IS4301 Agile IT with DevOps – Lecture 3

Adjunct Professor Foong Sew Bun

Department of Information Systems and Analytics

National University of Singapore

Learning Objectives

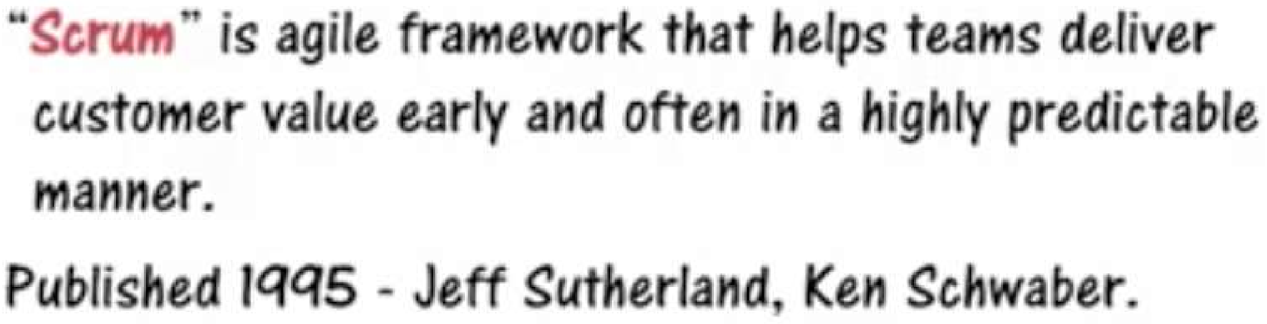
At the end of this lecture, you will understand:

- Essential concepts of different agile IT methods

Agile Methods

- Iterative strategies for managing software development projects, such as Scrum, the Dynamic Systems Development Method (DSDM), Feature Driven Development™ (FDD™), the Agile Unified Process (AUP) and Lean Development
- Strategies for optimizing software development work, such as eXtreme Programming™ (XP) and the Rational Unified Process® (RUP®)
- Strategies for managing software maintenance and support activities, such as Kanban.

SCRUM



“Scrum” is agile framework that helps teams deliver customer value early and often in a highly predictable manner.

Published 1995 - Jeff Sutherland, Ken Schwaber.

- Scrum Alliance: *www.scrumalliance.org*

Roles in SCRUM

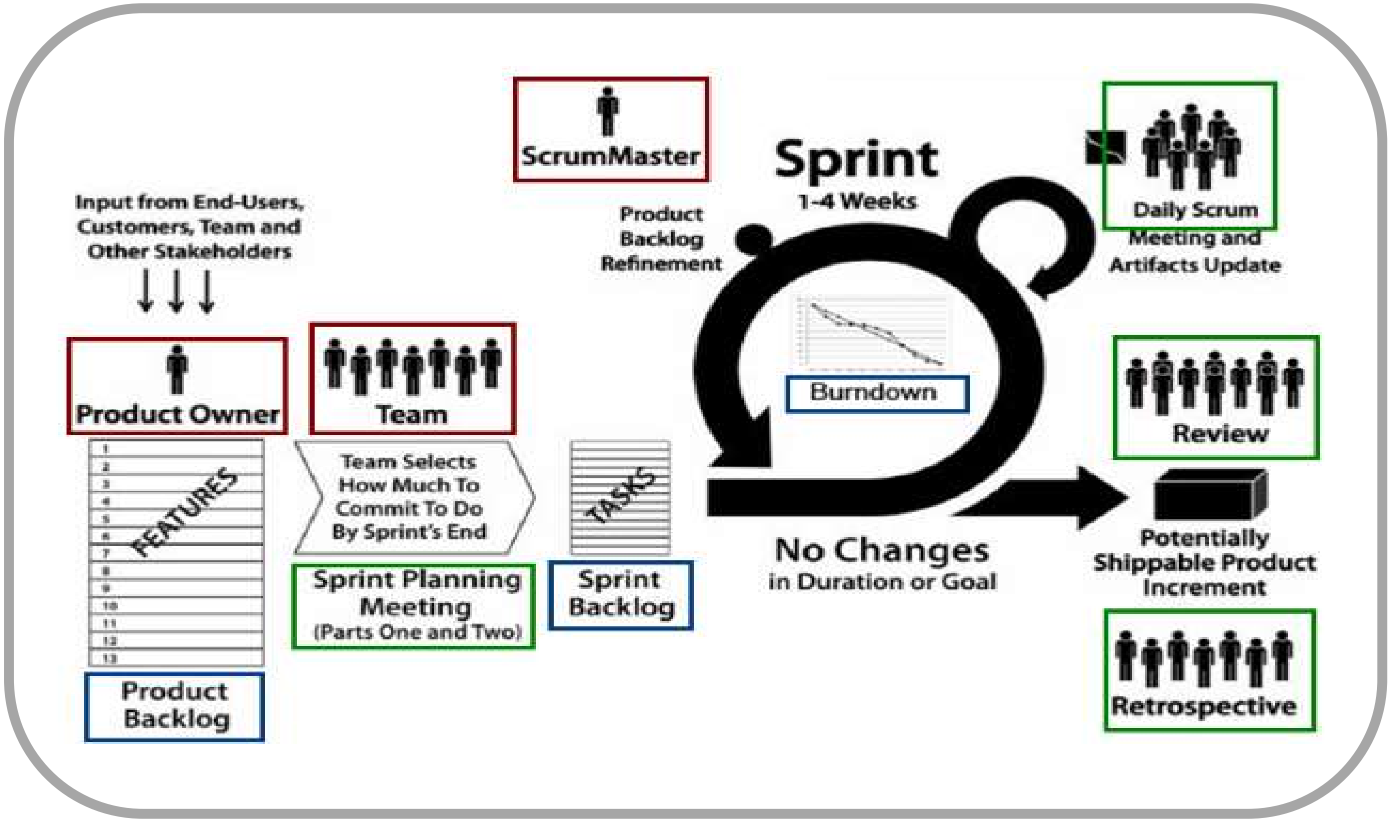
THE PRODUCT OWNER: WHO REPRESENTS THE NEEDS OF THE BUSINESS AND IS RESPONSIBLE FOR DOCUMENTING AND PRIORITIZING HIGH-LEVEL REQUIREMENTS AS INPUT INTO ONGOING PLANNING

- **THE SCRUM TEAM:** A CROSS-DISCIPLINARY TEAM CHARGED WITH UNDERTAKING THE AGREED WORK IN EACH SPRINT

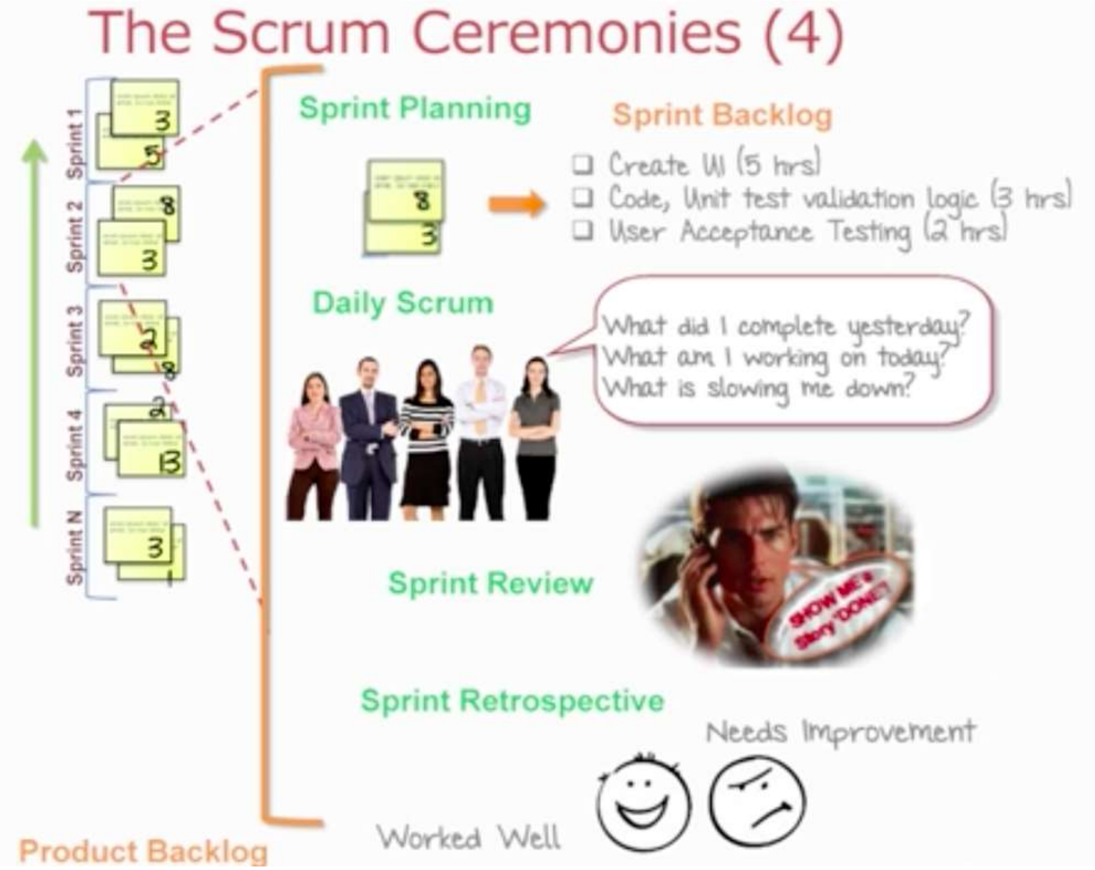
- **THE SCRUMMASTER:** WHO FACILITATES THE TEAM'S WORK BY REMOVING PROJECT IMPEDIMENTS AND ENSURING THAT APPROPRIATE SCRUM PRACTICES ARE BEING FOLLOWED BY THE TEAM.

Agile LifeCycle and Scrum

- **The sprint planning meeting:** held at the beginning of each sprint; this is where the Product Owner, ScrumMaster and Scrum Team review the highest-priority items identified by the Product Owner and agree on the subset of priority items that will be included in the forthcoming sprint
- **The sprint review:** held at the end of each sprint; this includes a demonstration of work completed in that sprint and a retrospective review of the work undertaken to enable continuous improvement for subsequent iterations.


Input from End-Users, Customers, Team and Other Stakeholders
Product Owner
FEATURES
1
2
3
4
5
6
7
8
9
10
11
12
13
Product Backlog
Team
Team Selects How Much To Commit To Do By Sprint's End
Sprint Planning Meeting (Parts One and Two)
TASKS
Sprint Backlog
ScrumMaster
Product Backlog Refinement
Sprint
1-4 Weeks
Burndown
No Changes
in Duration or Goal
Daily Scrum Meeting and Artifacts Update
Review
Potentially Shippable Product Increment
Retrospective

SCRUM Ceremonies



The Scrum Ceremonies (4)

Sprint Planning

Sprint Backlog

- ☐ Create UI (5 hrs)
- ☐ Code, Unit test validation logic (3 hrs)
- ☐ User Acceptance Testing (2 hrs)

Daily Scrum

What did I complete yesterday?
What am I working on today?
What is slowing me down?

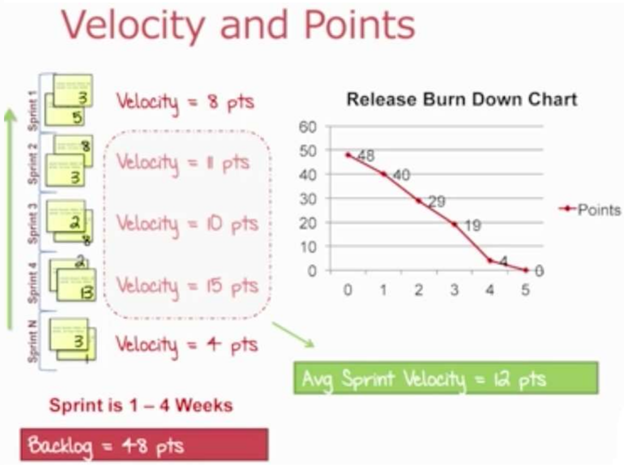
Sprint Review

Sprint Retrospective

Worked Well

Needs Improvement

Product Backlog



Velocity and Points

Velocity = 8 pts

Velocity = 11 pts

Velocity = 10 pts

Velocity = 15 pts

Velocity = 4 pts

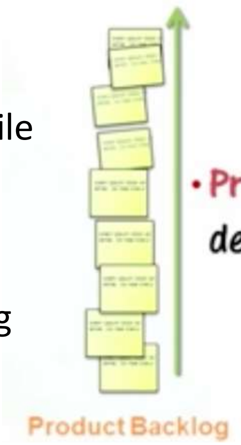
Avg Sprint Velocity = 12 pts

Sprint is 1 – 4 Weeks

Backlog = 48 pts

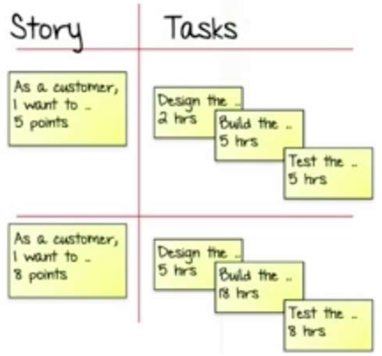
SCRUM Progress Tracking

- **Executive dashboard:** a report that summarizes the work within (and across) Agile teams for easy progress monitoring across the department
- **Product backlog:** a reporting tool that enables both stakeholders and project teams to monitor the progress of work against the agreed business requirements
- **Sprint backlog:** a reporting tool that enables project teams to monitor and manage their actual day-to-day work.

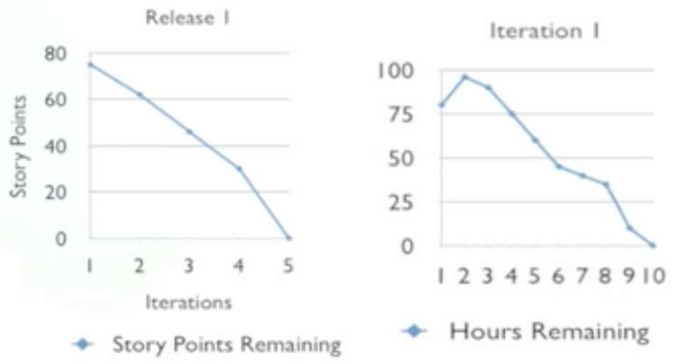


- **Product Backlog :** prioritized list of desired project outcomes/features.

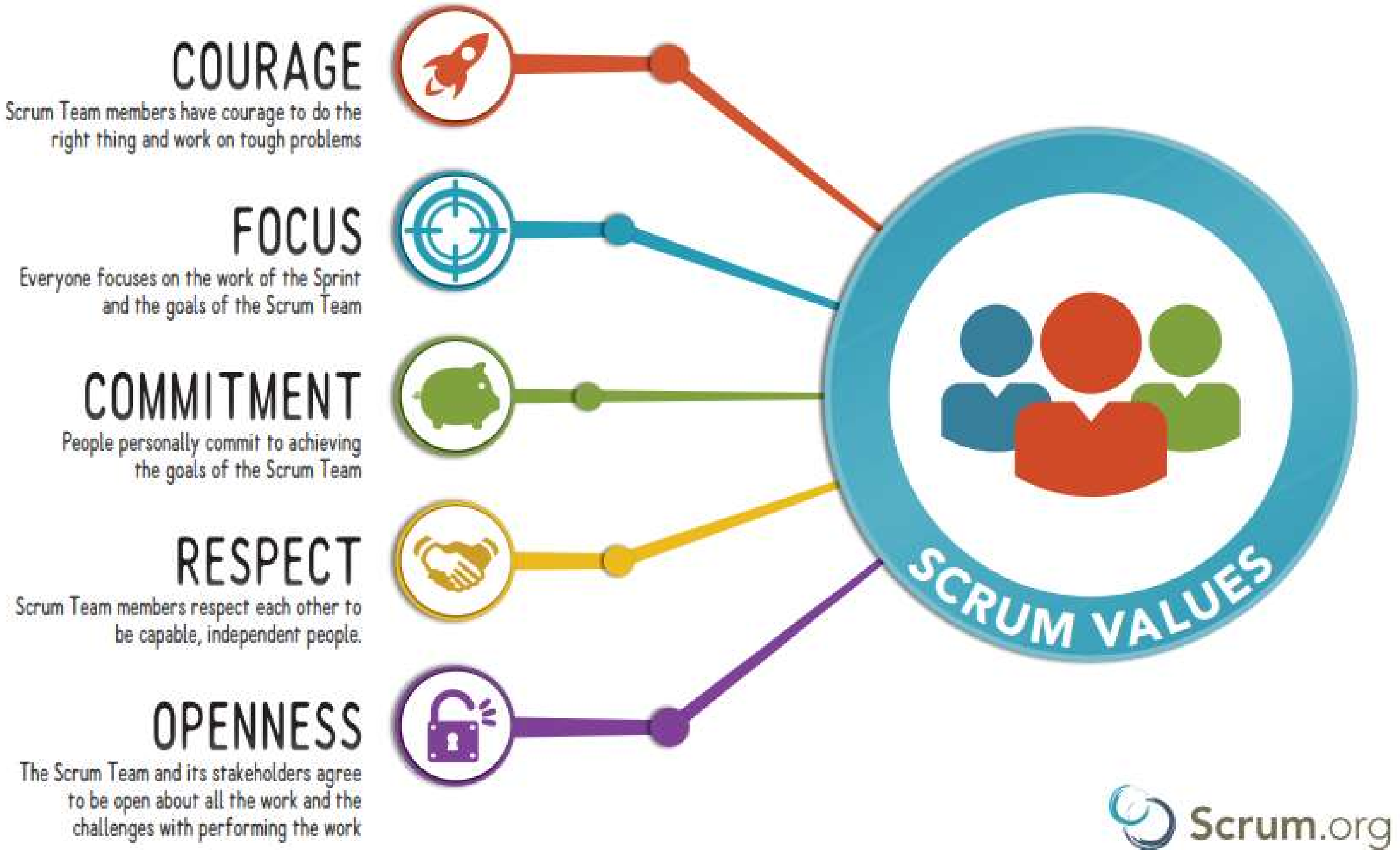
- **Sprint Backlog:** set of deliverables from the product backlog that the team agrees to complete in a sprint usually broken into work items.



- **Burndown chart:** at-a-glance look at the work remaining at the Release or Sprint level.



SCRUM Values



COURAGE
Scrum Team members have courage to do the right thing and work on tough problems

FOCUS
Everyone focuses on the work of the Sprint and the goals of the Scrum Team

COMMITMENT
People personally commit to achieving the goals of the Scrum Team

RESPECT
Scrum Team members respect each other to be capable, independent people.

OPENNESS
The Scrum Team and its stakeholders agree to be open about all the work and the challenges with performing the work

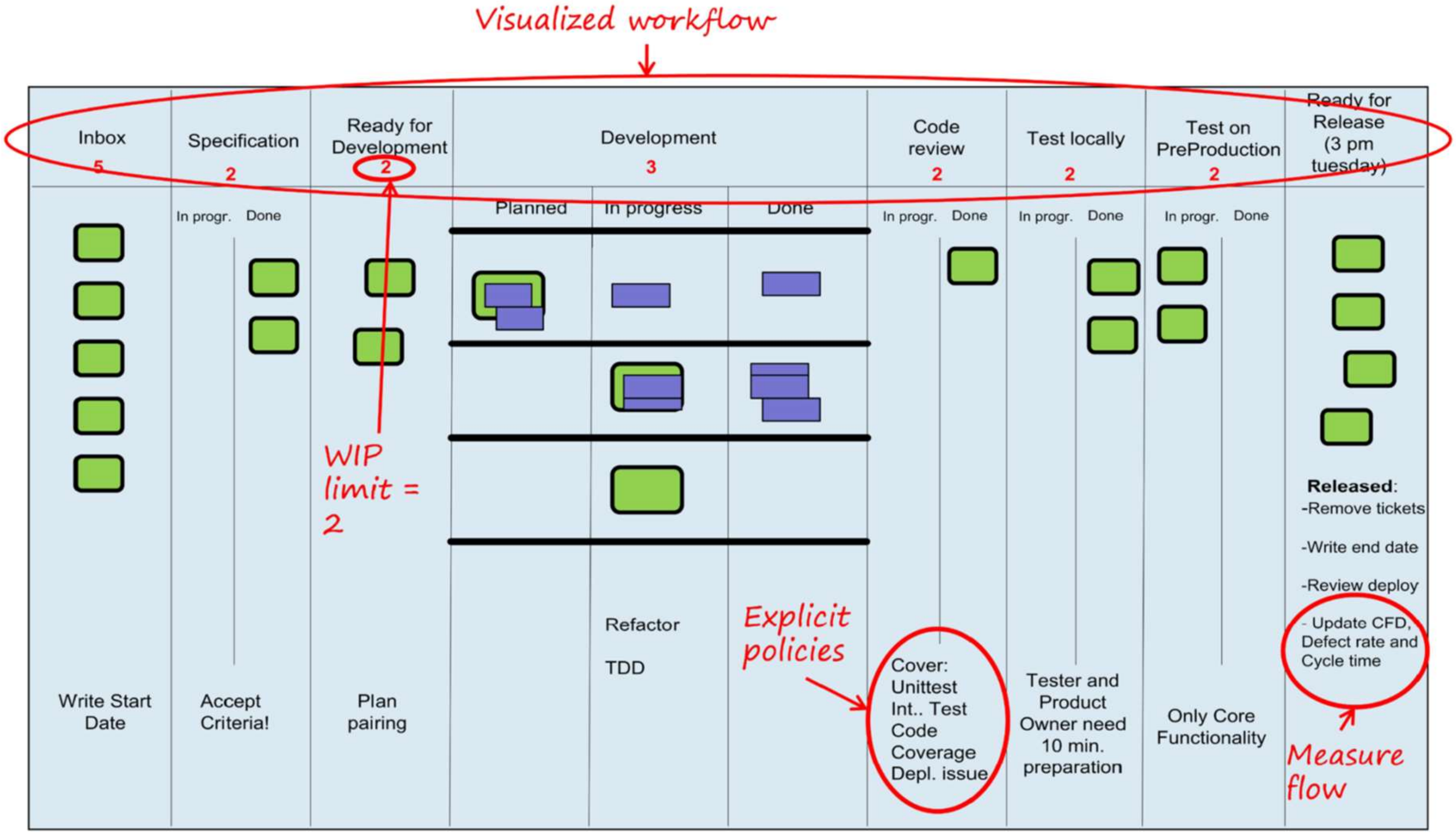
Kanban

- An approach to incremental, evolutionary process and systems change for organizations
- Five core properties of Kanban:
  - Visualize Workflow
  - Limit Work-in-Progress based on Capacity
  - Measure and Manage Flow
  - Make Process Policies Explicit
  - Use Models* to Recognize Improvement Opportunities

Kanban (overview): www.crisp.se/kanban

Kanban and Scrum – making the most of both, Kniberg H and Skarin M (2010): www.infoq.com/minibooks/kanban-scrum-minibook

Kanban



Visualized workflow

WIP limit = 2

Explicit policies

Measure flow

Inbox	Specification		Ready for Development	Development			Code review		Test locally		Test on PreProduction		Ready for Release (3 pm tuesday)
5	2		2	3			2		2		2		
	In progr.	Done		Planned	In progress	Done	In progr.	Done	In progr.	Done	In progr.	Done	
Write Start Date	Accept Criteria!		Plan pairing		Refactor TDD		Cover: Unittest Int.. Test Code Coverage Depl. issue		Tester and Product Owner need 10 min. preparation		Only Core Functionality		**Released**: -Remove tickets -Write end date -Review deploy - Update CFD, Defect rate and Cycle time

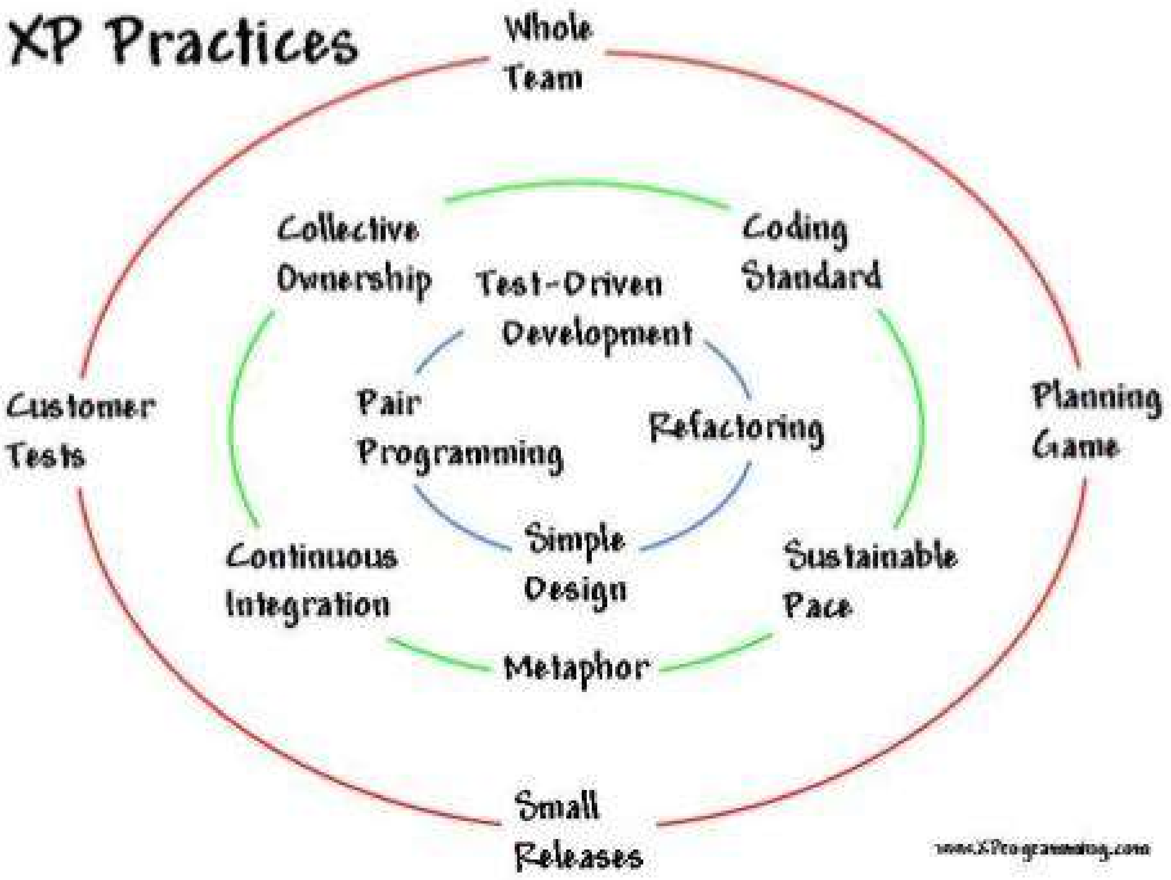
eXtreme Programming (XP)

eXtreme Programming™ (XP) is an Agile methodology for software development work that is based on:

- Delivering the *simplest possible technical solution* required to meet the customer’s objectives
- Anticipating that *requirements will change* once the customer has had an opportunity to work with the delivered software
- Encouraging the *ongoing improvement and optimization of the software* based on customer feedback.

eXtreme Programming™: a gentle introduction:
www.extremeprogramming.org

Practices of eXtreme Programming (XP)



Key Practices of XP

Pair programming: having two members of the project team work together on assigned tasks to increase accountability and knowledge sharing

Automated testing: using an automated testing harness (or equivalent) to regularly confirm the integrity of developed software – in particular, to see if code changes have impacted other functions in the solution

Continuous integration: integrating newly developed code into the code base of the working system, so that updated capabilities are continuously available for production release. Note that continuous integration is generally used *in conjunction with* automated testing to ensure that updated code has not introduced errors into the existing system.

Feature Driven Development (FDD)

- Feature Driven Development™ is an Agile methodology that combines elements of iterative project management with practices that are specific to software development. The basic driver of FDD™ is to provide incremental value to the business by delivering complete, working product capabilities (i.e. software "features" and "feature sets") in every iteration.
- Combines iterative project delivery with software development practices by:
    - Having teams model the business problem upfront
    - Decomposing the model into smaller features and feature sets
    - Integrating selected feature sets into the overall software solution through iterative releases
    - Keeping a strong focus on collaboration with users, production of tangible outputs, and quality management throughout the process.

An Overview of FDD™ – Web Development Methodology: *www.influxive.com/fdd-overview.html*

Feature Driven Development™ (FDD™) and Agile Modeling, Ambler S W: *www.agilemodeling.com/essays/fdd.htm*

Dynamic Systems Development Method (DSDM)

The **Dynamic Systems Development Method (DSDM®)** is an Agile methodology for project delivery that involves:

- Delivering software in time-boxed iterations
- Prototyping and documenting the software solution prior to undertaking full development activities
- Collaborating with users, producing tangible outputs, and ensuring quality management throughout the process.

Scrum provides a high-level framework for achieving this objective, and relies on communication between the participants to ensure that work undertaken meets ongoing business needs. DSDM provides a more structured framework to achieve this objective, requiring proposed work to be documented and confirmed prior to continuing to the next stage.

What is DSDM? Clifton M, Dunlap J (2003): *www.codeproject.com/KB/architecture/dsdm.aspx*

DSDM Explained, Davies R, JAOO (2004): *www.agilexp.com/presentations/DSDMexplained.pdf*

Lean Development (Lean)

Lean Development (Lean) is an Agile methodology that combines elements of iterative project management with best practices in software design and development by:

- Using value stream mapping to deliver the highest business-value features within the most efficient software development process
- Incorporating pull techniques and optimal capacity planning to deliver results as quickly as possible
- Enforcing stringent quality management through integrity checking and continuous improvement
- Empowering skilled cross-functional teams to deliver the highest-value outcomes to the department.

Lean stems from the Lean manufacturing processes that originated as early as 1922, but most closely aligns with the principles in KAIZEN-and Total Quality Management (TQM).

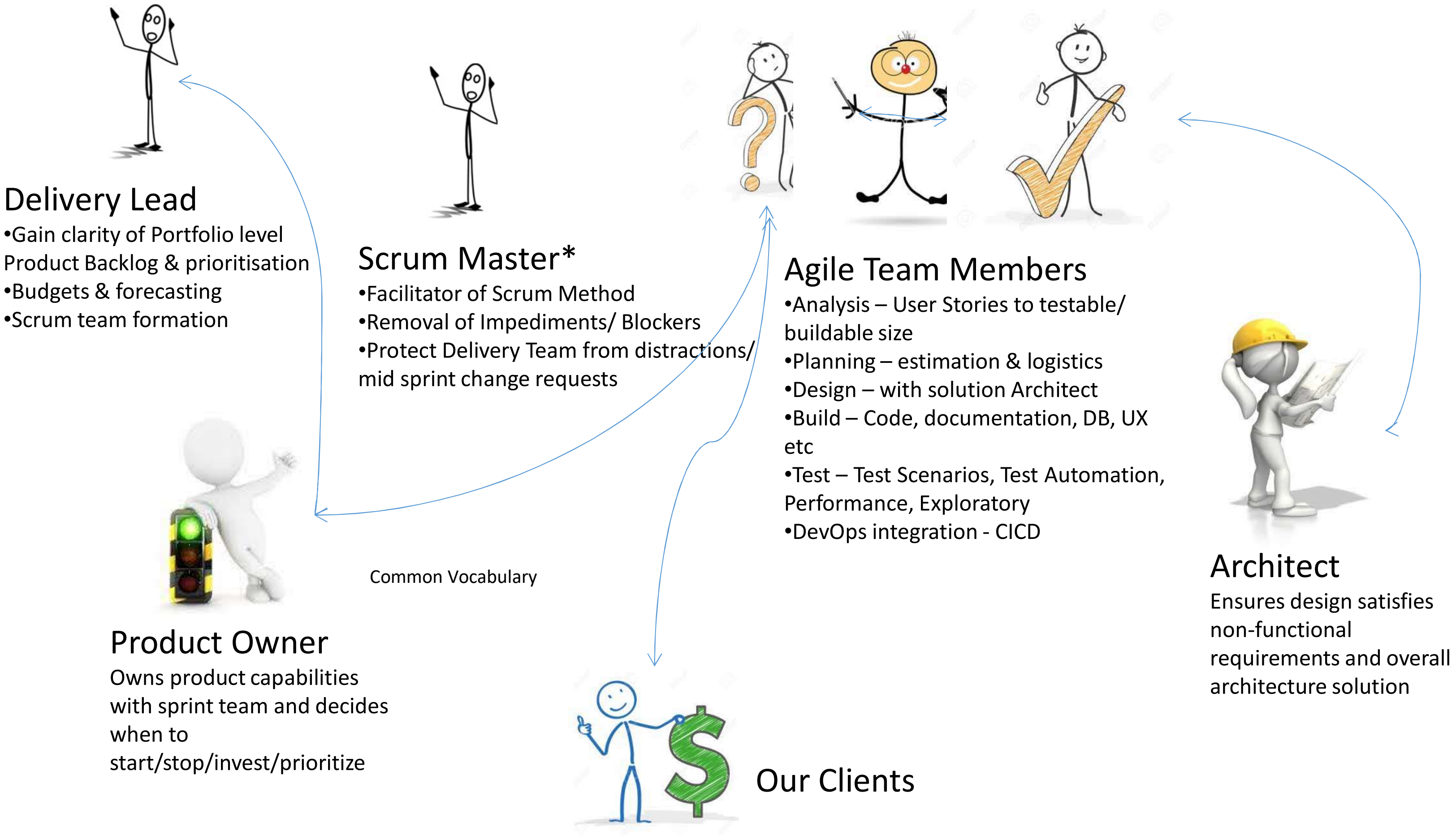
Lean Primer, Larman C & Vodde B (2009): *www.leanprimer.com/downloads/lean_primer.pdf*

Leading Lean Software Development: Results Are Not the Point, Poppendieck.LLC (2009):

www.poppendieck.com/pdfs/LLSD_intro.pdf

Running Agile: A Practitioner's View to Lean and Agile: http://runningagile.com/

Roles in an Agile Team



Delivery Lead

- Gain clarity of Portfolio level Product Backlog & prioritisation
- Budgets & forecasting
- Scrum team formation

Scrum Master*

- Facilitator of Scrum Method
- Removal of Impediments/ Blockers
- Protect Delivery Team from distractions/ mid sprint change requests

Agile Team Members

- Analysis – User Stories to testable/ buildable size
- Planning – estimation & logistics
- Design – with solution Architect
- Build – Code, documentation, DB, UX etc
- Test – Test Scenarios, Test Automation, Performance, Exploratory
- DevOps integration - CICD

Architect

Ensures design satisfies non-functional requirements and overall architecture solution

Product Owner

Owns product capabilities with sprint team and decides when to start/stop/invest/prioritize

Common Vocabulary

Our Clients

Product Owner

A Product Owner owns product capabilities with sprint team and decides when to start / stop / invest / prioritizes.

- Responsibilities:
  - Clearly communicates project vision, business case and goals to the team and stakeholders
  - Collates, creates and maintains the product backlog
  - Prioritizes and sequences the product backlog according to business value
  - Assists and contributes with the elaboration of epics and user stories
  - Defines MVP in collaboration with stakeholders
  - Engages in sprint planning, sprint demo and sprint retrospectives
  - The only team member empowered to accept stories-are-done
  - Reports progress to the programme managers and stakeholders regularly

Scrum Master

- A Scrum Master is an experienced agile practitioner who has shared responsibility with the team for the delivery of the sprint scope

- Responsibilities:
  - Coordinates and facilitates agile ceremonies (sprint planning, sprint demo, sprint retrospective and daily stand-up meetings)
  - Facilitate the team to self-organize, self-manage and deliver
  - Address scrum team’s impediments
  - Holds team, product owner and stakeholders accountable for their commitments
  - Protects team from distractions, such as mid-sprint change-requests
  - Maintains RAID log

Architect

An Architect is a technical specialist responsible for defining the solution architecture

Responsibilities:

- Defines the high-level solution architecture and review with stakeholders
- Get approval for the solution architecture from the required architecture boards
- Defines a first indicative roadmap of how the solution evolves
- Ensures that the product is developed in accordance with the overall architectural strategy and the development team adheres to valid design guidelines and standards
- Works with the technology services team to ensure non-functional requirements are captured and implemented as expected

Team Members

- Team members estimate user stories, identify tasks, “sign-up” for tasks, design, code, test, and update their status towards completion

- Responsibilities:
  - Attend all agile ceremonies (sprint planning, sprint demo, sprint retrospective and daily stand-up meetings)
  - Demonstrate done-user-stories in sprint demo
  - Contribute in defining DoD and DoR for each user story
  - Help create and maintain the sprint backlog, sprint burndown chart and task board

Role	Responsibilities
Business Analysts	• Work with the product owner, elaborate user stories and define acceptance criteria (providing inputs and expected outputs)
Developers	• Analyze, design and implement user stories • Fix defects • Perform unit testing using TDD / BDD
Testers	• Perform functional and non-functional testing including automation • Prepares test status reports

Stakeholders

Stakeholders are representatives from business, end users, support / operations and other impacted teams.

Responsibilities:

- Carry out timely decisions regarding project scope and funding
- Review and approve strategies to resolve project risks
- Involved in user acceptance testing
- Participates in programme steering committee meetings
- Agree and sign off scope, MVP and release
- Provide acceptance and sign off for their areas of responsibilities

Agile Coach

Agile coach is an experienced agile practitioner with solid experience in coaching / mentoring teams

Responsibilities:

- Assess team’s Agile competency to identify opportunities for improvement
- Bring learnings from the Agile community into the team
- Help the team adhere to the Agile principles and values
- Coaching on the levels of planning and team structure required for a large enterprise effort
- Help the team to define appropriate metrics based on the Agile method used
- Helps teams to transition from waterfall to agile methodology

Specialists

Specialists increase efficiency in agile teams. They solve key problems and execute plans with deep understanding of a specific discipline (i.e. Domain SME, Test Environment, CI/CD, etc…)

Role	Responsibilities
Test Environment Manager	• Analyze and translate environment requirements from test strategy in order to establish environment specifications. • Plan for key activities, resources, dependencies, risks and timelines necessary to deliver the environment and communicate progress to relevant stakeholders • Collaborates with DevOps and team members in implementing CI/CD
Domain Expert	• Interprets and understands real business requirements and issues which are critical for operational decision making • Provides the knowledge and expertise in a specific subject for a project
UX Designer	• Creates user experience designs which encompasses all aspects of end users interaction with the product
GIS	• Analyze systems to identify system security risks/vulnerability and perform security testing
Tech. Operations	• Responsible for OAT
Tech. Support Engineer	• Represents PSS and Change management teams. Provide support after go-live to user community

Program Governance

The Program Governance team manage end-to-end delivery of a project from securing funding to release. In addition, they manage collective activities, issues and risks of concurrent projects.

Role	Responsibilities
Tech. Delivery lead	• Leads the technology delivery 'work stream' /team of a project
Programme Manager	• Delivers the programme within agreed finances • Track and coordinate the program portfolio and its dependencies • Manage collective activities, issues, and risks • Run the programme steering committee meetings • Accountable to the business stakeholders for delivering the change
Portfolio Test Manager	• Manage and set strategic direction and end-to-end delivery within the given portfolio • Work with product owner and scrum master to ensure the portfolio’s applications are tested according to the standards and policy of the bank as specified in SDF • Serve as escalation point for all test related matters and a point of reference for complex problem resolution within the given portfolio • Plan and coordinate test resources across projects within CIO and the given portfolio • Prepare Test Strategy and Test Closure reports