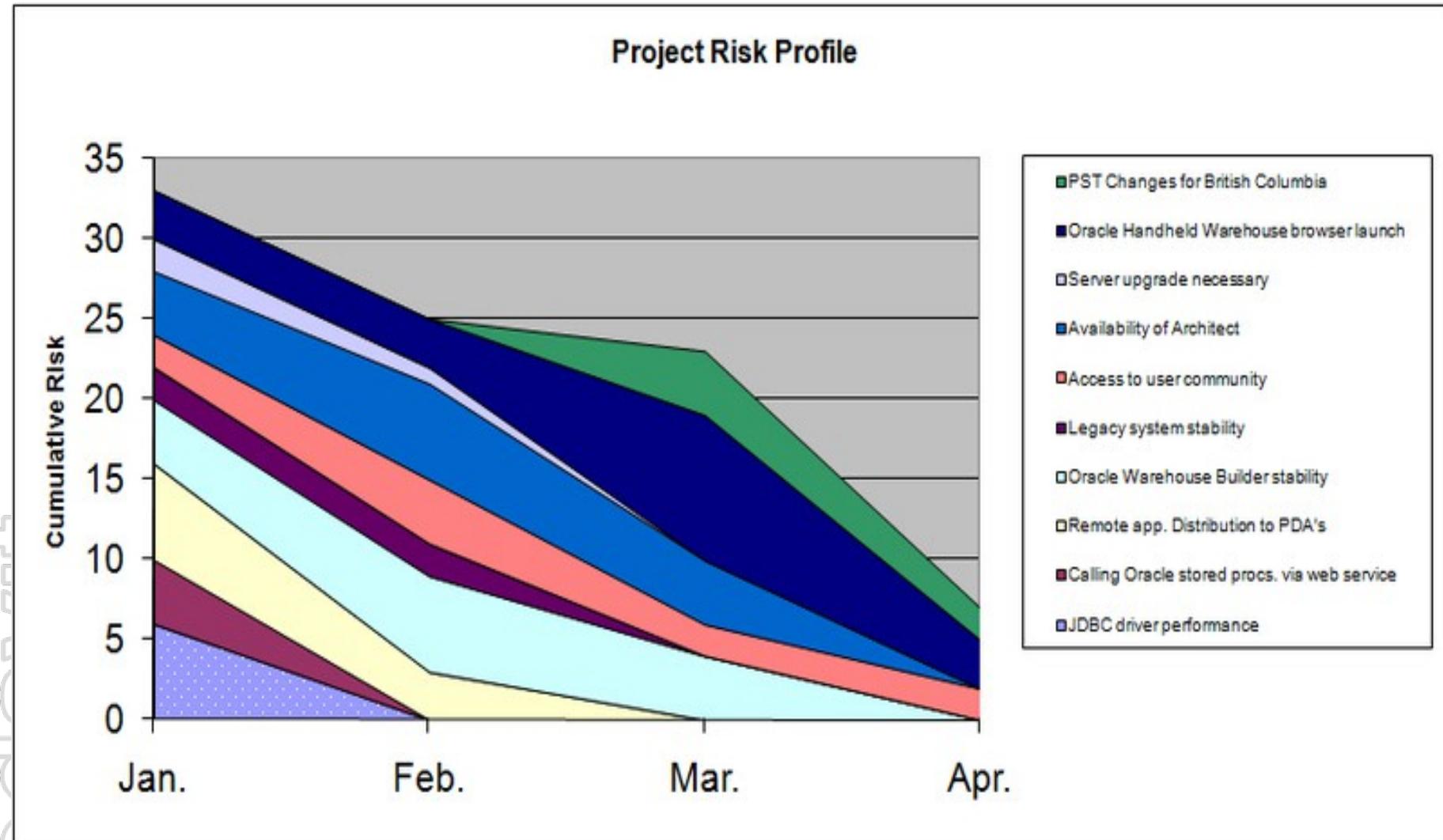
# Risk Management Terminologies

- Assumptions, Dependencies, Constraints are **root cause of risk**
- **Sources of risks** : Depending upon business domain of the project its sources may be technology, skills, resource availability, project management practices etc.
- **Proximity of risk** : In future how soon risk is going to occur
- **Risk Area**: What are geographies, department, components, services, server etc. will be affected by the risk
- **Risk Urgency** : If it occurs how soon it will impact the project objective: immediately, in short run, in long run

# Plan Risk Management

- Setup Risk Management Framework
- Identify tools & templates
- Conduct awareness training for risk management activities
- Establish responsibilities for Risk Management Activities
- Establish risk reporting formats and frequency
- Determine who will conduct risk audit and when
- Define risk attitude and risk appetite of key stakeholders

# Project Risk Profile

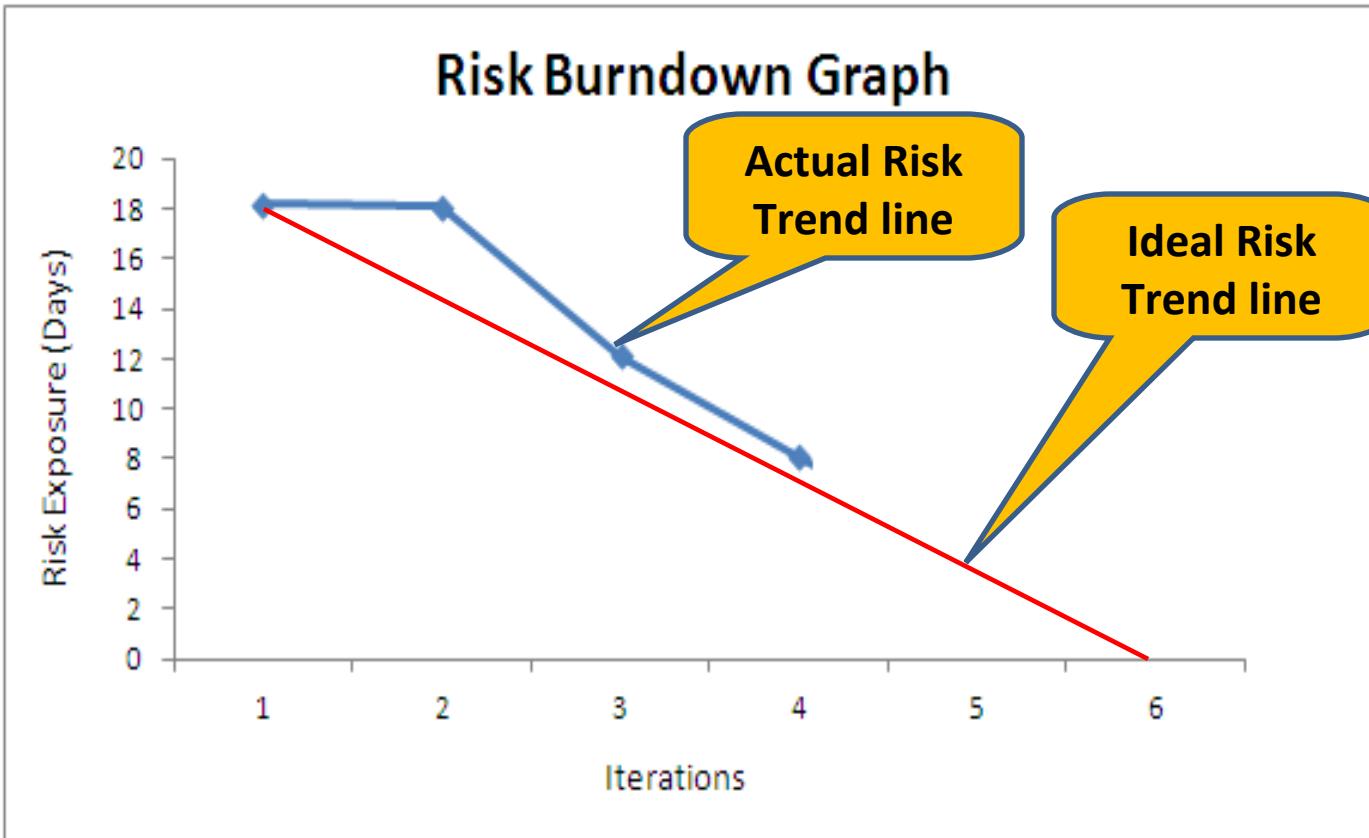


# Risk Adjusted Backlog

Risk  
Score

Risk	Probability	Impact(Days)	Exposure
Internet may be down while working	10%	1	0.1
Lack of experience may take time in implementing functionality	60%	10	6
Backup/store may required additional hardware	80%	5	4
Resource R may not join to support on nn date	50%	8	4
Partner P employee may not be available to validate new features	30%	3	0.9
Component A may not work with the product	40%	8	3.2
Total Risk Impact (Days)			18.2

# Risk Burn-down Graph



If actual risk trend line is above than ideal trend line then it means risks are not coming down at the appropriate rate

# Identify Risks

- Risk identification is performed for each product backlog items
- Every team member is responsible for identifying risk and updating related data/graph
- Risk Identification techniques
  - Brainstorming Technique
  - Delphi Technique
  - Checklist Analysis

# Probability and Impact Matrix

Probability	VL (0.9)	H (0.7)	M (0.5)	L (0.3)	VL (0.1)
Impact	VL (0.05)	L (0.1)	M (0.2)	H (0.4)	VH (0.8)
4.5%	9%	18%	36%	72%	
3.5%	7%	14%	28%	56%	
2.5%	5%	10%	20%	40%	
1.5%	3%	6%	12%	24%	
0.5%	1%	2%	4%	8%	

Which risk you would chose to respond or which you can afford to accept depends upon your risk appetite. Your risk appetite for each risk depends upon the exposure of the risk under consideration.

# Qualitative Risk Analysis

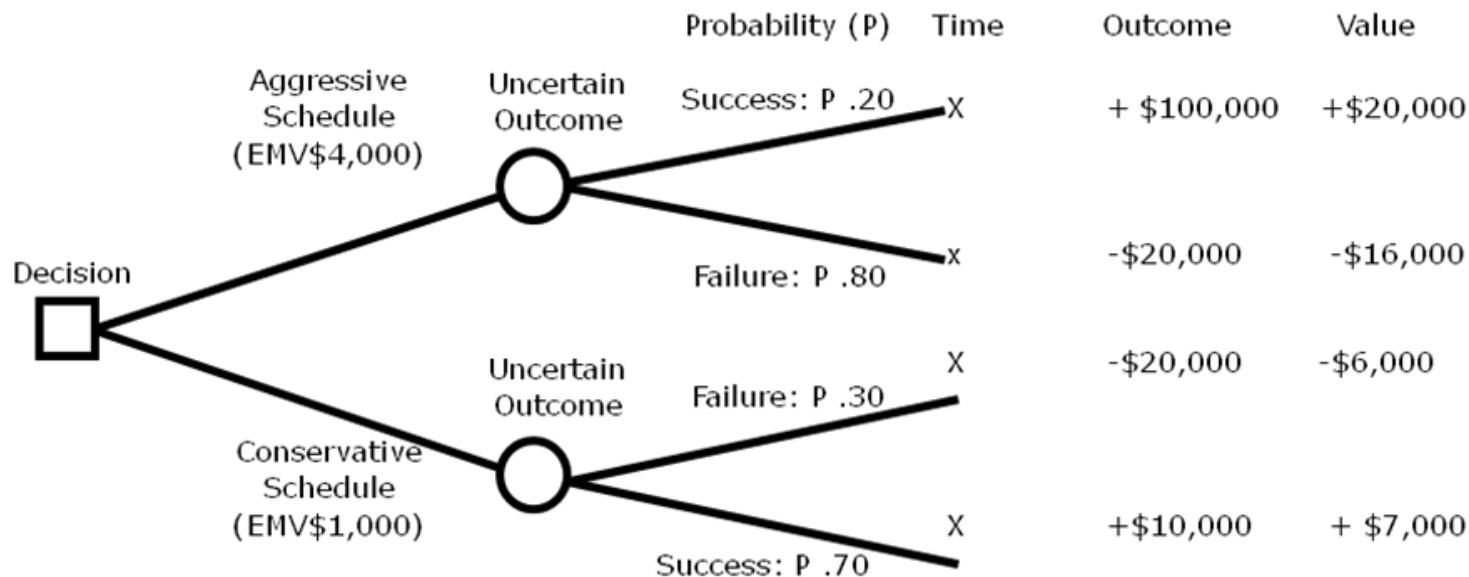
- Agile project management prefers Qualitative Risk Analysis method for analysis and prioritization of the risk
- This is the quick and simplest way of risk analysis
- In this method you need to determine probability and impact. Exposure can be calculated by multiplying probability and impact. This step is called probability and impact assessment.
- Rank risks based on their exposure and pick up those top risks which you cannot afford to forego
- Even if you have 50 risks in your risk register do not choose to work upon more than 10 on any point of time.

# Risk Categorization

- Risk can categorized based on
  - Ranking of risk
  - Cause of risk
  - Response type
  - Priority of risk
  - Any other specific trend

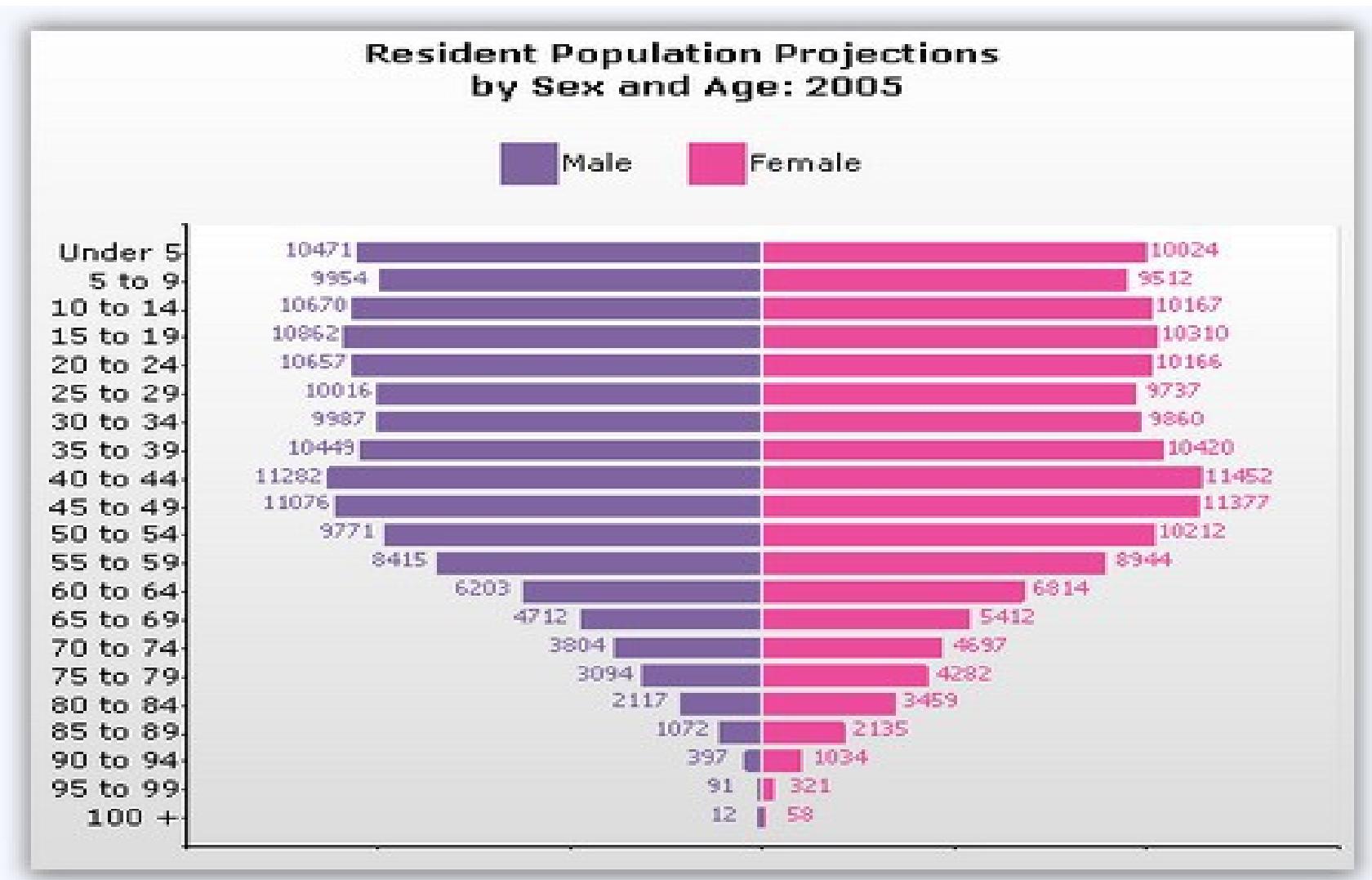
# Decision Tree

**A decision tree is a diagram that describes a decision under consideration and the implications of choosing one or other available alternatives**



- Expected monetary value (EMV) of result Outcome  $\times$  Probability of that outcome
- Expected monetary value of a decision sum of EMVs of all Outcomes stemming from that decision
- Aggressive schedule has expected monetary value of \$4,000 and is “preferred” over conservative schedule with expected monetary value of \$1,000

# Sensitivity Analysis



# Addressing an Unknown

In the product backlog sometimes agile team may find some epic/feature/user story which they are not able to understand. Therefore it is difficult to estimate those epic/features/ stories. Keeping the stories in the product backlog as estimated and in priority order is essential, as it gives the product owner insight into approximately which features will be completed by a certain date. There are 2 ways of solving this problem

- Spike Solution
- Tracer Bullet Solution

# Addressing an Unknown

- A Spike Solution
  - Spikes allows a team to set some amount of time to research and explore some unknown factor about an upcoming story.
  - Create an experimental solution that cuts through all the “layer”. The code is thrown away once solution is verified. No time is estimated for this kind of solutions but typically 1-2 days time is allowed to create a spike solution. Spike solution can be developed using A technology and actual implementation can be done using B technology.
- Tracer Bullet
  - Create an experimental solution that cuts through all the “layer”. The code is not thrown away once solution is verified but it is extended. This is not time-boxed solution and final implementation should be in the same language in which tracer bullet solution was developed.

# Risk-based Spikes

- Spike is a technical investigation to research an answer to a problem
- Unlike Technical spike risk-based spikes gets the answer through analysis. Technical spike depends upon technical research and feasibility study.
- Using Risk-based spikes there are four ways to know the answer of any complex problem
  - Sensitivity Analysis
  - Decision Tree Analysis
  - Expected Monetary Value (EVM) Analysis
  - Simulation

# Risk Response Plan

1. A Glance on Risk Response Plan
2. Strategies for Negative Risk
3. Strategies for Positive Risk
4. Common Response for +/- Risks

# A Glance on Risk Response Plan

Threats or Negative Risk	Opportunities or Positive Risk
Avoid	Exploit
Transfer	Share
Mitigate	Enhance
Accept	Accept

# Strategies for Negative Risk

## Avoid

Risk avoidance involves changing the project management plan to eliminate the threat posed by an adverse risk, to isolate the project objectives from the risk's impact or to relax the objective that is in jeopardy, such as extending the schedule or reducing scope.

## Transfer

It requires shifting the negative impact of a threat, along with ownership of the response, to a third party.

## Mitigate

It implies a reduction in the probability and / or impact of an adverse risk event to an acceptable threshold. Transference is a form of mitigation.

# Strategies Positive Risk

## Exploit

It seeks to eliminate the uncertainty associated with a particular upside risk by making the opportunity definitely happen. It may be selected for opportunities where the organization wishes to ensure that it is realized.

Examples include assigning more talented resources to the project to reduce time to completion or to provide better quality than originally planned.

- It is analogous to 'avoidance'

## Share

It involves allocating ownership to a third-party who is best able to capture the opportunity for the benefit of the project.

Examples include forming risk-sharing partnerships or joint ventures.

- It is analogous to 'transference'.

# Strategies Positive Risk

Tools & Techniques / Agile Project Risk Management/ Risk Response Plan

## Enhance

It modifies the ‘size’ of an opportunity by Increasing probability and / or positive impacts and identifying and maximizing key drivers of positive-impact risks.

It seeks to facilitate or strengthen the cause of the opportunity and proactively target and reinforce it’s trigger conditions to increase the probability.

It is analogous to ‘mitigation’.

# Common Response for +/- Risks

## Accept

It is a strategy that is adopted because it is seldom possible to eliminate all risk from a project.

It indicates that the project team

1. Has decided not to change the project management plan to deal with risk or
2. Is unable to identify any other suitable response strategy.

- ***Active Acceptance*** most commonly involves establishing a '***contingency reserve***', including amounts of time, money or resources to handle known or unknown threats or opportunities.

- ***Passive Acceptance*** requires no action, leaving the project team to deal with threats or opportunities as they occur.

# Monitoring & Controlling Risk

Tools & Techniques / Agile Project Risk Management

## Tools and techniques to monitor and control risks

- Risk Audit
- Secondary Risk
- Residual Risk
- Workarounds
- Reserve Analysis

# Understanding Reserves

## Contingency reserves : Known – Unknown

- It is designed for use only if certain events occur or only under certain predefined conditions, provided there is sufficient warning to implement the response.
- Examples of events that may trigger the contingency response include missing intermediate milestones or gaining higher priority with a supplier.
- Events triggering the contingency response should be triggered and tracked.

## Management reserves: Unknown – Unknown

- It is defined for use only if ‘the events that occur or only under certain conditions’, where information about the event & its occurrence is absolutely NOT available.

# Recap

1. What is Risk?
2. Risk Attitude
3. Risk Management Terminologies
4. Plan Risk Management
5. Project Risk Profile
6. Risk Adjusted Backlog
7. Risk Burn-down Graph
8. Identify Risks
9. Probability and Impact Matrix
10. Qualitative Risk Analysis
11. Risk Categorization
12. Decision Tree
13. Sensitivity Analysis
14. Risk-based Spikes
15. Risk Response Plan
16. Monitoring & Controlling Risks
17. Understanding Reserves

# **Discussions !**

# Topics

## Day 1

- ✓ History of Agile
- ✓ Agile Certifications
- ✓ The Most General APM Certificate
- ✓ The Foundation of Agile Project Management
- ✓ Project Management in General
- ✓ Agile Project Management
- ✓ Agile : Developing Mindset
- ✓ Agile : Methodologies
- ✓ Agile : Metrics & Estimation

vedavit

## Day 2

- ✓ Agile : Planning, Monitoring & Adapting
- ✓ Agile : Communication
- ✓ Agile : Analysis and Design
- ✓ Agile : Product Quality
- ✓ Agile : Value Prioritization

## Day 3

- ✓ Agile : Risk Management
- Soft Skills for Agile Project Manager
- Agile : Compliance
- Agile : Roles
- Scaling Agile
- Executing an Agile Iteration
- Professional Responsibility

# Agile : Soft Skills for APM

# Quotes Relevant to the Topic

Education is not the piling on of learning, information, data, facts, skills, or abilities - that's training or instruction - but is rather making visible what is hidden as a seed

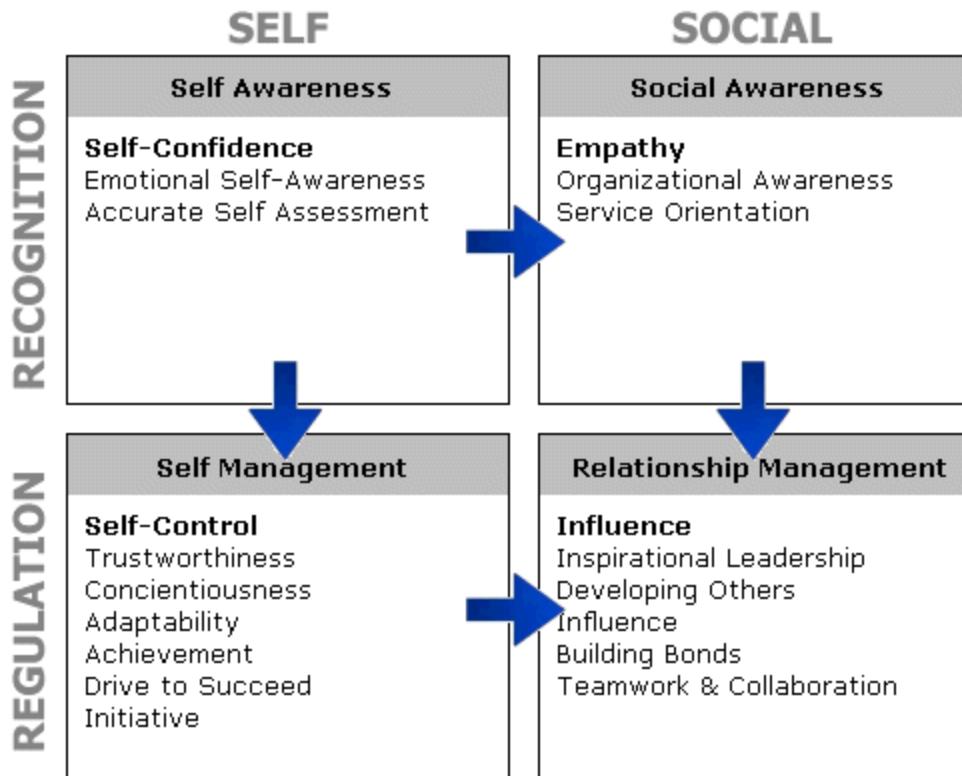
# Topics

Tools & Techniques / Soft Skills

1. Emotional Intelligence
2. Self Organizing Team
3. Shu-Ha-Ri
4. Active Listening
5. Collaboration
6. Adaptive Leadership
7. Interpersonal Skills of a Project Manager
8. Negotiation Strategies
9. Type of Powers
10. Group Decision Making Techniques
11. Conflict Management
12. 5 Stages of Team Development
13. Ground Rule
14. Mentoring & Coaching Team
15. Learning and Working
16. BART Analysis
17. Team Self-Assessment Chart
18. Servant Leadership

# Emotional Intelligence

- Emotional intelligence (EI) is the ability to identify, assess, and control the emotions of oneself, of others, and of groups
- One of the values of agile project management is “**Individuals and interactions over processes and tools**”. Thus the project success heavily depends upon people management



# Self Organizing Team

- Agile team is a team of highly self motivated, organized, disciplined individuals. They manage complete project without any extra motives. The team has strong will to make project successful.
- In all project activities like planning, execution, monitoring, adapting they participate willingly, and live with values and principles of Agile manifesto.
- No work is assigned to agile team members but they identify the most important task to deliver maximum business value in planning sessions and picks up the task on their own while sprint work is on.
- When the team was previously commanded and controlled, the general tendency is to wait for instructions than figuring out ourselves. This is anti-pattern of self-organizing team. It is very difficult to build a self organizing team under the same leadership, who used to command and control earlier

# Shu-Ha-Ri

Shu-Ha-Ri is Japanese term according to this a student or team goes through three stages of learning. Whatever you teach them it takes time to master the art. Three stages are

- Shu: This is the initial obedient stage, when the student just follows the rules.
- Ha: The student questions the rules, understands their importance, and makes innovations within the rules.
- Ri: The student breaks the rules and creates his own rules, thereby escalating to the master level.

# Active Listening

"Listen what I mean, not what I say" But how to do that?

Words are used for communication and if people are not choosing right words for communication due to poor subject understanding, poor grip on the language in which they are communicating, hidden intention or any other reason then how to listen them?

## To Actively Listen you should

- Pay Attention
- Show that you're listening
- Provide verbal and non-verbal feedback
- Defer Judgement
- Respond Appropriately

## Acronym for listen can be

- L - Look Interested - get interested
- I - Involve yourself by responding
- S - Stay on target
- T - Test your understanding
- E - Evaluate the message
- N - Neutralise your feelings

# Collaboration

- Collaboration is animated two way conversation with real understanding and progress happening. Collaboration means whole is more than the sum.
- Eight agile practices which helps team and stakeholders in effective collaboration are
  - **Trust** => to thrive on relations
  - **Sitting together** => for quick and accurate communication
  - **Real customer involvement** => to make sure that team understand what to build
  - **A ubiquitous language** => to help team members, customer understand each other
  - **Stand up meeting** => to keep team member informed
  - **Coding standards** => to seamless integrate the work of individual's work
  - **Iteration demos** => to keep the team's efforts aligned with stakeholder goals
  - **Reports** => to help reassure the organization that the team is doing well

# Adaptive Leadership

- Organization does not work in incubator but in volatile, fragile, competitive world. It is not possible to deliver results if we do not learn from mistakes, accommodate change request or changed condition. It is impossible to know everything before we start.
- Adaptive leadership pays attention to **value delivery, flexibility, fluidity, cooperation, reprioritization, simplicity, self-organizing, local decision making, innovation, lifelong learning, discuss diverse and divergent views.**

# Interpersonal Skills of a Project Manager

Tools & Techniques / Soft Skills

## Ability to establish and maintain relationships with other people

1. Leadership
2. Team Building
3. Motivation
4. Communication
5. Influencing
6. Decision Making
7. Political and Cultural Awareness
8. Negotiation
9. Trust Building
10. Conflict Management
11. Coaching
12. Emotional Intelligence
13. Group Facilitation
14. Active Listening
15. Overcoming Resistance to Change

# Negotiation Strategies

1. **Fair and Reasonable**
2. **Personal Insults** : Force people to come out of comfort zone and make decision
3. **Deadline** : Show urgency so that decision can be made faster
4. **Lying** : To know more and not commit which is not possible
5. **Limited Authority** : I do not have full power but I will get back to you
6. **Missing Man** : The man who makes decision could not join today
7. **Delay** : We need more time, lets meet once more
8. **Withdrawal** : I am not interested
9. **Fait Accompli** : If X and Y is done then there is no problem. It will be done 100%

# Type of Powers

- Penalty (Coercive)
- Referent
- Expert
- Formal (Legitimate)
- Reward

# Group Decision Making Techniques

Tools & Techniques / Soft Skills

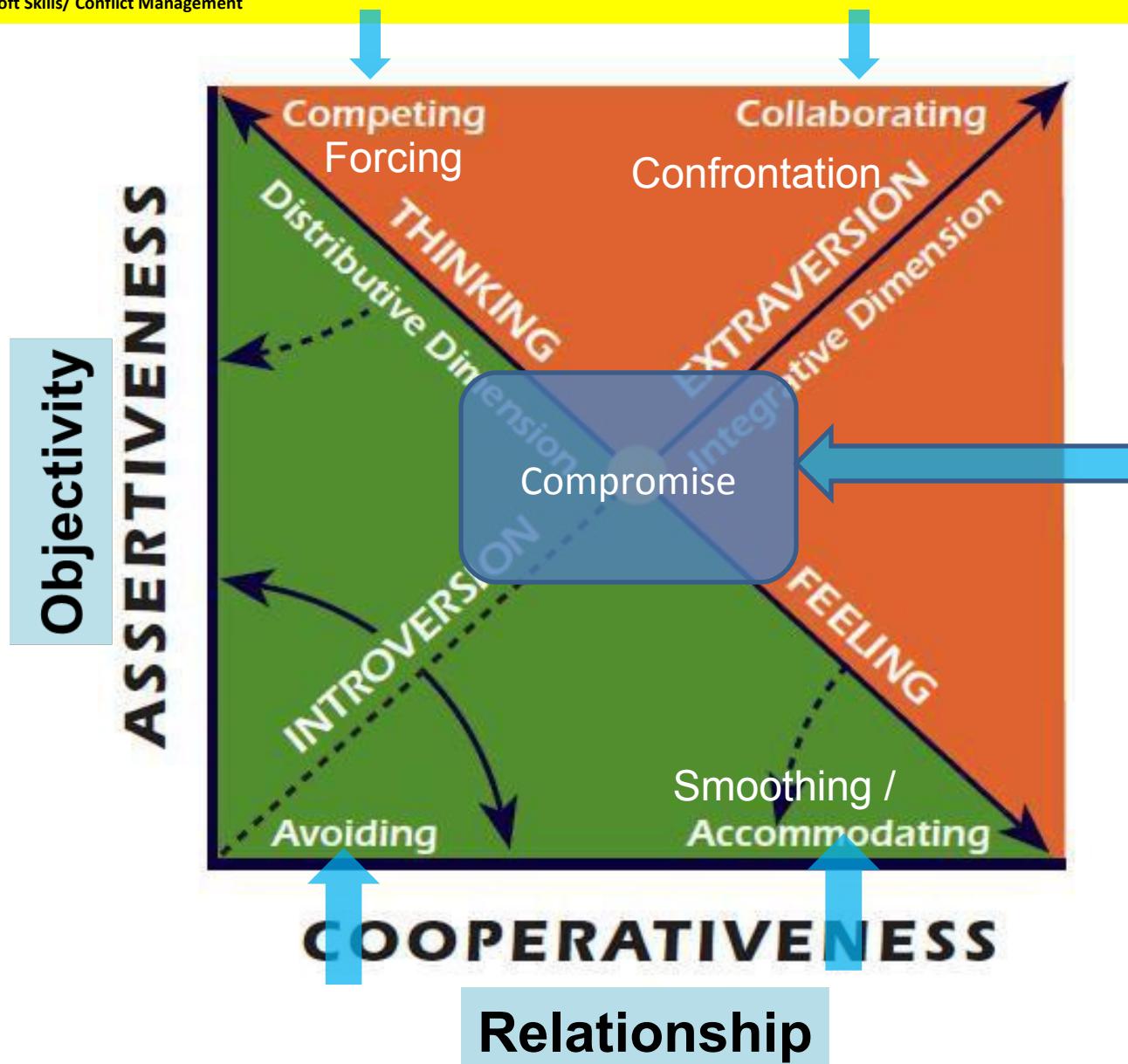
<b>ANARCHY</b> <ul style="list-style-type: none"><li>No known rules</li><li>Ambiguity is pervasive</li><li>Individual effort and thought are encouraged</li><li>Collective decision-making is impossible</li></ul> 	<b>CONSENSUS</b> <ul style="list-style-type: none"><li>Debate is encouraged</li><li>All factions are represented</li><li>Requires executive mandate and objectives</li><li>All team members agree to support decisions</li></ul> 	<b>DEMOCRATIC</b> <ul style="list-style-type: none"><li>All participants vote &amp; majority wins</li><li>"Losers" are allowed to complain</li><li>Optimum decisions are not always made</li><li>Majority of participants are happy</li></ul>    
<b>PRESIDENTIAL</b> <ul style="list-style-type: none"><li>Debate is encouraged</li><li>"Cabinet" is consulted</li><li>President makes major decisions</li></ul> 	<b>MILITARY</b> <ul style="list-style-type: none"><li>Debate is discouraged</li><li>Hierarchy is strong</li><li>Information flows up</li><li>Orders flow down</li><li>Orders are explicitly obeyed</li></ul> 	<b>REPRESENTATIVE</b> <ul style="list-style-type: none"><li>Each function provides a representative with decision-making authority</li><li>Representative core group makes decisions</li><li>All functions are represented</li><li>Core members communicate with their constituents</li></ul> 

# Conflict Management

- **Conflict an Idea**
- Conflict is Good or Bad?
- Gives you opportunity to learn about other's point of view
- It is about issues not about the people
- Remain focused in present and do not dig out dead
- Do not go with close mind
- Have trust in yourself and other that we can solve it
- Confront situation or issue not the people

# Conflict Management : Resolution Techniques

Tools & Techniques / Soft Skills/ Conflict Management

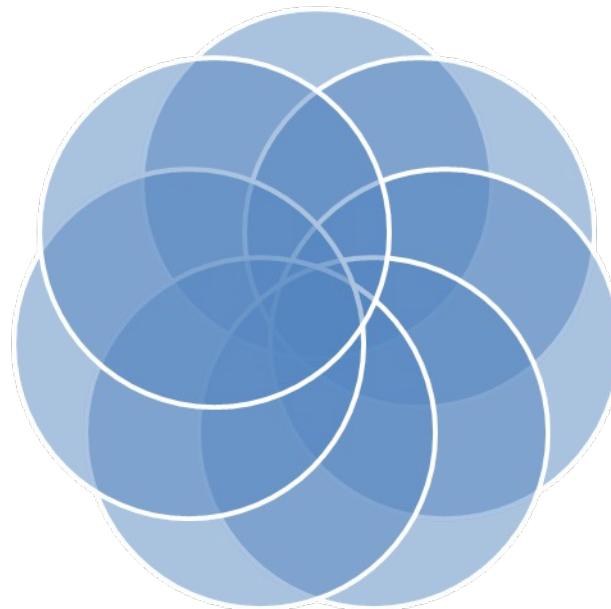


# Conflict Management : Sources

Personal  
work styles

Scarce  
resources

Cost



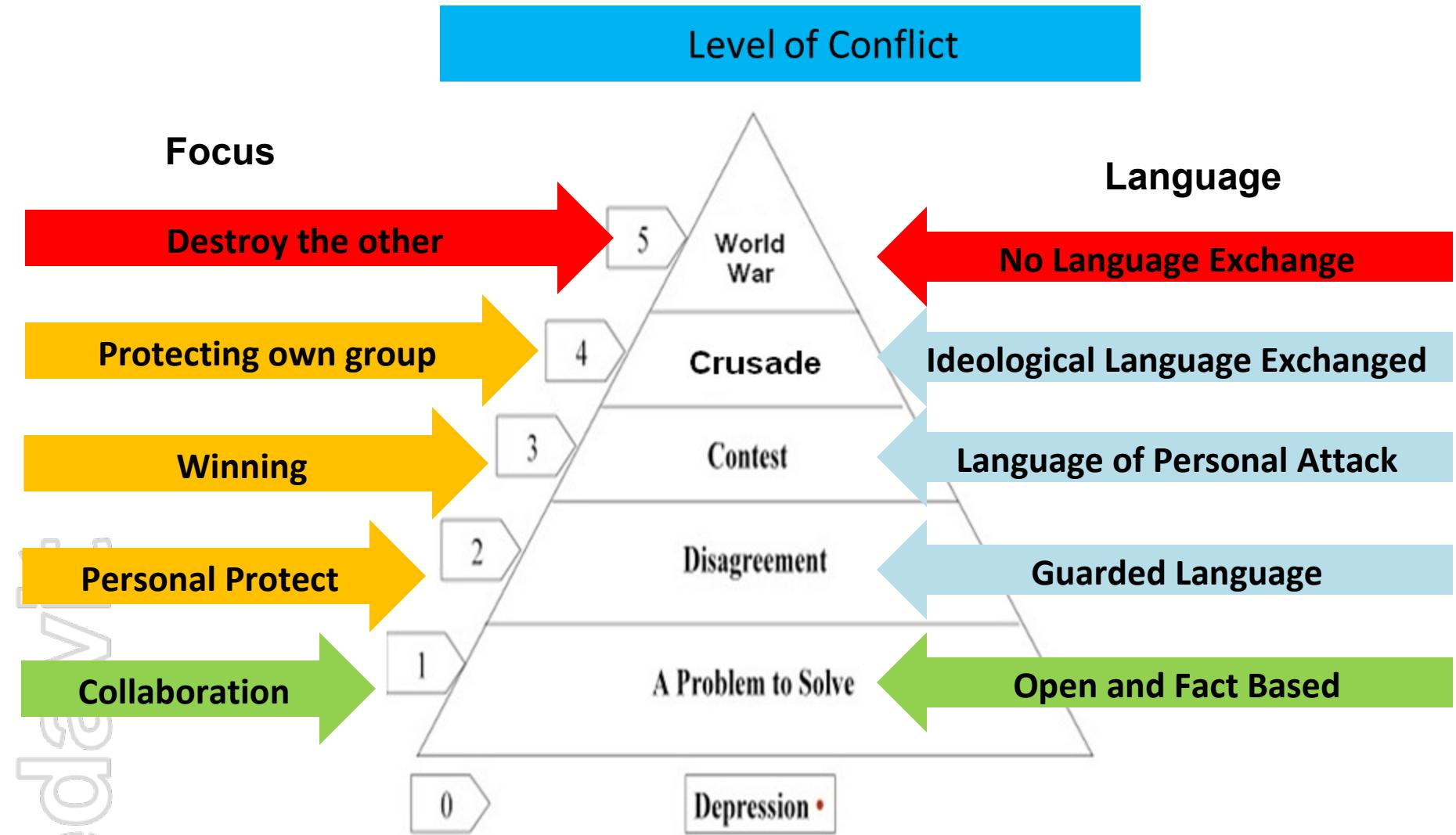
Schedules

Project  
Priorities

Technical  
Issues

Personality  
Conflict

# Conflict Management : Levels



# 5 Stages of Team Development

Tools & Techniques / Soft Skills

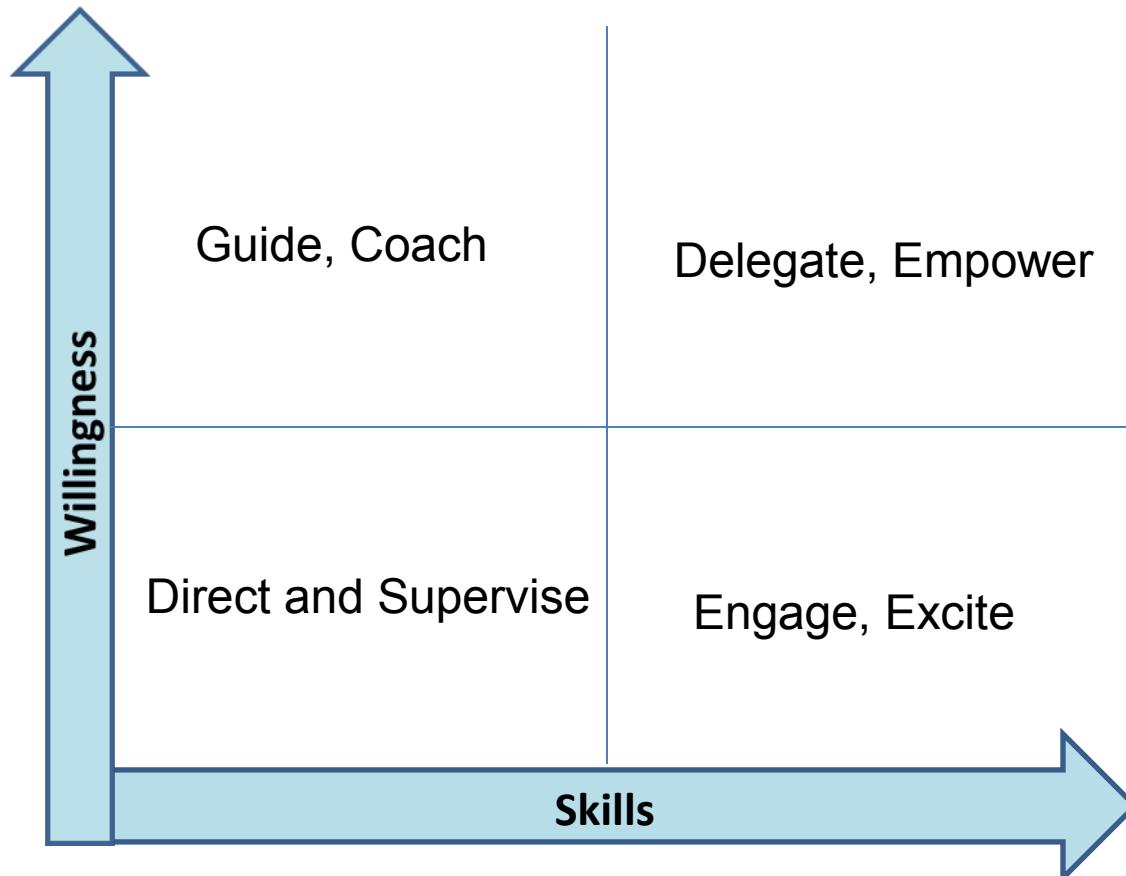
- Forming
- Storming
- Norming
- Performing
- Adjourning

# Ground Rules

- Layout ground rules which helps in team building, keeping the team intact as a whole, no partiality to anyone. Do not encourage heroic attitude. Team as a unit rise and fall not individual.

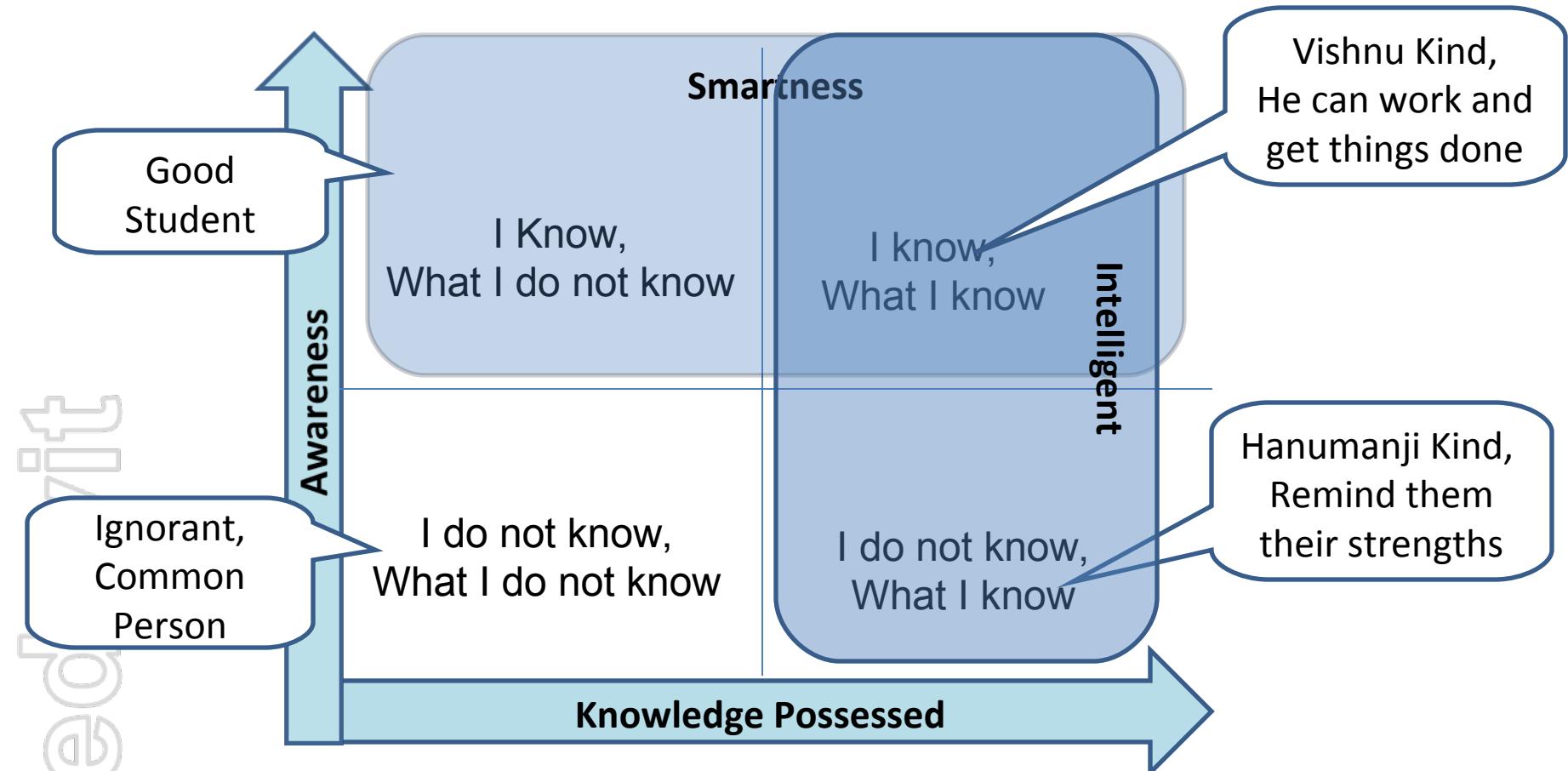
# Mentoring & Coaching Style

- Agile coaching is 40% doing and 60% being
- At the start of iteration or release team level coaching takes high importance
- In between the sprint or release individual level coaching takes high importance



# Learning & Working

- You can coach, guide, teach only to a good student
- You can guide and inspire only to Hanumanji Kind person
- Vishnu kind are leader, you need to give them vision and they can manage everything



# BART Analysis

- Boundary, Authority, Role and Task (BART)
- When people assemble to do work, clear definitions of Role, Task and Authority are essential.
- Unclear definitions of these items leads to all sorts of waste.
- Scrum's very clear Roles and associated Tasks and Authority are a big part of what makes actually Scrum 'tick'.
- A Boundary is 'collection' of attributes of the Role, Task and Authority. Boundaries can come in many forms, including: boundaries of time, boundaries in terms of access to resources, etc.

# Team Self-Assessment Chart

when people assemble to do work

Performance	Above Standard			Iteration4
	At Standard	Iteration2	Iteration3 Iteration5	Iteration6
	Below Standard	Iteration1		
		Below Standard	At Standard	Above Standard
	Behavior			

# Servant Leadership

Tools & Techniques / Soft Skills

**Leadership style where leader's focus is on creating conducive environment of performance and removing impediments. He works as fire wall for team and keep them away from day to day other distraction.**

## Characteristics of Servant Leader

- Listening
- Empathy
- Healing
- Awareness
- Persuasion
- Conceptualization
- Foresight
- Stewardship
- Commitment to the growth of people
- Building Community

# Recap

Tools & Techniques / Soft Skills

1. Emotional Intelligence
2. Self Organizing Team
3. Shu-Ha-Ri
4. Active Listening
5. Collaboration
6. Adaptive Leadership
7. Interpersonal Skills of a Project Manager
8. Negotiation Strategies
9. Type of Powers
10. Group Decision Making Techniques
11. Conflict Management
12. Team Management
13. Mentoring & Coaching Team
14. Learning and Working
15. BART Analysis
16. Team Self-Assessment Chart
17. Servant Leadership

# **Discussions !**

# Topics

## Day 1

- ✓ History of Agile
- ✓ Agile Certifications
- ✓ The Most General APM Certificate
- ✓ The Foundation of Agile Project Management
- ✓ Project Management in General
- ✓ Agile Project Management
- ✓ Agile : Developing Mindset
- ✓ Agile : Methodologies
- ✓ Agile : Metrics & Estimation

vedavit

## Day 2

- ✓ Agile : Planning, Monitoring & Adapting
- ✓ Agile : Communication
- ✓ Agile : Analysis and Design
- ✓ Agile : Product Quality
- ✓ Agile : Value Prioritization

## Day 3

- ✓ Agile : Risk Management
- ✓ Soft Skills for Agile Project Manager
- Agile : Compliance 
- Agile : Roles
- Scaling Agile
- Executing an Agile Iteration
- Professional Responsibility

# Agile : Compliance

# Agenda

1. Regulatory Compliance
2. Agile Compliance
3. Agile Self Assessment Chart

# Regulatory Compliance

Tools & Techniques / Value Based Prioritization & Value Streaming/ Compliance

- Agile regulatory compliance are taken care by Agile PMO
- Agile PMO conducts training for Agile Teams and ensure that they implement agile project management principles and the same time comply regulatory requirements
- Example of regulatory compliance
  - CMMI
  - ISO
  - Sarbanes Oxley
  - Basel II
  - HIPAA
  - ISO 27001

- SAMI : Sidky Agile Measurement Index
- SAMI is used for implementing agile in an organization and improving the maturity of agile project management by achieving higher levels
- SAMI is also used for audit and compliance purpose to ensure that maturity is achieved.

# SAMI : 5 Levels of Agile Implementation

The 5 Levels of Agility are designed to represent the core qualities of the **Agile Manifesto** rather than the qualities related to any particular agile method.

- **Level 1: Collaborative.** This level denotes the fostering of communication and collaboration between all stakeholders. Collaboration is the foundation of agile software development.
- **Level 2: Evolutionary.** Evolutionary development is the early and continuous delivery of software . It, too, is fundamental because every agile method assumes its presence.
- **Level 3: Effective.** The focus in this level is to increase efficiency of the development process by adopting engineering practices that will lead to the development of high quality working software . This is needed to prepare the development process for the next level where it can respond to constant change without jeopardizing the software system being developed.
- **Level 4: Adaptive.** This level constitutes establishing the agile quality of responding to change in the process . Defining and responding to multiple levels of feedback is essential in this level.
- **Level 5: Encompassing.** Agility is essentially a culture, and it is important to have an environment that is reflective and supportive of the agile nature of the software development process. This level concentrates on establishing an all-encompassing environment to sustain and foster agility throughout an organization.

# SAMI : 5 Principles

The Agile Manifesto outlines 12 principles that characterize agile development processes. After careful grouping and summarization, five agile principles emerged that capture the essence of the 12. These five principles guide the refinement or tailoring of the SAMI:

- Embrace change to deliver customer value .
- Plan and deliver software frequently .
- Human centric.
- Technical excellence.
- Customer collaboration.

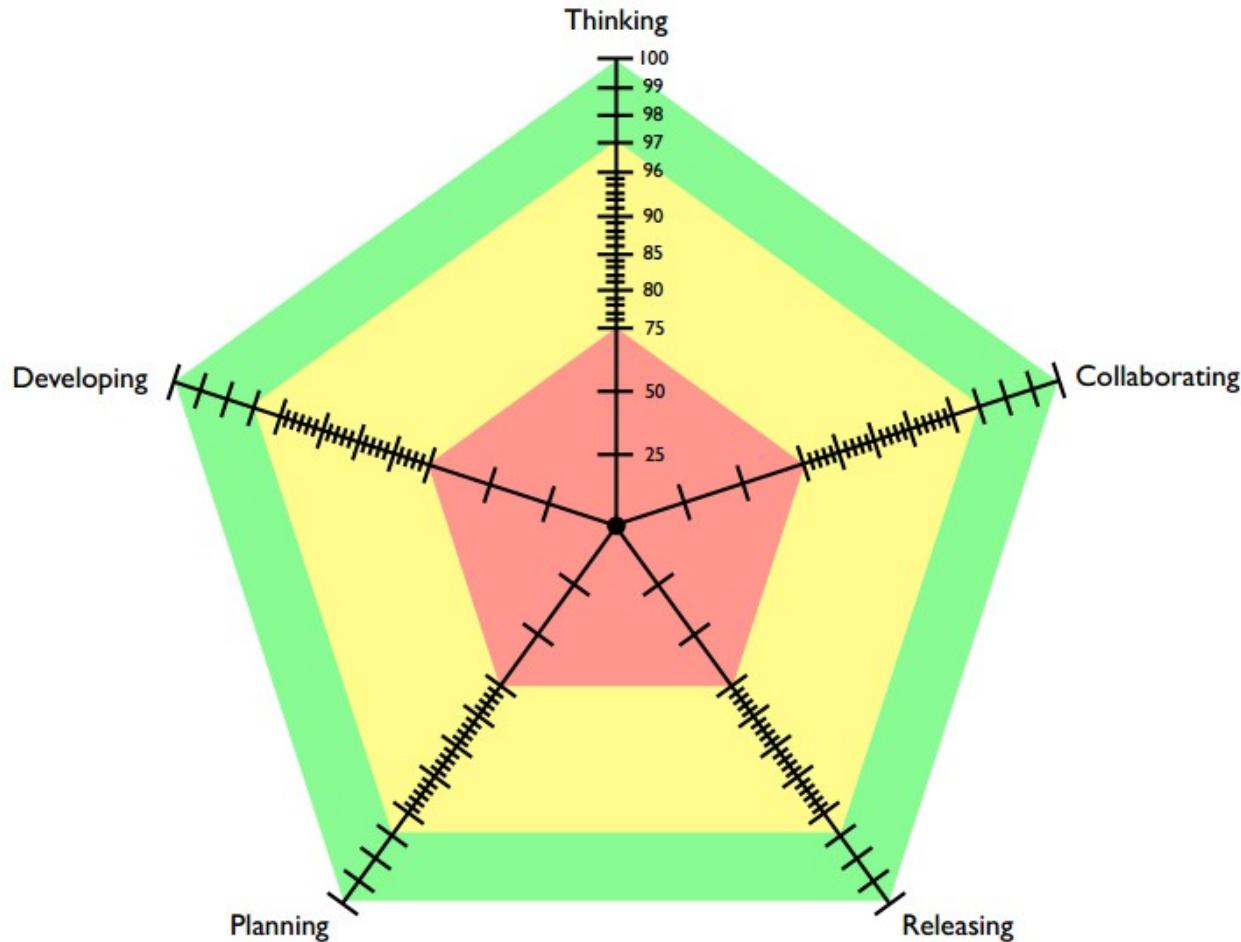
# SAMI : Agile Implementation

Tools & Techniques / Value Based Prioritization & Value Streaming/ Compliance

	<b>Embrace change to deliver customer value</b>	<b>Plan &amp; deliver learning processes frequently</b>	<b>Human Centric</b>	<b>Technical Excellence</b>	<b>Collaboration with business people</b>
<b>5. Encompassing</b>		Project estimation		Test driven	Frequent face-to-face interaction
<b>4. Adaptive</b>	Measure customer satisfaction often		Metalearning and metacognition are part of the learning processes	Design is "what is going to happen," not "what has happened"	After Action Reviews (AAR) are employed frequently
<b>3. Integrated</b>	Flow		Solution is a mix of Informal, Formal, & Nonformal learning	Solution is a blend of learning processes and delivery methods	
<b>2. Evolutionary</b>	Iterations	Time based iterations		No big design upfront	Milestones
<b>1. Collaborative</b>	User stories	Whole team planning	Empowered teams	Knowledge sharing tools (blogs, wikis, Twitter)	Customer commitment to work with design team (and vice versa)

# Agile Self-Assessment Chart

Tools & Techniques / Value Based Prioritization & Value Streaming/ Compliance



*The Art of Agile Development  
Self-Assessment Chart*

[http://www.jamesshore.com/Agile-Book/assess\\_your\\_agility.html](http://www.jamesshore.com/Agile-Book/assess_your_agility.html)

# Agile Roles

- Pigs vs Chickens
  - Pigs are those roles which are committed
  - Chickens are those roles which are involved
- Product Owner
  - Grooms product backlog, interface between product user and team
  - Prioritizes requirements based on value
  - Has authority to change requirement, reject product
  - Justify the importance of product and its features
- Scrum Master
  - Process owner, ensures that stakeholders follow the processes
  - Removes impediments
  - Works as a servant leader
- The Team
  - Includes architect, developer, tester, UI Designer
  - Cross functional multi skilled team
  - Responsible for quality of the product

# Agile Roles

- **Agile Tester**
  - Try to identify if a user does not do what he is suppose to do then what happens to the software? Report it.
  - Helps customer, product owner in writing acceptance test cases
- **Coach:** Identify and Improve work habits
- **Interaction Designer:** Learns and decides how users will interact with system.
- **SME (Domain Expert):** Understands the business domain, regulatory requirements and defines business rule for the product

# **Discussions !**

# Topics

## Day 1

- ✓ History of Agile
- ✓ Agile Certifications
- ✓ The Most General APM Certificate
- ✓ The Foundation of Agile Project Management
- ✓ Project Management in General
- ✓ Agile Project Management
- ✓ Agile : Developing Mindset
- ✓ Agile : Methodologies
- ✓ Agile : Metrics & Estimation

vedavit

## Day 2

- ✓ Agile : Planning, Monitoring & Adapting
- ✓ Agile : Communication
- ✓ Agile : Analysis and Design
- ✓ Agile : Product Quality
- ✓ Agile : Value Prioritization

## Day 3

- ✓ Agile : Risk Management
- ✓ Soft Skills for Agile Project Manager
- ✓ Agile : Compliance
- ✓ Agile : Roles
- Scaling Agile 
- Executing an Agile Iteration
- Professional Responsibility

# Scaling Agile & Agile Teams

# Agile Team

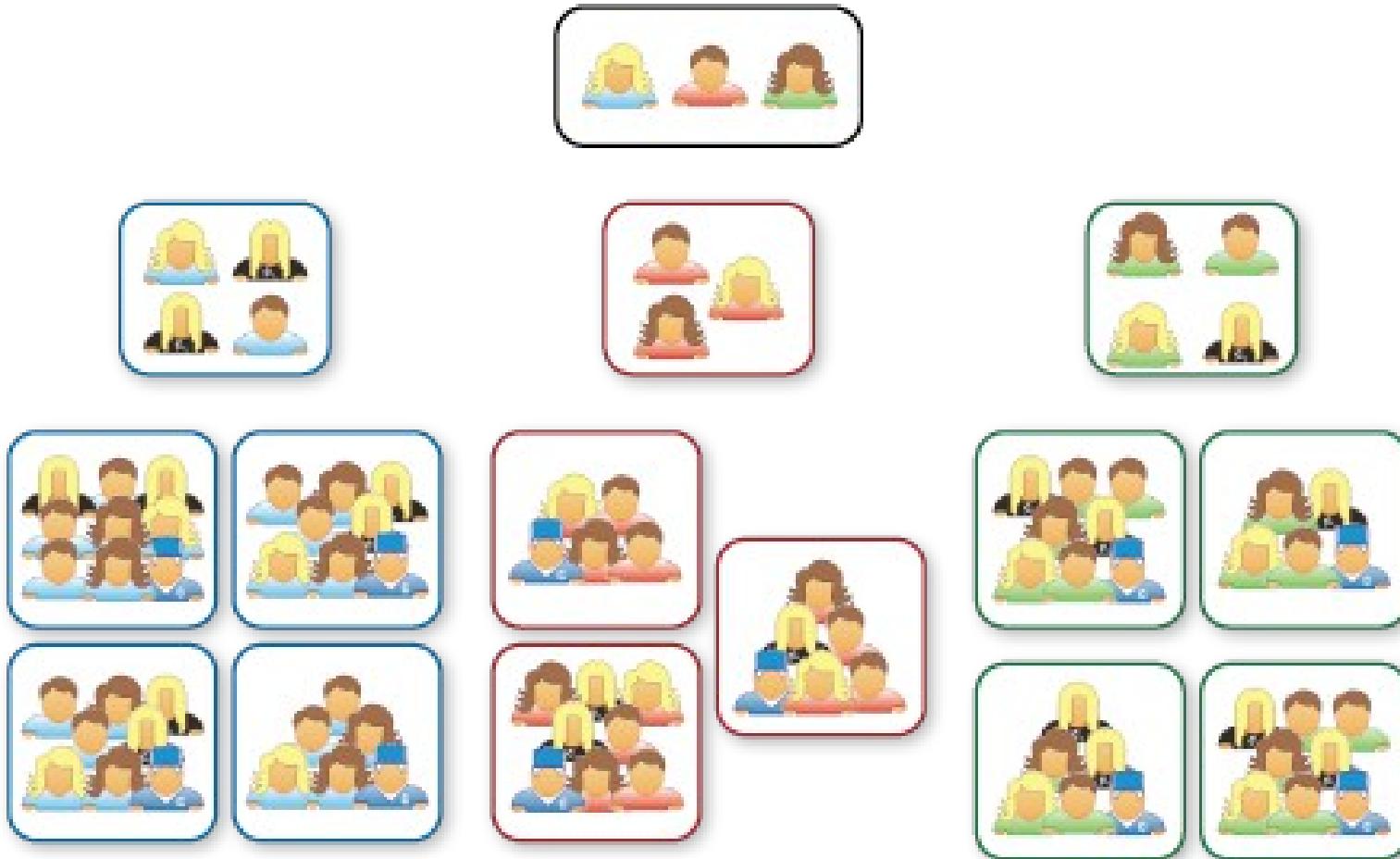
- Agile team is smaller compare to Waterfall team size
- Scrum suggests team size of  $7 \pm 2$ . But maximum team size recommended in Agile is 20
- If you need to add more resource than add more experienced people rather adding junior people
- If you further need to add more team member then use agile scaling principle

# Agile Team Scaling

- Create feature team (vertical team) do not create skill wise (horizontally).
- Create high level scrum of scrum teams
- Everyday standup meeting is joined by one designated person from each team
- Extra four questions for Scrum of Scrum Daily Standup meeting
  - What has your team done since we last met?
  - What will your team do before we meet again?
  - Is anything slowing your team down or getting in their way?
  - Are you about to put something in another team's way?

# Agile Team Scaling

Tools & Techniques / Soft Skills/ Team Management



# Type of Teams

	Manager lead team	Self-managing team	Self-organizing team	Self-governing team
<b>Setting overall direction for the team</b>				
<b>Designing the team and its context</b>				
<b>Managing work processes and monitoring progress</b>				
<b>Executing the team task</b>				

# Executing an Agile Iteration

- Update task board on regular basis
- Conduct daily scrum everyday without fail
- Keep all burndown and burnup charts updated
- Keep team self-assessment visible chart
- Keep performing agile health checkup on regular basis
- If sprint goal is not getting realized then take tough decision
- No extension or preponing of sprint in any case
- Testing should be part of day to day development no separate sprint for testing
- Inbuilt the concept of 10 minute built
- Fix bugs in the same iteration, no separate iteration for bug fixing
- In case bugs identified in sprint review then put those in product backlog and allow PO to prioritize
- Must perform sprint review even if you do not have anything to show
- Must perform retrospective

# **Discussions !**

# Topics

## Day 1

- ✓ History of Agile
- ✓ Agile Certifications
- ✓ The Most General APM Certificate
- ✓ The Foundation of Agile Project Management
- ✓ Project Management in General
- ✓ Agile Project Management
- ✓ Agile : Developing Mindset
- ✓ Agile : Methodologies
- ✓ Agile : Metrics & Estimation

vedavit

## Day 2

- ✓ Agile : Planning, Monitoring & Adapting
- ✓ Agile : Communication
- ✓ Agile : Analysis and Design
- ✓ Agile : Product Quality
- ✓ Agile : Value Prioritization

## Day 3

- ✓ Agile : Risk Management
- ✓ Soft Skills for Agile Project Manager
- ✓ Agile : Compliance
- ✓ Agile : Roles
- ✓ Scaling Agile
- ✓ Executing an Agile Iteration
- Professional Responsibility



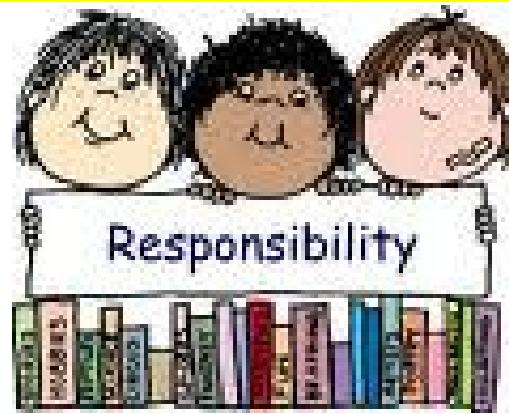
# Professional Responsibility



No use of running fast on wrong road. So, first choose your road.

**Selfless, Dedicated, Fair, Accepts Complete Responsibility, Passionate, Balance in life, Full of Love, Warmth & Respect** for all type of resources including natural resources entrusted

## Professional Responsibility As Integrated Part of Complete Project Life Cycle



# Honesty

**Honesty is our duty to understand the truth and act in a Truthful manner both in our communication and in our conduct**

1. We earnestly seek to understand the truth
2. We are truthful in our communications and conduct and provide in timely manner
3. We make commitments and promises in good faith (implied & explicit)
4. We do not engage on or condone behavior that is designated to deceive others
5. We do not engage in dishonest behavior with the intention of personal gain or at the expense of other.

# Responsibility

**Responsibility is our duty to take ownership for the decisions we make or fail to make, the actions we take or fail to take & the consequences that result**

1. Make decisions and take actions based on the BEST interests of society, Public safety and the environment
2. We accept those assignments that are consistent with our background, experience, skills and qualifications
3. We inform ourselves and uphold the policies, rules and regulations and laws that govern our work, professional and volunteer activities
4. We report unethical or illegal conduct to appropriate management & if necessary to those affected by the conduct
5. We bring violations of this Code to the attention of the appropriate body for resolution. We only file ethics complaints when they are substantiated by facts.

# Respect

**Respect is our duty to show a high regard for ourselves, others and the resources entrusted to us**

1. We inform ourselves about norms and customs of others and avoid engaging in behaviors they might consider disrespectful
2. We listen to others points of view , seeking to understand them
3. We approach directly those persons with whom we have conflict
4. We conduct ourselves in a professional manner
5. We negotiate in good faith, do not exercise the power of our expertise or position to influence decisions

# Fairness

**Fairness is our duty to make decisions and act impartially & Objectively. Our conduct must be free from competing self interest , Prejudice and favoritism.**

- 1.We demonstrate transparency in decision making and constantly re-examine our impartiality .
- 2.We provide equal access to information to those authorized & equal opportunities to qualified candidates
- 3.We do not discriminate against others based on, but not limited to , Gender, Race, Age, Religion, Disability, Nationality or Sexual orientation.

## Conflict of Interest:

1. We proactively and fully disclose any real or potential conflicts of interest to the appropriate stakeholders.
2. When we realize that we have a real or potential conflict of interest:
  - We refrain from engaging in the decision making process or otherwise
  - attempting to influence outcomes, unless or until we have made full
  - disclosure to the affected stakeholders.
3. We do not hire or fire, reward or punish, or award or deny contracts based on personal considerations. Including but not limited to Favoritism, Nepotism, or Bribery.

# Ethics Quick Test

Texas Instruments “Ethics Quick Test” for making ethical decisions

- ✓ Is the action legal?
- ✓ Does it comply with your understanding of company values?
- ✓ If you do it, will you feel bad?
- ✓ How will it look in the newspaper?
- ✓ If you know it is wrong, do not do it.
- ✓ If you are not sure, ask.
- ✓ Keep asking until you get an answer.

# **Agile Project Manager's Oath of Professional Responsibility**

# Professional Responsibility

- ✓ Ensure individual integrity
- ✓ Adhere to legal requirements and ethical standards
- ✓ Protect Stakeholders
- ✓ Share lessons learned and other relevant information
- ✓ Build capabilities of colleagues
- ✓ Advance project management professionalism
- ✓ Improve competencies as project manager
- ✓ Balance stakeholder interests in project
- ✓ Respect cultural ethnic and personal differences
- ✓ Ensure collaborative project management environment
- ✓ Comply with all organizational rules and policies

# Professional Responsibility

- ✓ Provide accurate and truthful representations in cost estimates
- ✓ Provide accurate and truthful representations in project reports
- ✓ Report violations of policies, procedures and code of ethics
- ✓ Strive for fair resolutions
- ✓ Satisfy competing needs and objectives
- ✓ Interact with others in a professional manner
- ✓ Be responsible for satisfying the complete scope and objectives of customer requirements
- ✓ Maintain and respect confidential information

# Professional Responsibility

- ✓ Ensure that a conflict of interest does not interfere with professional judgment
- ✓ Disclose conflict of interest to customer
- ✓ Disclose circumstances that could be construed as conflicts of interest
- ✓ Refrain from offering or accepting inappropriate payments, gifts, or other forms of compensation
- ✓ Adhere to all applicable laws or customs of the country where services are being provided
- ✓ Respect intellectual property developed or owned by others
- ✓ Act in an accurate, truthful and competent manner

# Topics

## Day 1

- ✓ History of Agile
- ✓ Agile Certifications
- ✓ The Most General APM Certificate
- ✓ The Foundation of Agile Project Management
- ✓ Project Management in General
- ✓ Agile Project Management
- ✓ Agile : Developing Mindset
- ✓ Agile : Methodologies
- ✓ Agile : Metrics & Estimation

vedavit

## Day 2

- ✓ Agile : Planning, Monitoring & Adapting
- ✓ Agile : Communication
- ✓ Agile : Analysis and Design
- ✓ Agile : Product Quality
- ✓ Agile : Value Prioritization

## Day 3

- ✓ Agile : Risk Management
- ✓ Soft Skills for Agile Project Manager
- ✓ Agile : Compliance
- ✓ Agile : Roles
- ✓ Scaling Agile
- ✓ Executing an Agile Iteration
- ✓ Professional Responsibility

# **Discussions !**



thank you!

## **Hari P Thapliyal,**

PMP, PMI-ACP, MCITP, PRINCE2 Practitioner, CSM, MBA, MCA, CIC, PGDFM  
PMO Architect & Project Management Evangelist

**Profile:** <http://in.linkedin.com/in/harithapliyal>

For content related queries please contact me via  
[hari.Prasad@vedavit-ps.com](mailto:hari.Prasad@vedavit-ps.com)