PMI Project Management Institute

GOLDENSOURCE

# Scrum: An Agile Method of Software Development

Presented to PMI Mumbai Chapter

May 10, 2009

Sabu Mohamed

MUMBAI PMI

*Making project management indispensable for business results.®*

# About the presenter

- Sabu Mohamed, B Tech., MS., PMP.
  - Head Application Development at GoldenSource, a software product company that offers EDM platform for the securities and investment management industry.
  - 20+ years in various management and leadership positions in public sector (HSCL, NHPC), Govt. Of India (CBI, Indian Railways), and private sector organizations.

# Presentation Summary

- Scrum an agile method of software development with simple set of practices and rules encompassing **transparency**, **frequent inspection,** and **adaptation**, deals with the inherent complexities of software development better.
- This presentation will cover a case study of a scrum implementation, transitioning scrum in enterprise, and discuss about project management in scrum.

# Agenda

- Evolution of agile methods
- GoldenSource and scrum
  - Results
- What is scrum?
  - Roles, Terms and Artifacts of Scrum.
  - Transition @ GoldenSource
  - Lessons learned
  - Release planning
- The enterprise and scrum - Discussion
- Project management and scrum – Discussion.

# Waterfall Model: The Last Few Decades

1970's

1970 - Winston Royce:
Waterfall description -
Ironi

1976 – H
hum
capa
man
contr

1980's

1984 - Carolyn Wong: *"Software development is a complex continuous, iterative and repetitive process. The water fall model does not reflect this complexity"*

1986 – F Brooks: *The assumption that one can specify a satisfactory system in advance, have it built and install it is fundamentally wrong.*

1986 – David Parnas and Paul Clemens: *"Wa*
*many reasons.*

1988 – DOD-STD-2167 A – 2167 that *required s*
*waterfall model was amended - for life cy*

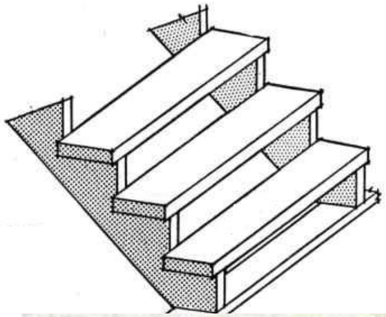
1994 - DOD was still experiencing failures. Consequently, Mil-Std-498 replaced 2167A - "Removing the Waterfall Bias. - Describes software development in one or more incremental builds.

1998 – Standish Group report (CHAOS – Charting the Seas of Information Technology) – analyzed 23000 projects – top reason for failure was waterfall practices.

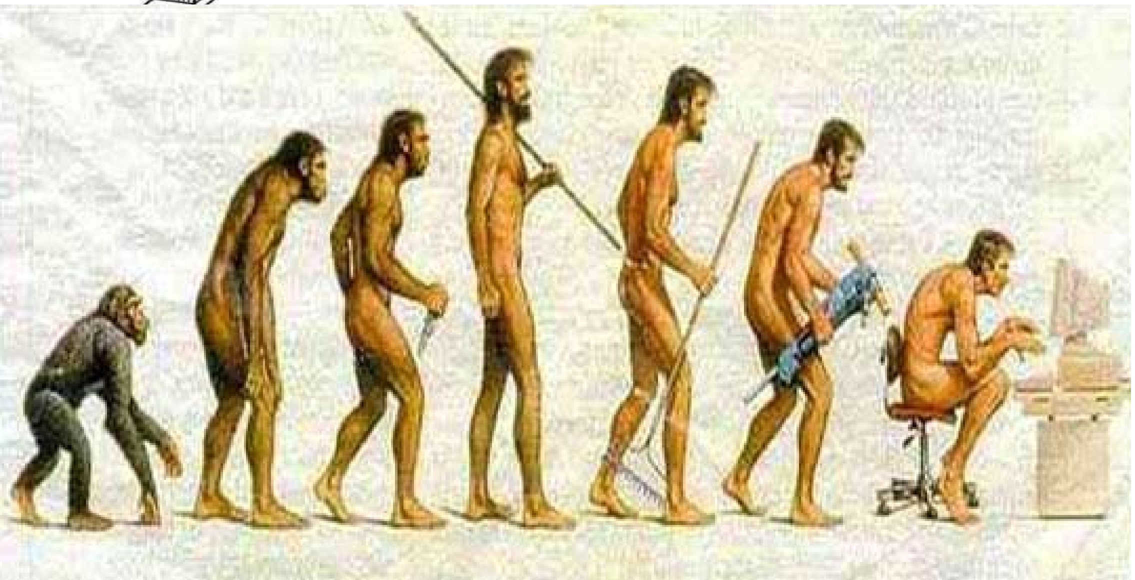
2000 – DoD replaced 498 with 5000.2 -Adopting evolutionary acquisition practices.

# Alternative - Agile

- Making informed adjustments on feedback
- Corner Stones:
  - Iterative
  - Incremental
  - Evolutionary



- In Line With Reality
  - Learning
  - Innovation
  - Change



# Agile

Pre 1970's

1930s- \
Do

1940's –
pro

1950's: X15 Hypersonic .
(Non Software)

1960's: NASA's project
Mercury (short half c
iterations)

1972- IBM FSD - Command & Control System of Trident submarine (4 iterations of 6 months each).

*- TRW Ballistic missile defense system (100 Mln $ - 5 iterations)*

*1975 - FSD - LAMPS (Helicopter to Ship weapon system) – 4 year (45 time boxed itera*

*1977 -80: NASA*
*primary ava*
*(17 iteratio*

1980's – Prototyping

1982 –100 Mln $ Military (
Control Project (IBM

1985 – T Gilb – recommen
delivery of useful res

1985- Barry Boehm: *"A Sp*
*Software Developme*
iterations.

1987 – TRW – Command
(time boxed iterations
– RUP.

*1988 – Tom Gilb –Principl*
*Engineering Manage*

## 1990 - Present

*1990's* – Paradigm shift from a preliminary major specification stage – to an evolutionary analysis approach.

*1994* – Defense Science Board Task Force – DOD - *to manage programs using iterative, evolutionary development – with rapid deployment of initial functional capability.*

*1994- SCRUM -Jeff Sutherland and Ken Schwaber - 30 day iterations based on Shashimi and a Scrum 1986 – Non Software Products – Honda, Canon,& Fujitsu.*

*- Later Refined in 1999*

*1994 – RAD – defined standards later became DSDM*

*1995 – Microsoft – Daily Build and Smoke test.*

*- Rational Unified Process (Kruchten & W Royce)*

*1996 – XP Practices matured - emphasis on communication, simplicity, and testing (Kent Beck)*

*1997- Jeff De Luca -described FDD an iterative process*

*2001- Agile Alliance and the Agile methods.*

# The Agile Manifesto

| | | |
|---|---|---|
| Individuals and interactions | over | Process and tools |
| Working software | over | Comprehensive documentation |
| Customer collaboration | over | Contract negotiation |
| Responding to change | over | Following a plan |

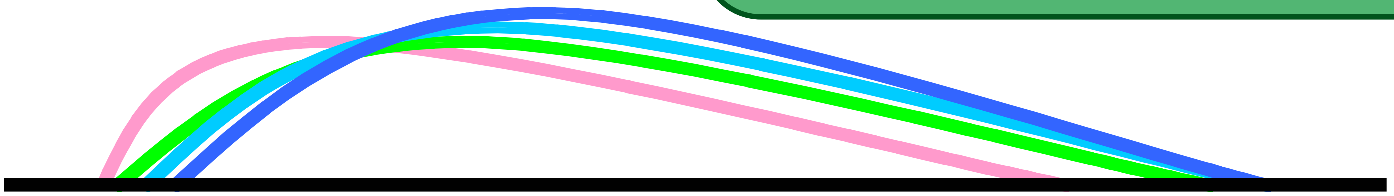
*Source: www.agilemanifesto.org*

# Waterfall Vs Agile

Requirements | Design | Code | Test

Rather than doing all of one thing at a time...

...Agile teams do a little of everything all the time

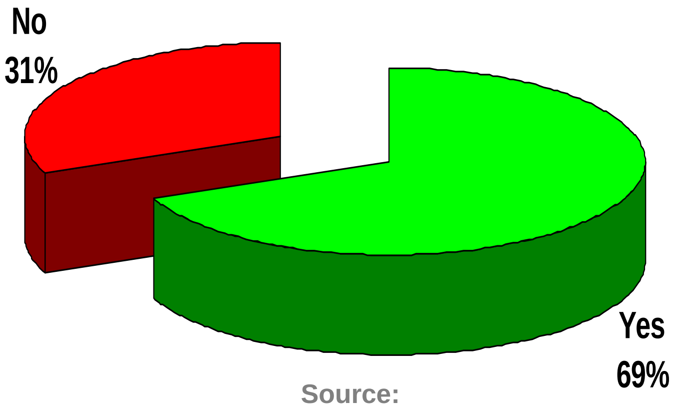


Source: “The New New Product Development Game” by Takeuchi and Nonaka. *Harvard Business Review,* January 1986.

# Agile Adoption & Drivers

- Changing Requirements
- Accelerate Time to Market
- Increase Productivity
- Enhance Software Quality

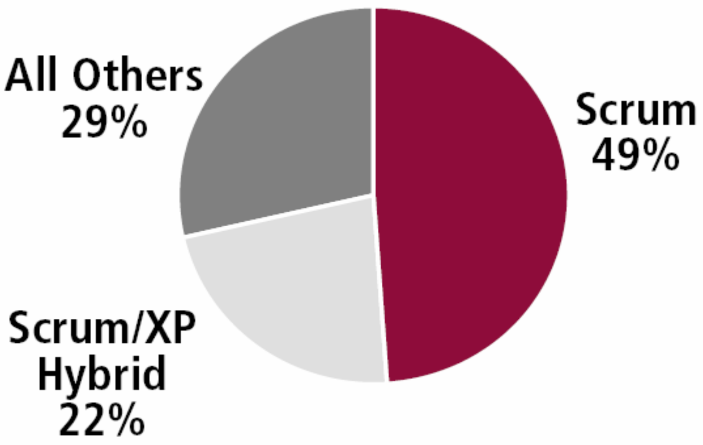
Source:
Version One Inc



Source:
www.ambysoft.com

# Popular Agile Methods

Which Agile methodology do you follow most closely?



| | |
|---|---|
| Scrum | 49.1% |
| Scrum/XP Hybrid | 22.3% |
| Extreme Programming (XP) | 8.0% |
| Custom/Hybrid | 5.3% |
| Don't Know | 3.7% |
| Agile Unified Process (AgileUP) | 2.2% |
| Other | 2.2% |
| Feature-Driven Development (FDD) | 2.1% |
| Lean Development | 1.9% |
| Dynamic Systems Development Method (DSDM) | 1.4% |
| OpenUP | 0.6% |
| Agile Modeling | 0.6% |
| Crystal | 0.5% |

3rd Annual Survey: 2008

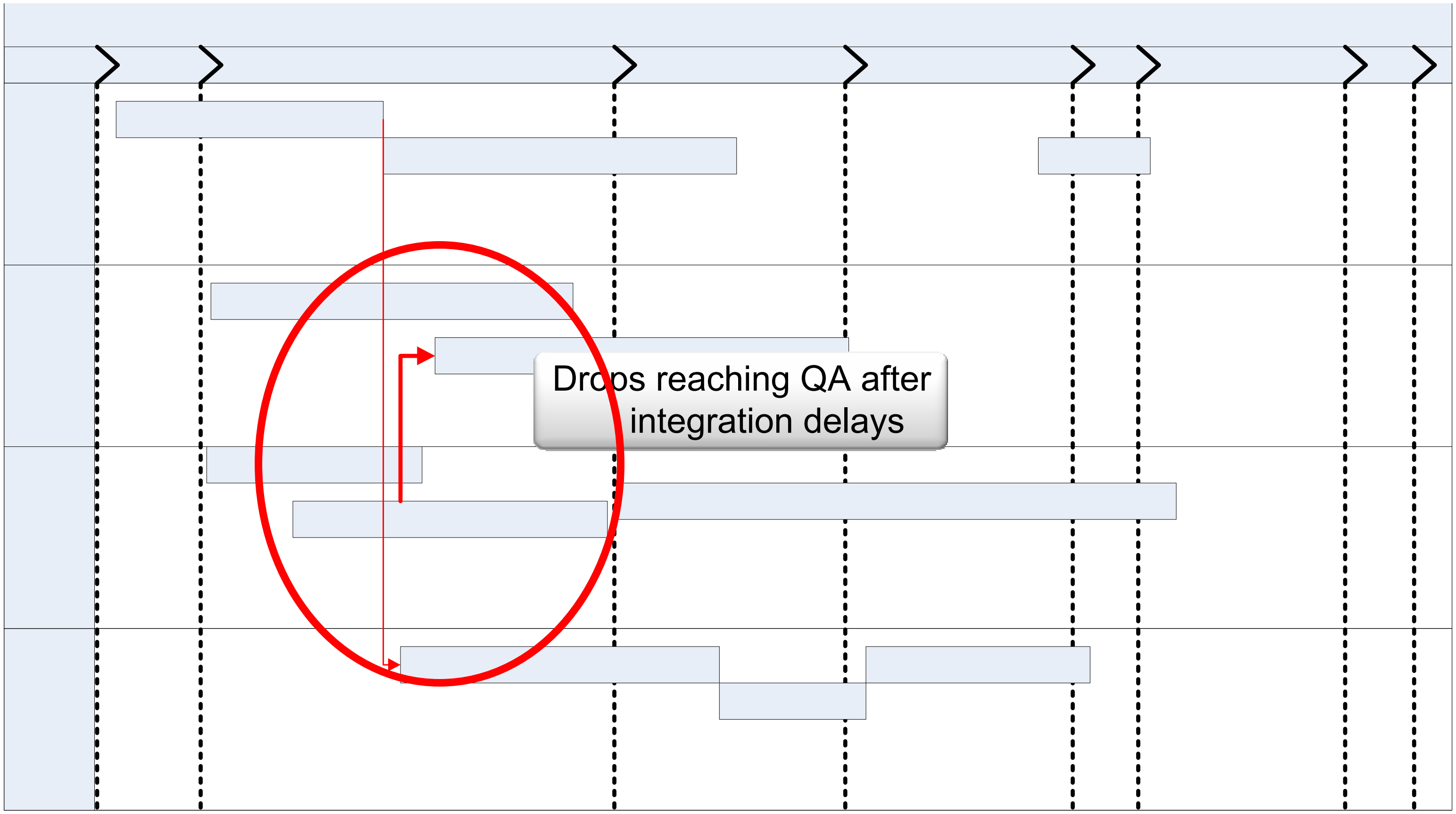
"The State of Agile Development"

Conducted: June-July, 2008

-VersionOne, Inc

# Scrum @ GoldenSource

# Why ?

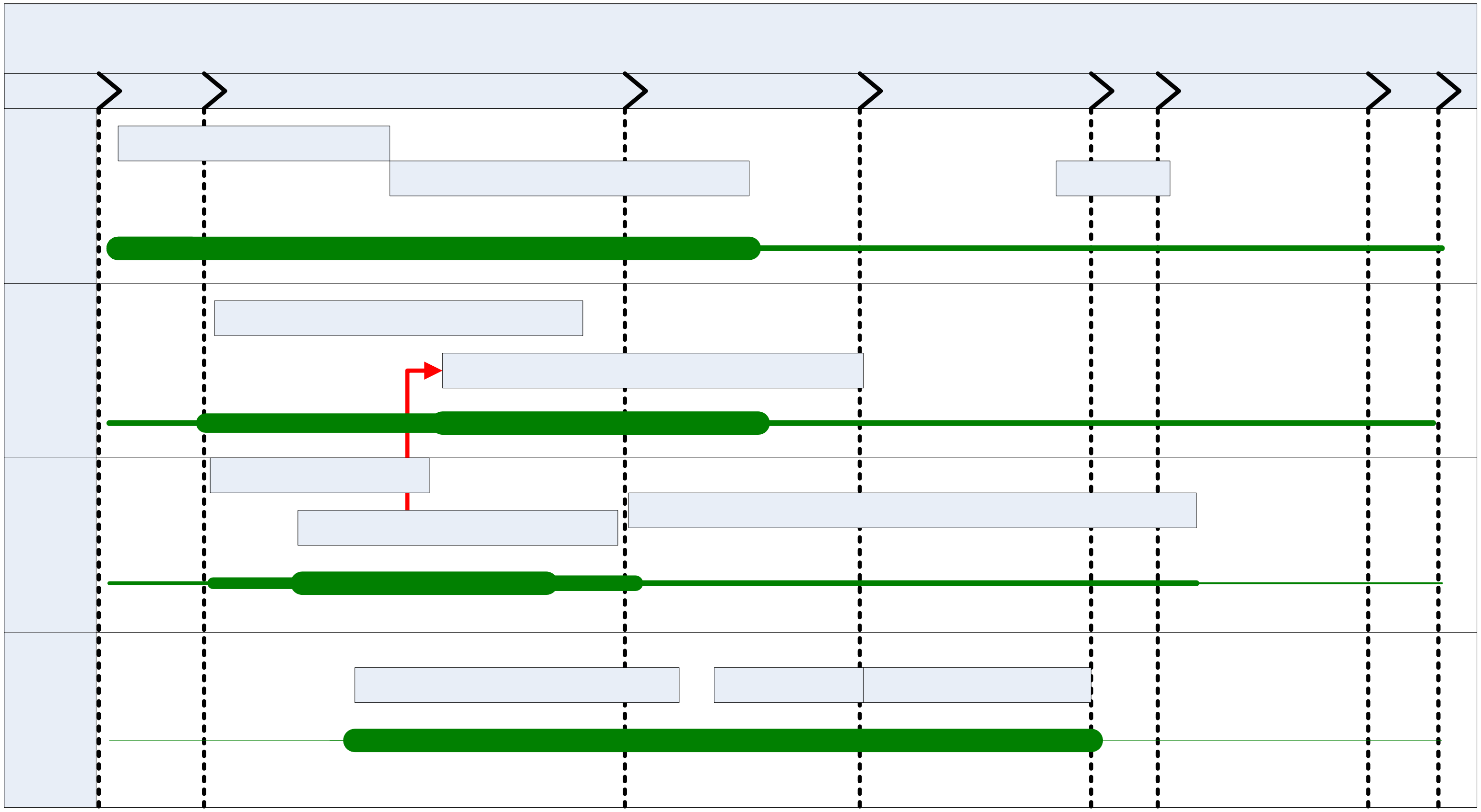


Drops reaching QA after integration delays

Integration Delays for QA

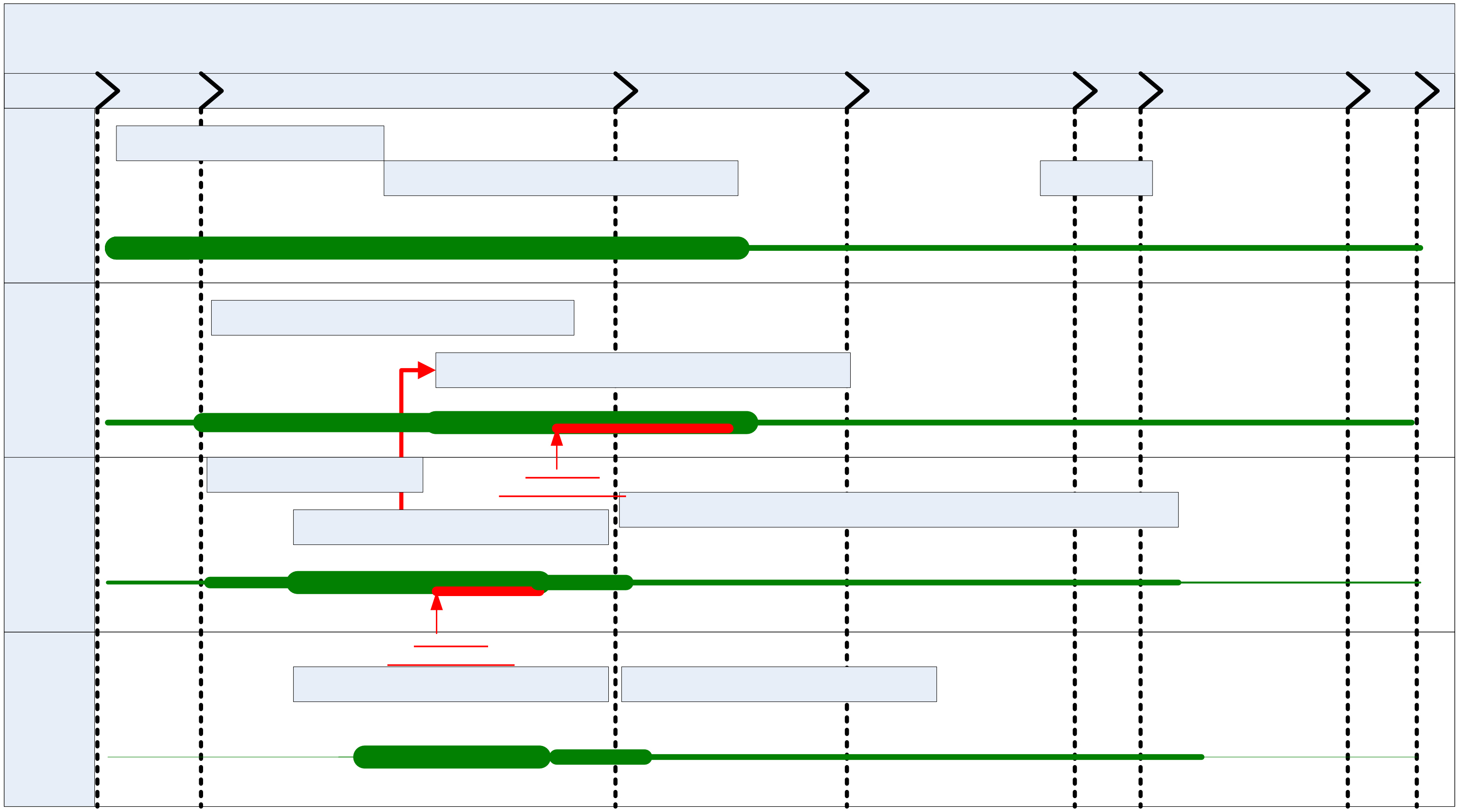
Start FSWT

# Why ?



Ineffective Utilization of Resources

# Why?



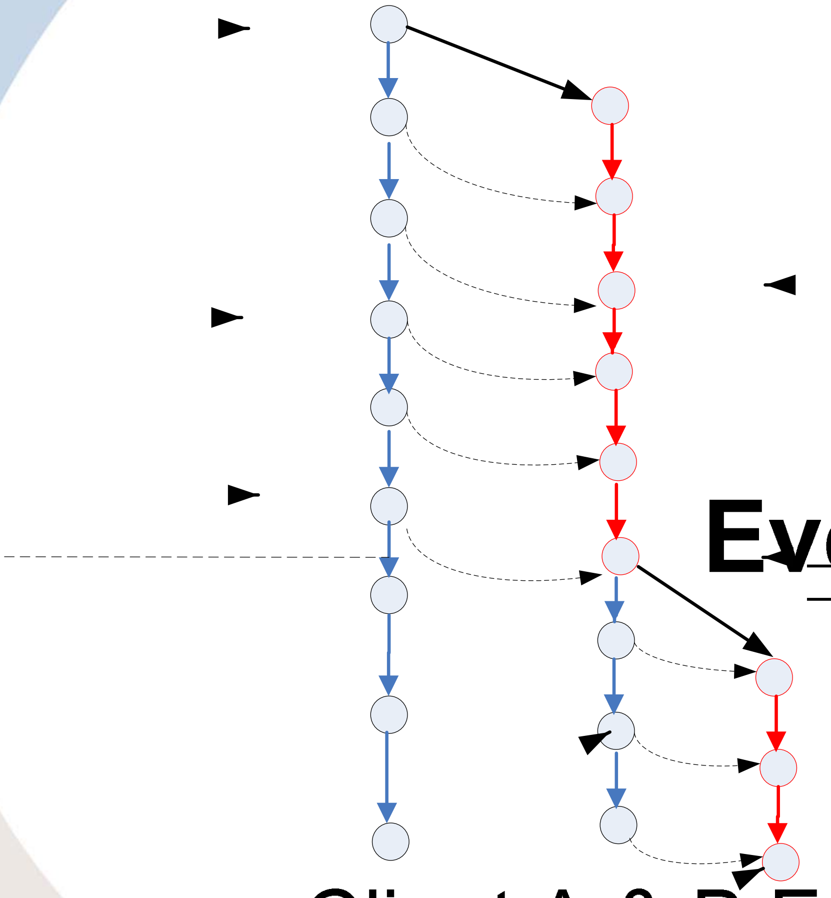
*Making project management indispensable for business results.*

Impact of Dependent Component

# Why?

- Integration delays
- Ineffective utilization of resources
- Impact of dependent components
- Lengthy release cycles – redundant QA cycles
- Resources contentions – concurrent activities
- Quality issues – requirements
- Lack of flexibility – customer needs
- Patch release process - standardization

# Results …



Client A & B Entry

- Faster turnaround
- Customer satisfaction
- Higher productivity
- Better visibility and communication
- Flexibility
- Builds only what is needed

**Every Sprint Delivery – A F**

**Version**

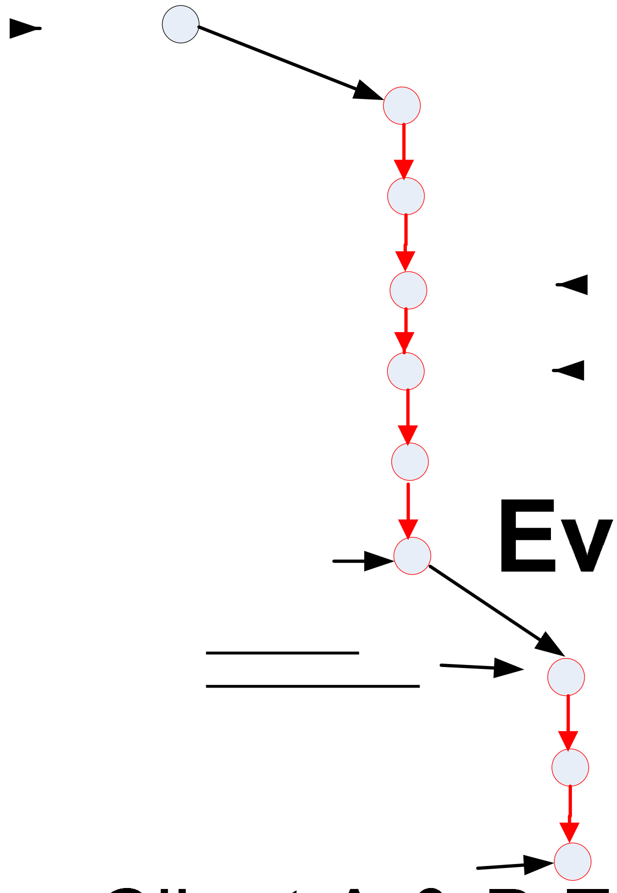
*8.2.0 Support Branch*

Ver 8.2.0.1

Ver

# Goal ?

- Less upgrade hassles
- Fewer versions to maintain
- Rapid feature updates
- Gives value sooner



**Every Sprint Delivery – A F**

**Versio**

*8.2.0 Support Branch*

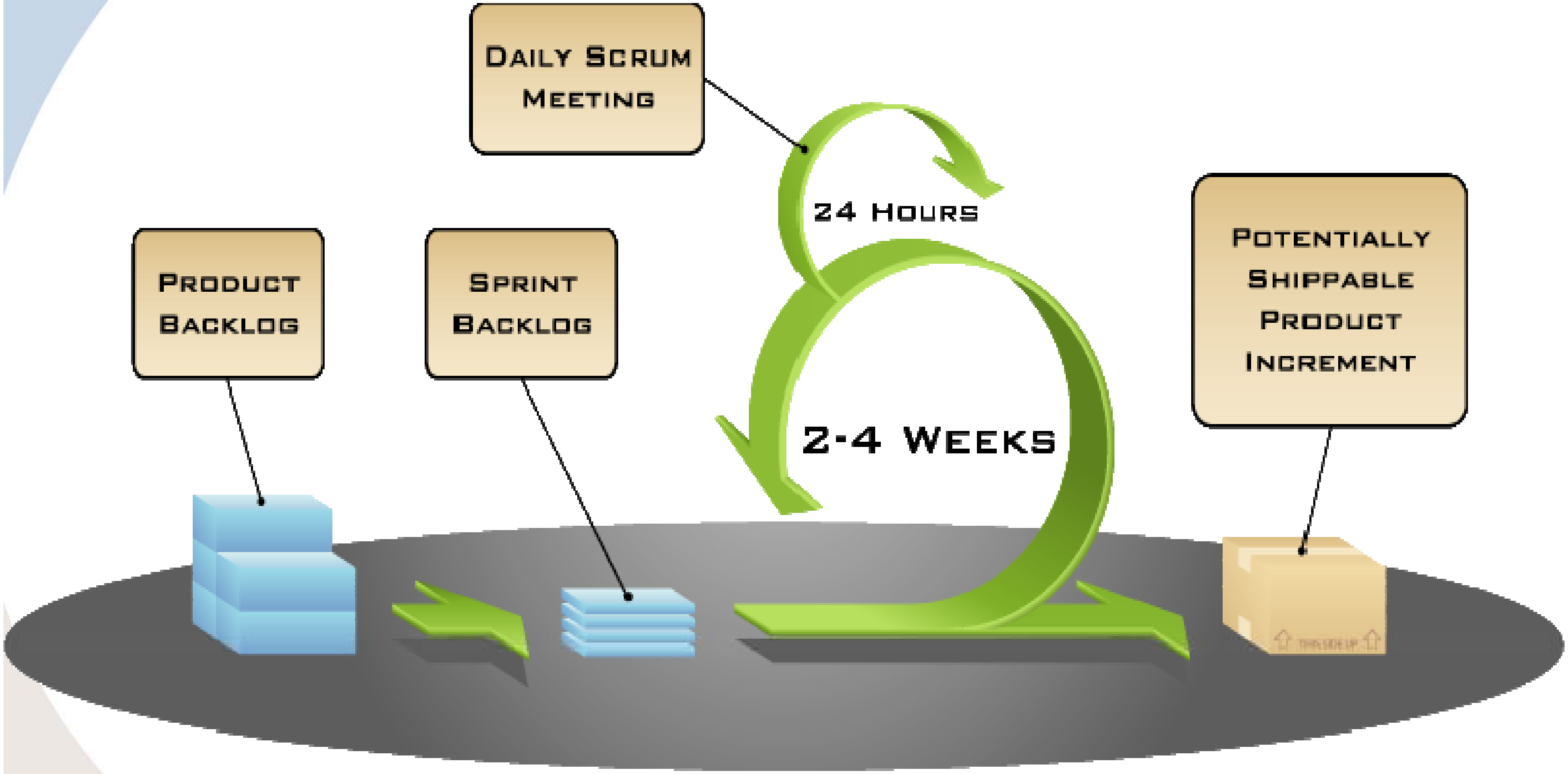
Ver 8.2.0.1

Client A & B Entry

# Scrum

## What is scrum ?

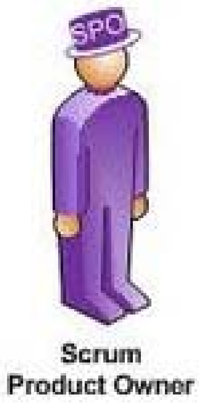
# What is Scrum ?



Copyright © 2005, Mountain Goat Software

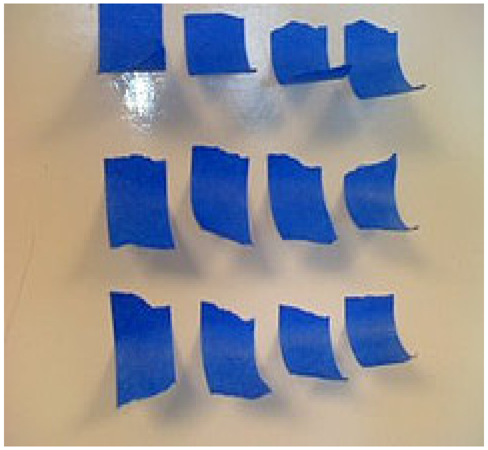
# Scrum: Roles

- Product Owner
  - Represents the interests of stakeholders
  - Responsible for requirements & resulting product
  - Ensure the most valuable functionality is built first
- Scrum Master
  - Ensures rules and practices are followed
  - Responsible for the scrum process
- Team
  - Responsible for developing the functionality
  - Collective responsibility
  - Self managing, self organizing and cross functional



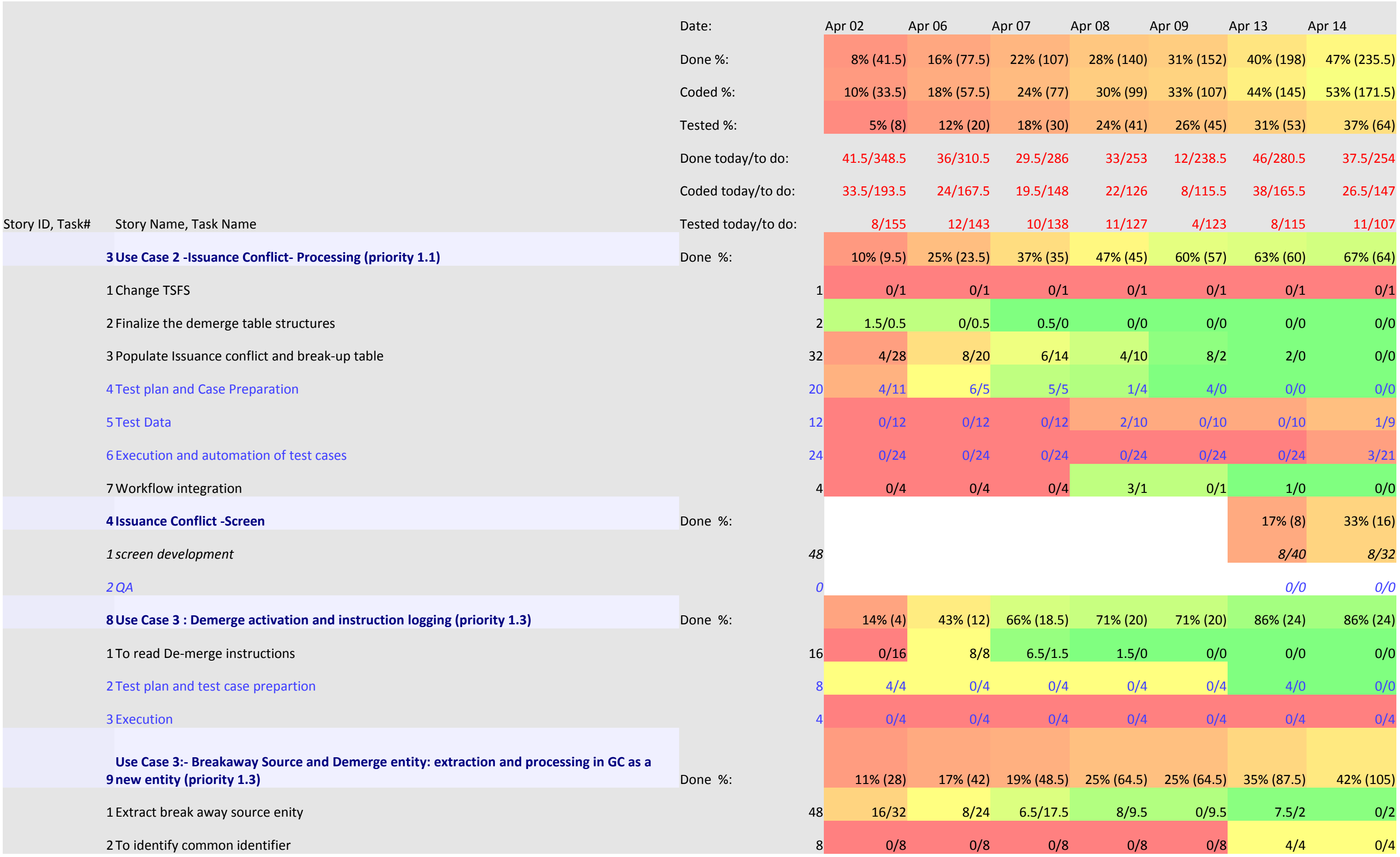
# Scrum: Artifacts

- Product Backlog:
  - The list of prioritized requirements /user stories
    - Dynamic
    - Potentially shippable user stories
    - User story template: As a **<role>** I want **<functionality>** so that **<business value>**



- Sprint Backlog:
  - Tasks that the team defines for doing the selected product backlog items.
  - Sprint backlog emerges as sprint evolves
  - Each task to take 4 to 16 hours to complete

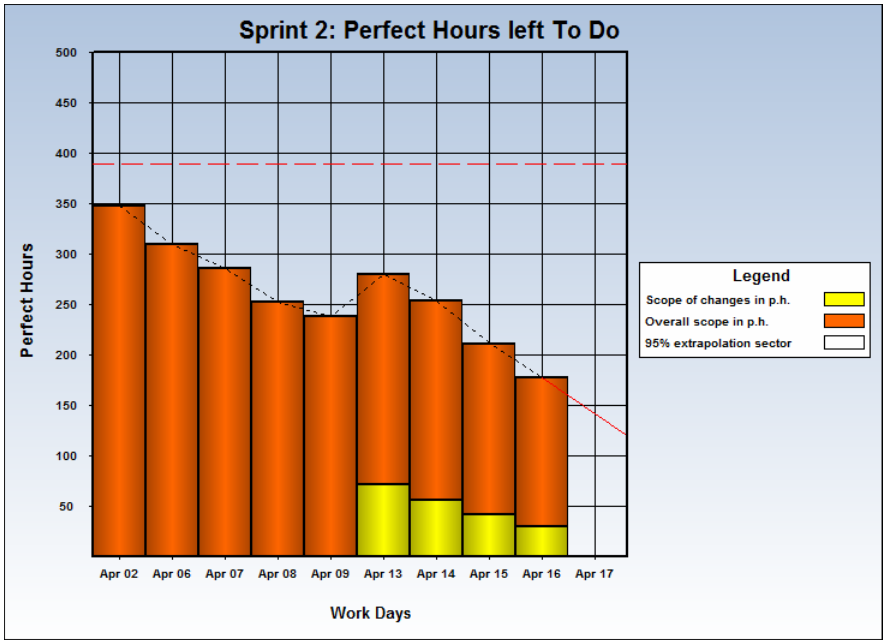
# Sprint Backlog – A Sample



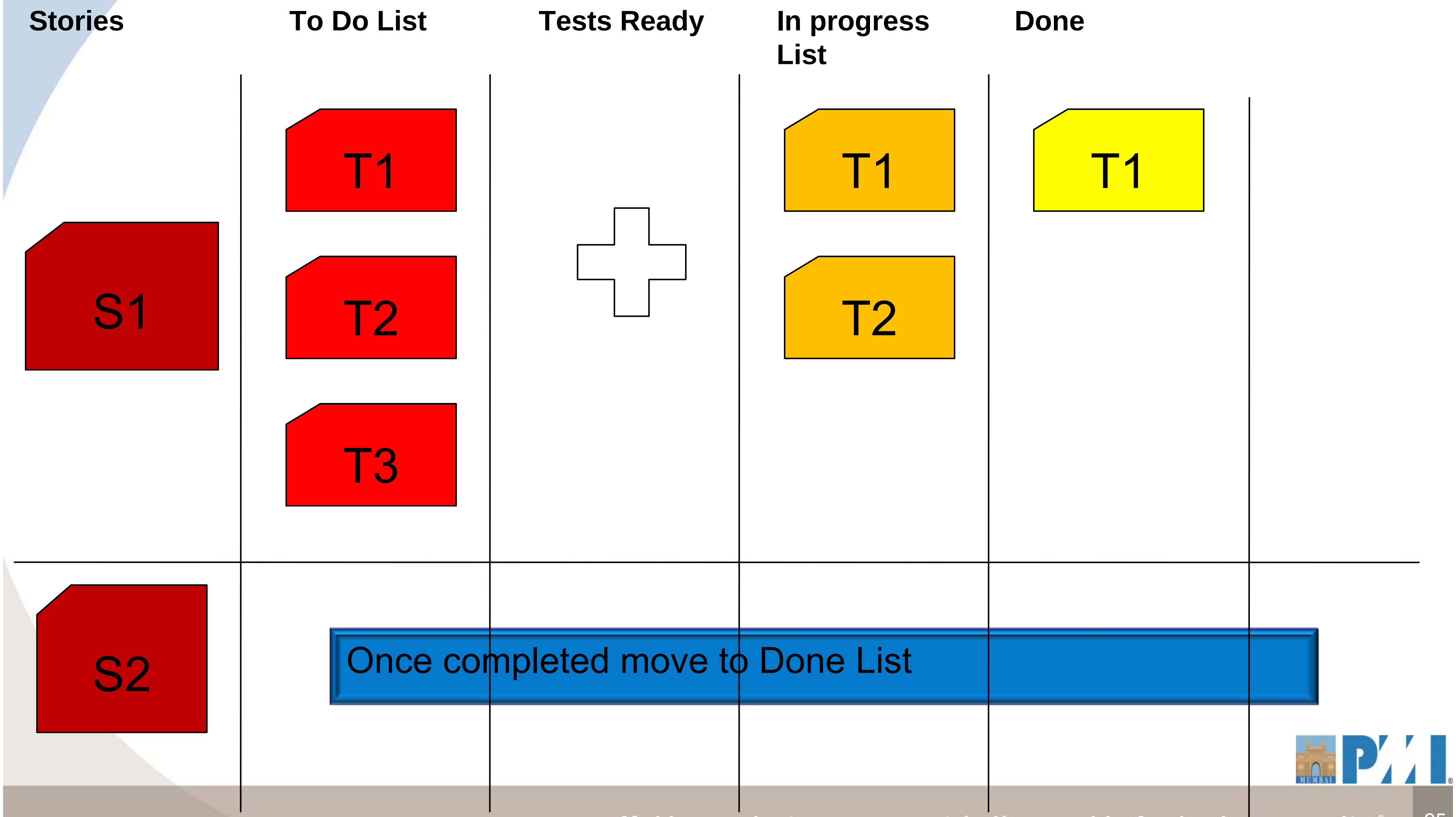
| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Date: | | Apr 02 | Apr 06 | Apr 07 | Apr 08 | Apr 09 | Apr 13 | Apr 14 |
| | | Done %: | | 8% (41.5) | 16% (77.5) | 22% (107) | 28% (140) | 31% (152) | 40% (198) | 47% (235.5) |
| | | Coded %: | | 10% (33.5) | 18% (57.5) | 24% (77) | 30% (99) | 33% (107) | 44% (145) | 53% (171.5) |
| | | Tested %: | | 5% (8) | 12% (20) | 18% (30) | 24% (41) | 26% (45) | 31% (53) | 37% (64) |
| | | Done today/to do: | | 41.5/348.5 | 36/310.5 | 29.5/286 | 33/253 | 12/238.5 | 46/280.5 | 37.5/254 |
| | | Coded today/to do: | | 33.5/193.5 | 24/167.5 | 19.5/148 | 22/126 | 8/115.5 | 38/165.5 | 26.5/147 |
| Story ID, Task# | Story Name, Task Name | Tested today/to do: | | 8/155 | 12/143 | 10/138 | 11/127 | 4/123 | 8/115 | 11/107 |
| 3 | **Use Case 2 -Issuance Conflict- Processing (priority 1.1)** | Done %: | | 10% (9.5) | 25% (23.5) | 37% (35) | 47% (45) | 60% (57) | 63% (60) | 67% (64) |
| 1 | Change TSFS | | 1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 |
| 2 | Finalize the demerge table structures | | 2 | 1.5/0.5 | 0/0.5 | 0.5/0 | 0/0 | 0/0 | 0/0 | 0/0 |
| 3 | Populate Issuance conflict and break-up table | | 32 | 4/28 | 8/20 | 6/14 | 4/10 | 8/2 | 2/0 | 0/0 |
| 4 | Test plan and Case Preparation | | 20 | 4/11 | 6/5 | 5/5 | 1/4 | 4/0 | 0/0 | 0/0 |
| 5 | Test Data | | 12 | 0/12 | 0/12 | 0/12 | 2/10 | 0/10 | 0/10 | 1/9 |
| 6 | Execution and automation of test cases | | 24 | 0/24 | 0/24 | 0/24 | 0/24 | 0/24 | 0/24 | 3/21 |
| 7 | Workflow integration | | 4 | 0/4 | 0/4 | 0/4 | 3/1 | 0/1 | 1/0 | 0/0 |
| 4 | **Issuance Conflict -Screen** | Done %: | | | | | | | 17% (8) | 33% (16) |
| *1* | *screen development* | | *48* | | | | | | *8/40* | *8/32* |
| *2* | *QA* | | *0* | | | | | | *0/0* | *0/0* |
| 8 | **Use Case 3 : Demerge activation and instruction logging (priority 1.3)** | Done %: | | 14% (4) | 43% (12) | 66% (18.5) | 71% (20) | 71% (20) | 86% (24) | 86% (24) |
| 1 | To read De-merge instructions | | 16 | 0/16 | 8/8 | 6.5/1.5 | 1.5/0 | 0/0 | 0/0 | 0/0 |
| 2 | Test plan and test case prepartion | | 8 | 4/4 | 0/4 | 0/4 | 0/4 | 0/4 | 4/0 | 0/0 |
| 3 | Execution | | 4 | 0/4 | 0/4 | 0/4 | 0/4 | 0/4 | 0/4 | 0/4 |
| 9 | **Use Case 3:- Breakaway Source and Demerge entity: extraction and processing in GC as a new entity (priority 1.3)** | Done %: | | 11% (28) | 17% (42) | 19% (48.5) | 25% (64.5) | 25% (64.5) | 35% (87.5) | 42% (105) |
| 1 | Extract break away source enity | | 48 | 16/32 | 8/24 | 6.5/17.5 | 8/9.5 | 0/9.5 | 7.5/2 | 0/2 |
| 2 | To identify common identifier | | 8 | 0/8 | 0/8 | 0/8 | 0/8 | 0/8 | 4/4 | 0/4 |

# Scrum Artifacts (contd.)

- Burndown chart
  - Shows amount of work remaining across time
  - Indicates most probable completion of work
  - Helps "What if analysis".
- Defect Backlog



# Iteration Planning Board



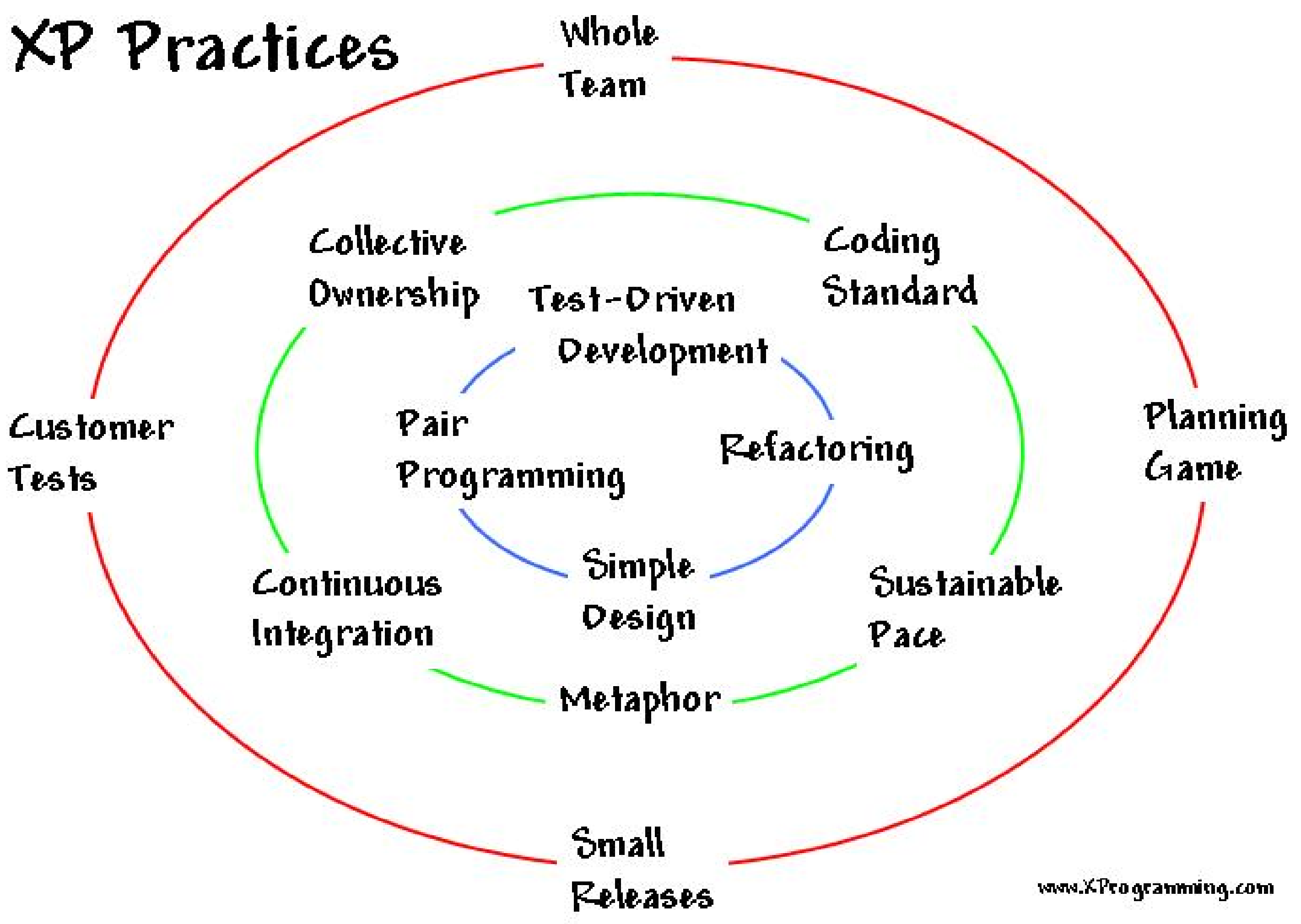
| Stories | To Do List | Tests Ready | In progress List | Done |
|---|---|---|---|---|
| S1 | T1<br>T2<br>T3 | ✚ | T1<br>T2 | T1 |
| S2 | Once completed move to Done List | | | |

# Scrum: Meetings

- Sprint Planning
  - What & How?
  - What do we include in the sprint
  - Divide the requirements to tasks
- Sprint Retrospective
  - What went wrong
  - How can we improve
- Sprint Review
  - What is completed in the sprint
  - Demonstrate what completed
  - What is left out

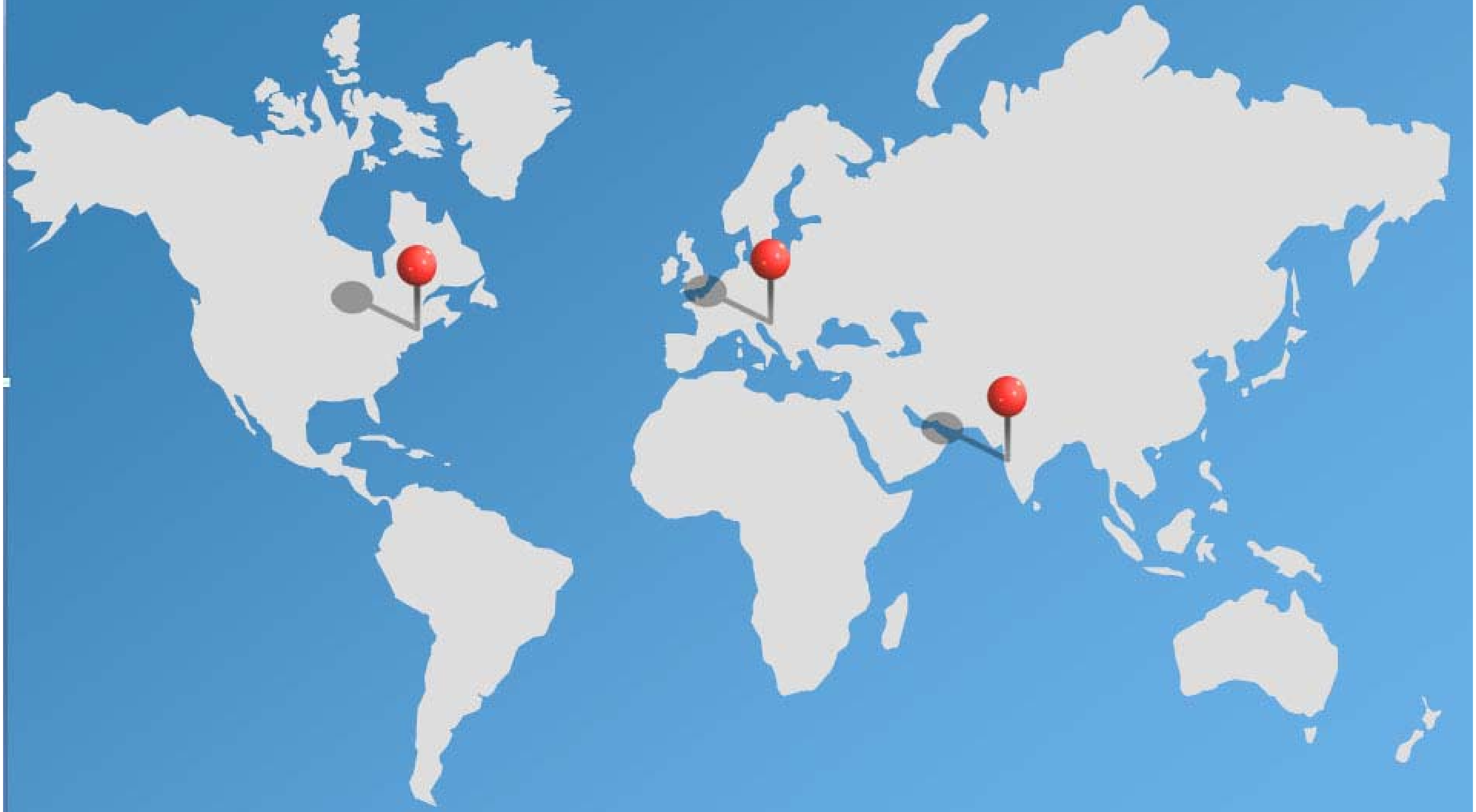
- Daily Scrum Meeting
  - What was done so far in the sprint
  - What needs to be done today
  - Any obstacles

# XP – Another Agile Method



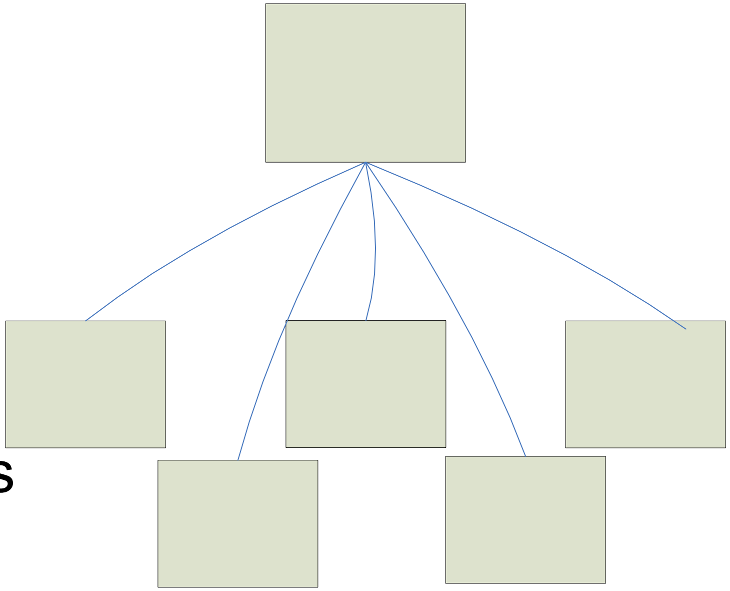
# Transition @ GoldenSource

# GoldenSource Development



*Making project management indispensable for business results.®*

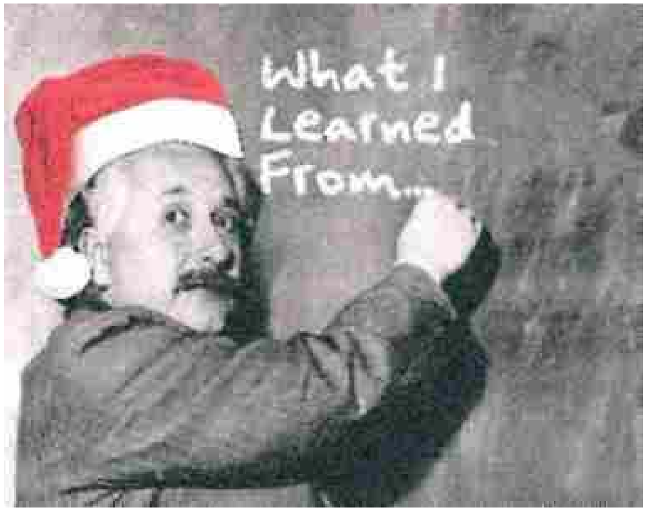
# Implementation ?



- Pilot project in application development
- Transition Team
  - Transition Product Backlog
  - Aligning management & teams
- Initially feature based scrum
- Product based scrum teams
  - Scrum of scrums (Scrum integration team)
- Release planning
- Buy-in and roll out in development
- Roll out in other departments

# Lessons Learned

- Traditional team and reporting structure
  - Unrelated work– team structure
- Sprint Planning:
  - Planning before sprint start
  - Detailed planning
- Tendency to avoid unit testing
  - Prepare unit tests before coding
- Failing to repeat tests – regressions
  - Prepare and automate tests before development
- Part timers & Junior resources



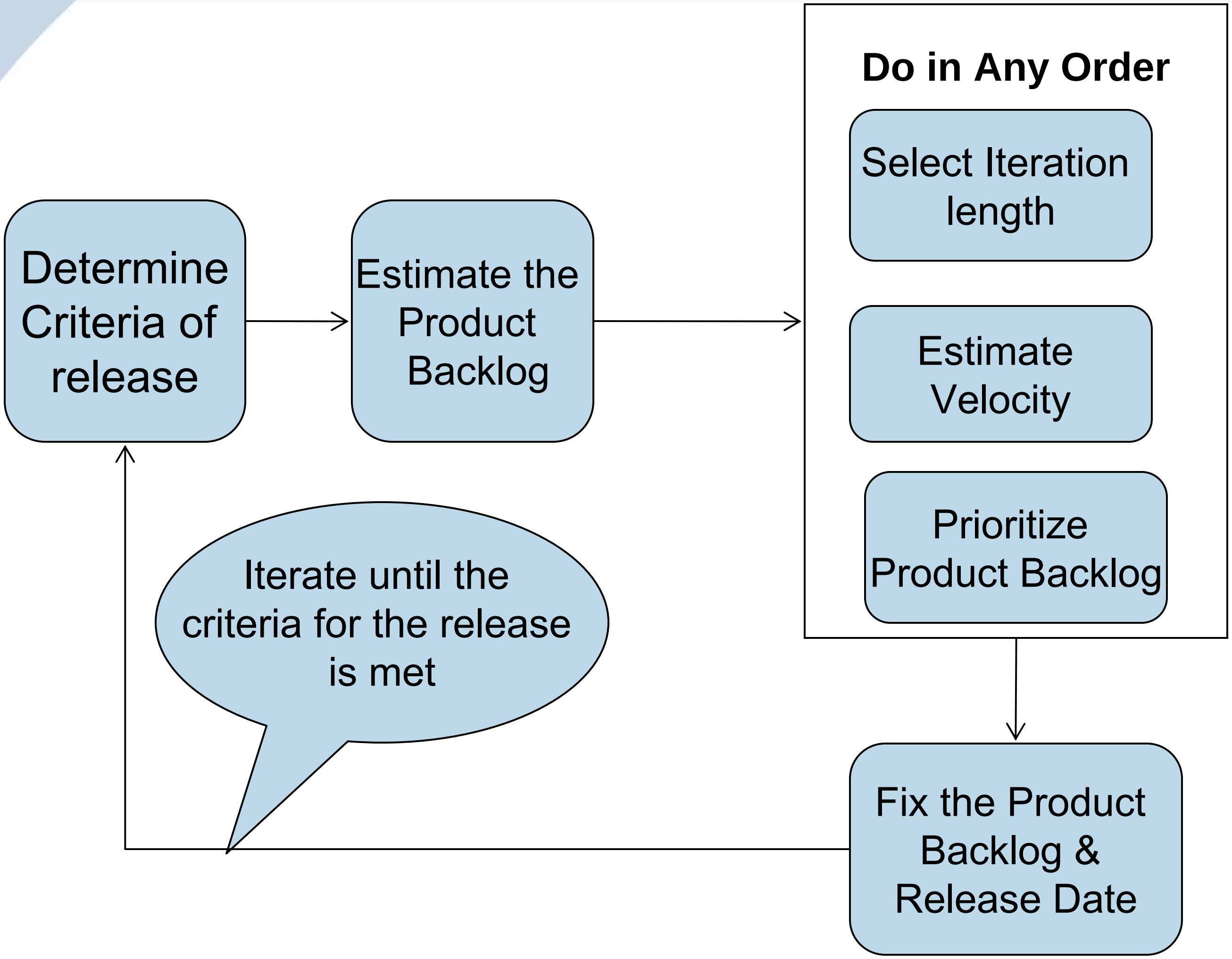
# Lessons Learned (Contd.)

- Training and learning
  - Even breaking story in to small shippable chunks
- Missed deliverables
- Skipping processes
  - Rework and impacting quality
- Resistance to change
- Defect sprint

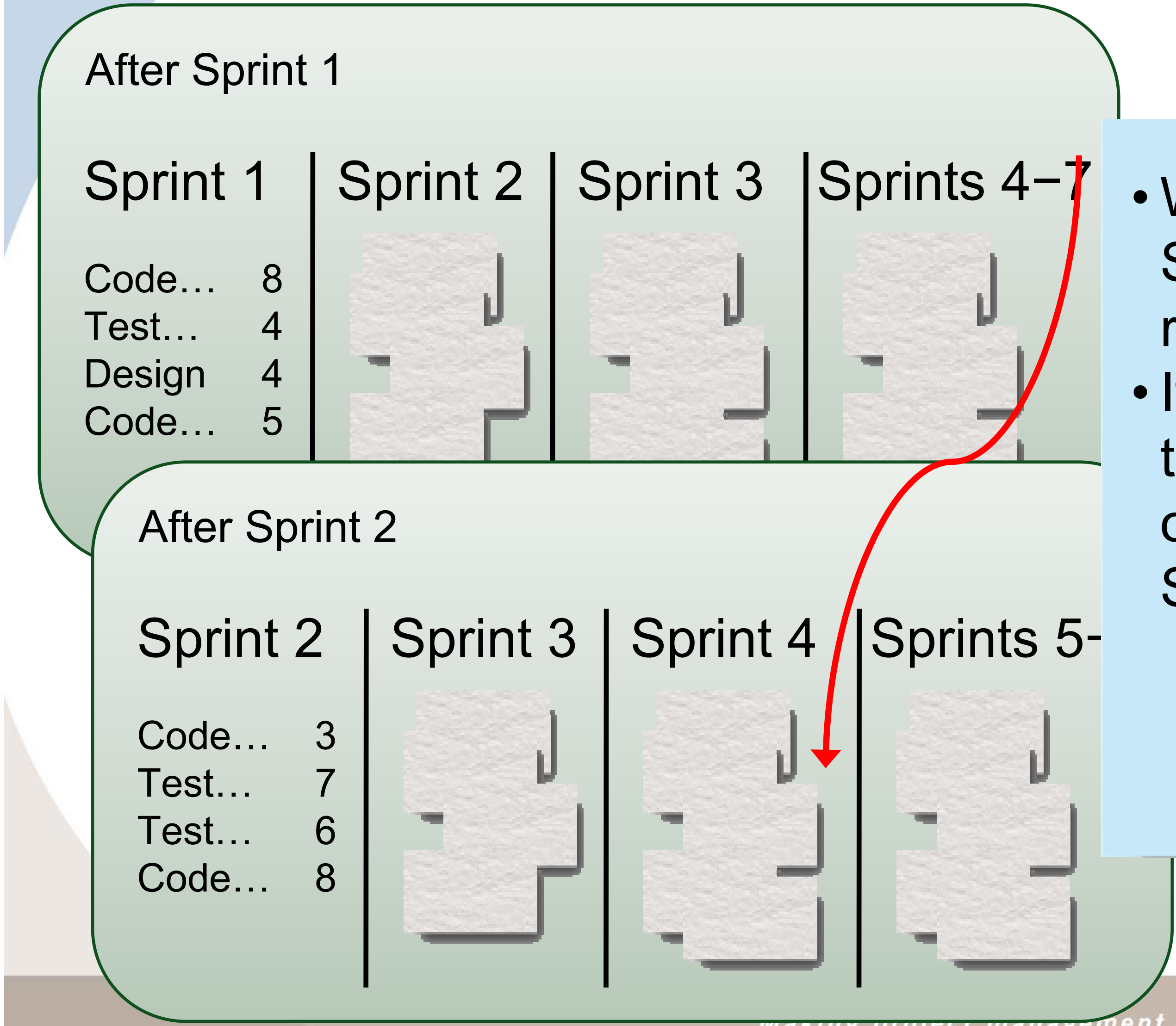
# Release Options

- Dynamic Software Release
  - Every iteration is release quality – system tests
  - PM decides on whether to release or not
  - If decided run System and Performance Tests and Release
- Static Software Release
  - Fix time line for release
  - Pick the features that can be included – priority & estimation
  - Iteratively Plan and Develop
  - Defined Phases for System QA (Defect sprint)
  - Release at the end of the time line

# Release Planning - Estimating


Determine Criteria of release
Estimate the Product Backlog
Do in Any Order
Select Iteration length
Estimate Velocity
Prioritize Product Backlog
Fix the Product Backlog & Release Date
Iterate until the criteria for the release is met

# Rolling Look ahead Planning



After Sprint 1

Sprint 1 | Sprint 2 | Sprint 3 | Sprints 4−7

| Sprint 1 | |
|---|---|
| Code… | 8 |
| Test… | 4 |
| Design | 4 |
| Code… | 5 |

After Sprint 2

Sprint 2 | Sprint 3 | Sprint 4 | Sprints 5-

| Sprint 2 | |
|---|---|
| Code… | 3 |
| Test… | 7 |
| Test… | 6 |
| Code… | 8 |

- While planning Sprint 2, Sprint 4 is rolled into view.
- If any dependency. the other team work on that item during Sprint 3.

# Enterprise and Scrum

- Why you want to be agile? Belief in scrum
- Pilot projects
- Enterprise transition team
  - Goal setting
  - Establish metrics (ROI, Productivity)
  - Training, coaching and communication
- Invest in test automation and tools
- Scrum roll out teams
- Start up in phases
- Use scrum retrospectives to improve

# Project Management and Scrum : Discussion

- Each Release / Sprint
  - Initiating
  - Planning
  - Executing
  - Monitoring and Controlling
  - Closing
- Repeat…

***Executing: Integration Management (D****irect and Manage Project Execution), Human Resource Management (Acquire Project Team, Develop Project Team), Quality Management (Perform Quality Assurance), Communications Management (Information Distribution), Procurement Management (Request Seller Responses, Select Sellers)*

***Monitoring & Controlling**: **Integration Management** (Monitor & Control Project Work, Integrated Change Control), **Scope Management** (Scope  Verification , Scope Control), **Time Management** (Schedule Control), **Cost Management** (Cost Control), **Risk Management** (Risk Monitoring & Control), **Quality Management** (Perform Quality Control), **Communications Management** (Manage stakeholders), **Human Resource Management** (Manage Project Team), **Communications Management** (Performance Reporting), **Procurement  Management** (Contract Administration)*

# Project Management and Scrum : Discussion

- Project Manager Tasks:
  - Project Management Plan – Agreed to and realistic
  - Replan (on going)
- Triple Constraint – ongoing throughout
  - Time, Cost, Scope, Quality, Customer Satisfaction & Risk
- Program Management - Scrum of Scrums
- Product Backlog – Scope Statement – dynamic
- Organizational Structure ?
  - Functional / Projectized / Matrix
  - Role of Scrum master and Product owner

# References

- K. Schwaber (2007) -The Enterprise and Scrum
- DeGrace, P., & Stahl, H. L. (1990). *Wicked Problems Righteous Solutions: a Catalogue of Modern Software Engineering Paradigms.* Yourdon Press.
- Dr. Sutherland, J. (2004). Agile Development: Lessons learned from the first scrum.
- Mills, H. (1976, December). Software Development. *IEEE Trans. Software Eng.* , 265 -273.
- Takeuchi, H., & Nonaka, I. (1986, January). The New New Product Development Game. *Harvard Business Journal* , 137-146

# References (Contd.)

- Larman, C.& Basili, V., (2003, June) Iterative and Incremental Development : A Brief History. IEEE Computer, 47-66
- Martin, R., & Cohn, M. (2006) - Agile Estimating and Planning.

Thank you!