COMP41670 Software Engineering

12. Agile Methods including Scrum

Dr Avishek Nag

UCD School of Computer Science. **Scoil na Ríomheolaíochta UCD.**

Table of Contents

Introduction

Agile Methods

- With a plan-driven approach it is difficult to accommodate late changes in system requirements.
- Agile development methods emerged in the late 1990s with the goal of reducing delivery time for working software systems.
- In the plan-driven approach, activities are performed sequentially. It is assumed that the output of each activity is largely correct and does not have to be re-visited.
- In the agile approach, activities are interleaved and iterated. The working product is built up incrementally.

The Agile Manifesto

- “On February 11-13, 2001, at The Lodge at Snowbird ski resort in the Wasatch mountains of Utah, seventeen people met to talk, ski, relax, and try to find common ground—and of course, to eat. What emerged was the Agile ‘Software Development’ Manifesto. Representatives from Extreme Programming, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, Pragmatic Programming, and others sympathetic to the need for an alternative to documentation driven, heavyweight software development processes convened.”
- https://agilemanifesto.org

The Agile Manifesto: 12 Principles

Our highest priority is to satisfy the customer
through early and continuous delivery
of valuable software.

Welcome changing requirements, even late in
development. Agile processes harness change for
the customer's competitive advantage.

Deliver working software frequently, from a
couple of weeks to a couple of months, with a
preference to the shorter timescale.

Business people and developers must work
together daily throughout the project.

Build projects around motivated individuals.
Give them the environment and support they need,
and trust them to get the job done.

The Agile Manifesto: 12 Principles

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

The Agile Manifesto: 12 Principles

At regular intervals, the team reflects on how
to become more effective, then tunes and adjusts
its behavior accordingly.

The Agile Manifesto in a Nutshell

- Early and continuous delivery of valuable software
- Welcome changing requirements, even late in development
- Deliver working software frequently
- Business people and developers must work together daily throughout the project
- Face-to-face conversation
- Self-organising teams
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

Agile Project Management

Scrum

Agile Project Management

- The principal responsibility of software project managers is to manage the project so that the software is delivered on time and within the planned budget for the project.
- The standard approach to project management is plan-driven. Managers draw up a plan for the project showing what should be delivered, when it should be delivered and who will work on the development of the project deliverables.
- Agile project management requires a different approach, which is adapted to incremental development and the practices used in agile methods.

Scrum

- “Scrum is the most popular Agile framework”
- Scrum is a framework for organising and managing work
- The concepts have been refined a lot since, but the original paper was “The New Product Development Game” by Takeuchi and Nonaka, Harvard Business Review, 1986:
  - “The traditional sequential or ‘relay race’ approach to product development - exemplified by the National Aeronautics and Space Administration's phased program planning (PPP) system - may conflict with the goals of maximum speed and flexibility. Instead, a holistic or ‘rugby' approach - where a team tries to go the distance as a unit, passing the ball back and forth - may better serve today's competitive requirements.”
- https://www.scrumalliance.org
- https://www.scrum.org

Scrum Roles

- A Scrum development consists of one or more Scrum teams.
- Each team has:
  - 1 Product Owner
  - 1 ScrumMaster
  - 5-9 developers

- Stakeholders: These are people that are not on the team but have a vested interest in the project, i.e. future users of the application, business managers who are paying for the development.

The Product Owner

- Single authority responsible for deciding which features to build and the order in which to build them.
- Does this by:
  - communicating with the stakeholders (e.g. managers, people paying the bills)
  - communicating with users
  - understanding from the development team what the options are (e.g. how long things will take)
- Collaborates closely with the ScrumMaster and the development team

The ScrumMaster

- A leader, not a manager (can’t hire and fire the developers).
- As a facilitator, helps the team resolve issues.
- Champions the Scrum approach
- Responsible for removing impediments to team productivity.

Development Team

- Cross-functional (architect, programmer, database designer, user interface designer) team who design, build and test the product
- Self-organised to accomplish the goal set out by the product owners

Scrum

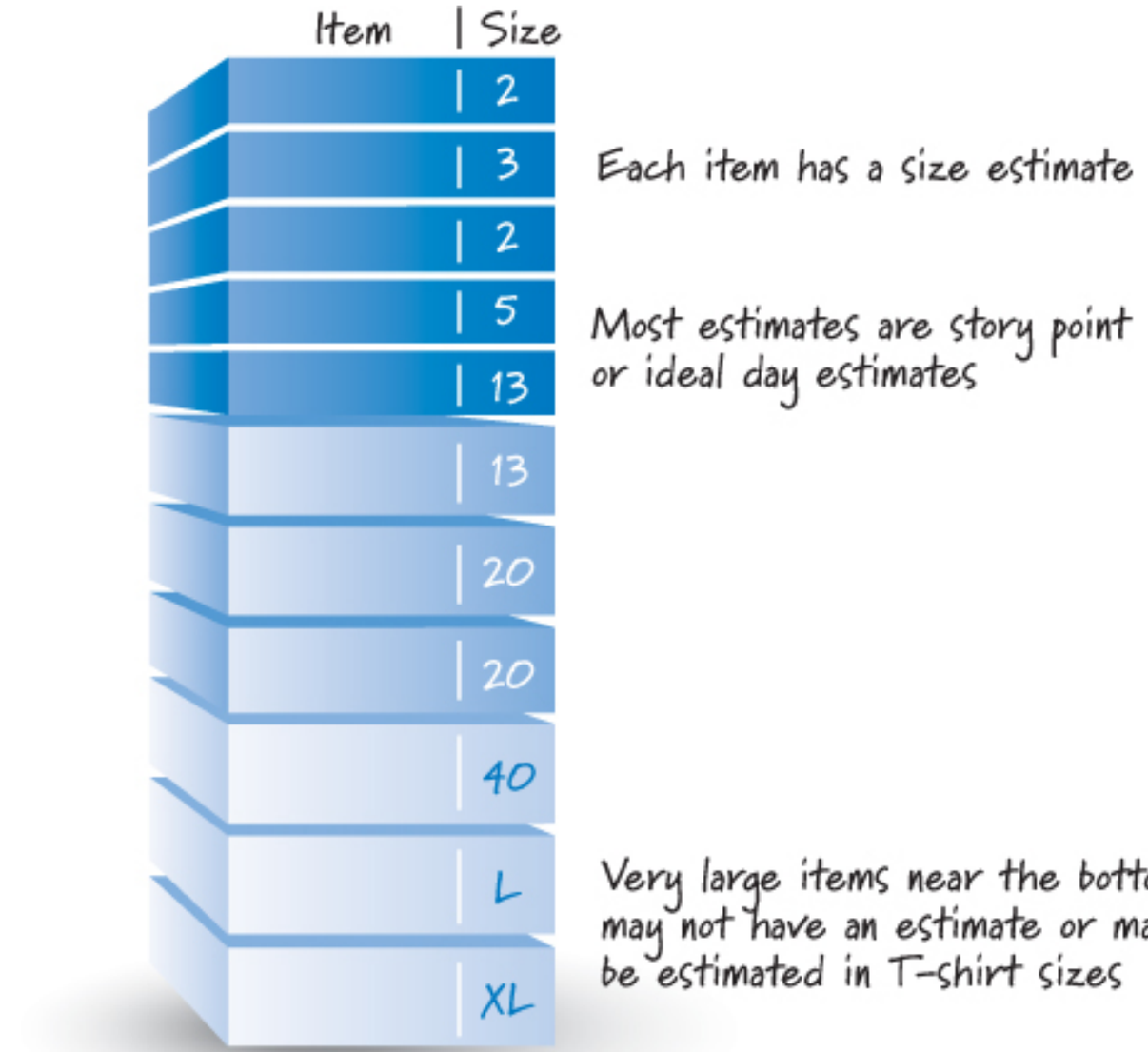
- There are three phases in Scrum:
  1. The initial phase is an outline planning phase where you establish the general objectives for the project, capture the user requirements, and establish the Product Backlog.
  2. This is followed by a series of sprint cycles, where each cycle develops an increment of the system.
  3. The project closure phase wraps up the project, completes required documentation such as system help frames and user manuals and assesses the lessons learned from the project.

Initial Phase

- Uses a Requirements Gathering approach (see later).
- From this, the team makes a Product Backlog.
- The **Product Backlog** is the list of features to be built on the project.

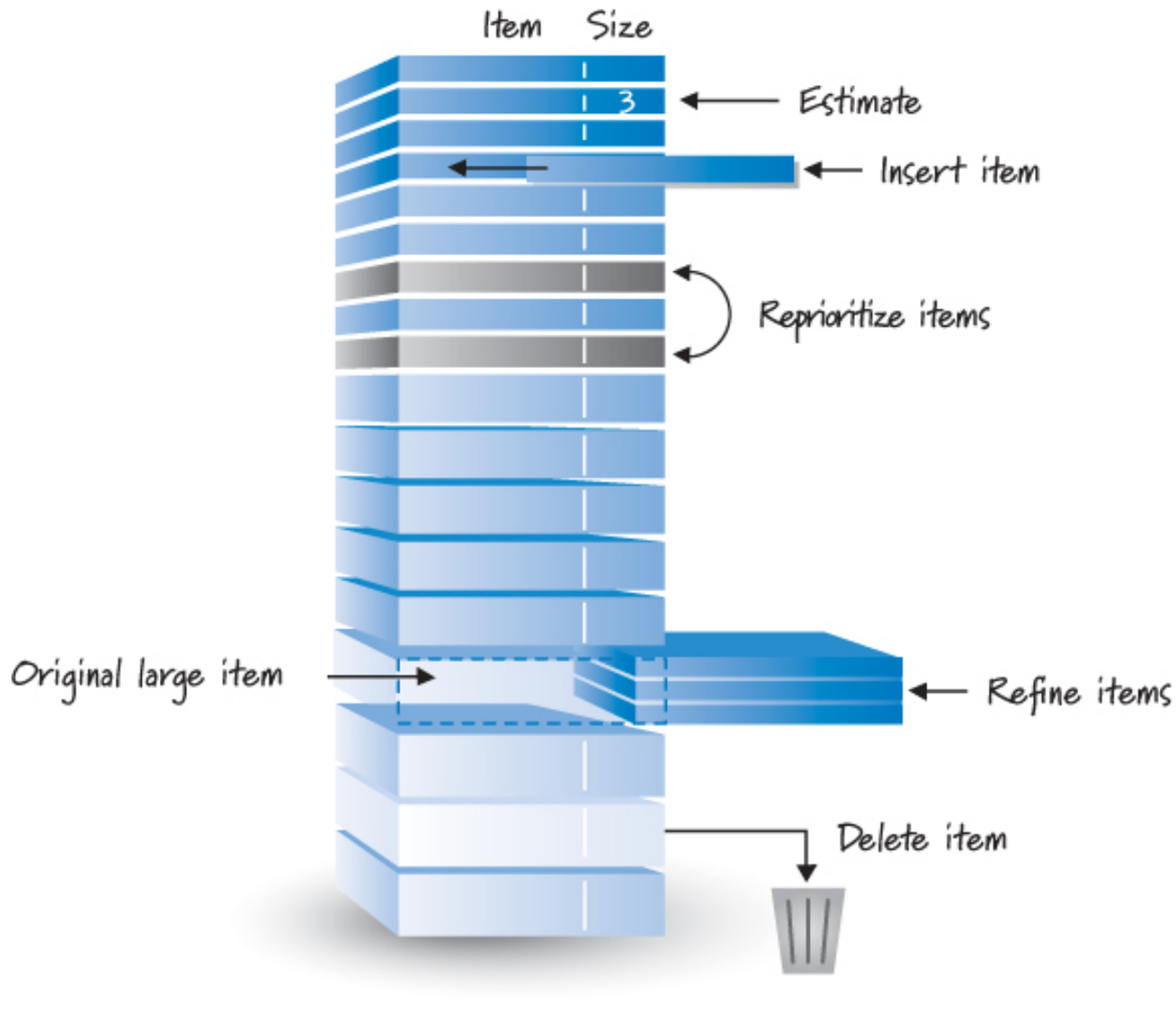
Product Backlog

- The development team lists the items to be built and estimates the time that each item will take to do (in days).



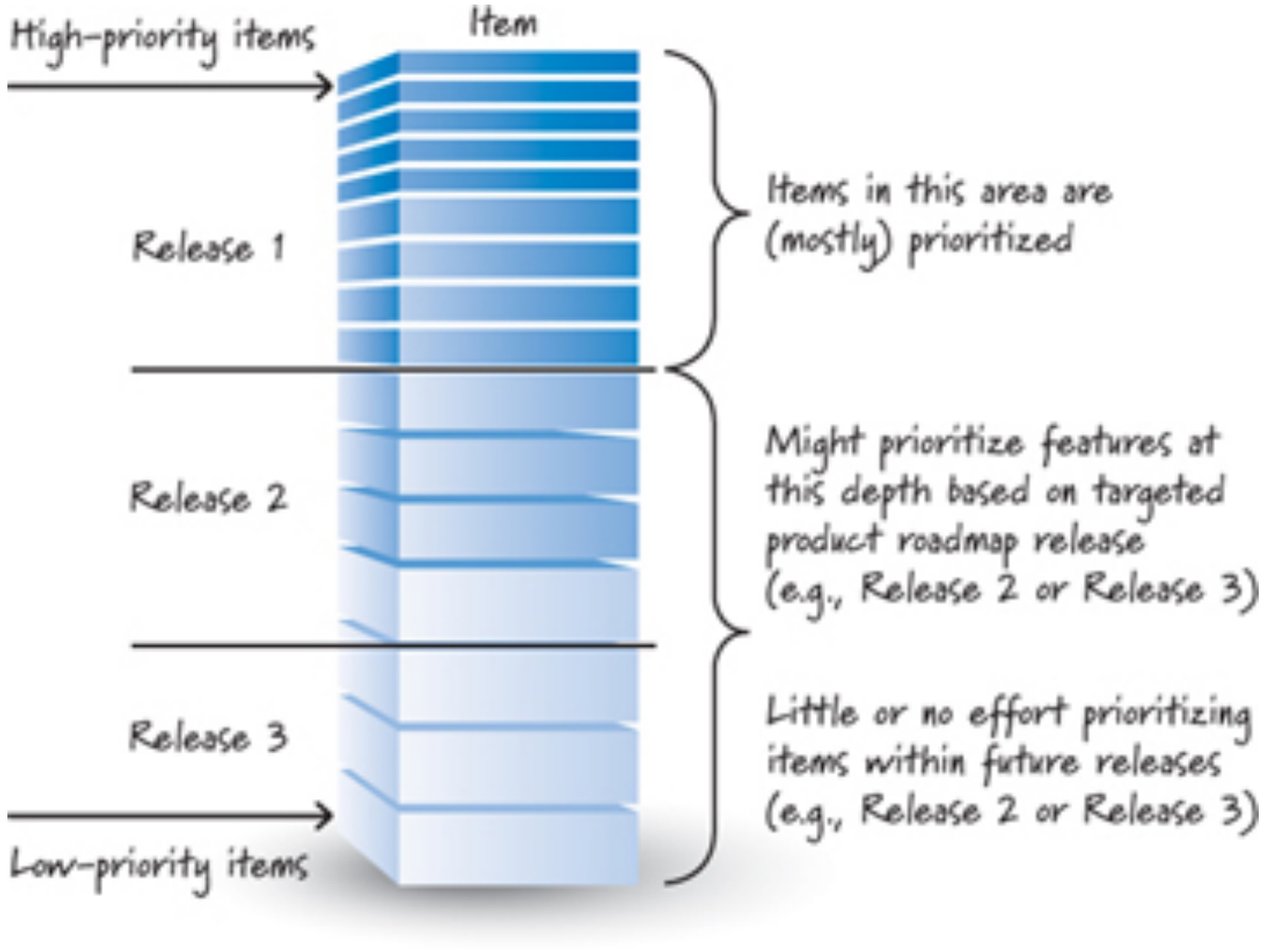
Product Backlog

- The Product Backlog changes over the duration of the project as:
  - Items get done and removed from the backlog
  - New items are identified and added
  - The team spots defects that need fixes
  - Vague long term items get clarified and split into many simpler, do-able, near term items



Product Backlog

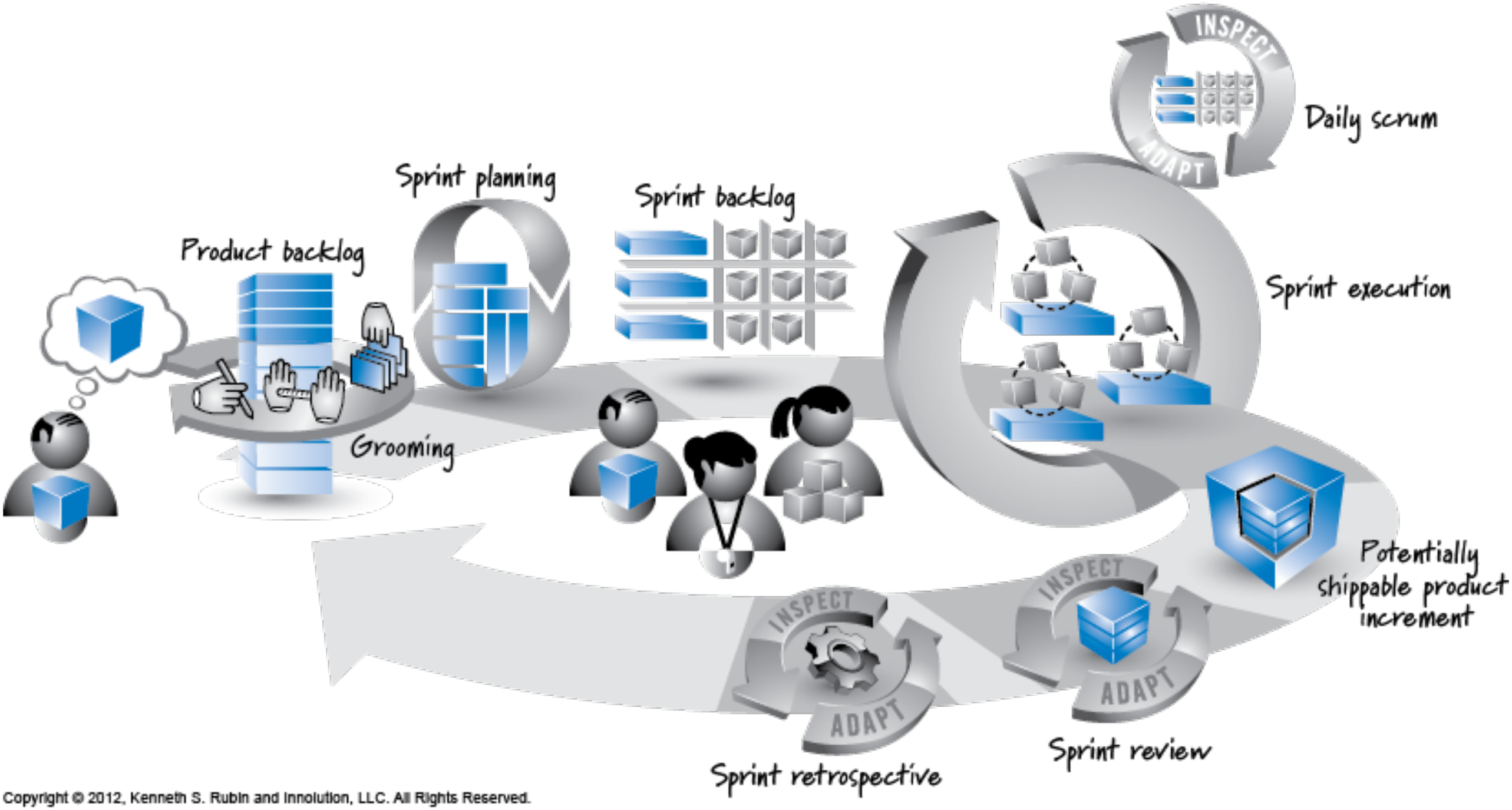
- The items are prioritised according to:
  - the logical build sequence
  - value of the feature
  - cost to build
  - risk to the project
  - development sequence
- Want to get risky, high impact items out of the way early
- Need to build the foundations first



Sprints

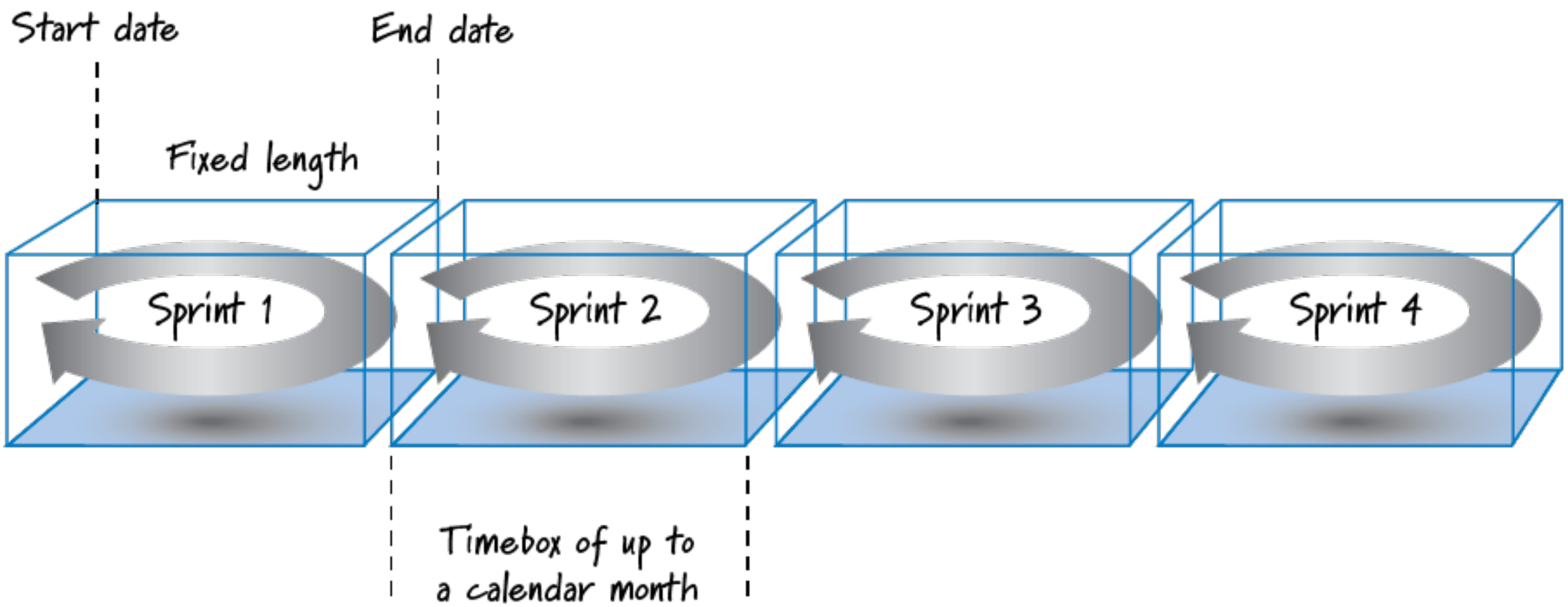
- The development team build the product in a series of **sprints**.
- The sprints are of **fixed duration** (usually 2-4 weeks).
- Each Sprint includes:
  - Sprint Planning to create the **Sprint Backlog**.
  - Sprint execution
  - Release of a **Potentially Shippable Product**.
  - A Sprint **Review**.
  - A Sprint **Retrospective**.
- The team keeps doing Sprints until the product development is finished.

Scrum Framework



https://www.scrumalliance.org/why-scrum

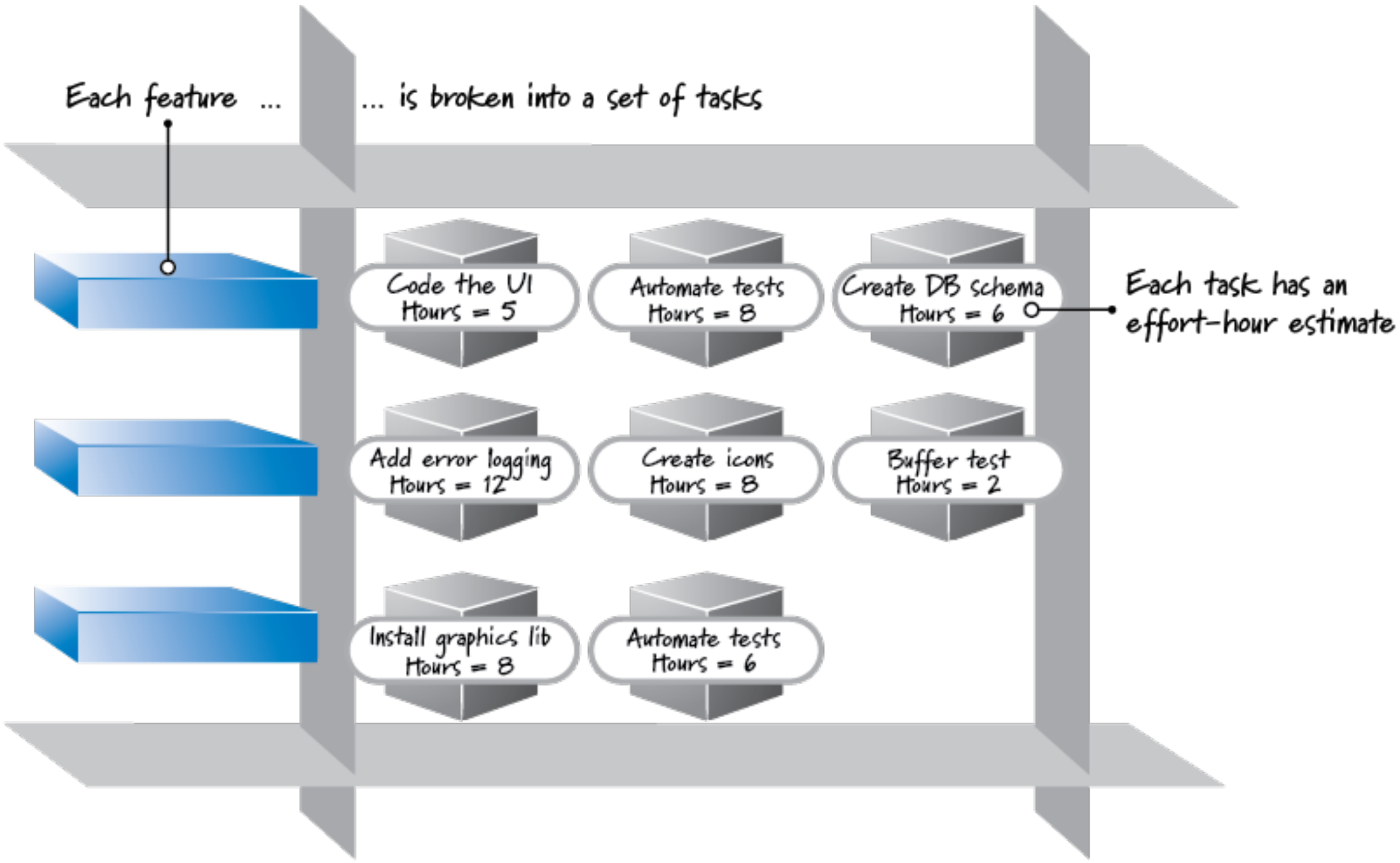
Sprints



Sprint Planning

- The Product Backlog is generally more than can be done in a single sprint.
- So, in Sprint Planning, the team:
  - in negotiation with the Product Owner, selects the subset of the Product Backlog that they will do in this sprint
  - creates a **Sprint Backlog** which is the list of **tasks** which must be done to build this subset
    - decides who does what task (assigning the tasks)
    - estimates how long each task will take (in hours)
    - decide the sequence of tasks (dependencies)
    - re-plan until everyone is happy with the **Product Backlog** subset and the **Sprint Backlog**
    - everyone then **commits** to the sprint
- After planning, the subset of the Product Backlog selected for building in this Sprint DOES NOT change.

Sprint Planning



Sprint Execution

- The **Daily Scrum** is a stand up in-person team meeting where team members sync up on progress so far and plan the day.
- The Scrum meeting is at a fixed time every day and only lasts about 10 minutes.
- Everyone answers the questions:
  - What did I accomplish since my last Scrum?
  - What do I plan to work on until the next Scrum?
  - Are there, or do I foresee, any impediments?
- Status is updated on the Scrum Board.
- The team agrees who tackles any impediments. The team does not seek to solve problems at the Scrum meeting.

Scrum Board

- The team keeps track of progress on that tasks on a Scrum Board (a.k.a. Task or Kanban Board)

To Do	Doing	Done
Task 1 2 day Task 2 1 day		
Task 3 4 day		

Scrum Board

- Team members pick up tasks and add theirs name and deadlines.
- Any delays are notified.
- New tasks are added as the come up.

To Do	Doing	Done
Task 2 1 day	Task 1 ~~2~~ 4 day SC Mon	
Task 3 4 day		
Task fix 1 day		

Scrum Board

Parking Lot
Task 3 4 day

- Eventually everything ends up Done or Parked to do in a later Sprint.

To Do	Doing	Done
		Task 1 2 day | SC Mon
		Task 2 1 day | LH Mon
		Bug fix 1 day | LW Mon

Potentially Shippable Product Increment

- The **Potentially Shippable Product** is a working prototype.
- It could be internally (to stakeholders & test users) or externally (to customers) shippable
- A feature is considered done if the product is tested and the specification is met.

Sprint Review

- All stakeholders, the Product Owner and the development team inspect the Potentially Shippable Product.
  - What features are working well?
  - What features are not working so well?
  - Should the Product Backlog be changed?
- The Product Owner updates the Product Backlog based on the review outcome
- The Product Owner sorts the Product Backlog according to the latest priorities

Sprint Retrospective

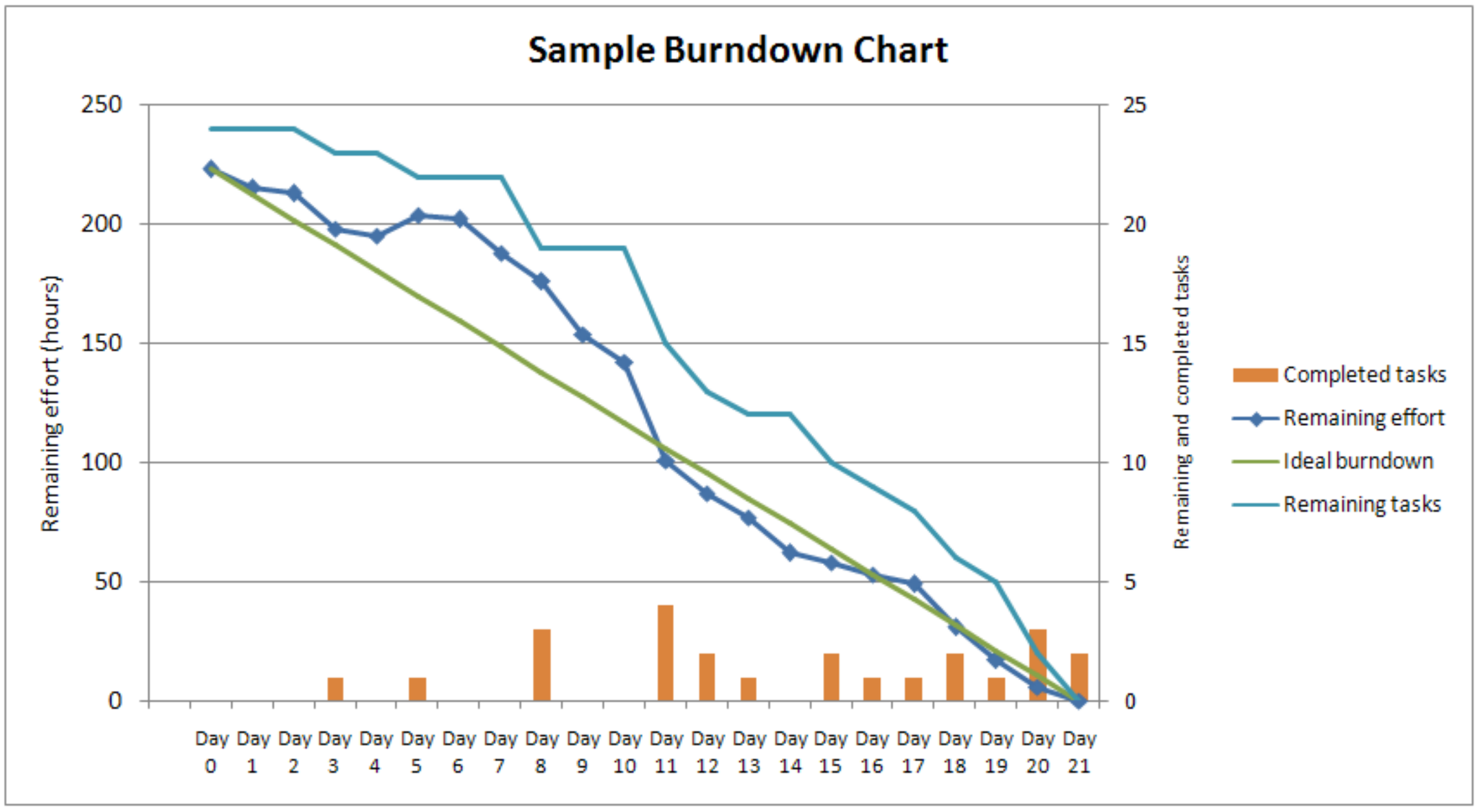
- The Product Owner and development team consider the development process used during the sprint
  - What worked well?
  - What did not work so well?
  - What will we do differently, if anything, in the next sprint?
- This ensures that the process is being continually improved

Scrum

- Scrum is a project management framework, it isn’t just for software development projects.

Sprint Burndown Chart

- Add up the number of estimated days work left in the 'To Do' and 'Doing' columns. Update at every Scrum.



Project Closure

- Complete documentation
- Lesson learned
- Hand over to operations team

Scrum Examples

Agile Simulation: The Daily Standup, Agile Training Videos, YouTube

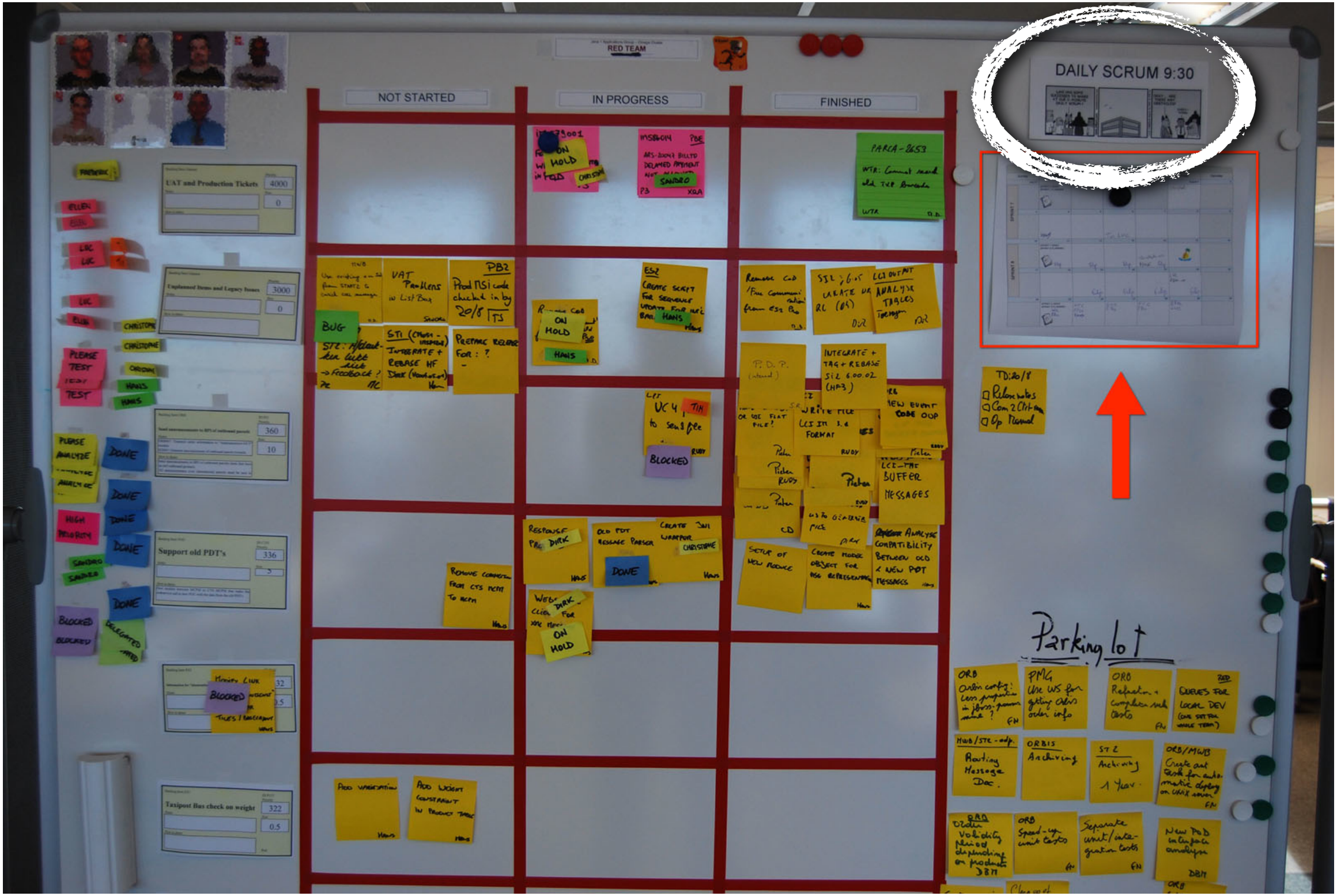
Scrum Board


RED TEAM
NOT STARTED
IN PROGRESS
FINISHED
DAILY SCRUM 9:30
UAT and Production Tickets
Unplanned Items and Legacy Issues
Support old PDT's
Taxipost Bus check on weight
PARCA-2653
Parking lot

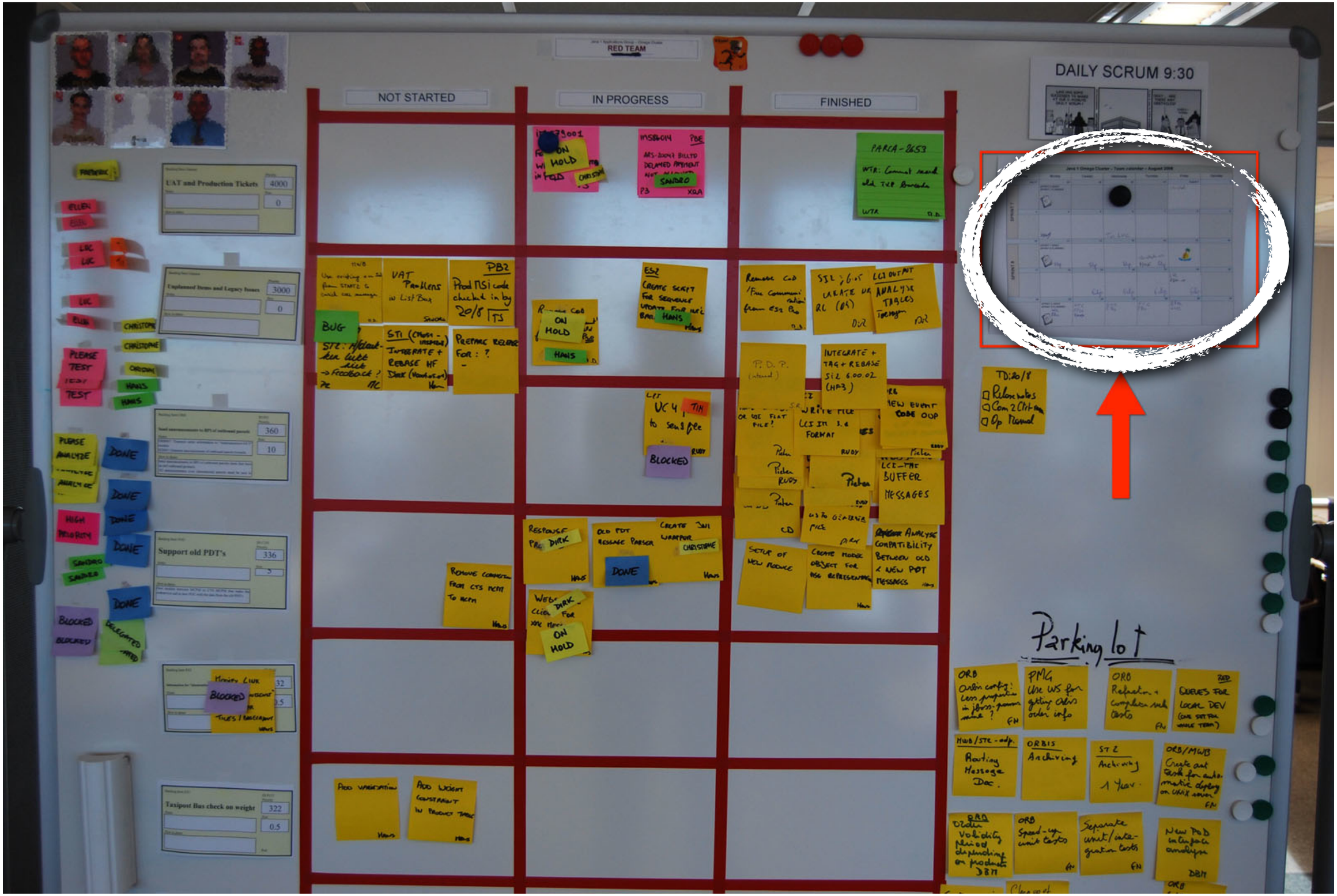
Scrum Board


RED TEAM
NOT STARTED
IN PROGRESS
FINISHED
DAILY SCRUM 9:30
Parking lot

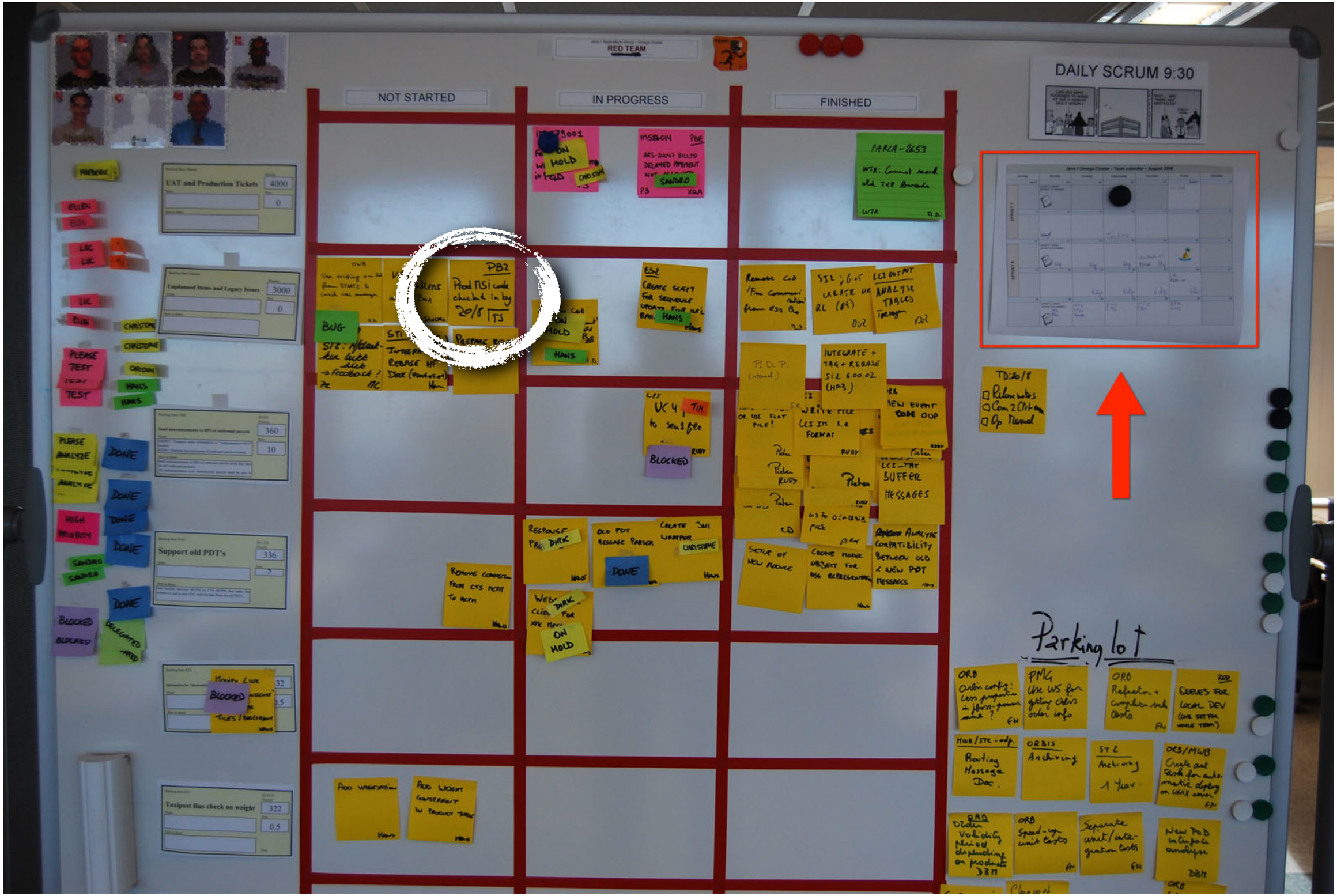
Scrum Board


RED TEAM
NOT STARTED
IN PROGRESS
FINISHED
DAILY SCRUM 9:30
Parking lot

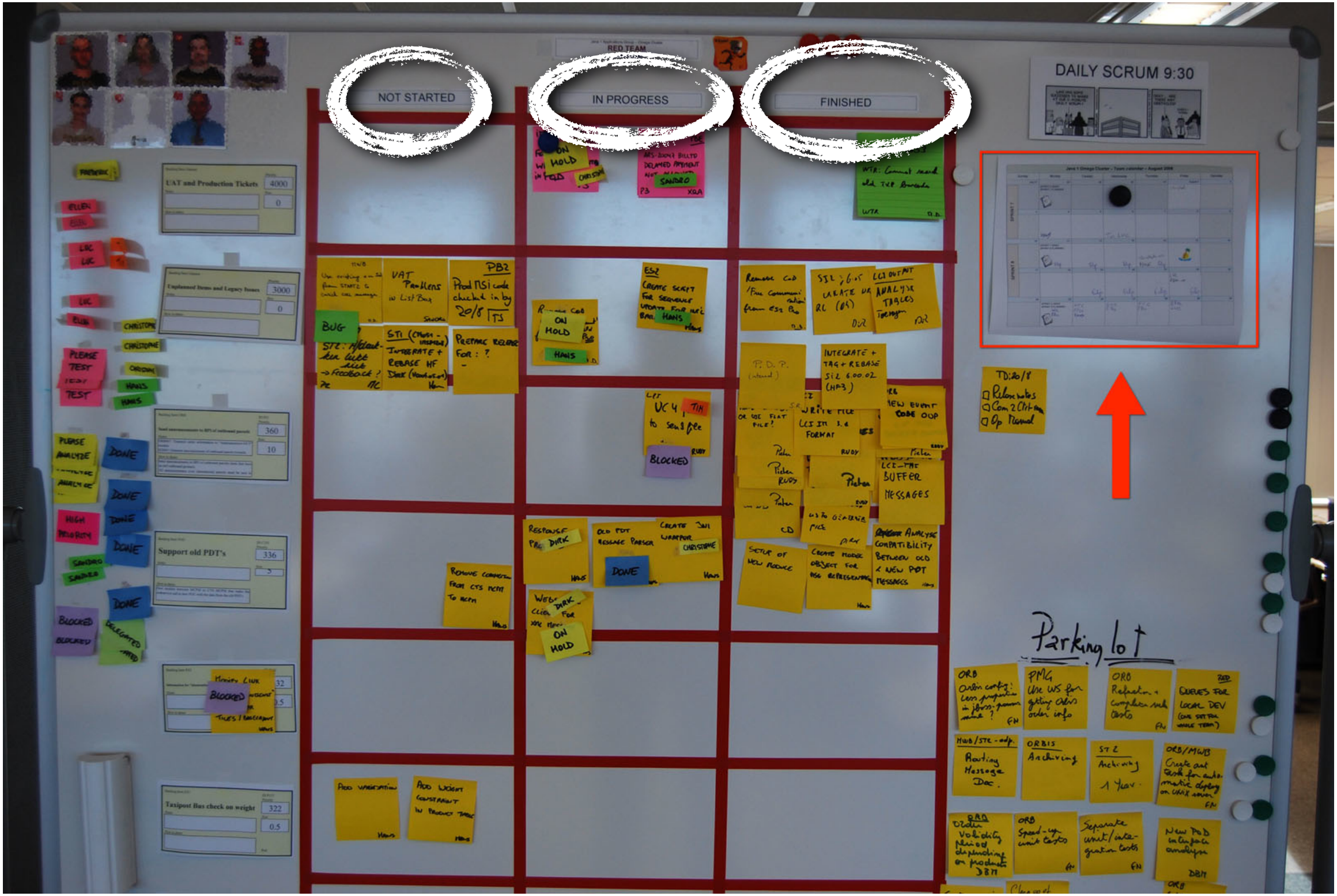
Scrum Board


RED TEAM
NOT STARTED
IN PROGRESS
FINISHED
DAILY SCRUM 9:30
UAT and Production Tickets
Unplanned Items and Legacy Issues
Support old PDT's
Taxipost Bus check on weight
Parking lot

Scrum Board


RED TEAM
NOT STARTED
IN PROGRESS
FINISHED
DAILY SCRUM 9:30
UAT and Production Tickets
Explained Items and Legacy Issues
Support old PDT's
Taxipost Bus check on weight
Parking lot

Scrum Board


NOT STARTED
IN PROGRESS
FINISHED
DAILY SCRUM 9:30
Parking lot

Scrum Board


RED TEAM
NOT STARTED
IN PROGRESS
FINISHED
DAILY SCRUM 9:30
UAT and Production Tickets
Explained Items and Legacy Issues
Support old PDT's
Taxipost Bus check on weight
Parking lot

Scrum Board


RED TEAM
NOT STARTED
IN PROGRESS
FINISHED
DAILY SCRUM 9:30
UAT and Production Tickets
Unplanned Items and Legacy Issues
Support old PDT's
Taxipost Bus check on weight
PARCA-2653
BUG
VAT Problems
PB2
Prod NSI code checked in by 20/8 TJ
ON HOLD
BLOCKED
DONE
Parking lot

GitHub

This repository Search Pull requests Issues Gist

ChrisBleakley / Test

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 1 Wiki Pulse Graphs Settings

Game

Details Add cards Fullscreen

To Do 2

Enter a note

Add note

Write Test Plan

Added by ChrisBleakley

Write main ()

Added by ChrisBleakley

In Progress 0

Finished 0

Parking Lot 0

Problems with Scrum

- Often works well for software development, doesn’t work well for software maintenance because the development team isn’t available.

Business Aspects

- Most software contracts for custom systems are based around a specification, which sets out what has to be implemented by the system developer for the system customer.
- However, this precludes interleaving specification and development as is the norm in agile development.
- A contract that pays for developer time rather than functionality is required. However, this is seen as a high risk my many legal departments because what has to be delivered cannot be guaranteed.

Agile and Plan-Driven Methods

- Most projects now use a balance between plan-driven and agile methods.

Extensions of Scrum

- Distributed Scrum allows for remote workers. Relies on good development and communication tools.
- Multi-team Scrum. Used when the project is so large that it is split across several teams. There is a daily Scrum of Scrums where representatives from each team meet.

Summary

- Agile methods are incremental development methods that focus on rapid software development, frequent releases of the software, reducing process overheads by minimising documentation and producing high-quality code.
- Scrum is the most popular, other agile methods include Kanban, DevOps, Design Thinking, eXtreme Programming (XP), Test-Driven Development (TDD).