Agile-Scrum
1 Day- Introductory Training

Prepared By: Hari P Thapliyal

Mobile: +91 9535999336

email : hari.prasad@vedavit-ps.com

Copyright 2013 Vedavit Project Solutions

Faculty & Participants An Introduction

Name, Company, Current Designation, Why Certification, Hobb(y/ies)

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

Workshop Objective

- ✓ ***Learn Agile Project Management best Practices*** as per PMI-ACP certification requirements
- ✓ **Getting accustomed to new terminology** of Agile Project Management

Agenda

- What is Project?
- What is Agility?
- Why Agile?
- Estimation in Scrum
- Scrum Roles
- Scrum Artifacts
- Release Planning
- Sprint Planning
- Tracking in scrum
- Exercises
  - Velocity calculation, Planning Poker
  - Scrum Simulation Exercise(Sprint Planning, Execution)

Copyright 2013 Vedavit Project Solutions

What is Project???

Copyright 2013 Vedavit Project Solutions

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*

Source PMBOK Guide Version 5.0

Copyright 2013 Vedavit Project Solutions

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

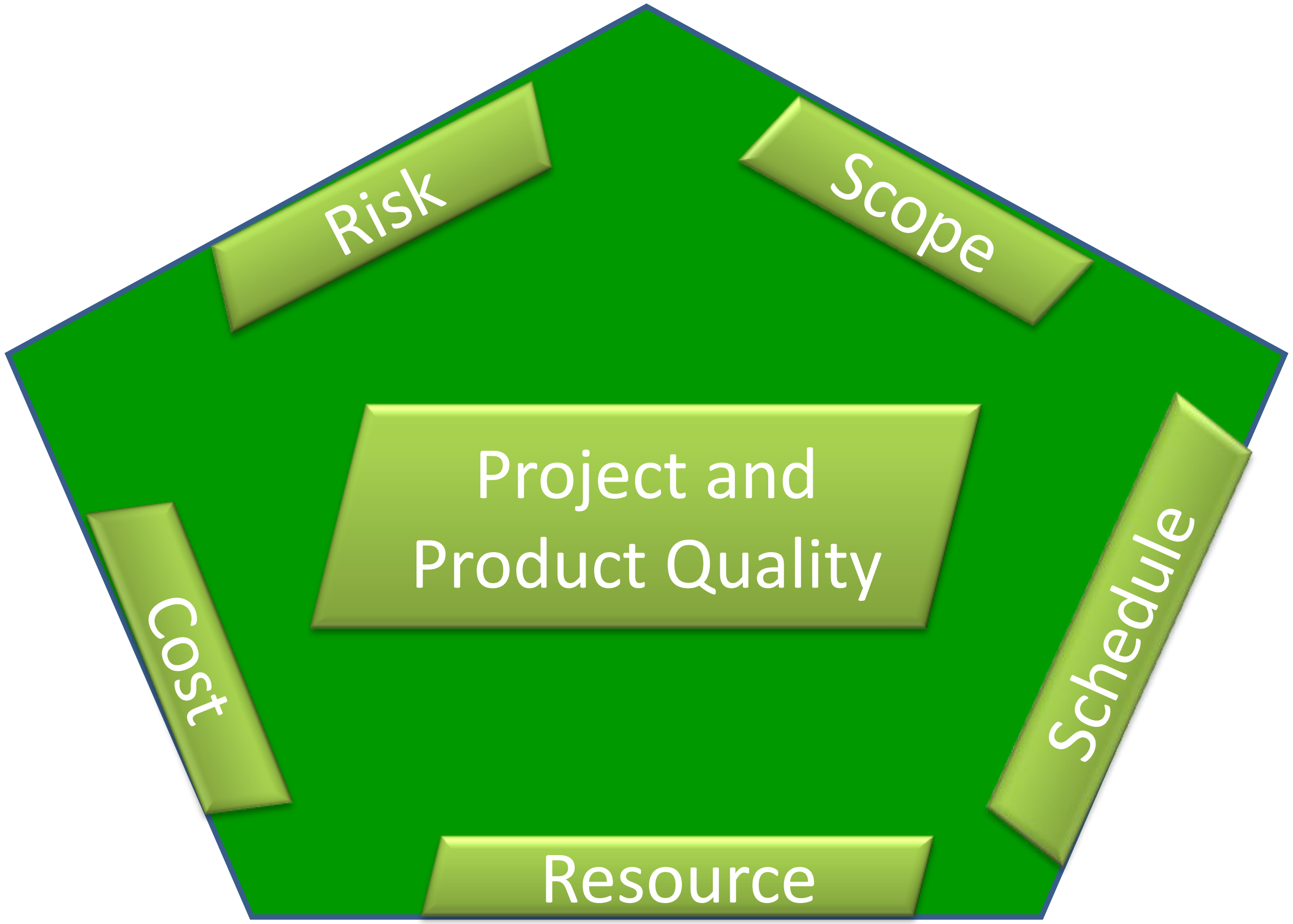
Copyright 2013 Vedavit Project Solutions

“Projects” different from “operations”?

Projects	Operations
➢ Permanent Project Charter	➢ Semi-permanent charter
➢ Catalyst for change	➢ Maintains status quo
➢ Unique product or service	➢ Standard product or service
➢ Heterogeneous teams	➢ Homogeneous teams
➢ Start and end date	➢ Ongoing
➢ Progressive elaboration	➢ Predefined product

Copyright 2013 Vedavit Project Solutions

Project Constraints



Copyright 2013 Vedavit Project Solutions

Before we go ahead…

Lets see one case study on agile project management

Copyright 2013 Vedavit Project Solutions

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-usable 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.

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

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

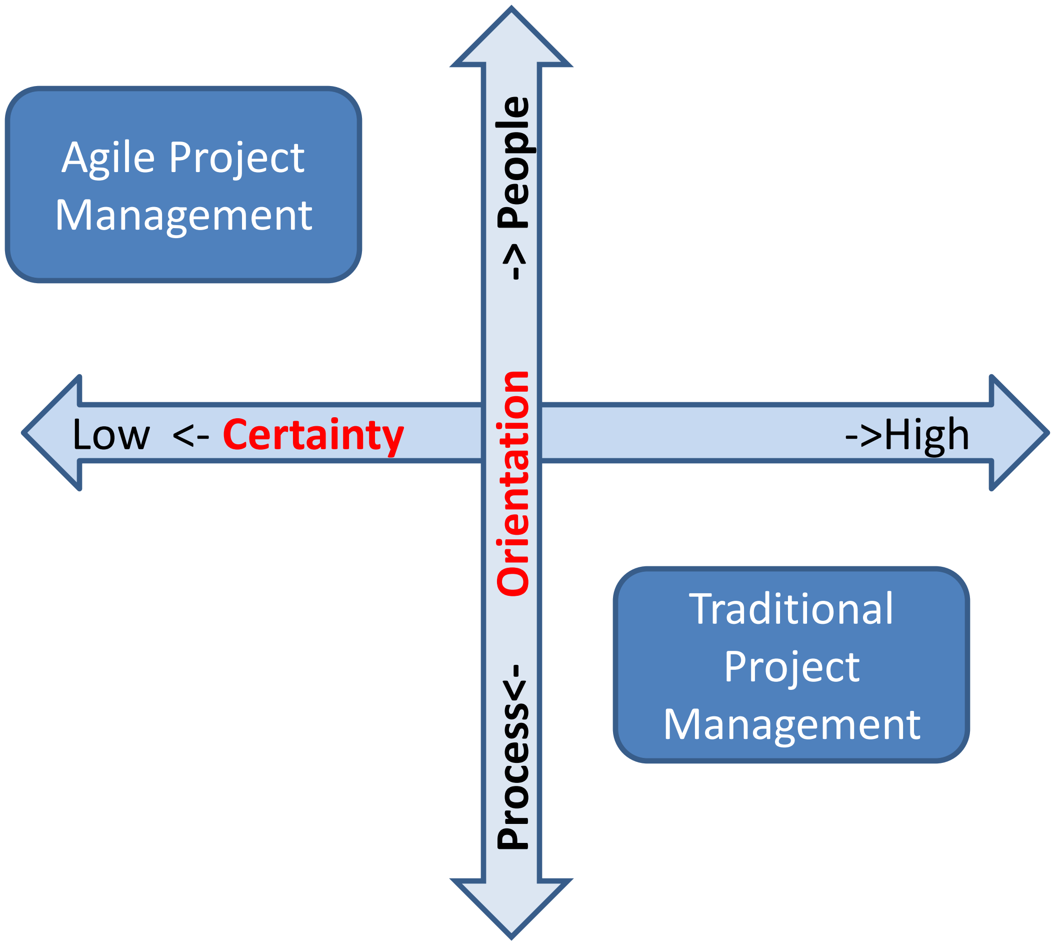
Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

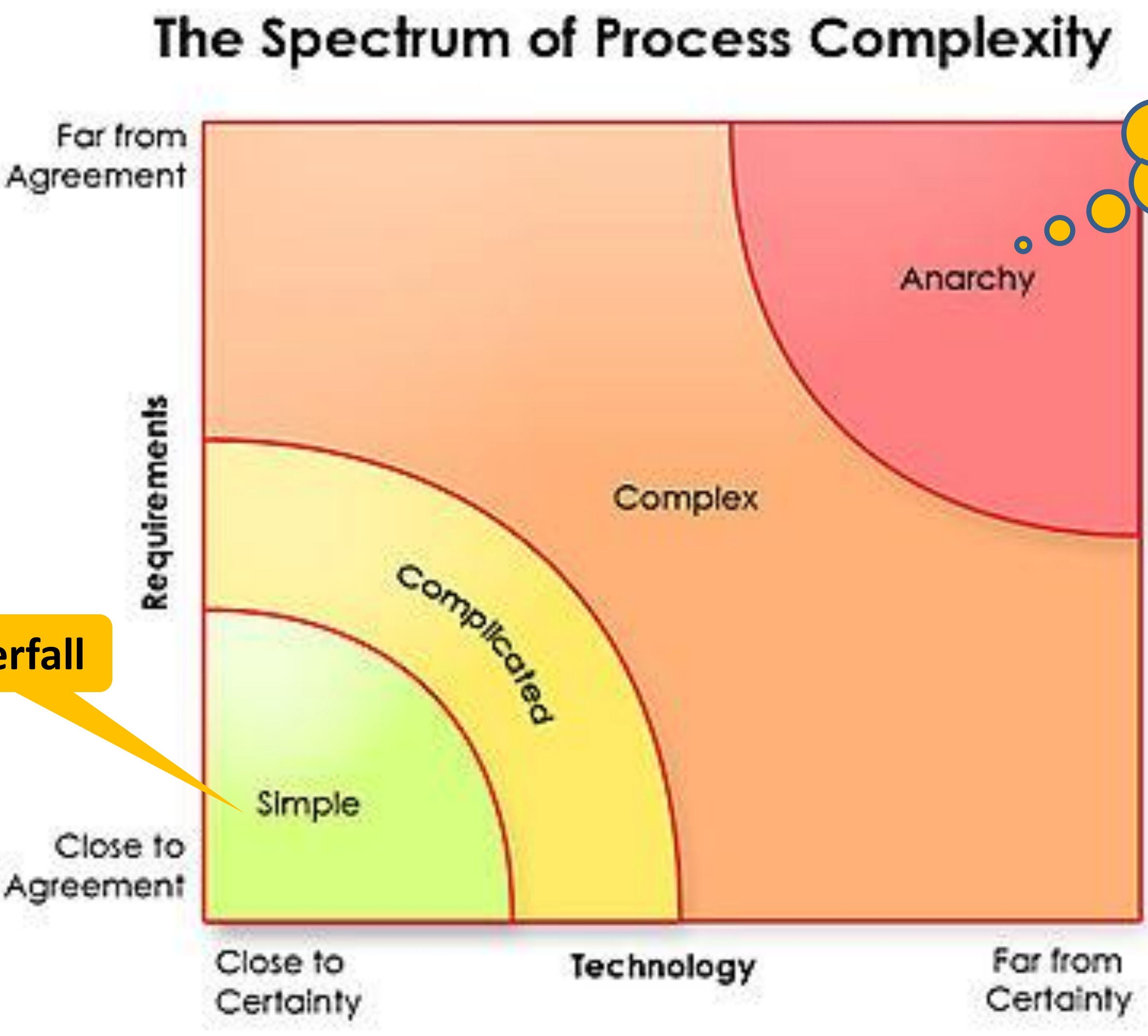
Which PM Methodology is best?



Copyright 2013 Vedavit Project Solutions

Which PM Methodology is best?

Ralph Stacey's complexity matrix



Waterfall

Nothing works here

Copyright 2013 Vedavit Project Solutions

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.

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

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.

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

Agile Methodologies- Scrum

- 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

Copyright 2013 Vedavit Project Solutions

Agile Methodologies- Scrum

Scrum Practices

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

Copyright 2013 Vedavit Project Solutions

Agile Methodologies-XP

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

Copyright 2013 Vedavit Project Solutions

Agile Methodologies- XP

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

Copyright 2013 Vedavit Project Solutions

XP Practices

Shared understanding

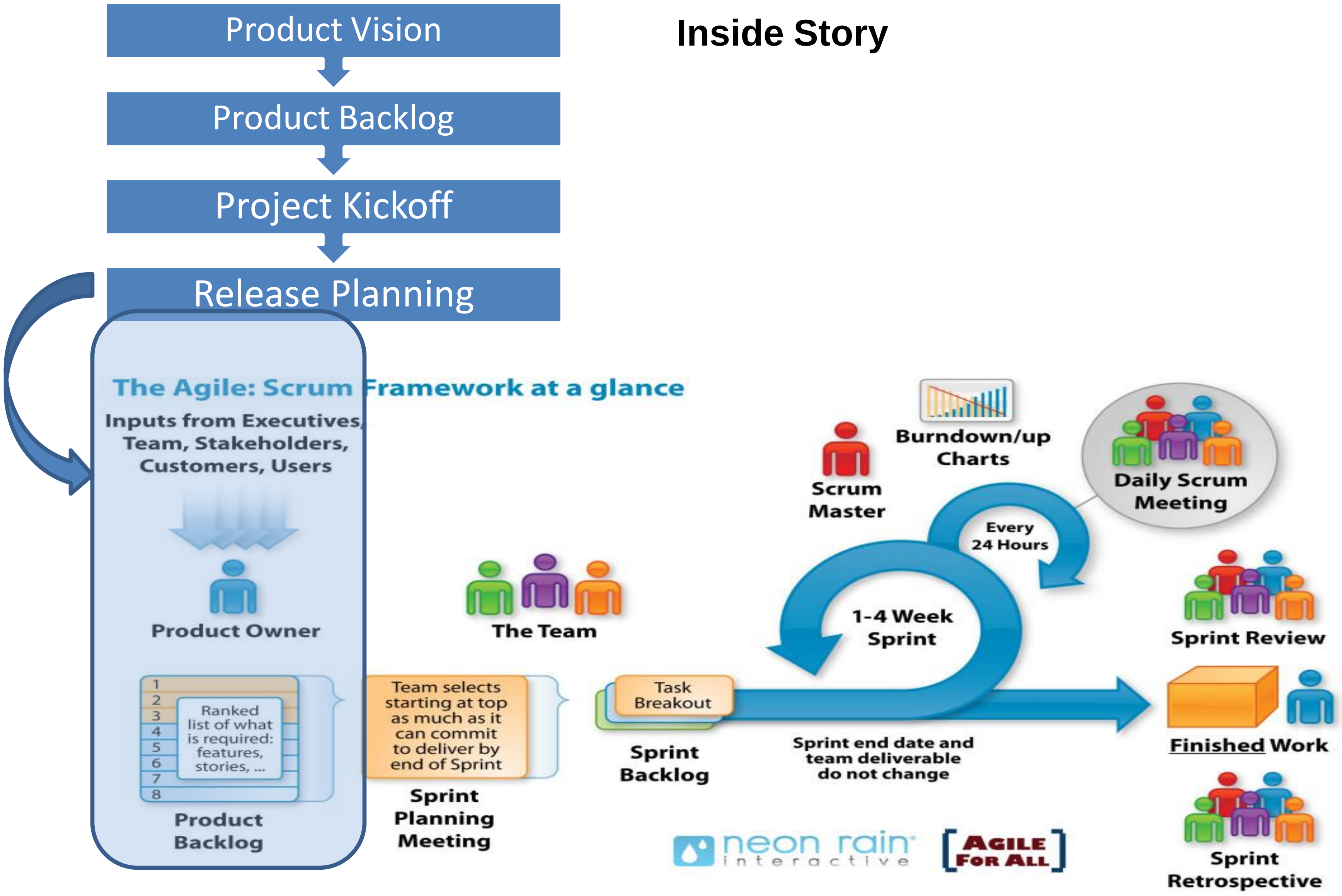
8. Simple Design (Do Simple Things, You Aren’t Gonna Need It **(YAGNI)**, Once And Only Once **(DRY)**, Simplify Vigorously)
9. System Metaphor
10. Collective Code Ownership
11. Coding Standard or Coding Conventions

Programmer welfare

12. Sustainable Pace

Copyright 2013 Vedavit Project Solutions

Agile Project Life-cycle

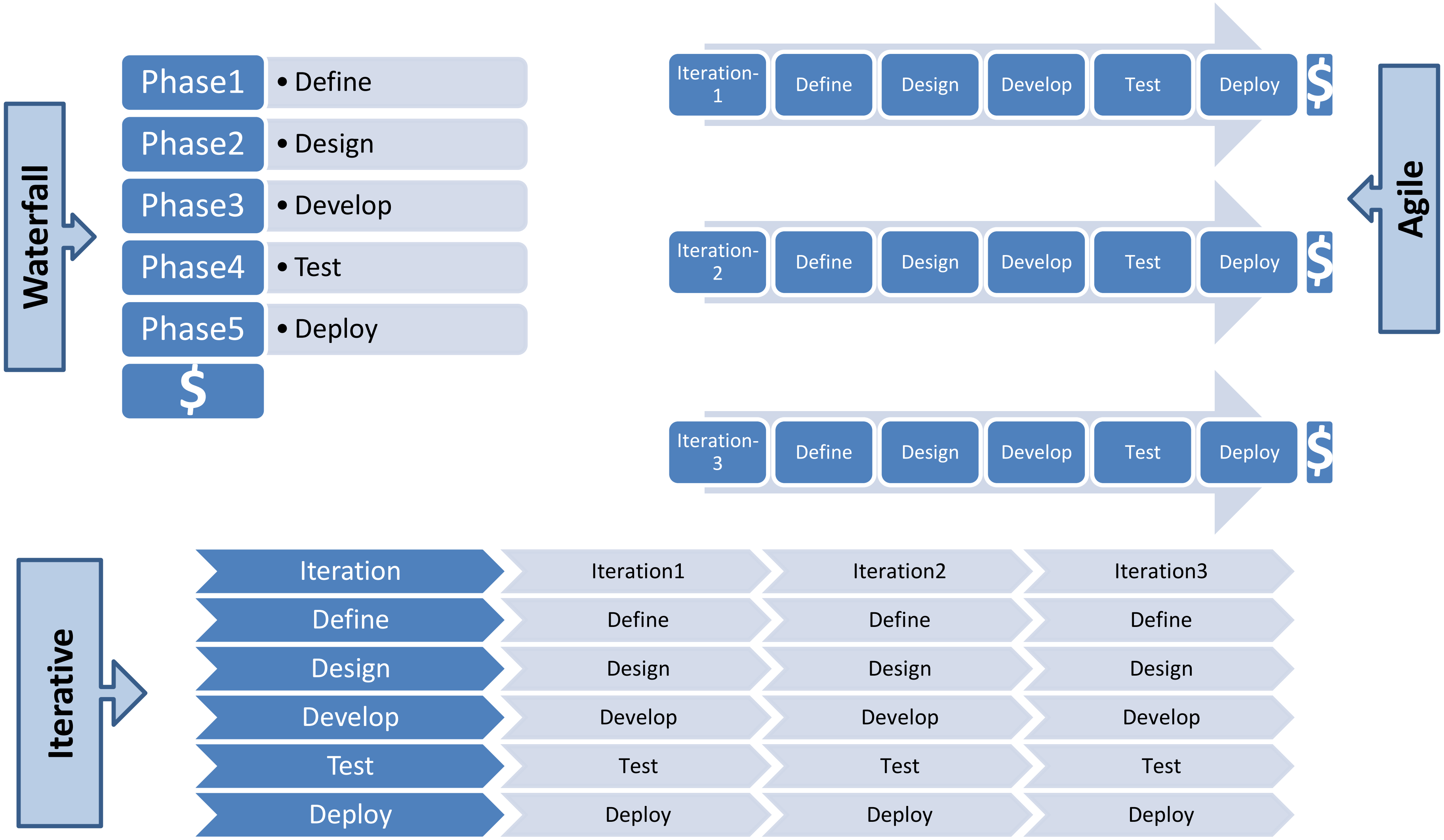


Inside Story

Copyright 2013 Vedavit Project Solutions

Value Delivery & Project Life Cycle

Lifecycles: Waterfall- Iterative- Agile



Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

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.

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

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?

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

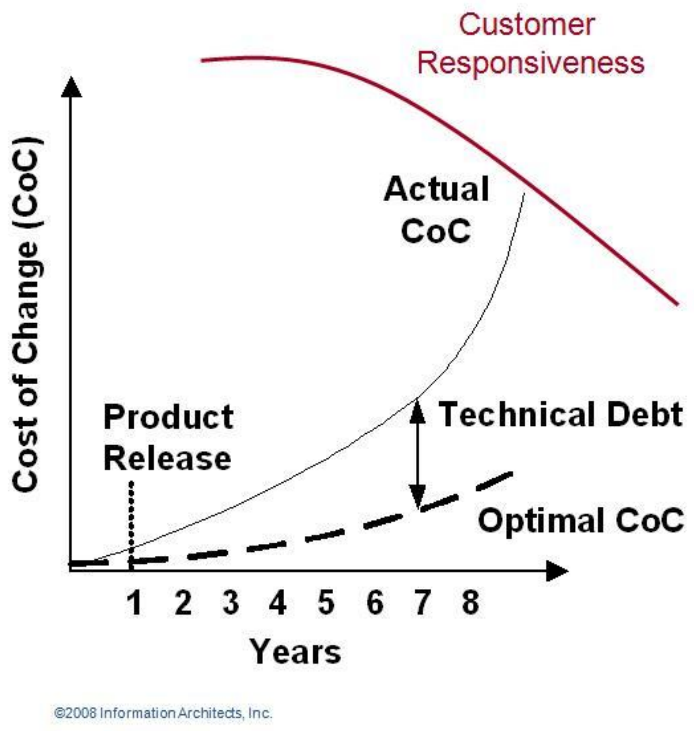
Last Responsible Moments

- 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

Copyright 2013 Vedavit Project Solutions

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



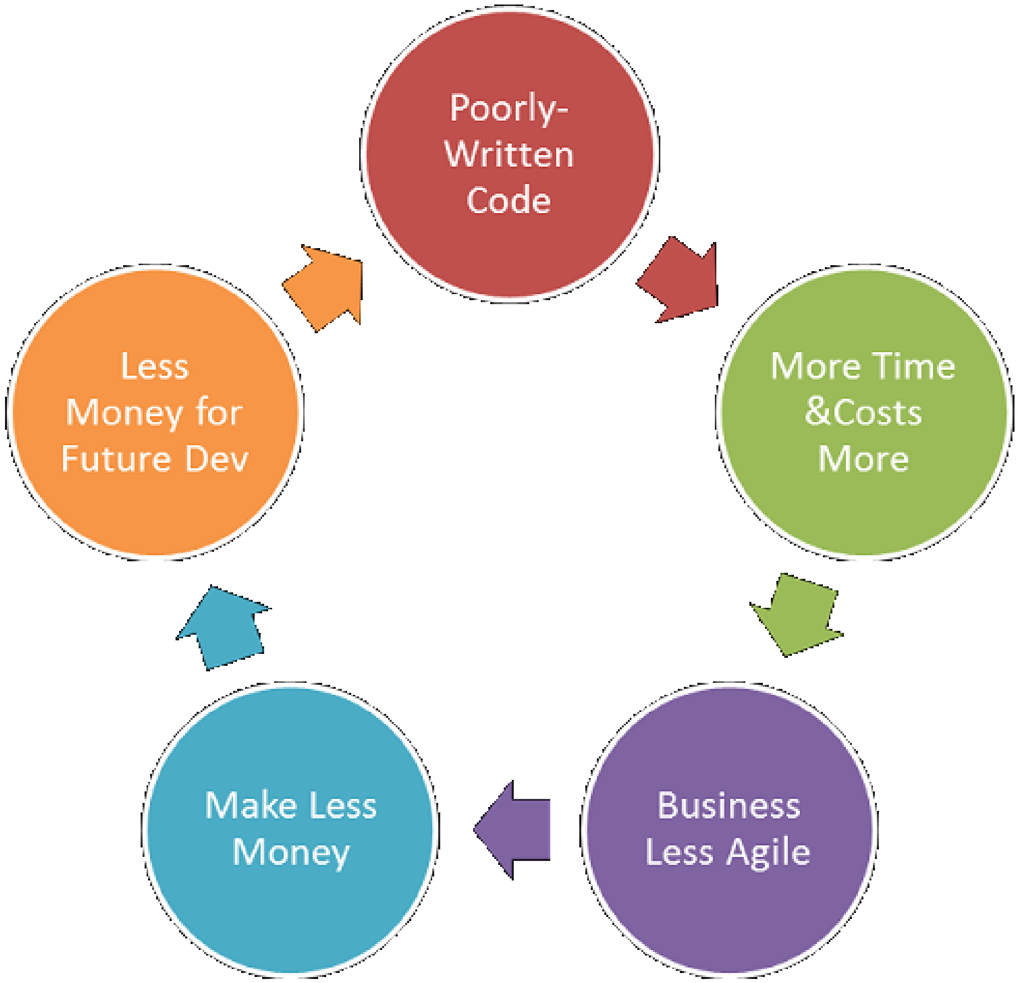
- Once on far right of curve, all choices are hard
- If nothing is done, it just gets worse
- In applications with high technical debt, estimating is nearly impossible
- Only 3 strategies
  - Do nothing, it gets worse
  - Replace, high cost/risk
  - Incremental refactoring, commitment to invest

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.

Copyright 2013 Vedavit Project Solutions

Vicious Cycle of Technical Debt



Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

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.

Copyright 2013 Vedavit Project Solutions

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 Desiger
  - Cross functional multi skilled team
  - Responsible for quality of the product

Copyright 2013 Vedavit Project Solutions

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.

Copyright 2013 Vedavit Project Solutions

Daily Stand-ups

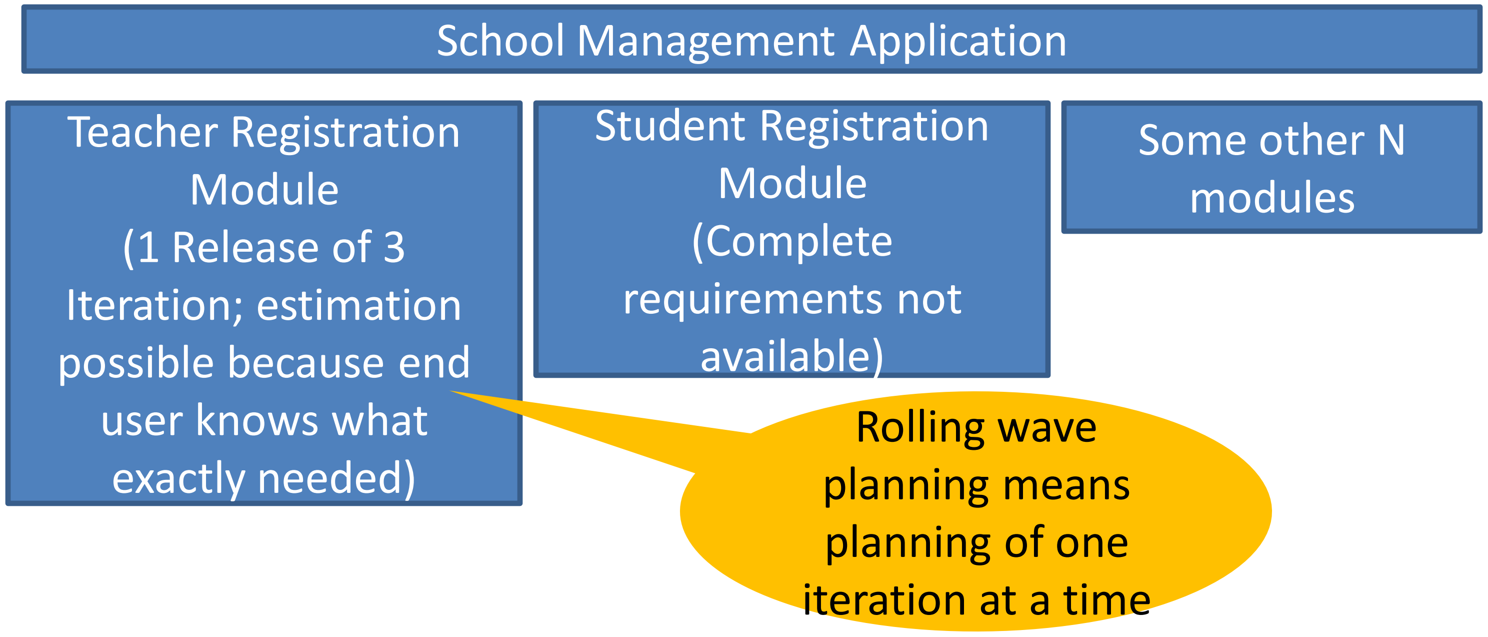
3 Questions of standup meeting which every team member must address are

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

Copyright 2013 Vedavit Project Solutions

Progressive Elaboration

- 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

Copyright 2013 Vedavit Project Solutions

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 21st century! Days of definitive processes and predictive planning are not many.

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

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”

Copyright 2013 Vedavit Project Solutions

3Cs of User Story

- Card
- Conversation
- Confirmation

Copyright 2013 Vedavit Project Solutions

INVEST Model of User Story

INVEST model defines following characteristics of a user story

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

Copyright 2013 Vedavit Project Solutions

Epic, Feature, Story, Task

- 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

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

Agile Planning

Release Planning

1. Product owner is ready with prioritized product backlog (prioritization is done based on business value)
2. Product owner defined release goals
3. During release planning product owner picks up only those user stories and features which helps him achieving release goals
4. All the selected user stories are pushed into release backlog
5. User stories in release backlog are size estimated using agile estimation techniques like story points
6. User stories in release backlog are again prioritized. It helps in creating number of iterations and iteration plan.

Agile Planning

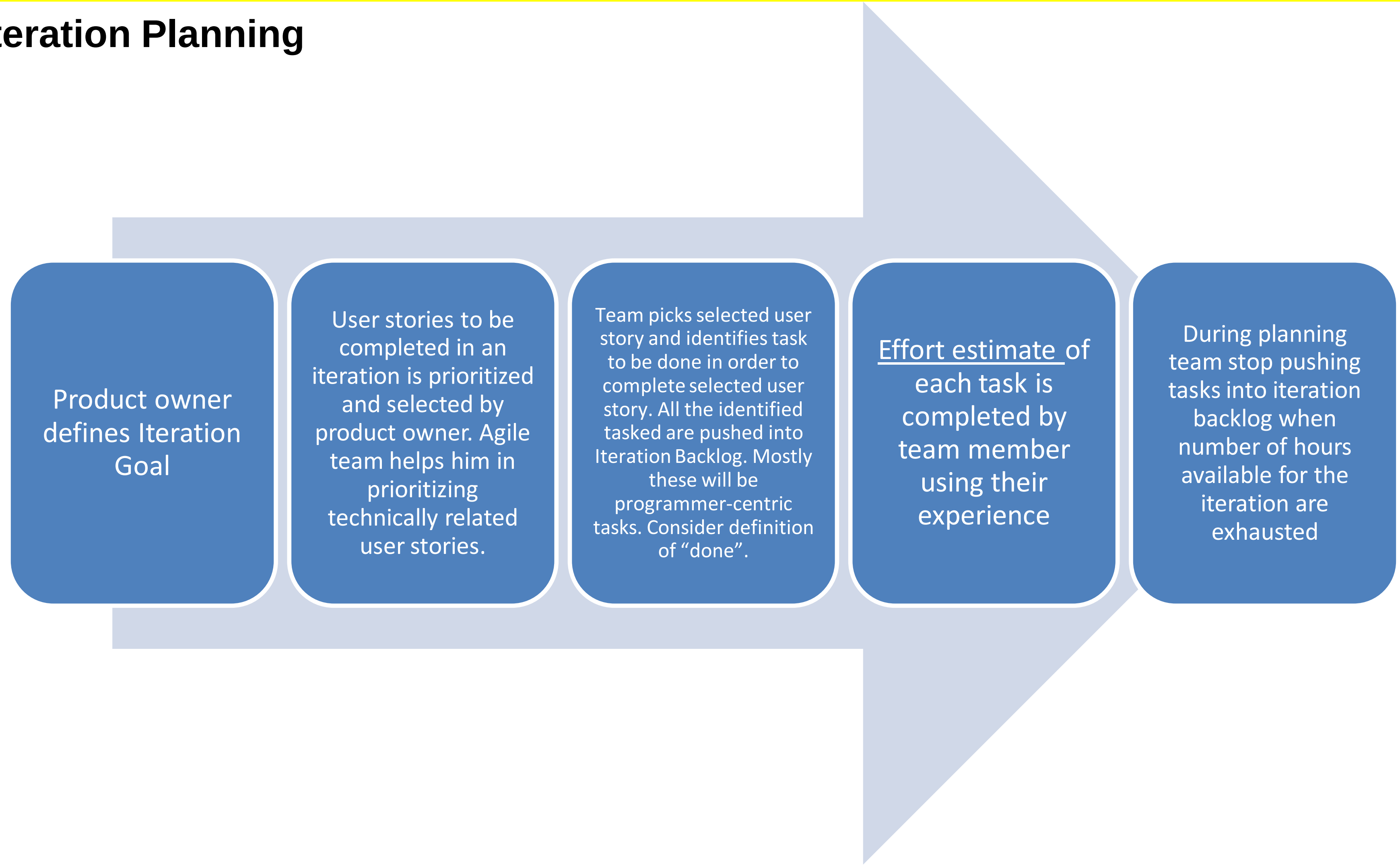
Iteration/Sprint Planning

- In iteration planning team identifies the task to be performed for each user story.
- Task 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 team start counting iteration from zero so the initial iteration is called **Sprint 0** or Iteration 0. In this iteration they take all technical spike 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.

Copyright 2013 Vedavit Project Solutions

Agile Planning

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 tasked 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

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

Copyright 2013 Vedavit Project Solutions

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±2)
  - Regular iteration work
  - Iteration Review (1 hour for every week of iteration)
  - Iteration Retrospective (1 hours for every week of iteration)

Copyright 2013 Vedavit Project Solutions

Minimum Marketable Features (MMF)

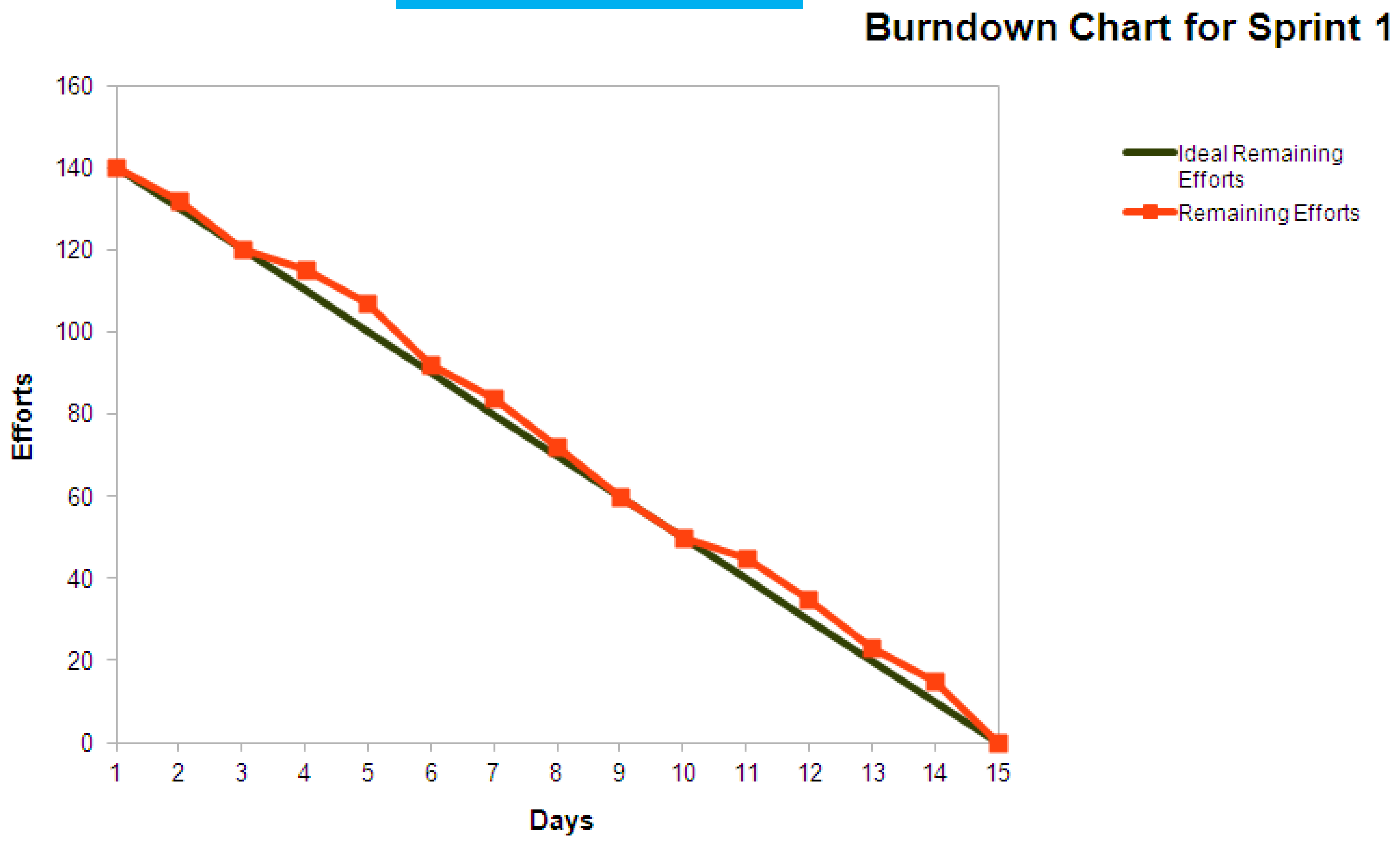
- MMF is basic minimum set feature of the product so that people can start using it
- MMF can be made of a single or combined of multiple user stories
- MMF has business value to the end user

Copyright 2013 Vedavit Project Solutions

Burn Chart

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

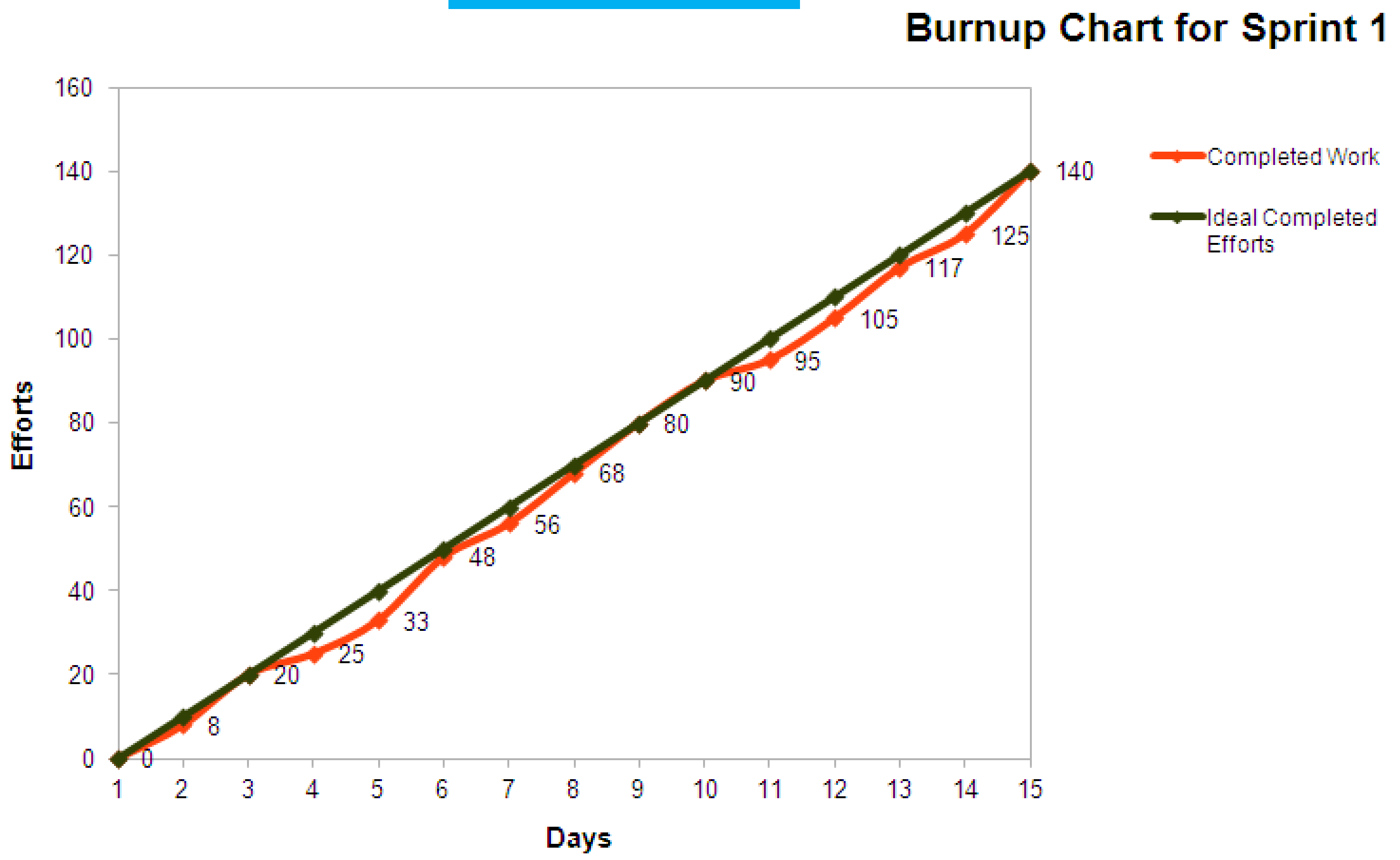
Burn Down Chart



Burn Chart

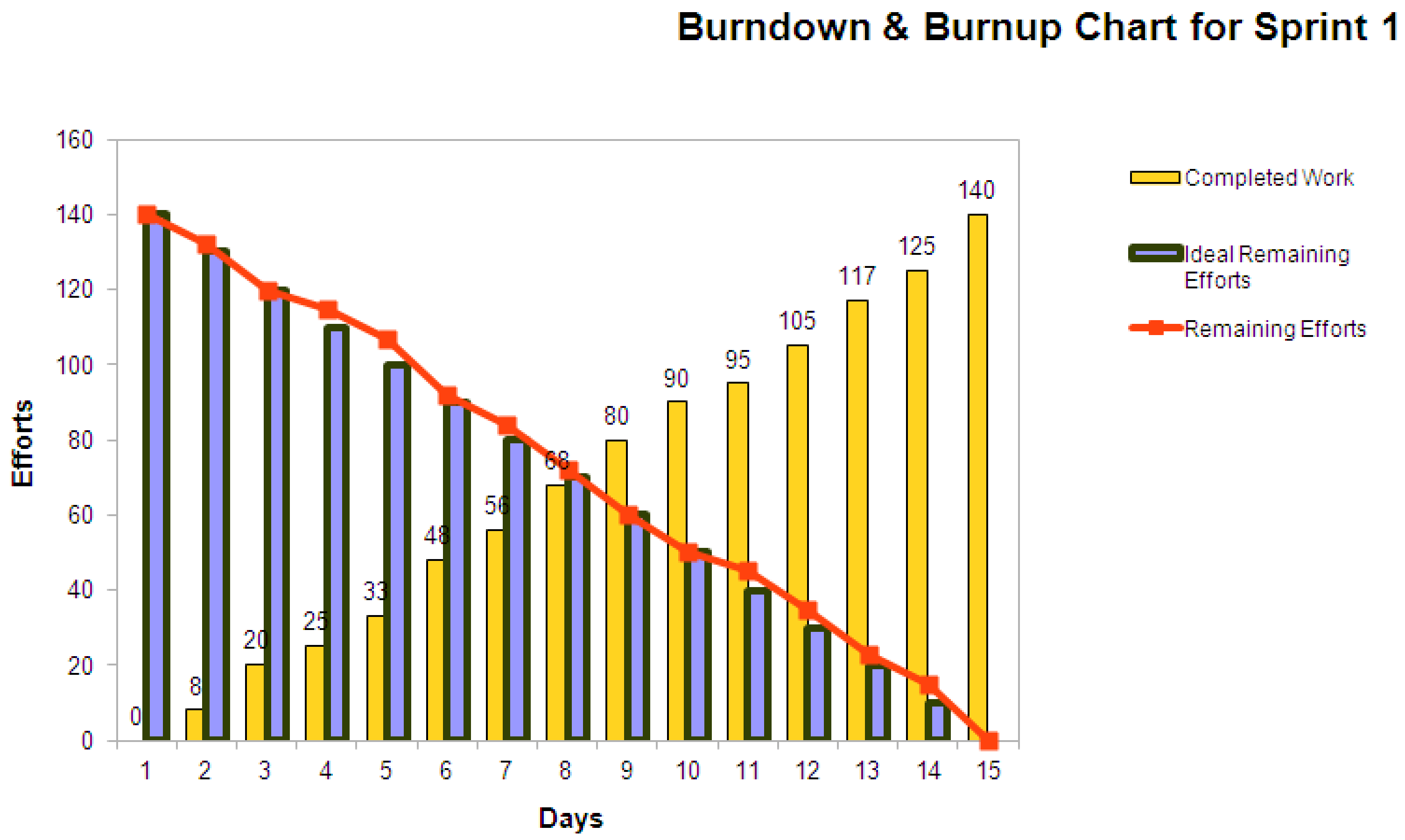
Tools & Techniques / Planning, Monitoring & Adapting/ Agile Project Tracking

Burn Up Chart



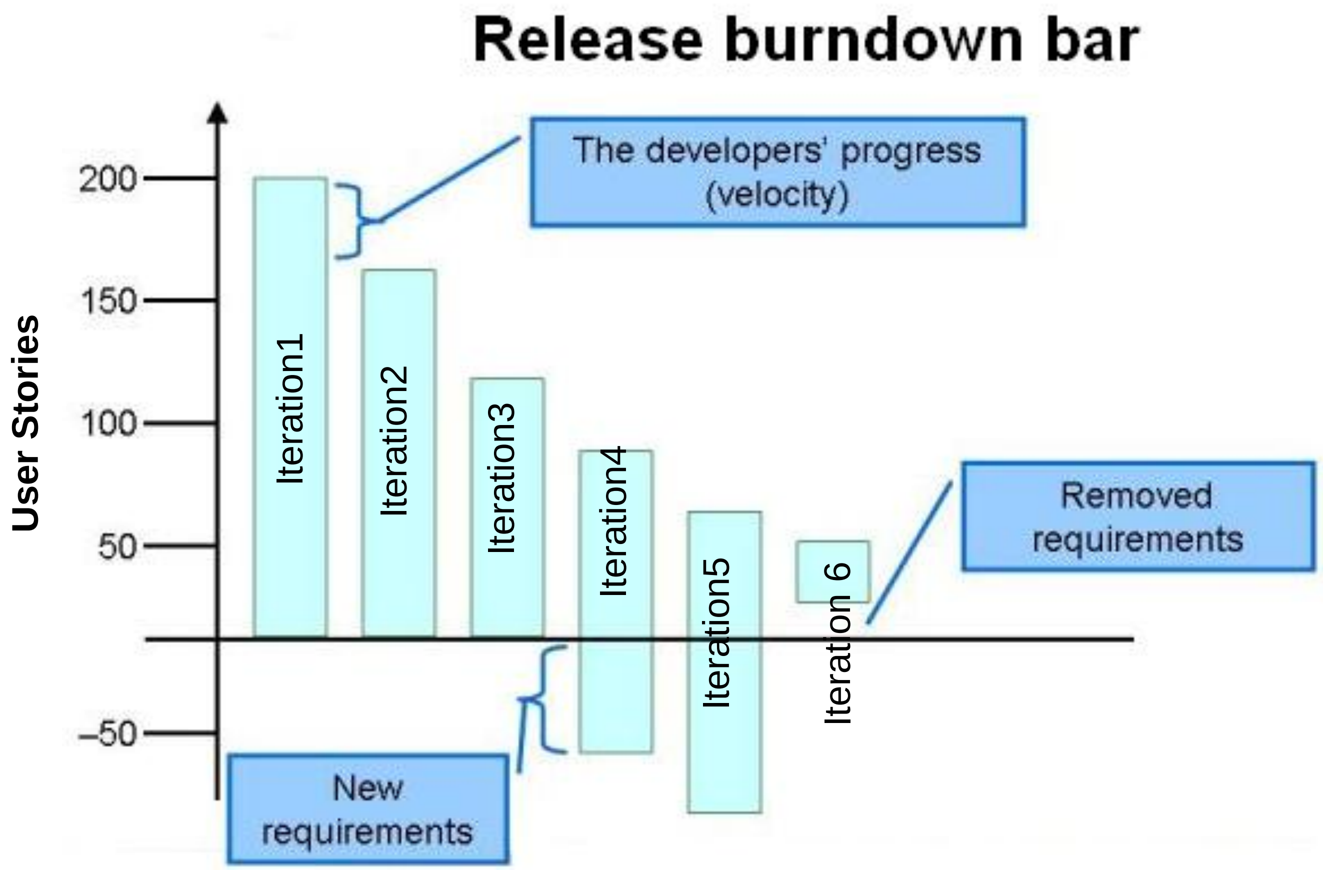
Burn chart

Burn-down & Burn-up combine chart



Burn Chart

Burn Down Bar Chart



Retrospectives

- 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

Copyright 2013 Vedavit Project Solutions

Retrospectives

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)

Copyright 2013 Vedavit Project Solutions

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 not it is worth to identify task to do quantitative estimation

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

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
- Les us assume that based on guess or some previous release teams knows that their velocity is 30 story points
- Total iterations required = 348 / 30 = **11 iterations**
- If iteration is of 2 weeks then **22 weeks** required to finish the work in product backlog

Copyright 2013 Vedavit Project Solutions

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

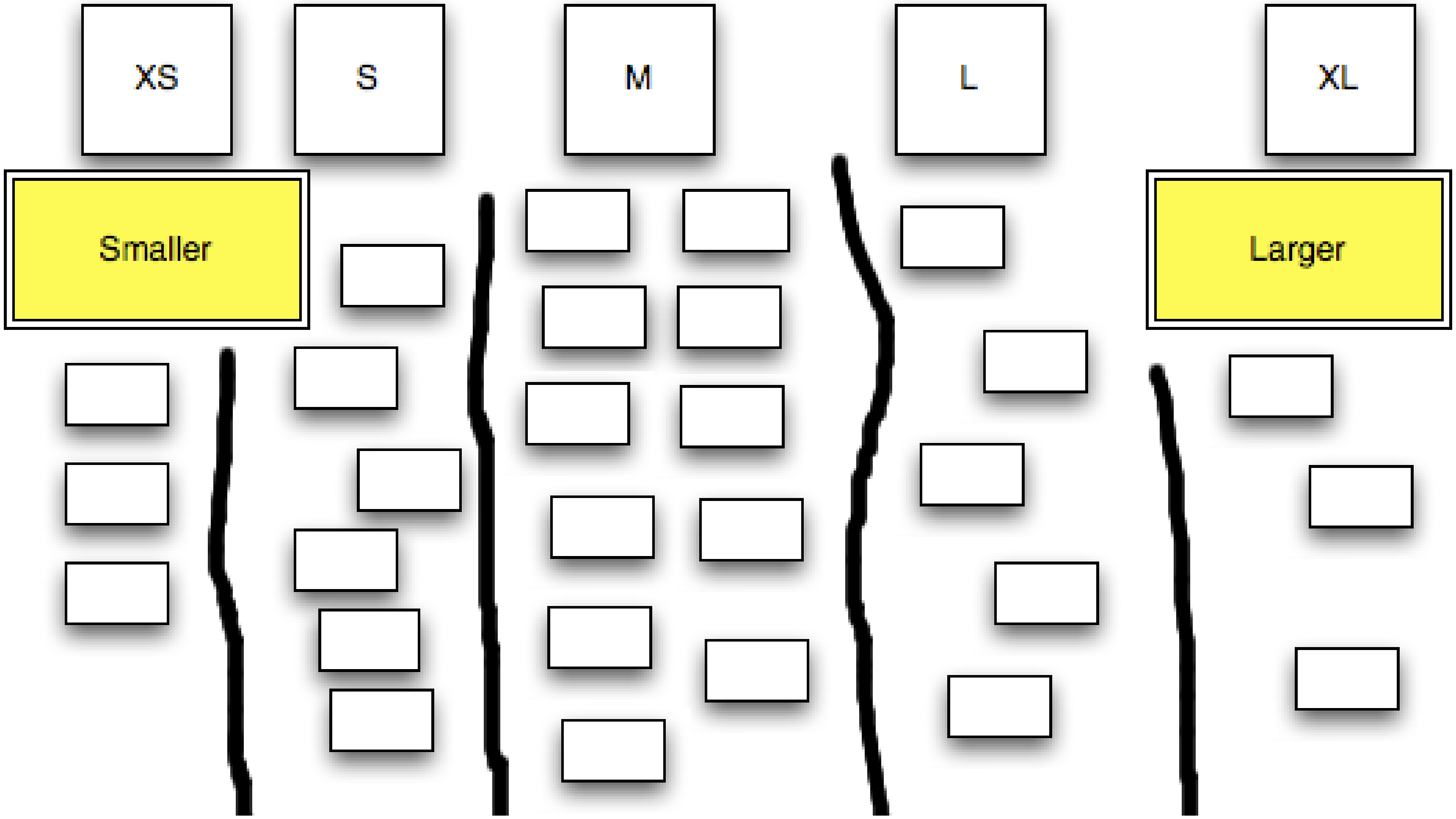
Copyright 2013 Vedavit Project Solutions

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
  - As may 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

Copyright 2013 Vedavit Project Solutions

Affinity Estimation



Copyright 2013 Vedavit Project Solutions

Metrics

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

Copyright 2013 Vedavit Project Solutions

Metrics

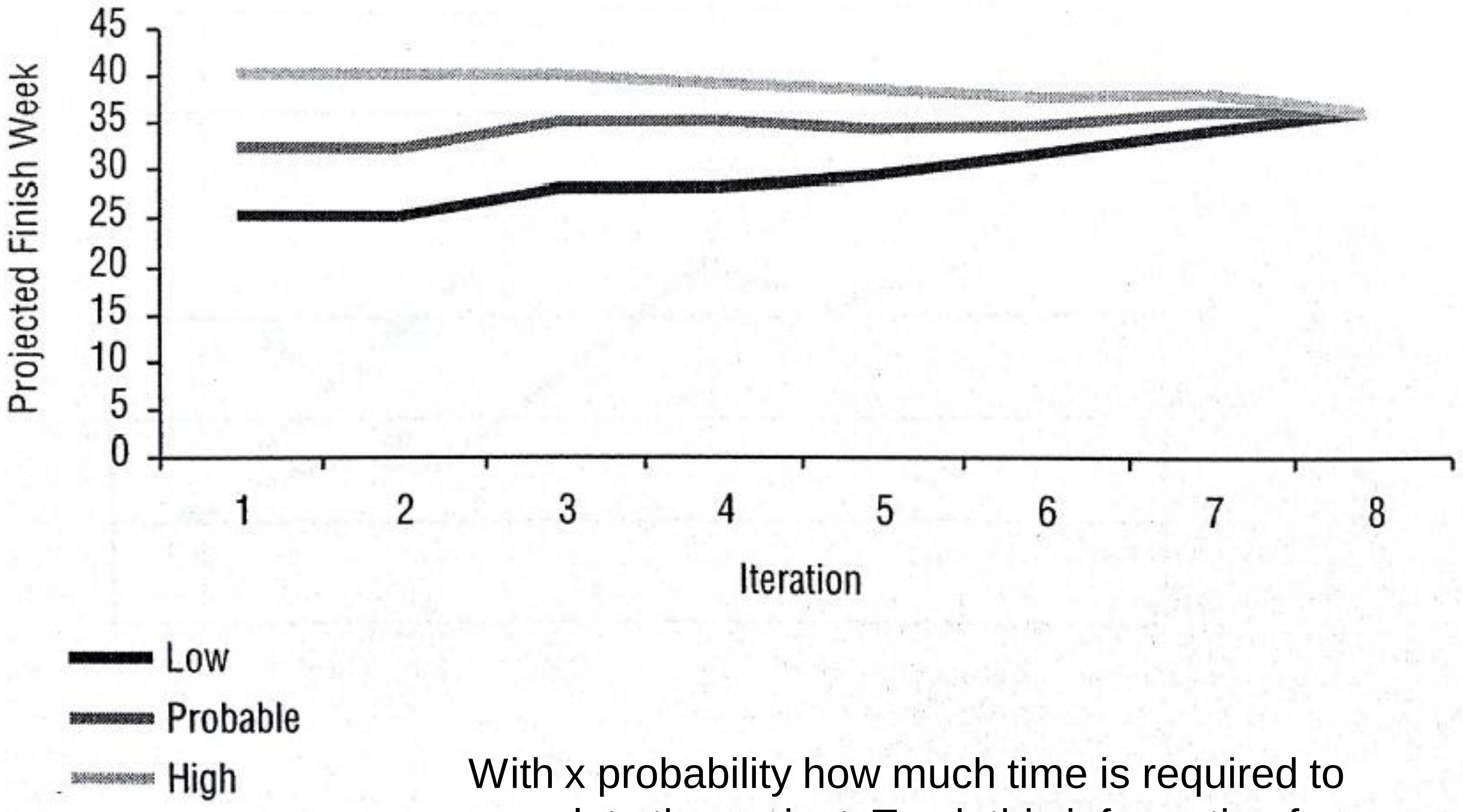
Escaped Defects

- 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

Copyright 2013 Vedavit Project Solutions

Metrics

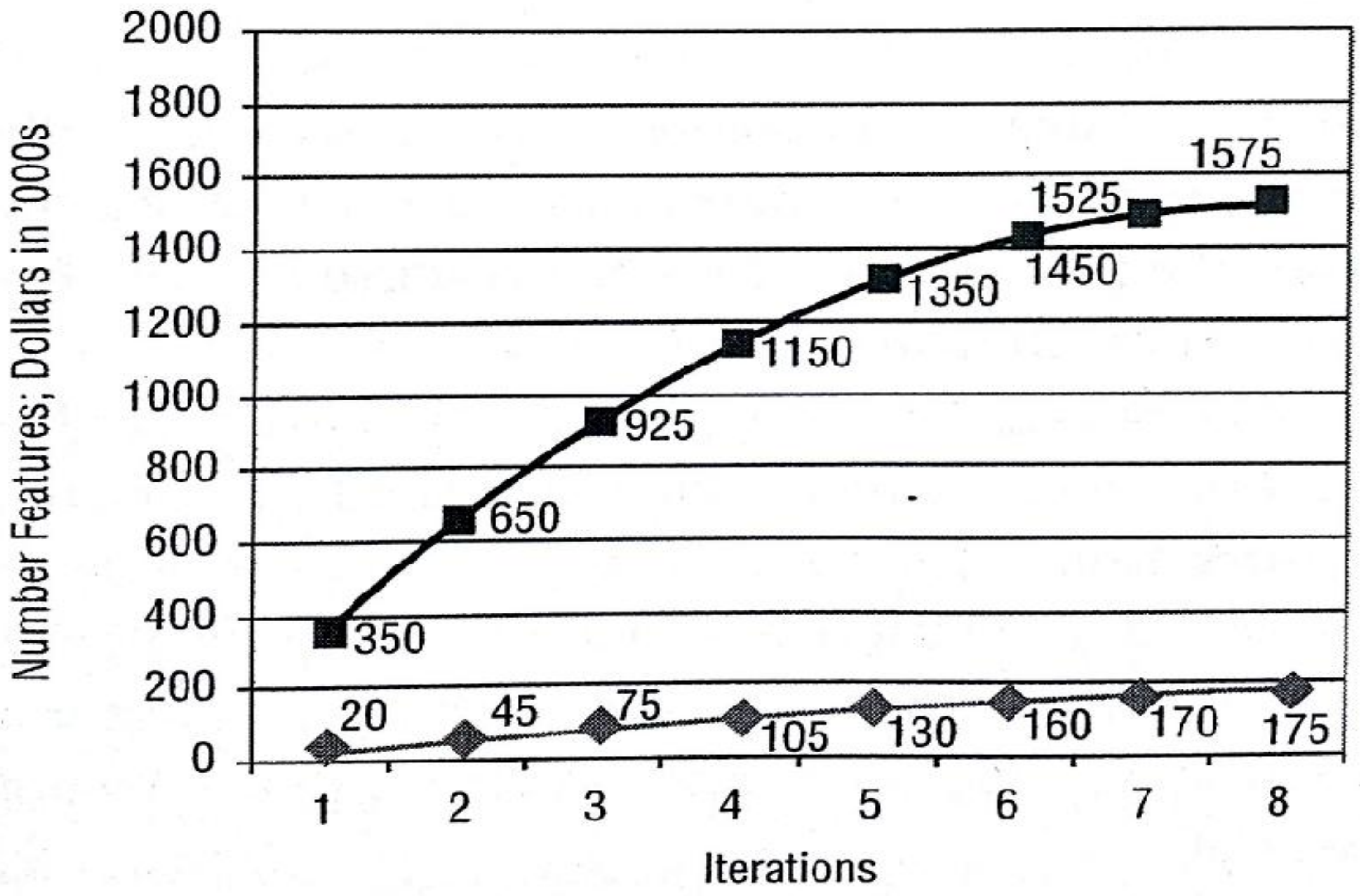
Projected Scheduled



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

Metrics

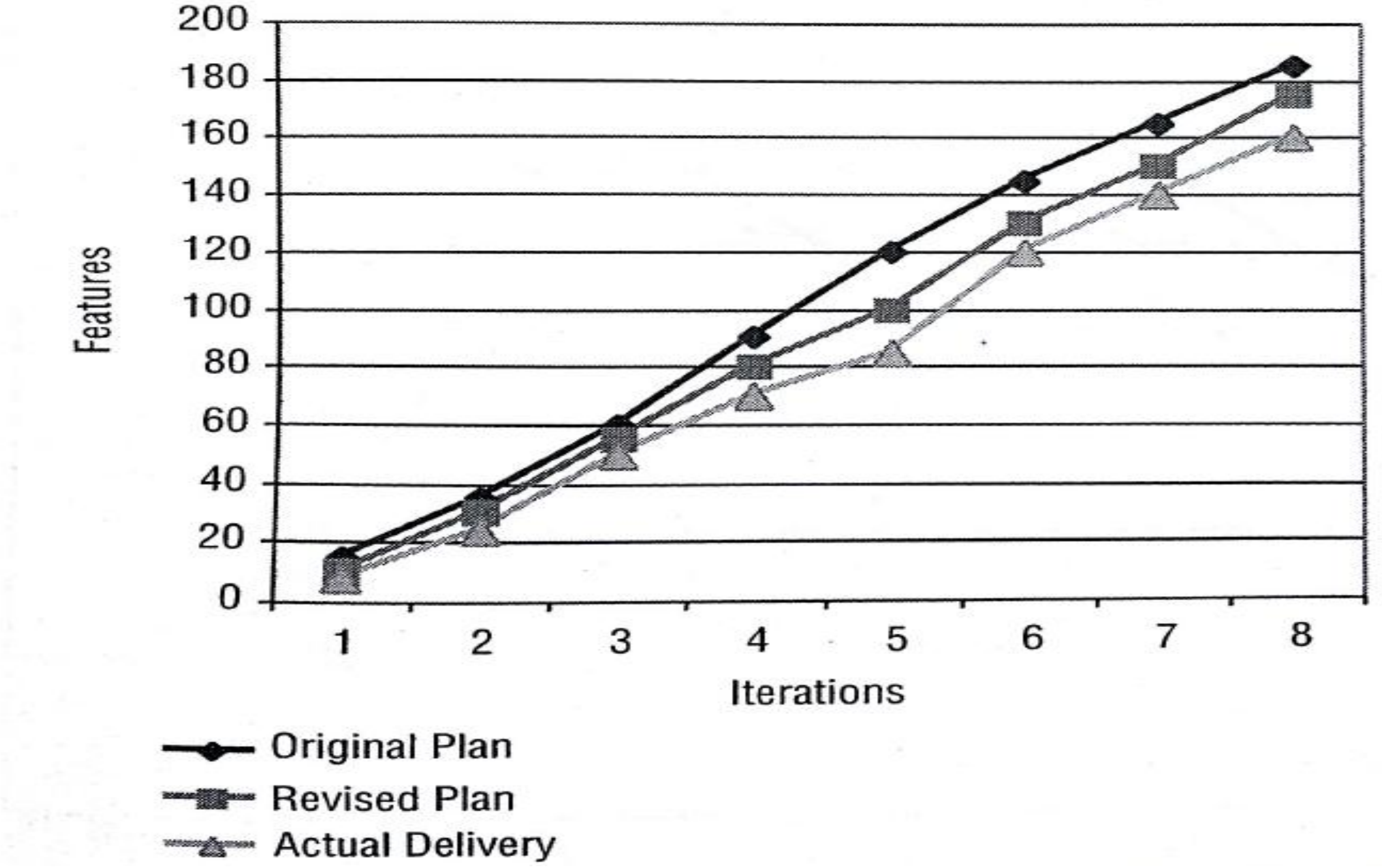
Features & Value Delivered



Copyright 2013 Vedavit Project Solutions

Metrics

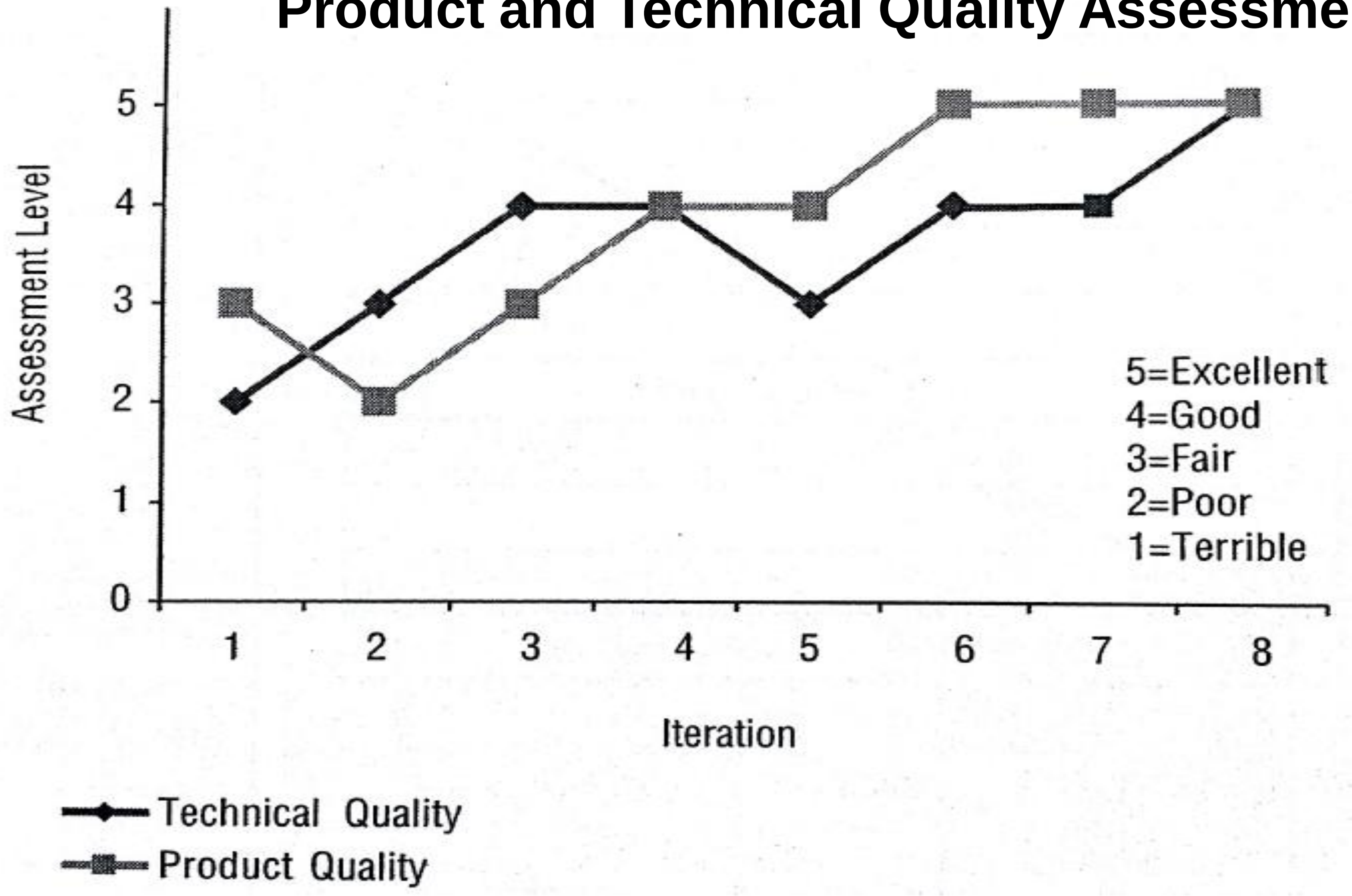
Delivery Performance



Copyright 2013 Vedavit Project Solutions

Metrics

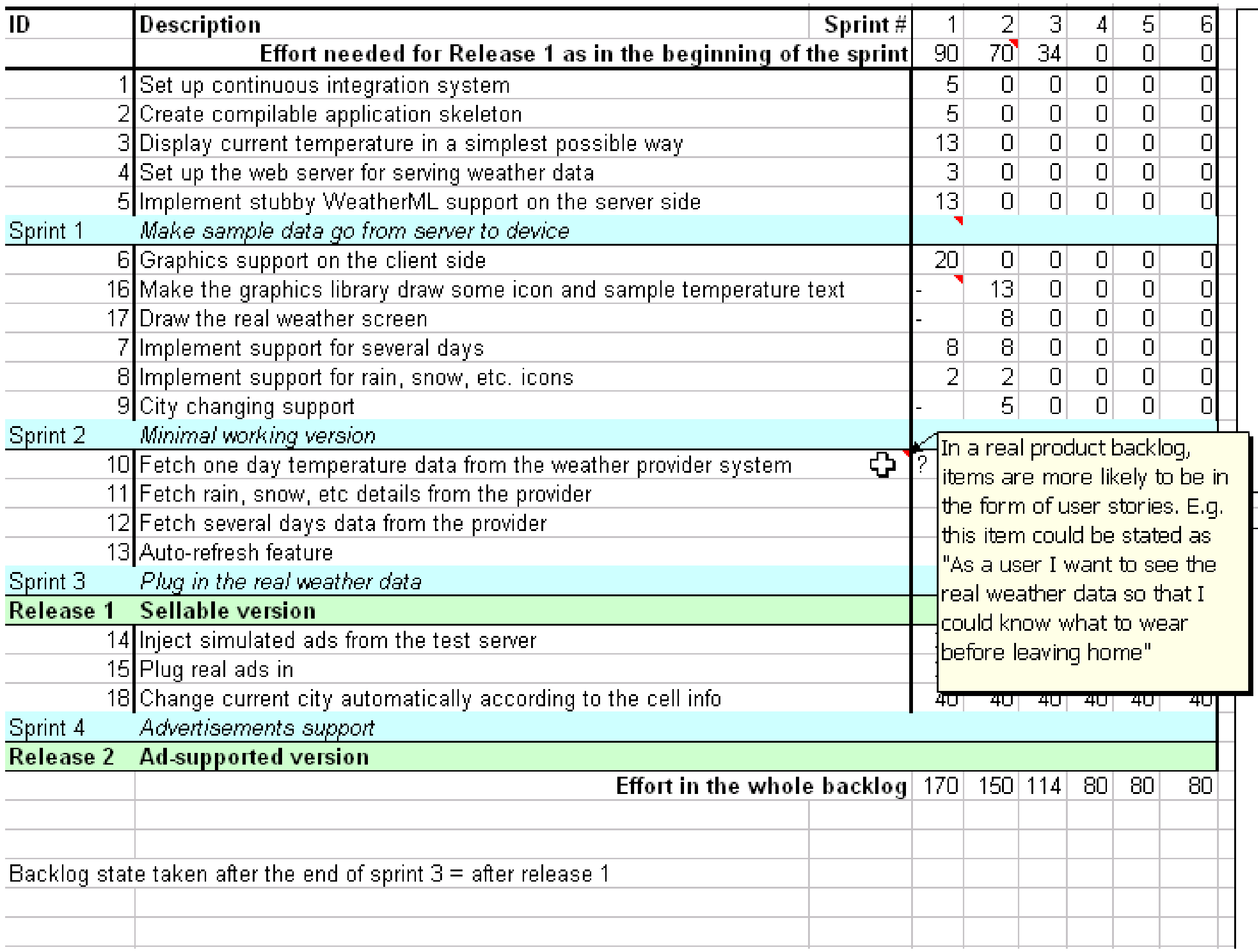
Product and Technical Quality Assessment



Copyright 2013 Vedavit Project Solutions

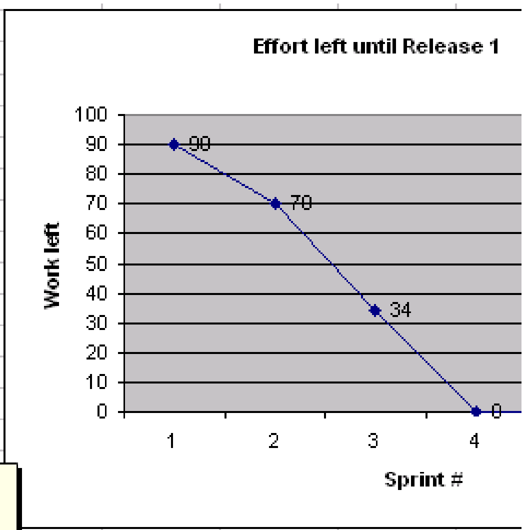
Backlog

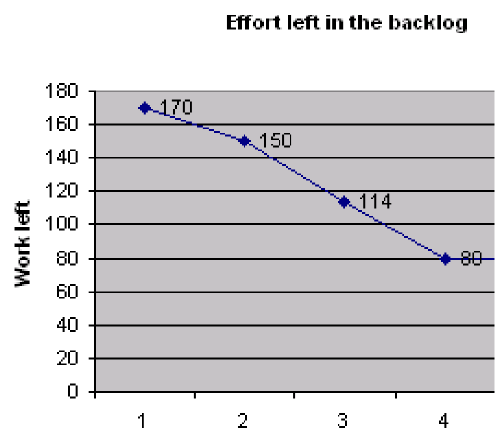
Release Backlog



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	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		40	40	40	40	40	40
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								

In a real product backlog, items are more likely to be in the form of user stories. E.g. this item could be stated as "As a user I want to see the real weather data so that I could know what to wear before leaving home"





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

Copyright 2013 Vedavit Project Solutions

Backlog

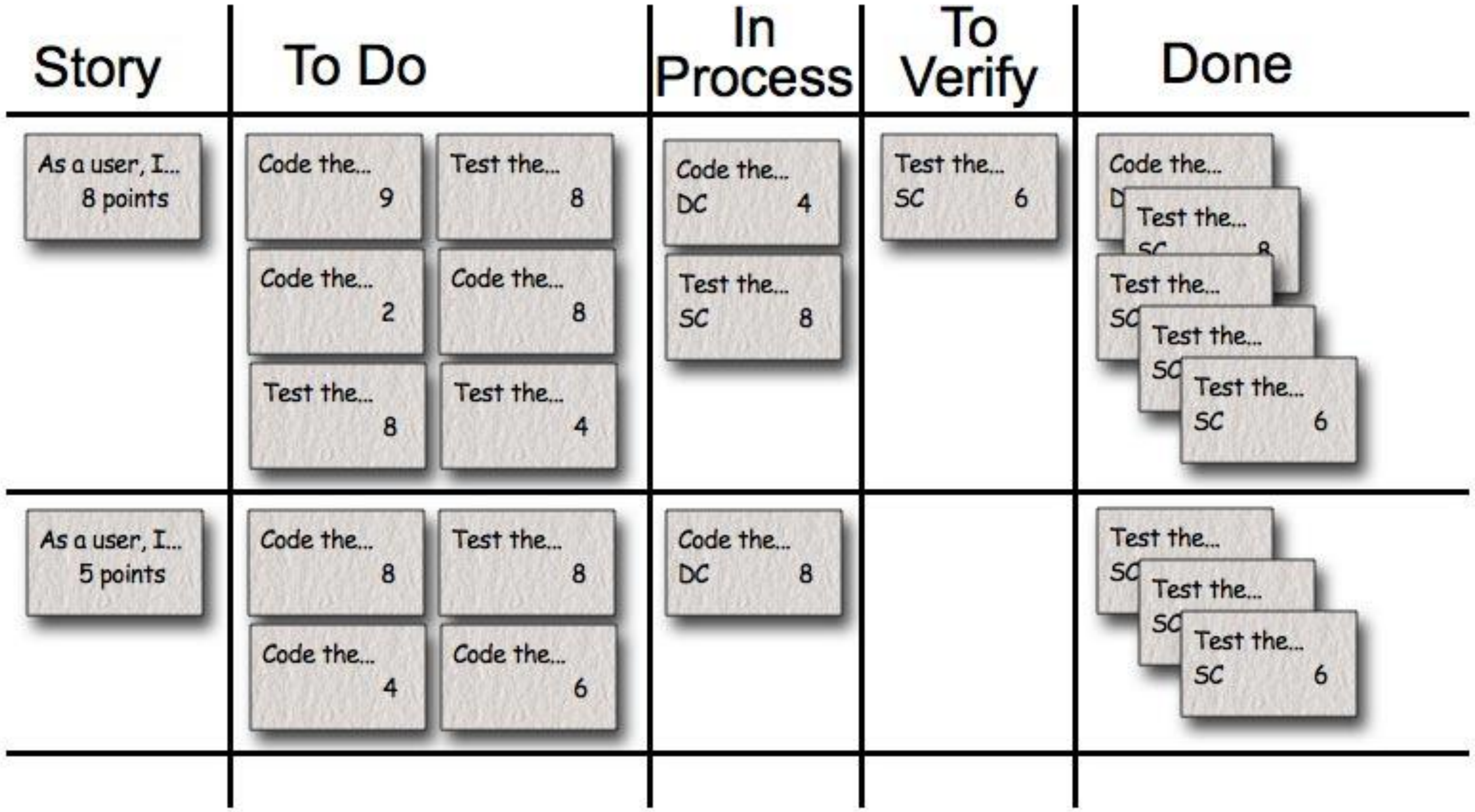
Sprint Backlog

- 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

Copyright 2013 Vedavit Project Solutions

Backlog

Sprint Backlog



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

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

Definition of Done (DoD)

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

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

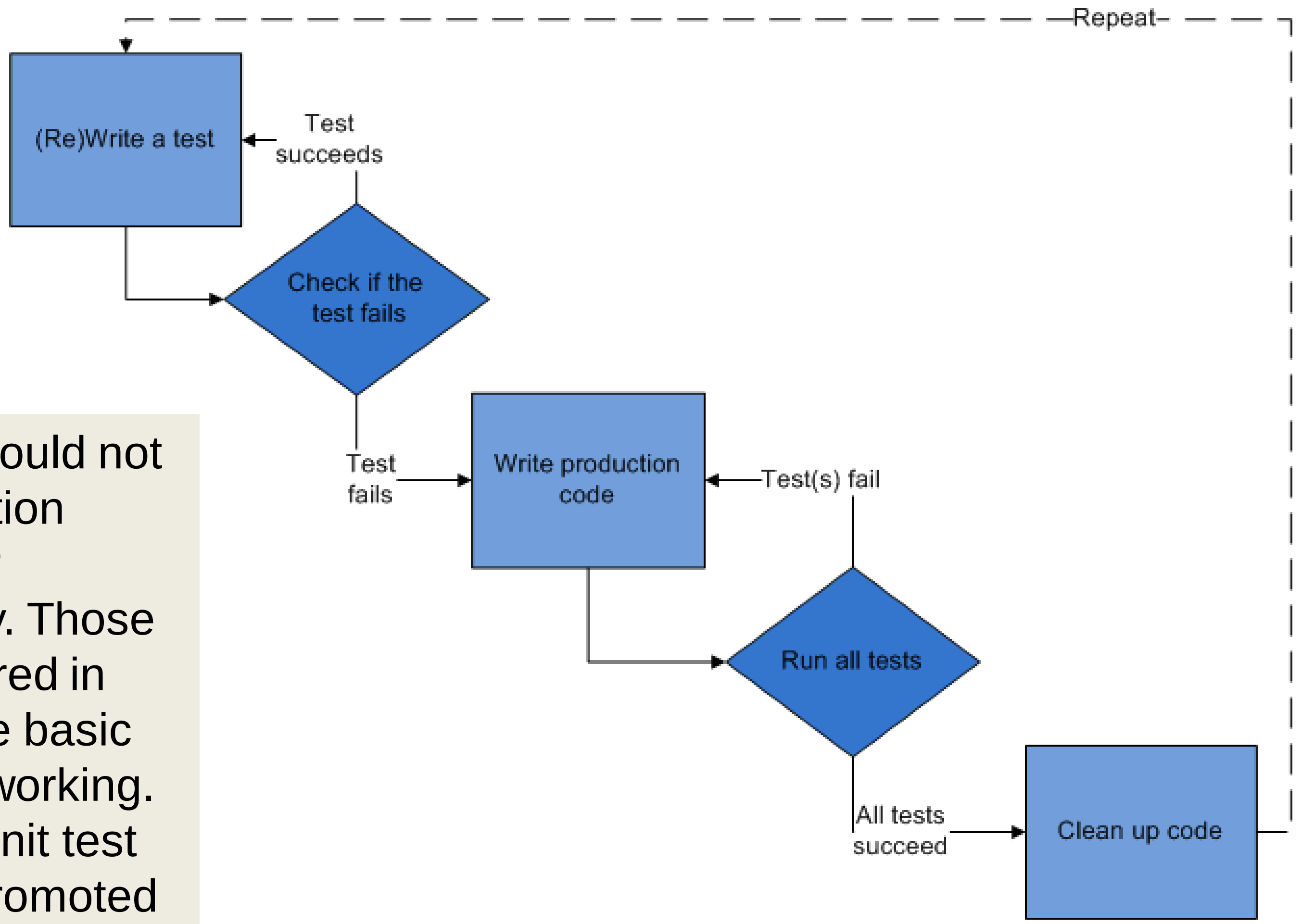
Definition of Done (DoD)

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

Copyright 2013 Vedavit Project Solutions

Test Driven Development (TDD)

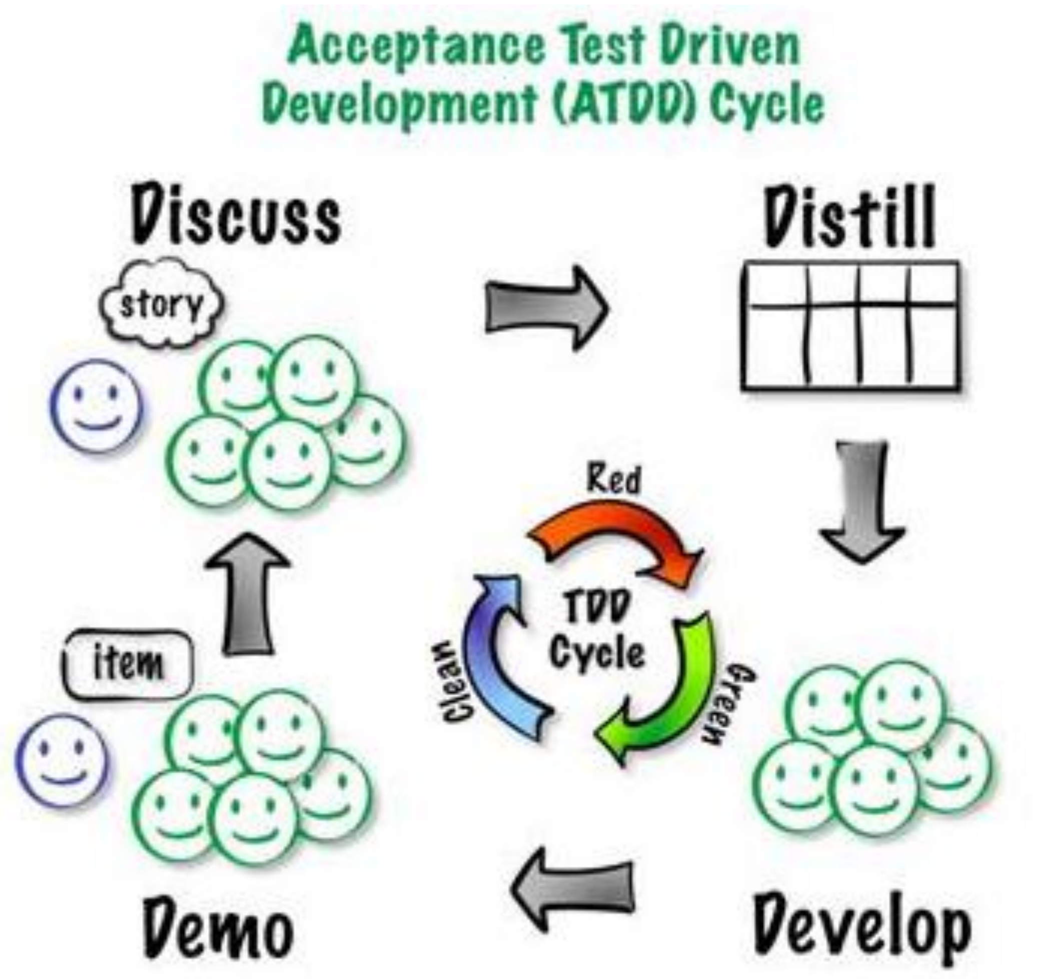


- Programmer should not consider exception cases and error handling initially. Those can be considered in next cycle, once basic functionality is working.
- If built passes unit test then only it is promoted to the next step as a candidate for acceptance test build

Copyright 2013 Vedavit Project Solutions

Acceptance Test Driven Development (ATDD)

- 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



Copyright 2013 Vedavit Project Solutions

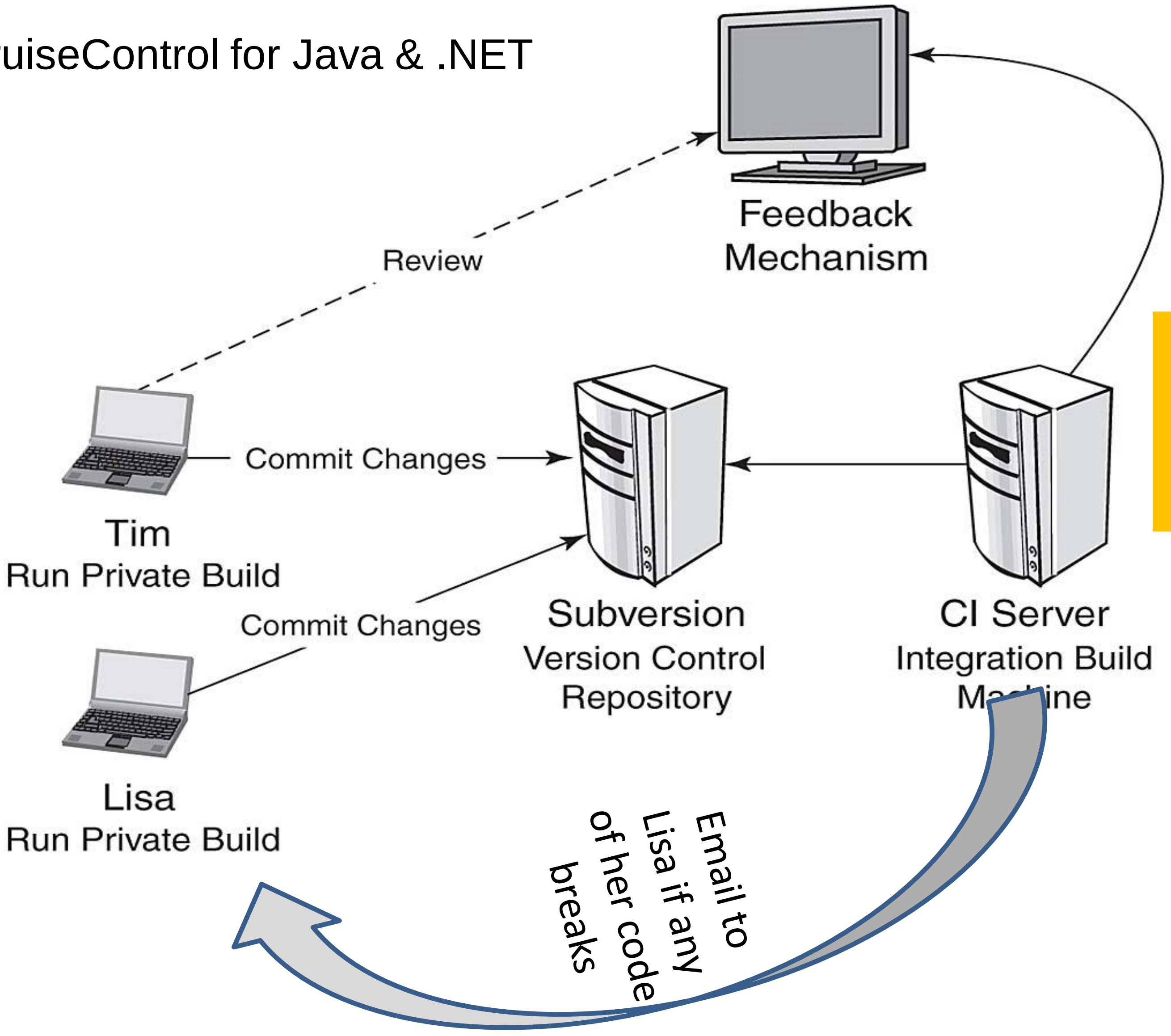
Exploratory Testing

- 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.

Copyright 2013 Vedavit Project Solutions

Continuous Integration

Tool: CruiseControl for Java & .NET



- Build should happen within 10 min, else change server
- No partial build

Copyright 2013 Vedavit Project Solutions

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.

Copyright 2013 Vedavit Project Solutions

Agile Team

- Agile team is smaller compare to Waterfall team size
- Scrum suggests team size of 7 ± 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

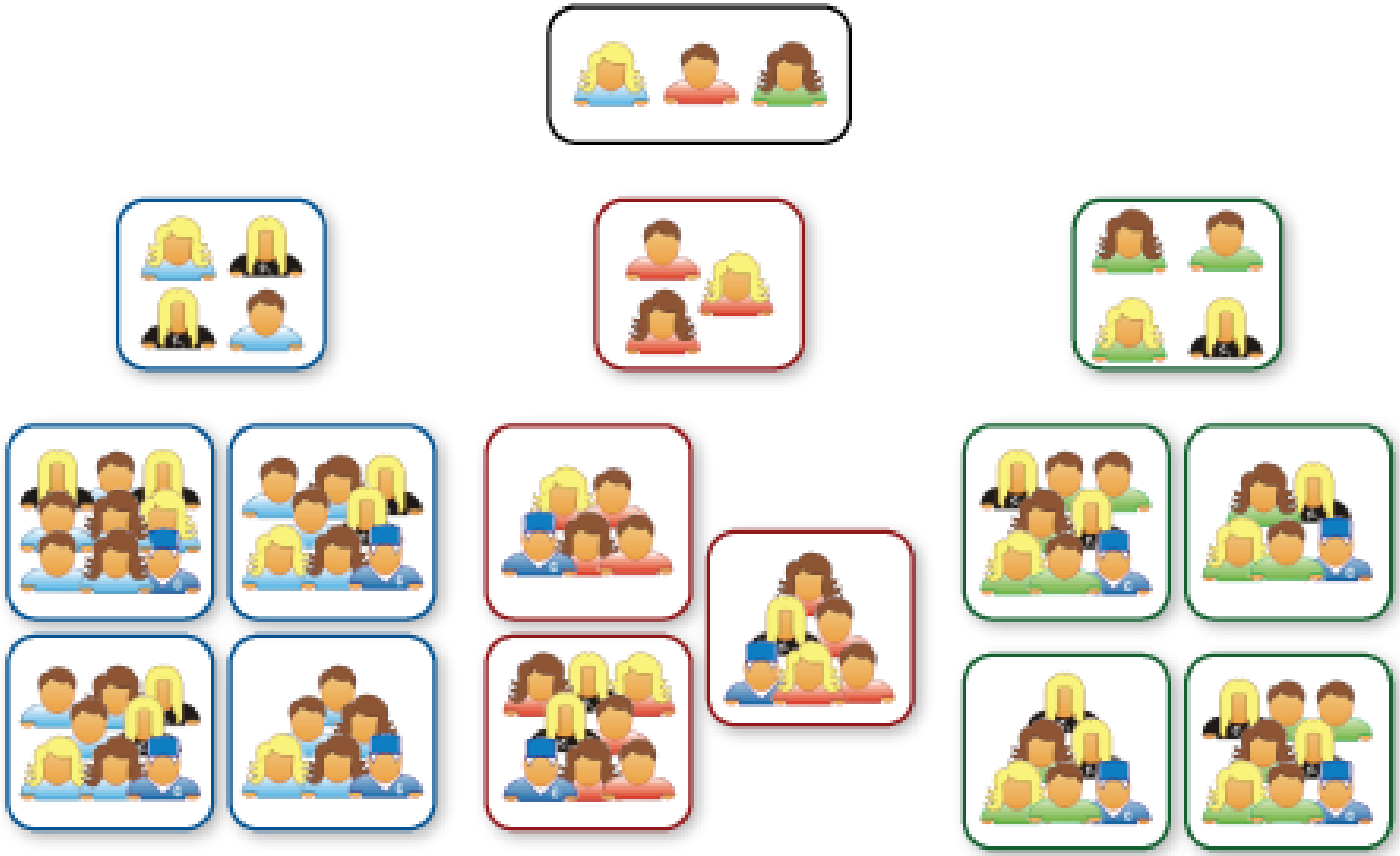
Copyright 2013 Vedavit Project Solutions

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?

Copyright 2013 Vedavit Project Solutions

Agile Team Scaling



Copyright 2013 Vedavit Project Solutions

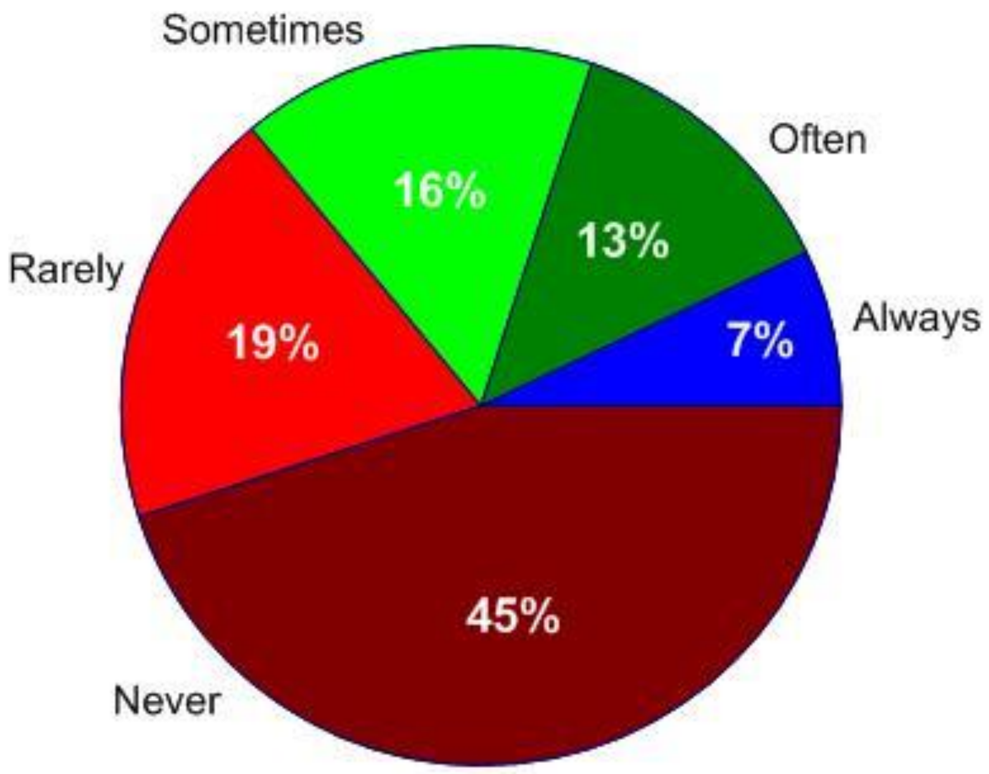
Type of Teams

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

Copyright 2013 Vedavit Project Solutions

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.

Copyright 2005-2006 Scott W. Ambler

Copyright 2013 Vedavit Project Solutions

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 3rd 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.

Copyright 2013 Vedavit Project Solutions

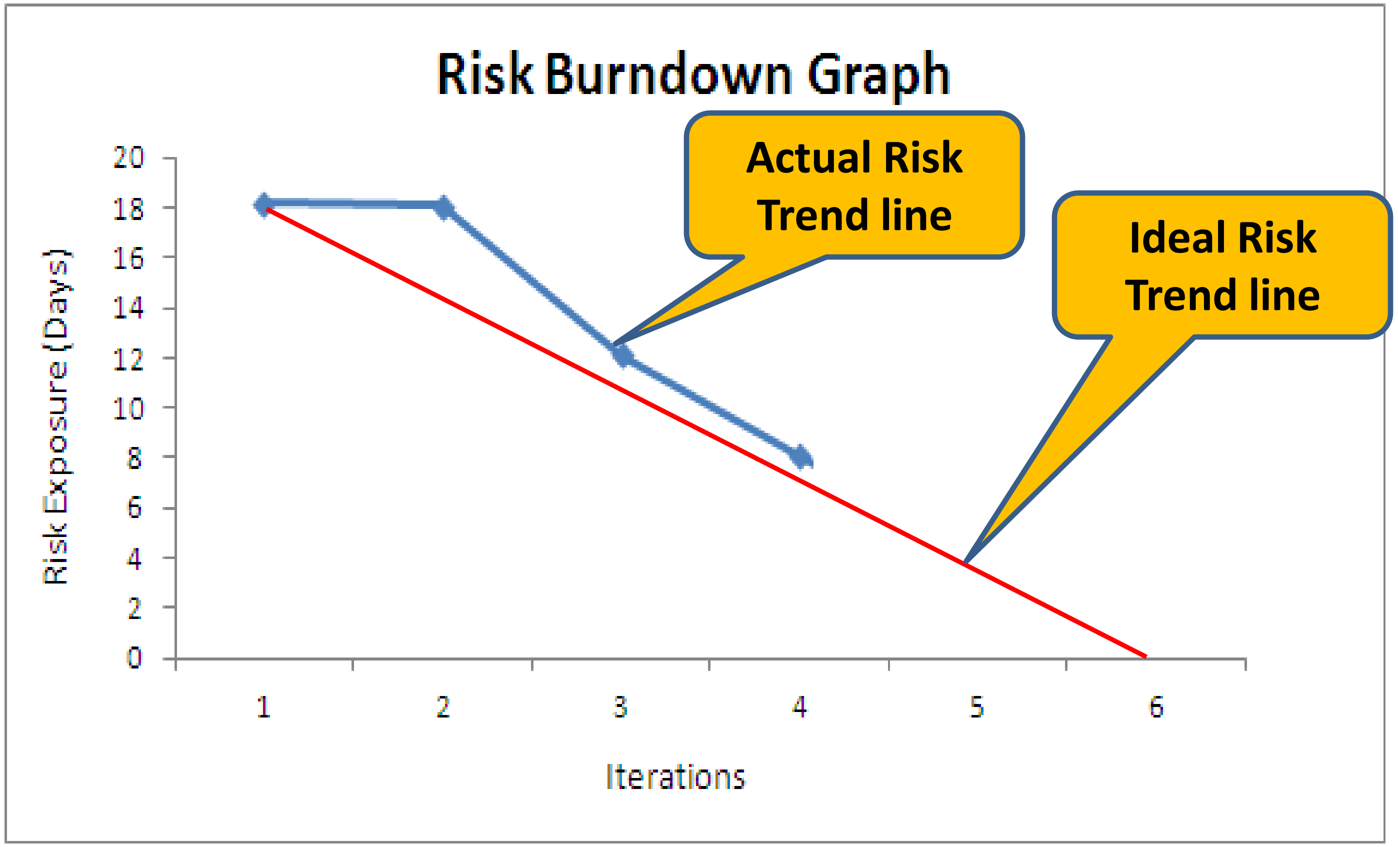
Risk Adjusted Backlog

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

Copyright 2013 Vedavit Project Solutions

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

Copyright 2013 Vedavit Project Solutions

State of Scrum

Survey Conducted by scrumalliance.org

ScrumAlliance.Org Survey in 2013

- 500 Participants from 70+ countries
- Majority of participants from USA, India, Canada, Austria, UK, Germany

Copyright 2013 Vedavit Project Solutions

Where are they now?

- 13% of those who started with waterfall now use Scrum exclusively
- Most ended up with a mix of waterfall and scrum
- 8% went back to Waterfall

Copyright 2013 Vedavit Project Solutions

Why Scrum?

- 41% Fulfilling customer needs
- 18% Meeting budget, time and scope constraints
- 9% Unknown
- 11% Adding new features and functionality
- 18% Completing projects that will drive innovation and market share
- 3% Others

Copyright 2013 Vedavit Project Solutions

State of Scrum

- 27% Frequently Used
- 25% Sometimes
- 13% Usually
- 15% Rarely
- 20% always

Copyright 2013 Vedavit Project Solutions

Who are using Scrum?

- 28% CSM
- 28% PMP
- 6% CSPO
- 3% CSP
- 3% PMI-ACP
- 1% CSD
- 1% PgMP

Copyright 2013 Vedavit Project Solutions

Which Industry Using?

- 41% IT
- 12% Finance
- 6% Government
- 6% Healthcare
- 5% Telecom
- 4% Construction
- 4% Education
- 2% Manufacturing
- 2% R&D
- 2% Retail
- 1$ Aerospace

Copyright 2013 Vedavit Project Solutions

Scrum Roles

- 41% projects have a dedicated ScrumMaster
- 24% projects have a dedicated Product Owner

Copyright 2013 Vedavit Project Solutions

Team Size

- 10% have <=3
- 71% have 4-9
- 19% have >=10

Copyright 2013 Vedavit Project Solutions

Sprint Duration

- 8% has 1 week
- 75% 2-6 weeks

Copyright 2013 Vedavit Project Solutions

Team Size

- 85% project has team size between 3-9

Copyright 2013 Vedavit Project Solutions

Recap

- What is Project?
- Why Agile?
  - Traditional v/s Agile Software Development
  - Agile Values and Principles
  - Key Benefits
  - Scrum Overview
  - Essence of Scrum
  - Scrum values
  - The key roles within SCRUM(Scrum Master, Product Owner , Agile Project Manager and Team)
  - The key artefacts within SCRUM
- Sprint Meetings
  - Understanding daily scrums
  - Understanding sprint review meetings

Copyright 2013 Vedavit Project Solutions

Recap Continue.1

- Estimation in Scrum
  - User stories and story points
  - Velocity? How to Compute Velocity
  - Planning poker for Product Back log size estimation
  - Estimation using Story Points
- Release Planning
  - Estimation of Product back Log Size
  - Estimating number of sprints
  - Deciding Release date
  - Creating Release Burn down
- Sprint Planning
  - Identifying Sprints Backlog Tasks from User stories
  - Creating Sprint Backlog
  - Creating and Updating Sprint Burn down

Copyright 2013 Vedavit Project Solutions

Recap Continue.2

- Tracking in scrum
  - Tracking releases, sprints
  - Accumulating Burn down
  - Measure sprint effectiveness

Copyright 2013 Vedavit Project Solutions

Hari P Thapliyal,

PMP, PMI-ACP, MCITP, Princ2 Practitioner, CSM, MBA, MCA, CIC, PGDFM

PMO Architect & Project Management Evangelist

Reach Me:

hari.prasad@vedavit-ps.com, hari.prasad@pm-learn.com

Skype: hari.thapliyal, YM: hari_thapliyal, Twitter: harithapliyal

Profile: http://in.linkedin.com/in/harithapliyal

Copyright 2013 Vedavit Project Solutions