Introduction to Agile

Lesson 2: Agile Methods and Practices - SCRUM

###### Lesson Objectives

* Introduction to SCRUM
* Scrum Framework
  + Scrum Roles
    - Product Owner
    - Scrum Master
    - Team
  + Ceremonies
    - Sprint planning
    - Sprint review
    - Sprint retrospective
    - Daily scrum meeting
* Artifacts
  + Product backlog
  + Sprint backlog
  + Burndown charts

###### Lesson Objectives

* Definition of “Ready”
* Definition of “Done”
* Introduction to Extreme Programming
* Introduction to Lean Software Development
* Principles of Lean Software Development
* What is Kanban?
  1. : Agile Methods and Practices - SCRUM

###### Introduction to SCRUM

* Agile way of project management
* A team based collaborative approach
* Iterative & incremental development
* Always focus to deliver “Business Value”

Wikipedia definition:

Scrum is an iterative and incremental agile software development framework for managing software projects and product or application development.

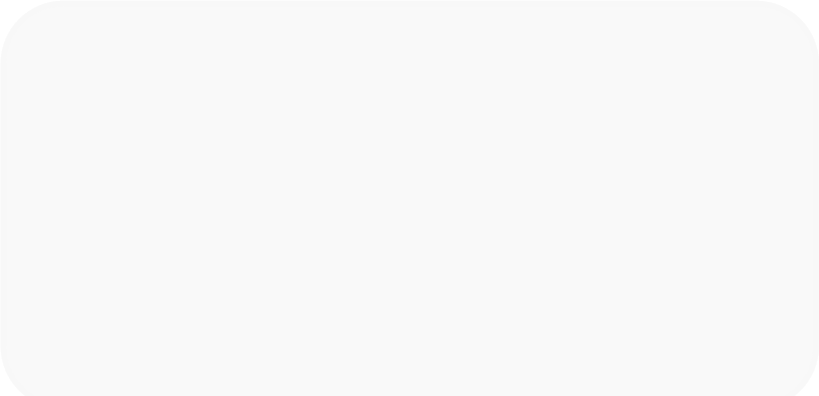
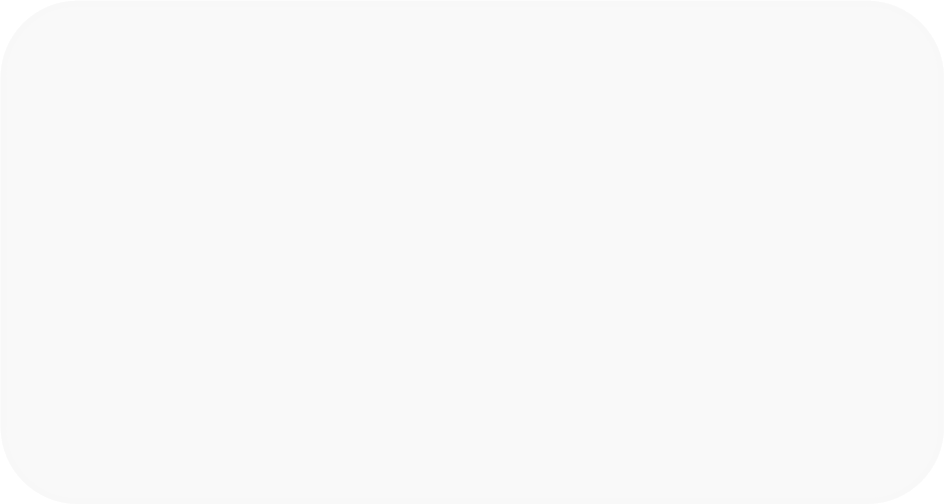
[www.scrumalliance.org:](http://www.scrumalliance.org/)

Scrum is an agile framework for completing complex projects. Scrum originally was formalized for software development projects, but works well for any complex, innovative scope of work. The possibilities are endless. The Scrum framework is deceptively simple.

* 1. : Agile Methods and Practices - SCRUM

###### Scrum framework

Roles



#### Product owner

* + - ScrumMaster

#### Team

Ceremonies

#### Sprint planning

* Sprint review

#### Sprint

retrospective

#### Daily scrum

meeting

Artifacts

#### Product backlog

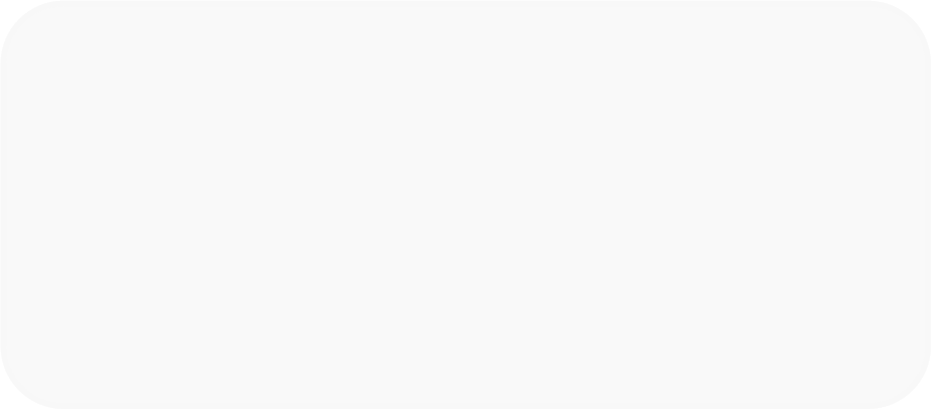
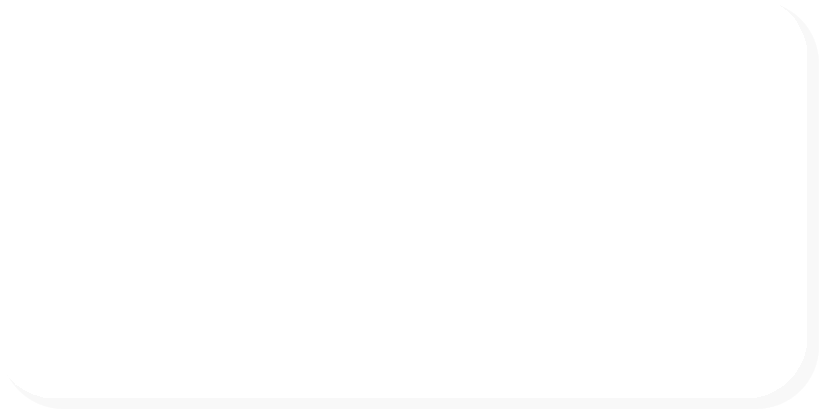
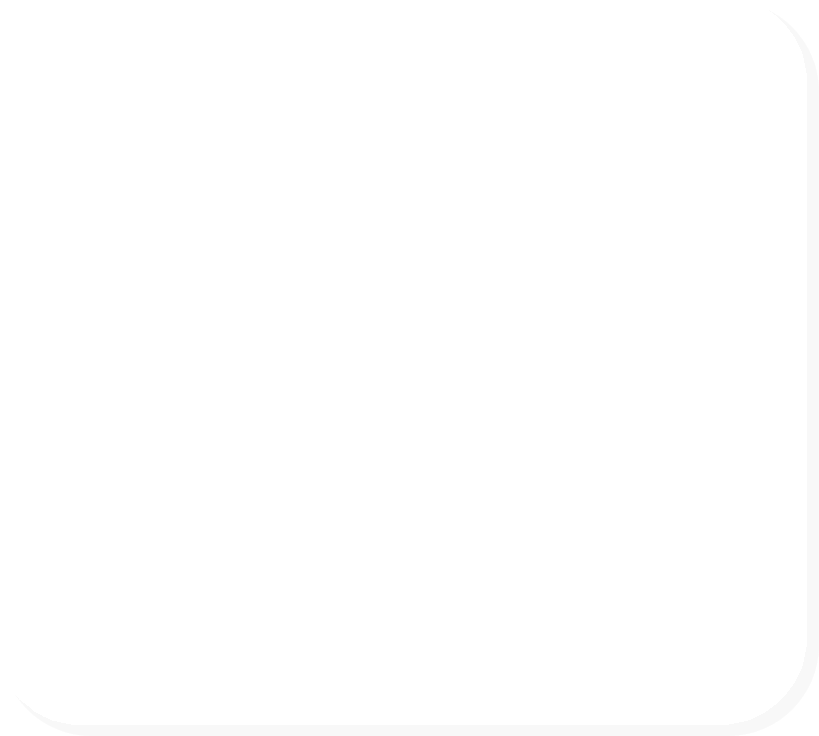
* + Sprint backlog

#### Burndown charts

* 1. Agile Methods and Practices - SCRUM

###### Scrum framework

Roles



#### Product owner

* + - ScrumMaster

#### Team

Ceremonies

* + - * Sprint planning

#### Sprint review

* + - * Sprint

#### retrospective

* + - * Daily scrum meeting

Artifacts

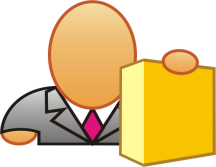
#### Product backlog

* Sprint backlog

#### Burndown charts

* + 1. : SCRUM Framework

###### Scrum Roles

* + - * Product Owner
        + Possibly a Product Manager or Project Sponsor
        + Decides features, release date, prioritization, $$$

###### Scrum Master

* + - * + Typically a Project & Process Co-ordinator or Team Leader
        + Responsible for enacting Scrum values and practices
        + Remove impediments / politics, keeps everyone productive

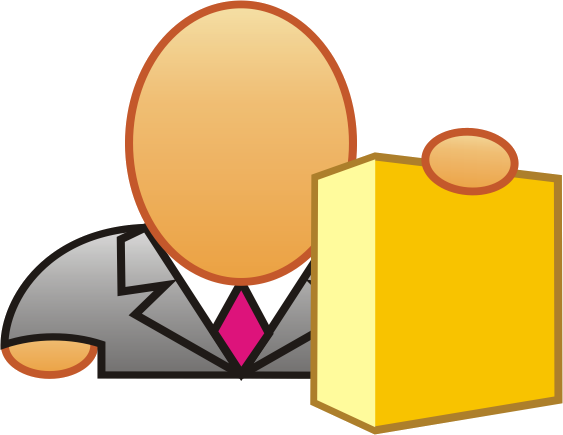
###### Project Team

* + - * + 5-10 members; Teams are self-organizing



* + - * + Cross-functional: QA, Programmers, UI Designers, etc.
        + Membership should change only between sprints

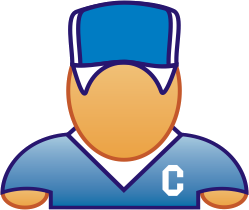
2.1.1.1: SCRUM Roles



###### Product owner

* Define the features of the product
* Decide on release date and content
* Be responsible for the profitability of the product (ROI)
* Prioritize features according to market value
* Adjust features and priority every iteration, as needed
* Accept or reject work results
* Responsible for:
  + Product Vision
  + Stakeholder management
  + Scope Management
  + Cost Management
  + Monitoring Release progress

2.1.1.1: SCRUM Roles



###### The ScrumMaster

* Responsible for facilitation of all ceremonies
* Responsible for enacting Scrum values and practices
* Removes impediments
* Ensure that the team is fully functional and productive
* Enable close cooperation across all roles and functions
* Shield the team from external interferences
  + - 1. : SCRUM Roles



###### The team

* + - * + Typically 5-9 people
        + Cross-functional:

Programmers, testers, user experience designers, etc.

* + - * + Members should be full-time

May be exceptions (e.g., database administrator)

* + - * + Teams are self-organizing

Ideally, no titles but rarely a possibility

* + - * + Membership should change only between sprints

2.1.1 SCRUM framework

###### Agreement - Definition of “Ready” (DoR)

* For a Story to be “Ready”, following criteria have to be met
  + The story should reasonably show INVEST characteristics

**I** – Independent / Immediately actionable

**N** – Negotiable

**V** – Valuable to the customer, user or product

**E** – Estimable **S** – Sized to fit **T** – Testable

* + The business implications of the story have been discussed, any

impacts to finance, customer care have been addressed

* + The User Interaction Design is ready (At the very least wireframes covering all interactions of the story should be available)
  + Any design assets needed for the story have been prepared to a reasonable degree (PSDs for some if not all pages in the Story should be available)

2.1.1 SCRUM framework

###### Agreement - Definition of “Done” (DoD)

* Definition of Done must describe exactly what “done” means
  + Product Owner must pay careful attention when defining the DoD
  + The scrum team must challenge the DoD, if necessary
  + *“What’s not in DoD, is not needed”*
  + Item is either “*done”* or “*not done”*

###### Example:

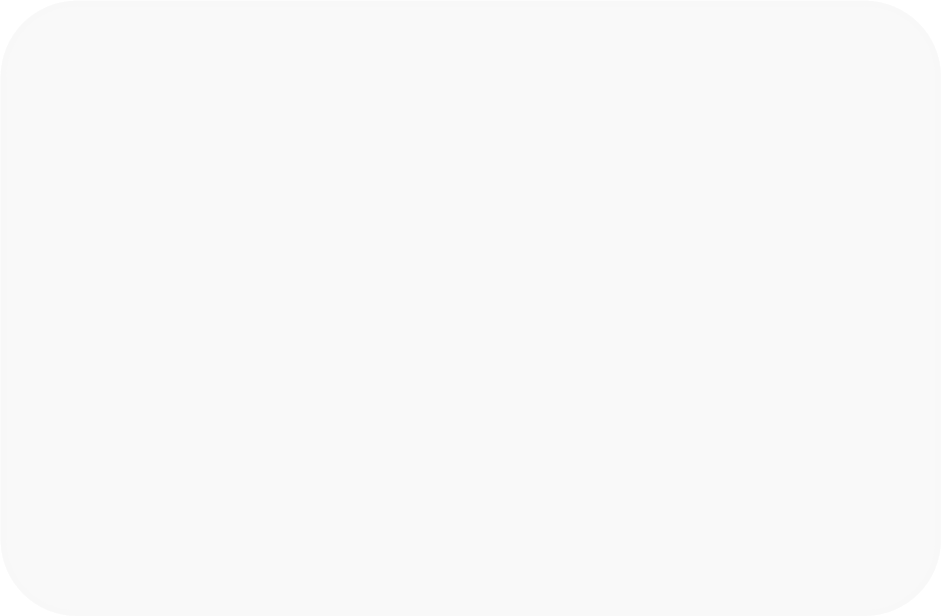
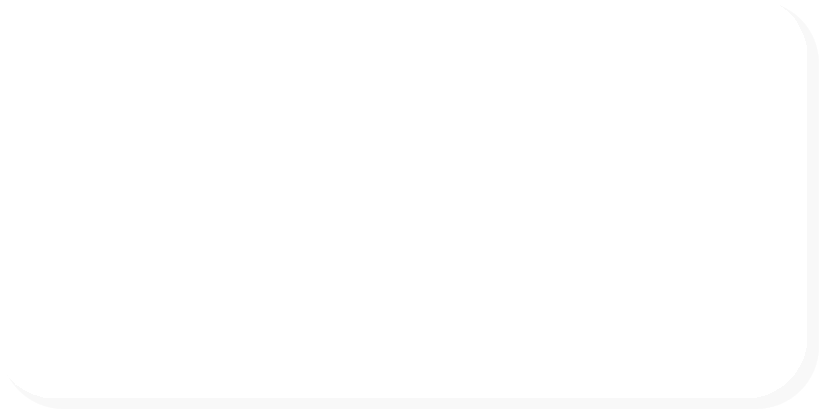
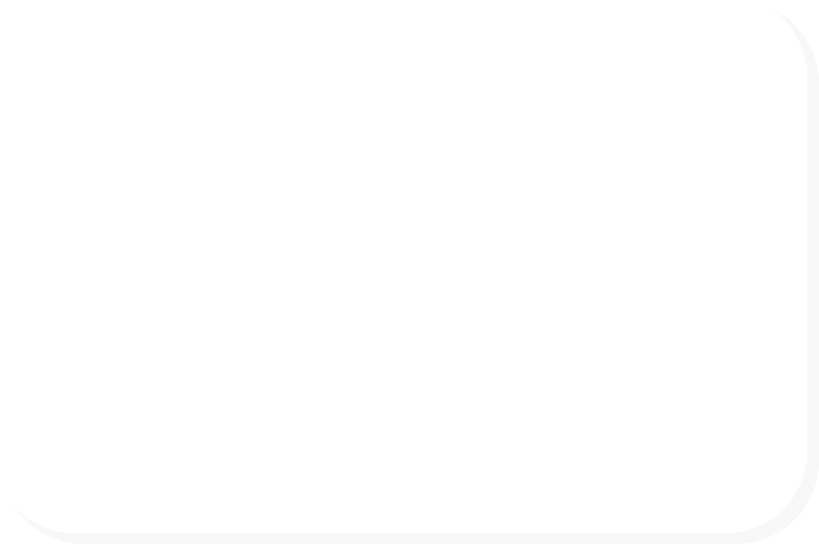
* + Story: Picture upload
    - end user can upload his/her picture from profile settings page
    - picture is shown on the left upper corner of the profile page
    - picture is scaled to fit the profile picture box on the profile page
    - functional tests are passed
    - regression tests are passed
    - design documents are updated
    - user's guide is updated

###### **Does not** define any details of the implementation!

* 1. : Agile Methods and Practices - SCRUM

###### Scrum framework

Roles



#### Product

owner

* + - ScrumMasterCeremonies

#### Team

* + - * Sprint planning

#### Sprint review

* + - * Sprint retrospective
      * Daily scrumArtifmacetseting

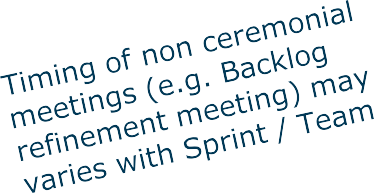
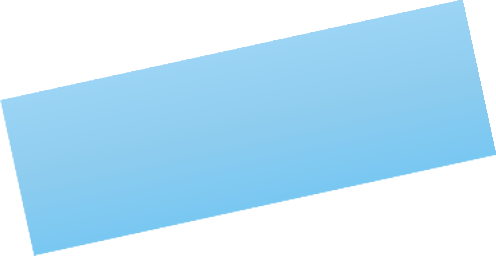
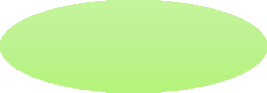
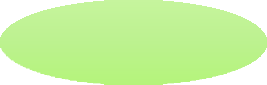
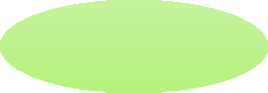
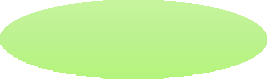
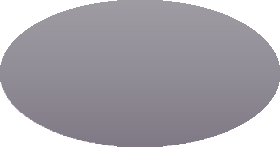
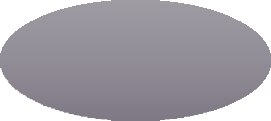
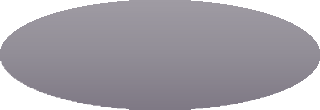
#### Product backlog

* + - * + Sprint backlog

#### Burndown charts

2.1.1 SCRUM framework

###### Scrum with 2 weeks Sprint Duration



9 am

**Sprint Planning**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Scrum Ceremonies and important meetings during one Sprint | | | | | | |
|  | Day 3 |  | Day 4 |  | Day 5 |  |

11

PO + SM + Dev

Team

am

1 pm

3 pm

5

pmv 9 am

Add on meetings

**Daily Scrum**

Day 6

11

SM + Dev + PO

am

1 pm

3 pm

5 pm

Day 1

**Mandatory**

Day 2

Add on meetings SM + Dev + PO

**Daily Scrum**

**Backlog Refineme**

Add on meetings

**Daily Scrum**

**nt**

PO + SM +

Dev

SM + Dev + PO

Day 7

**Optional**

Add on meetings SM + Dev + PO

**Daily Scrum**

**Sprint** (2 weeks)

Add on meetings

**Daily Scrum**

SM + Dev + PO

**nt**

PO + SM +

Dev

meetings

SM + Dev + PO

**Daily Scrum**

Day 8

Day 9

Add on

**Daily Scrum**

SM + Dev + PO

**Backlog Refineme**

Add on meetings

**Daily Scrum**

**Sprint Review**

(PO + SM +

Day 10

**Daily Scrum**

Add on meetings SM + Dev + PO

DevTeam) + key

Stakeholders

**Sprint Retrospection** (PO + SM +

DevTeam)

**meeting but not a scrum ceremony**



**Ceremonies**

**meeting and not a scrum ceremony**

Team capacity

#### Sprint planning



meeting

Sprint prioritization

Product backlog

* Analyze and evaluate product backlog
* Select sprint goal

Sprint goal

Business conditions

Current product

Technology

Sprint planning

* Decide how to achieve

sprint goal (design)

* Create sprint backlog (tasks) from product

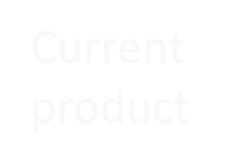
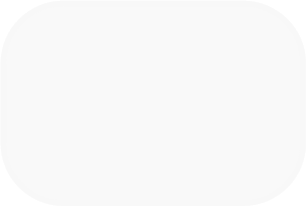
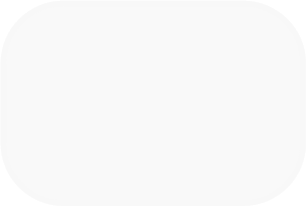
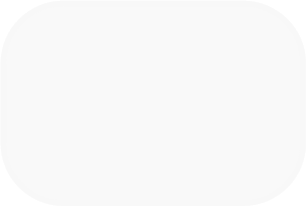
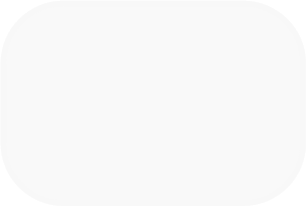
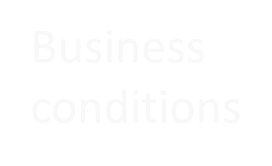
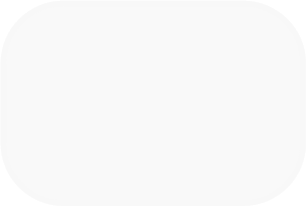
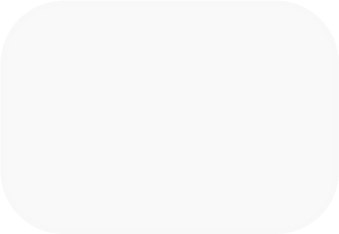
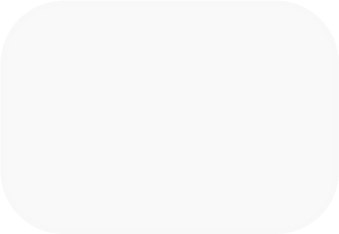
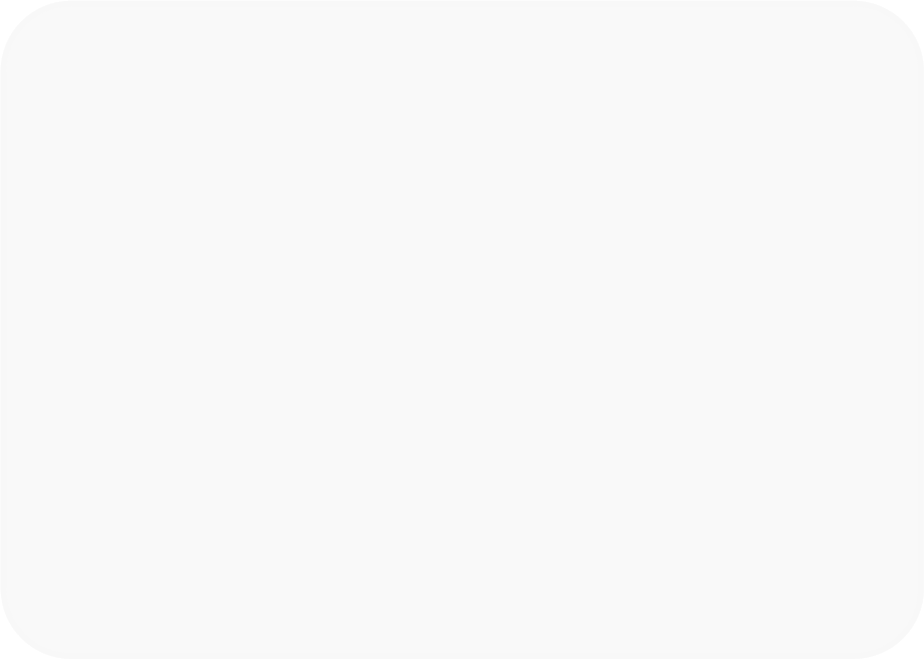
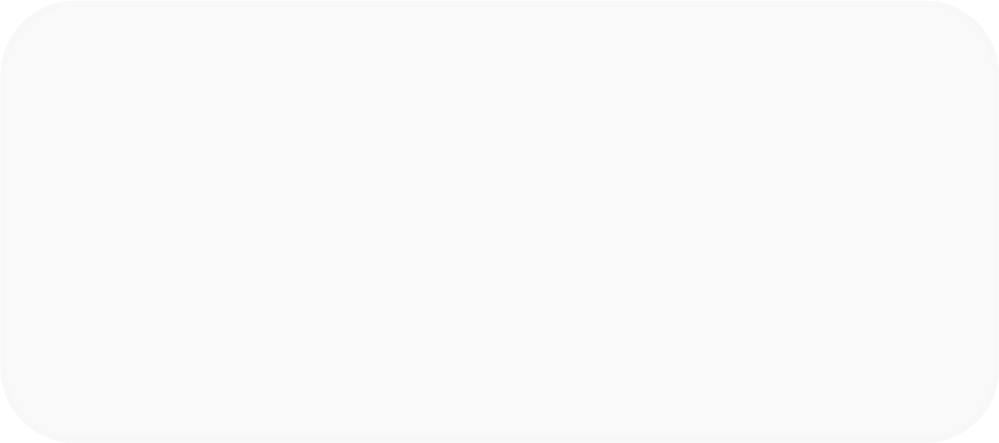
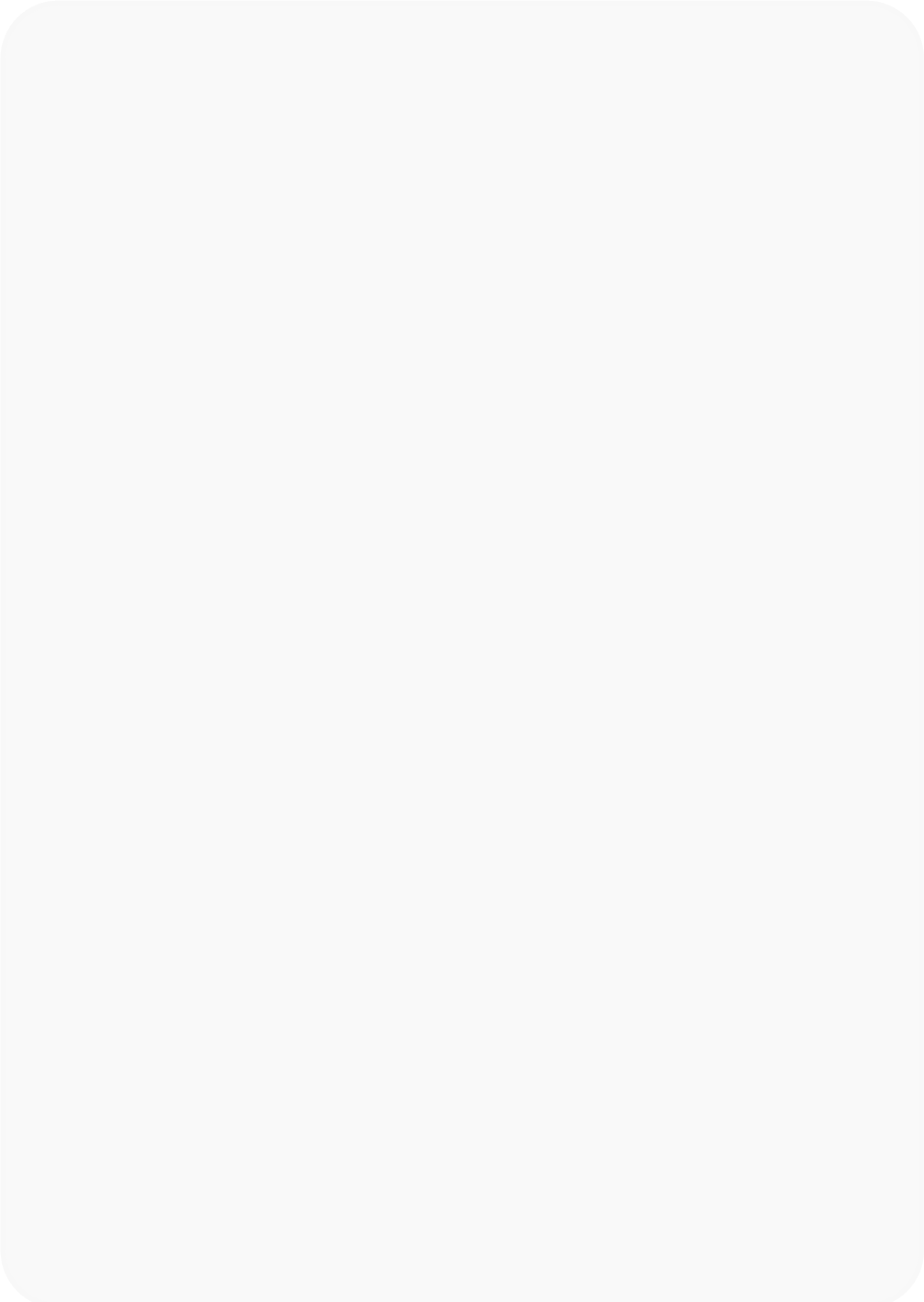
backlog items (user stories

/ features)

Sprint

backlog

* + Estimate sprint backlog in



hours

2.1.1 SCRUM framework

## Sprint planning

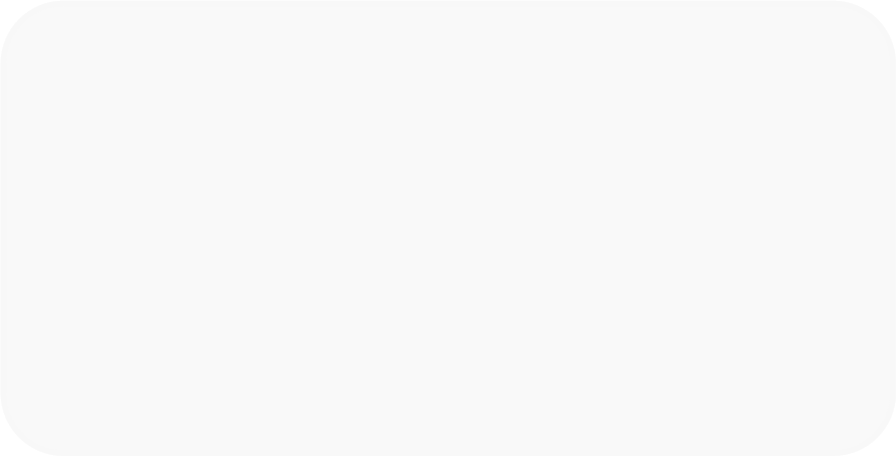
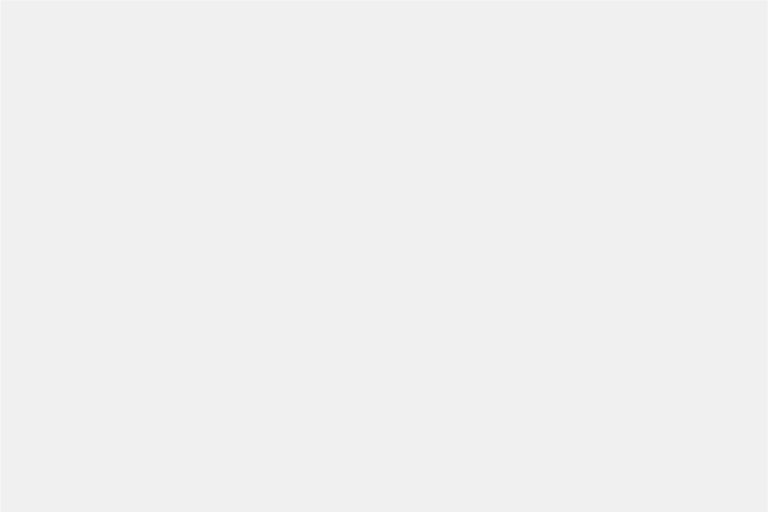
### Team selects items from the product backlog they can commit to completing

* Sprint backlog is created
  + Tasks are identified and each is estimated (1-16

hours)

* + Collaboratively, not done alone by the ScrumMaster

### High-level design is considered



As a vacation planner, I want to see photos of the hotels.

###### Code the middle tier (8

hours)

###### Code the user interface (4) Write test fixtures (4) Code the foo class (6)

Update performance tests (4)

* + 1. SCRUM framework

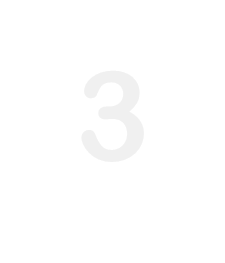
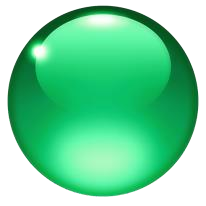
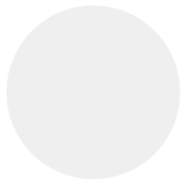
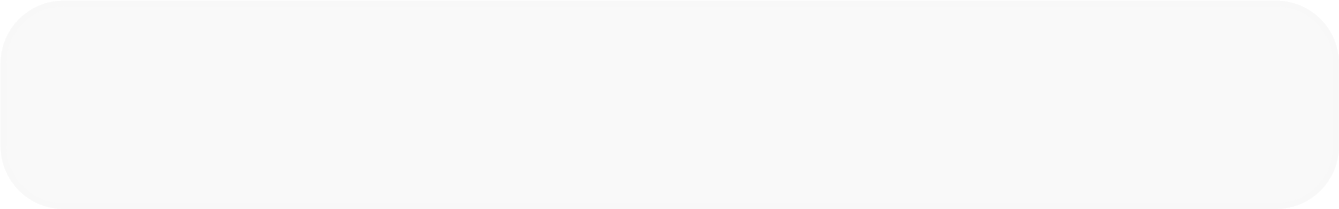
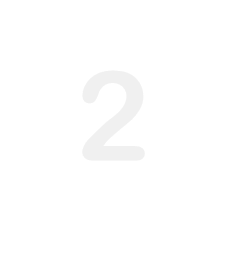
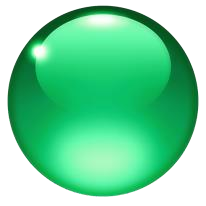
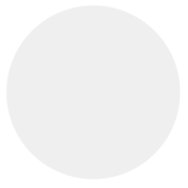
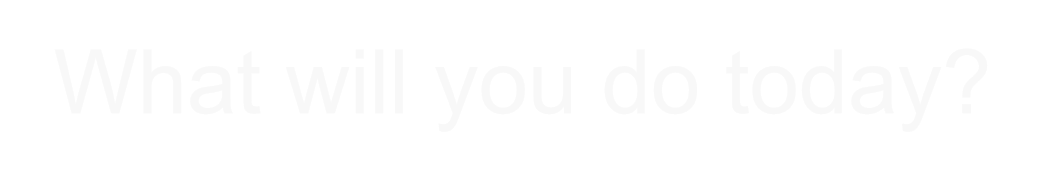
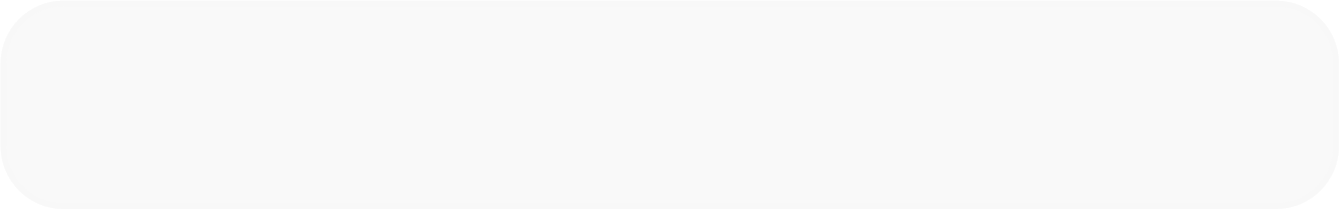
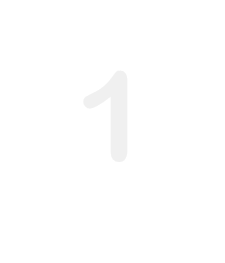
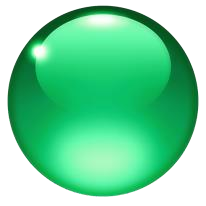
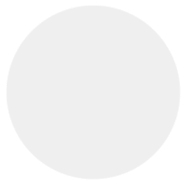
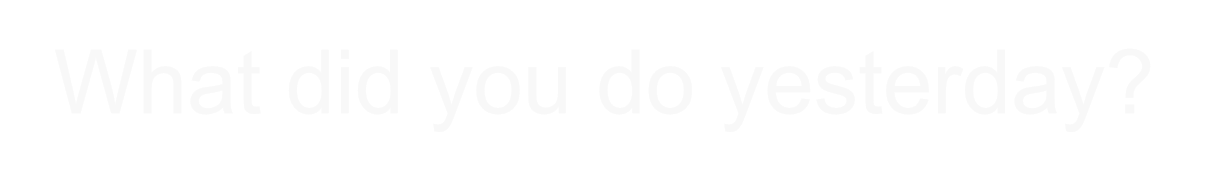
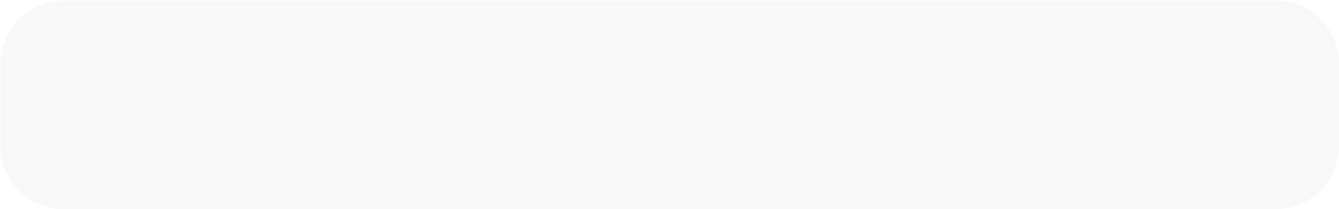
###### The daily scrum

* + - * Parameters
        + Daily
        + 15-minutes
        + Stand-up
      * Not for problem solving
        + Whole world is invited
        + Only team members, ScrumMaster, product owner, can talk
      * Helps avoid other unnecessary meetings

2.1.1 SCRUM framework

###### The daily scrum

Everyone answers 3 questions



# 1

What did you do yesterday?

# 2

What will you do today?

# 3

Is anything in your way?

* These are *not* status for the ScrumMaster
  + They are commitments in front of peers
    1. SCRUM framework

###### The sprint review

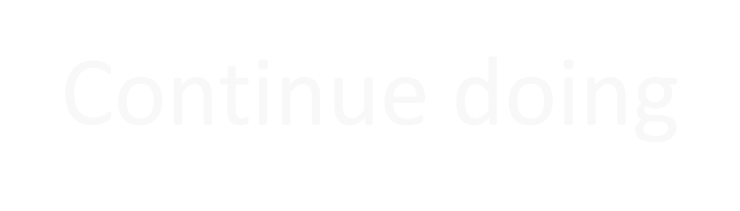
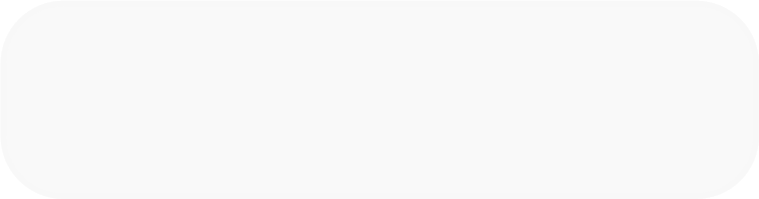
* + - * Team presents what it accomplished during the sprint
      * Typically takes the form of a demo of new features or underlying architecture
      * Informal
        + 2-hour prep time rule



* + - * + No slides
      * Whole team participates
      * Invite the world
    1. SCRUM framework

###### Sprint retrospective

* + - * Periodically take a look at what is and is not working
      * Typically 15–30 minutes
      * Done after every sprint
      * Whole team participates
        + ScrumMaster
        + Product owner
        + Team
        + Possibly customers and others



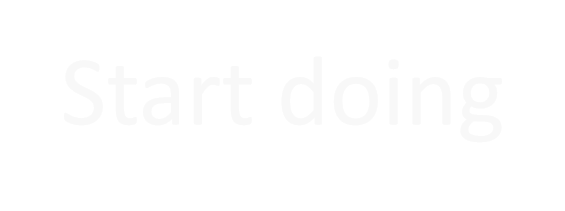
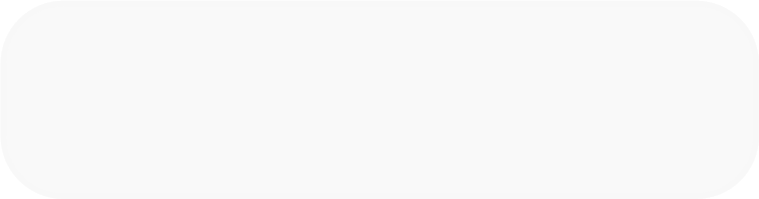
Continue doing

2.1.1 SCRUM framework

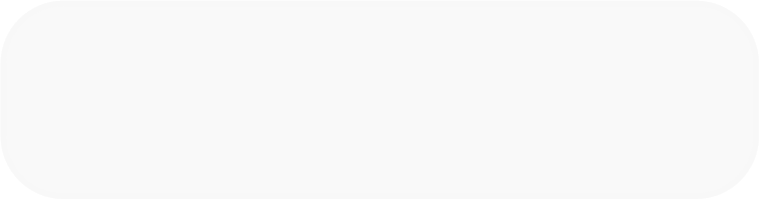
Sprint retrospective

Start / Stop / Continue

Whole team gathers and discusses what they’d like to:



Start doing



Stop doing

This is just one

of many ways to

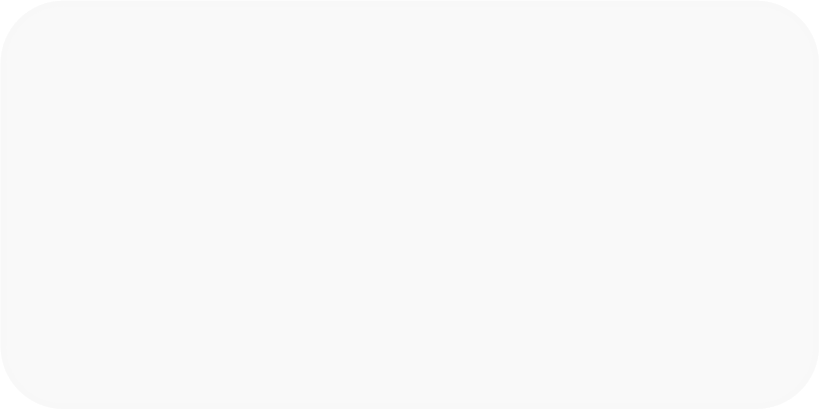
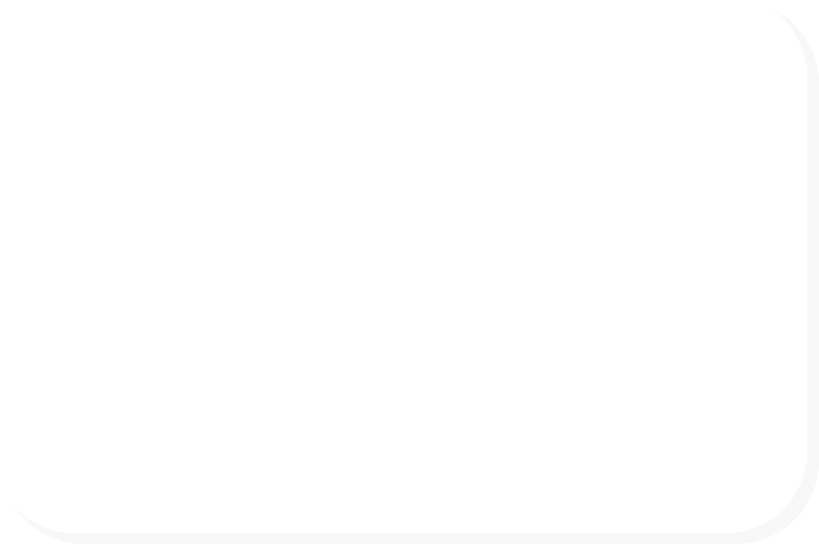
do a sprint

retrospective.

* 1. Agile methods and Practice :SCRUM

###### Scrum framework

Roles



#### Product

owner

#### ScrumMast

er

Ce

* + - Team

remonies

#### Sprint planning

* Sprint review

#### Sprint

e m

retrospectivArtifacts

u

#### Daily scr

meeting

#### Product backlog

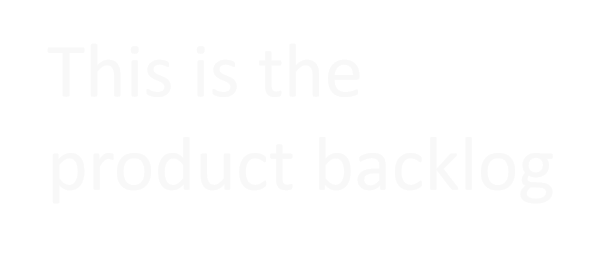
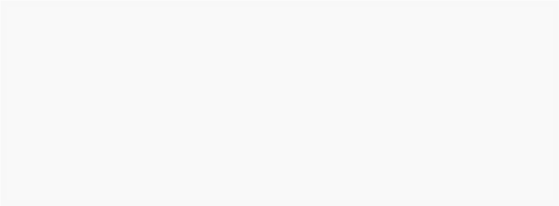
* + Sprint backlog

#### Burndown charts

2.1.1 SCRUM Framework

Product backlog

The requirements



•

•

•

•

•

This is the product backlog

A list of all desired work on the project

Ideally expressed such that each item has value to the users or customers of the product

Prioritized by the product owner

Reprioritized at the start of each sprint

2.1.1 SCRUM Framework

###### Sample Product Backlog

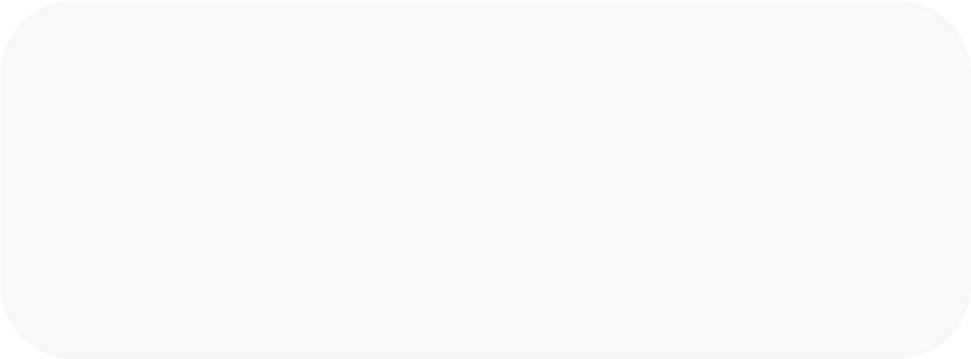
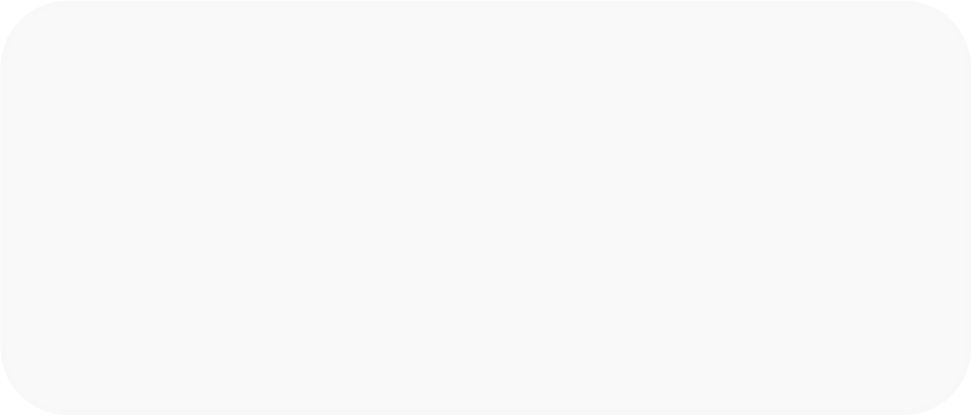
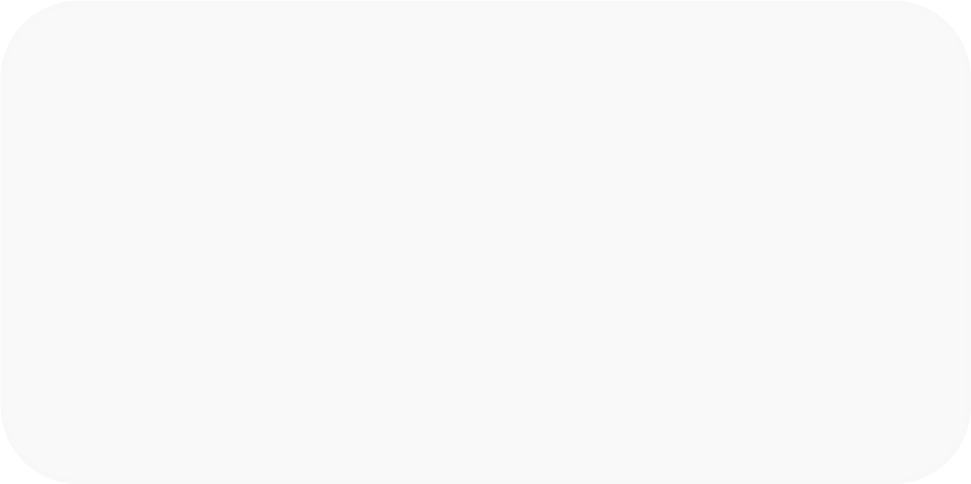
|  |  |
| --- | --- |
| Backlog item | Estimate |
| Allow a guest to make a reservation | 3 (story points) |
| As a guest, I want to cancel a reservation. | 5 |
| As a guest, I want to change the dates of a reservation. | 3 |
| As a hotel employee, I can run Revenue reports | 8 |
| Improve exception handling | 8 |
| ... | 30 |
| ... | 50 |

* + 1. SCRUM Framework

###### The sprint goal

* + - * A short statement of what the work will be focused on during the sprint

Life Sciences



Support features necessary

Database

Application

Make the application run on

SQL Server in addition to

Oracle.

for population genetics studies.

Financial services

Support more technical indicators than company ABC with real-time, streaming data.

2.1.1 SCRUM Framework

Sprint Backlog

* Individuals sign up for work of their own choosing

– Work is never assigned

##### Estimated work remaining is updated daily

* Any team member can add, delete change sprint backlog

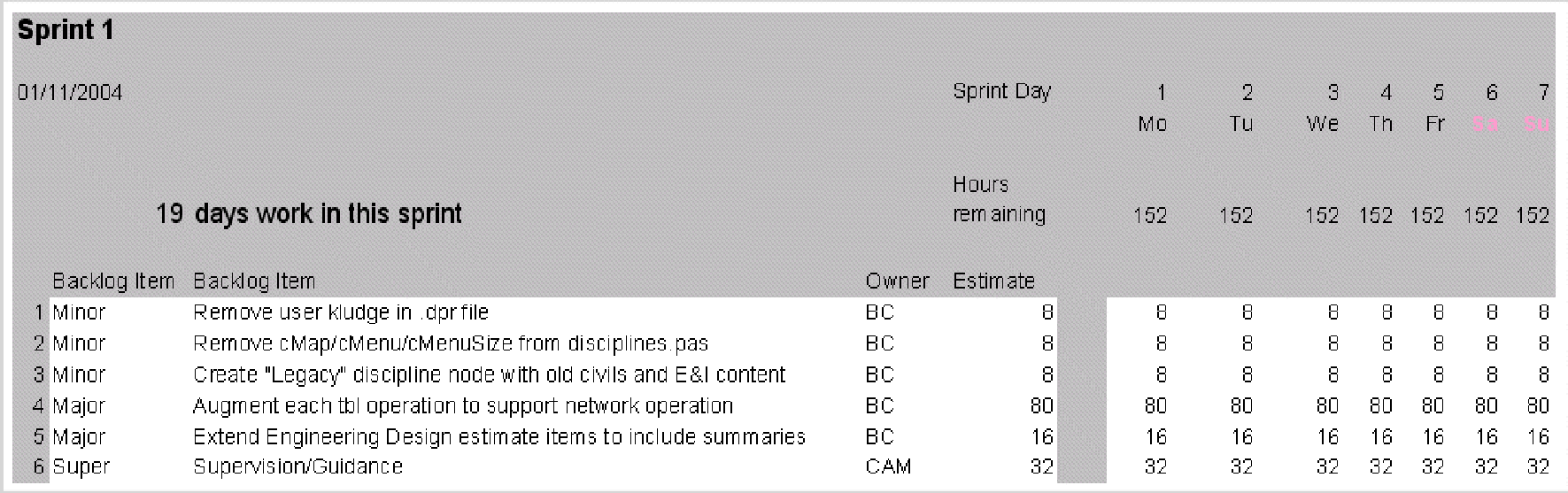
##### Work for the sprint emerges

* If work is unclear, define a sprint backlog item with a larger amount of time and break it down later

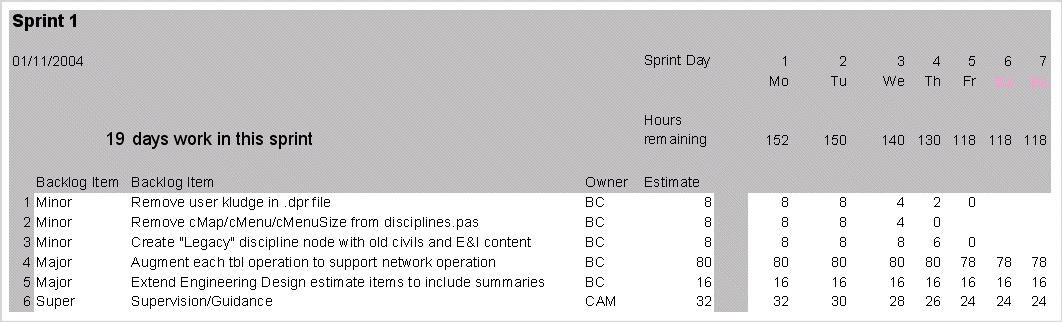
##### Update work remaining as more becomes known

2.1.1 SCRUM Framework

###### Sample Sprint Backlog



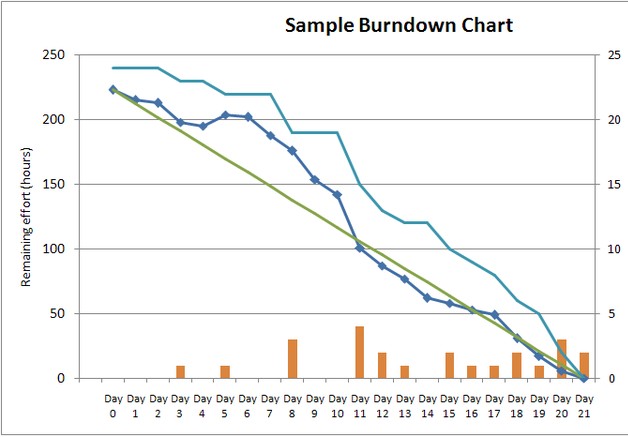
01/11/2015



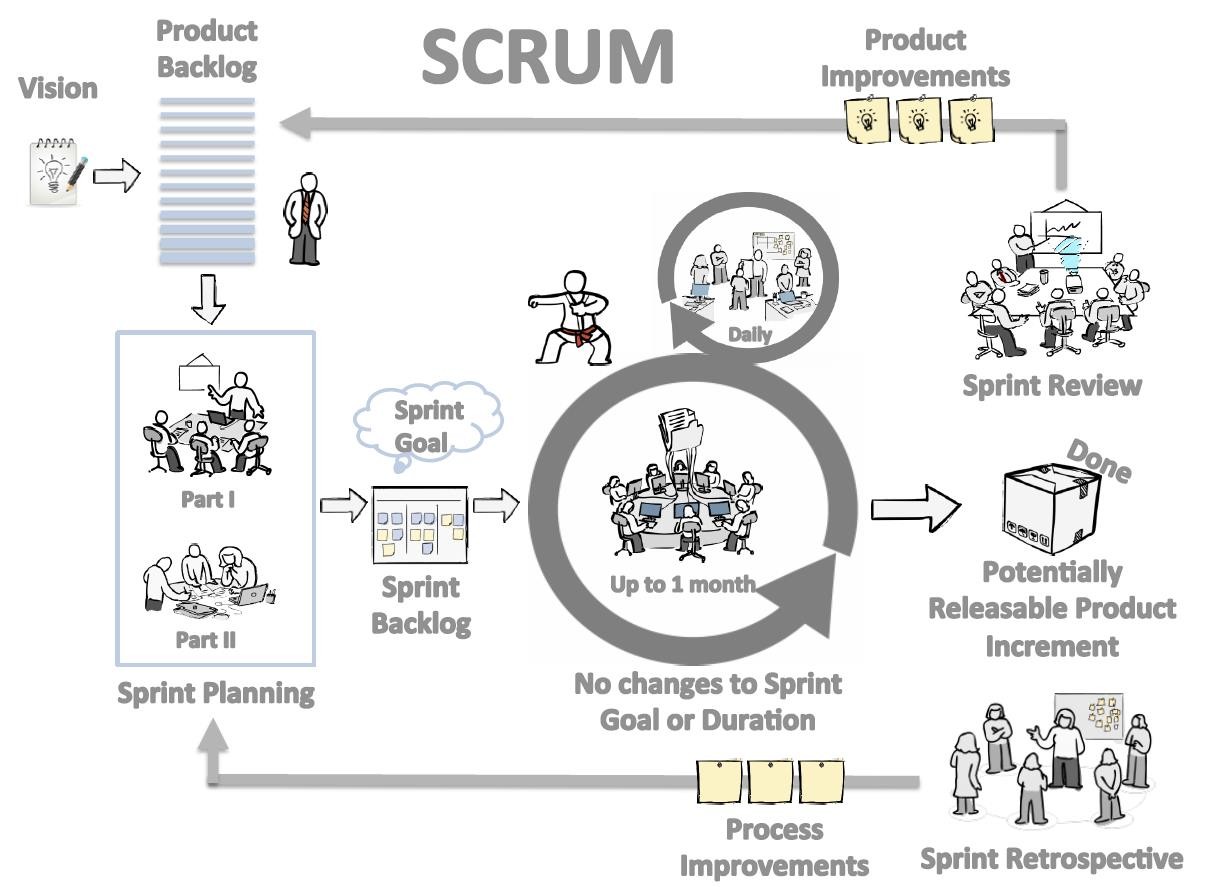
01/11/2015

2.1.1 SCRUM Framework

###### Sprint Burndown

* A display of what work has been completed and what is left to complete
  + one for each developer or work item
  + updated every day
  + (make best guess about hours/points completed each day)
* Gives indication to:
  + No work being performed
  + Not fast enough work
  + Too fast work
* *variation:* Release burndown chart
  + shows overall progress
  + updated at end of each sprint
  1. : Agile Methods and Practices - SCRUM

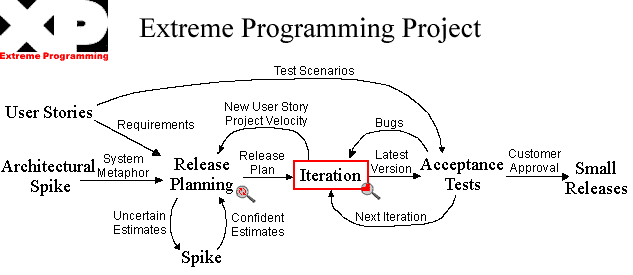
## Scrum at a Glance



* 1. : Agile Methods and Practices – Extreme Programming (XP)

###### Introduction to Extreme Programming

Wikipedia definition:



Extreme Programming (XP) is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements. As a type of agile software development, it advocates frequent "releases" in short development cycles, which is intended to improve productivity and introduce checkpoints where new customer requirements can be adopted.

2.3: Agile Methods and Practices – Extreme Programming (XP)

###### The Rules of Extreme Programming

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Planning** | **Managing** | **Coding** | **Designing** | **Testing** |
| * User stories are written * Release planning creates the release schedule * Make frequent small releases * The project is divided into iterations * Iteration planning starts each iteration | * Give the team a dedicated open work space * Set a sustainable pace * A stand up meeting starts each day * The Project Velocity is measured * Move people   around   * Fix XP when it breaks | * The customer is always available * Code must be written to agreed standards * Code the unit test first * All production code is pair programmed * Only one pair integrates code at a time * Set up a dedicated integration computer | * Simplicity * Choose a system metaphor * Use CRC cards for design sessions * Create spike solutions to reduce risk * No functionality is added early * Refactor whenever and wherever possible | * All code must have unit tests * All code must pass all unit tests before it can be released * When a bug is found tests are created. * Acceptance tests are run often and the score is published |

2.3: Agile Methods and Practices – Lean Software Development

###### Introduction to Lean Software Development

* Lean Software Development is the application of Lean Thinking to the

software development process

* Lean Software Development is more strategically focused than other Agile methodology
* The goals are to develop software in one-third the time, with one-third the budget, and with one-third the defect rate
* "Lean Software Development" is not a management or development methodology in itself, but it offers principles that are applicable in any environment to improve software development"
  1. : Agile Methods and Practices – Lean Software Development

###### Principles of Lean Software Development

* Eliminate waste: Do only what adds value for a customer, and do it

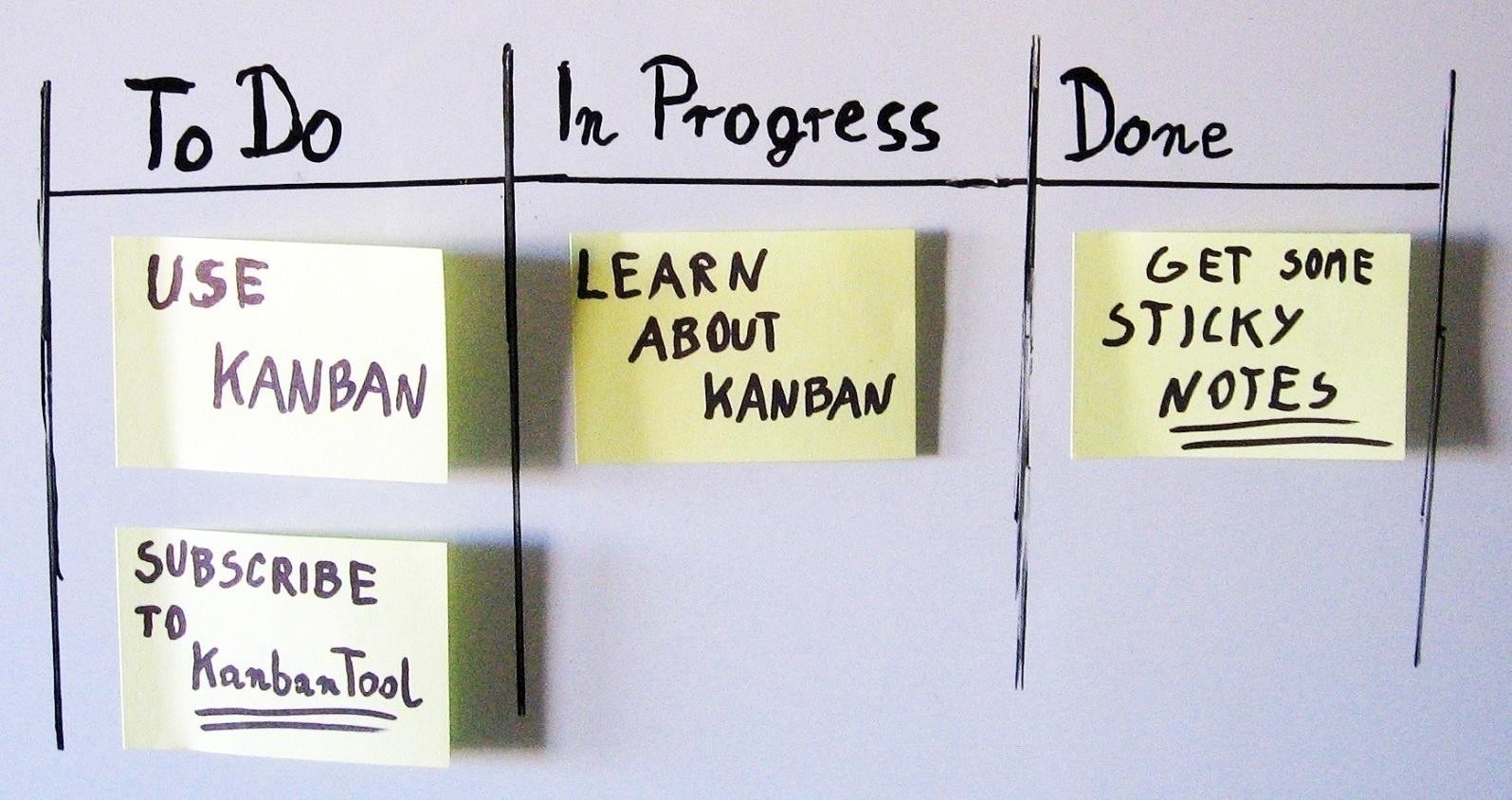
without delay

* Amplify learning: Use frequent iterations and regular releases to provide feedback
* Decide as late as possible: Make decisions at the last responsible moment
* Deliver as fast as possible: The measure of the maturity of an organization is the speed at which it can repeatedly and reliably respond to customer need
* Empower the team: Assemble an expert workforce, provide technical leadership and delegate the responsibility to the workers
* Build integrity in: Have the disciplines in place to assure that a system will delight customers both upon initial delivery and over the long term
* See the whole: Use measurements and incentives focused on achieving the overall goal
  1. : Agile Methods and Practices – Kanban

###### What is Kanban?

* The word Kan means "visual" in Japanese and the word "ban" means

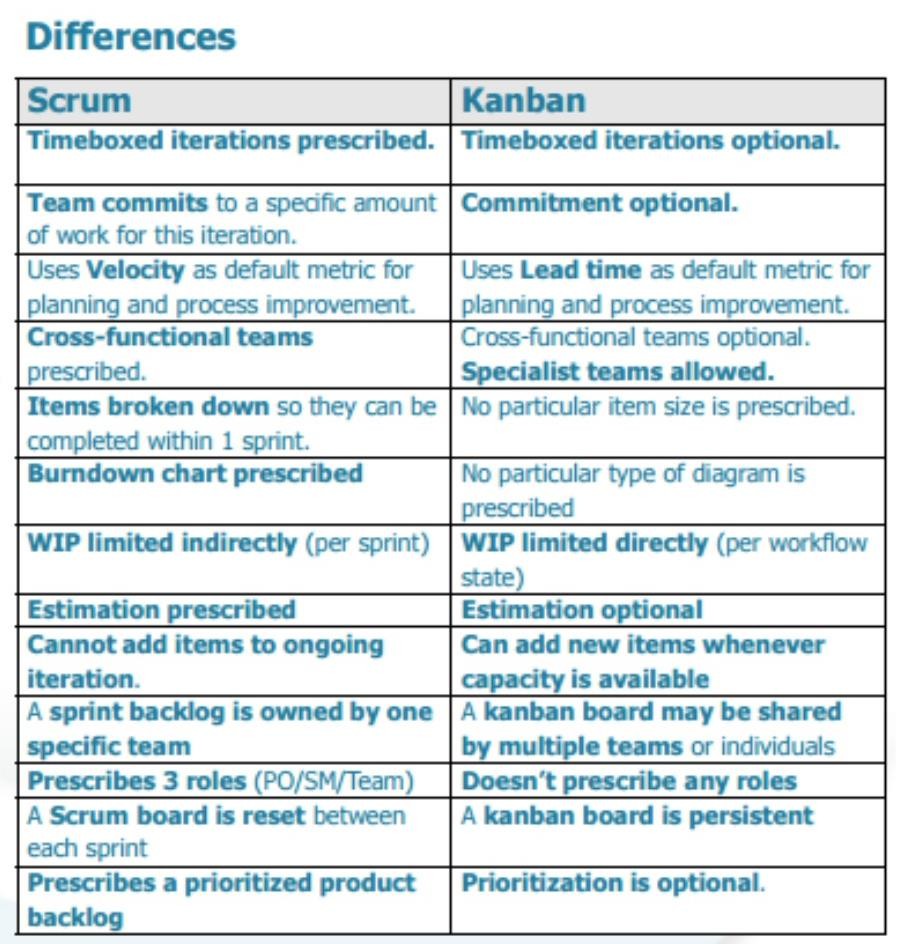
"card". So Kanban refers to "visual cards“



2.4: Agile Methods and Practices – Kanban

###### What is Kanban? (Cont.)

* Kanban is way for teams and organizations to visualize their work, identify and eliminate bottlenecks and achieve dramatic operational improvements in terms of throughput and quality
* Kanban is a method to gradually improve whatever you do – whether software development, IT/ Ops, Staffing, Recruitment, Marketing and Sales
* in fact, almost any business function can benefit from applying Kanban to bring about significant benefits such as reduced lead time, increased throughput and much higher quality of products or services delivered

###### Summary

* In this lesson, you have learnt
  + Introduction to SCRUM
  + Different Scrum Roles and Responsibilities in Agile
  + Scrum Core Practices and Artifacts
  + Definition of “Done”
  + An introduction to Extreme Programming
  + Lean Software Development
  + Kanban