# CptS 322- Software Engineering Principles I

## Agile and Scrum

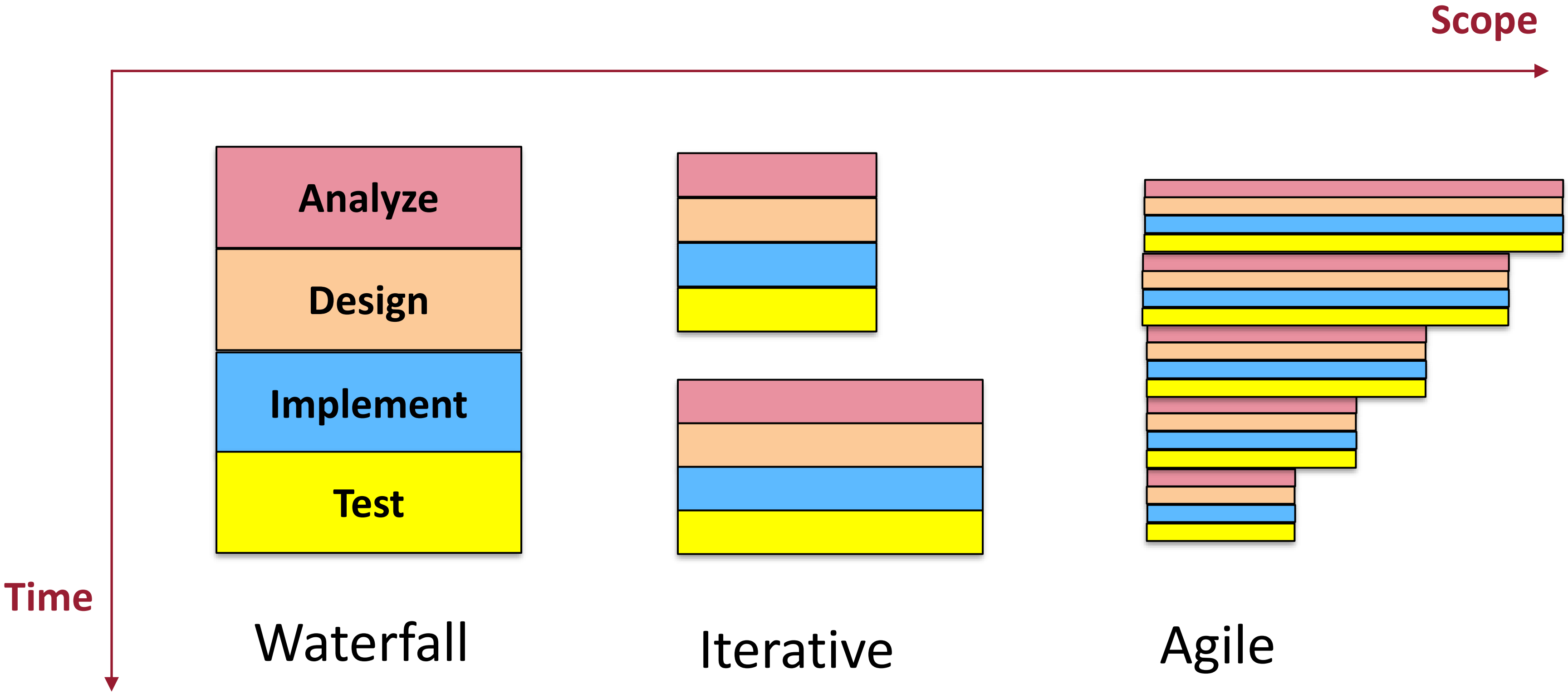
**Instructor: Sakire Arslan Ay**

**Fall 2021**

WASHINGTON STATE UNIVERSITY

World Class. Face to Face.

# Agile Processes

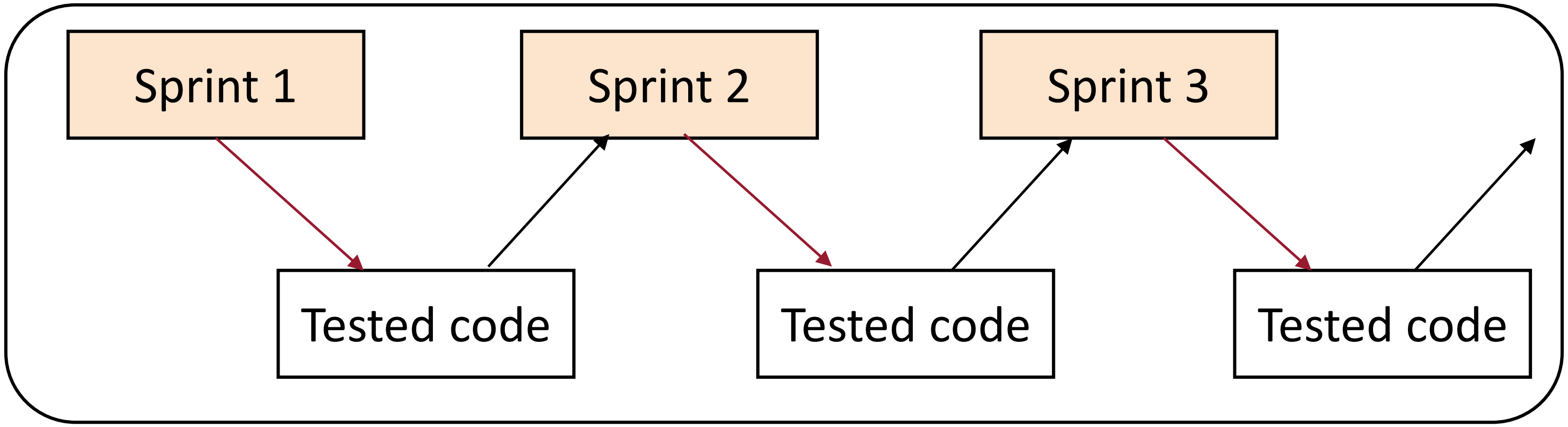


- Agile process is an iterative process with short iterations

# Definition: Sprint

- **Sprint**
  - A sprint is a pre-set period of time during which a team completes part of a software project.
  - Each sprint will go through most or all of the process steps.
  - A typical sprint might have a team of 4 to 10 people working for 1 to 4 weeks.

# Agile Process



- Development of a project is divided into a large number of sprints.
- For each sprint, a team works through a full software development cycle including planning, requirements analysis, design, coding, testing, and acceptance testing.
- This is a lightweight process with minimal documentation created during the process.
- Each sprint is completed in a fixed time period, e.g., four weeks. The size of an sprint is based on team size, e.g., 4-10 people.
- After each sprint the code may be:
  - released (original agile method)
  - combined with code from other sprints for subsequent release
  - incorporated into a larger code base (spiral development)

# 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.

# Agile – Advantages:

- Customer satisfaction by rapid, continuous delivery of useful software
- Continuous attention to technical excellence and good design; no premature optimization
  - Refactoring
  - Test driven development : Automated testing is a mandatory part of agile
- Regular adaptation to changing circumstances and requirements
- People and interactions are emphasized rather than process and tools

# Agile – Disadvantages:

- Difficult to assess the effort required at the beginning of the software development life cycle
- Can be very demanding on the customers time
- Project scope can go off-track
- Harder for new developers to integrate in the team
- Costs can increase

# What's the best model for …

- A system to control anti-lock braking in a car
- A learning management system similar to Blackboard Learn or Canvas.
- An interactive system that allows airline passengers to quickly find replacement flight times (for missed or bumped reservations) from terminals installed at airports
- A mobile app for finding romantic partners

# Choosing a Software Process

- Changes during the software development process are expensive.
  - If a big software system has many inter-related components without major changes to the design of a system during development. *Sequential process, such as the waterfall model.*
  - If the requirements are poorly understood, or expected to change, select a process that keeps flexibility. *Iterative and agile.*
  - If the market for the software is poorly understood, use a process that gets operational software in front of customers as quickly as possible. A*gile or evolutionary prototyping.*

# Corporate Processes

Large software development organizations have their own internal processes  that are designed for their needs.  For example:

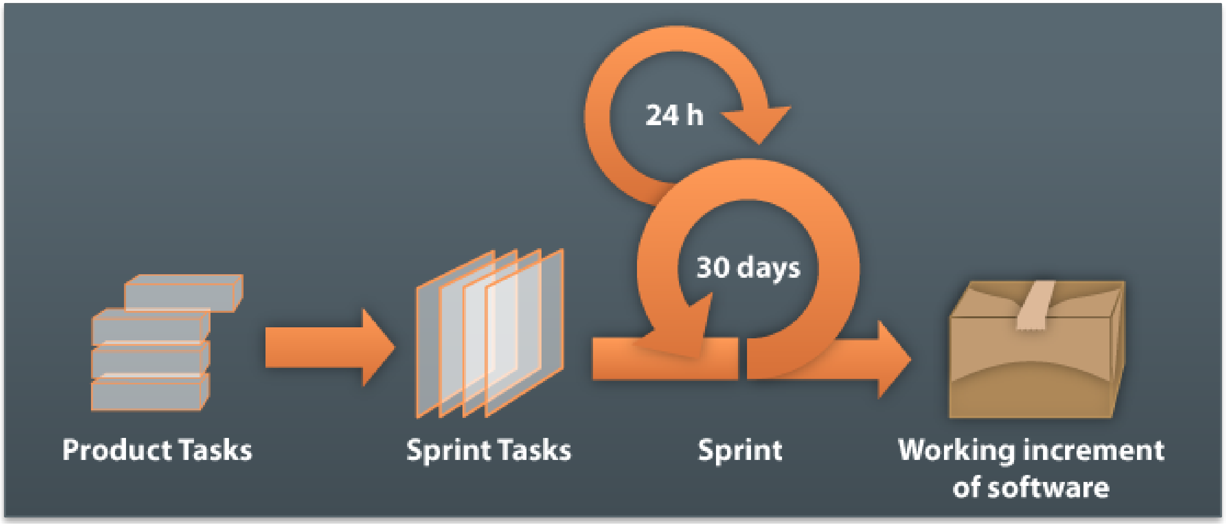
- Amazon.com (Internet commerce) makes extensive use of sprints.  Most software development is divided into increments of about four weeks elapsed time.
- SAP (business software) emphasizes the functionality that is seen by their  business customers.  Much of the development is suitable for a sequential process.
- Microsoft (PC software) places great emphasis on testing with a very wide variety of equipment and backward compatibility.  Much of the development uses spiral and agile processes.
- Lockheed Martin (government contractor) follows a sequential process that fits with the way that the US government manages software contracts.

# Variations on Agile Software Development

- There is a number of specific agile development methods widely used today.
  - Scrum, Extreme Programming, Crystal, Adaptive Software Development, Lean Software Development and etc.

- The Agile Methodologies differ in the approaches to software development and management they propose.
  - Some focus on project management and collaboration practices:
    - Scrum, Adaptive Software Development (ASD), and Lean Development.
  - Some concentrate on software implementation practices:
    - Extreme Programming (XP), Agile Modeling (AM), and Feature-driven Development (FDD).

# Scrum Framework



## Roles

- Product Owner
- Scrum Master
- Development Team

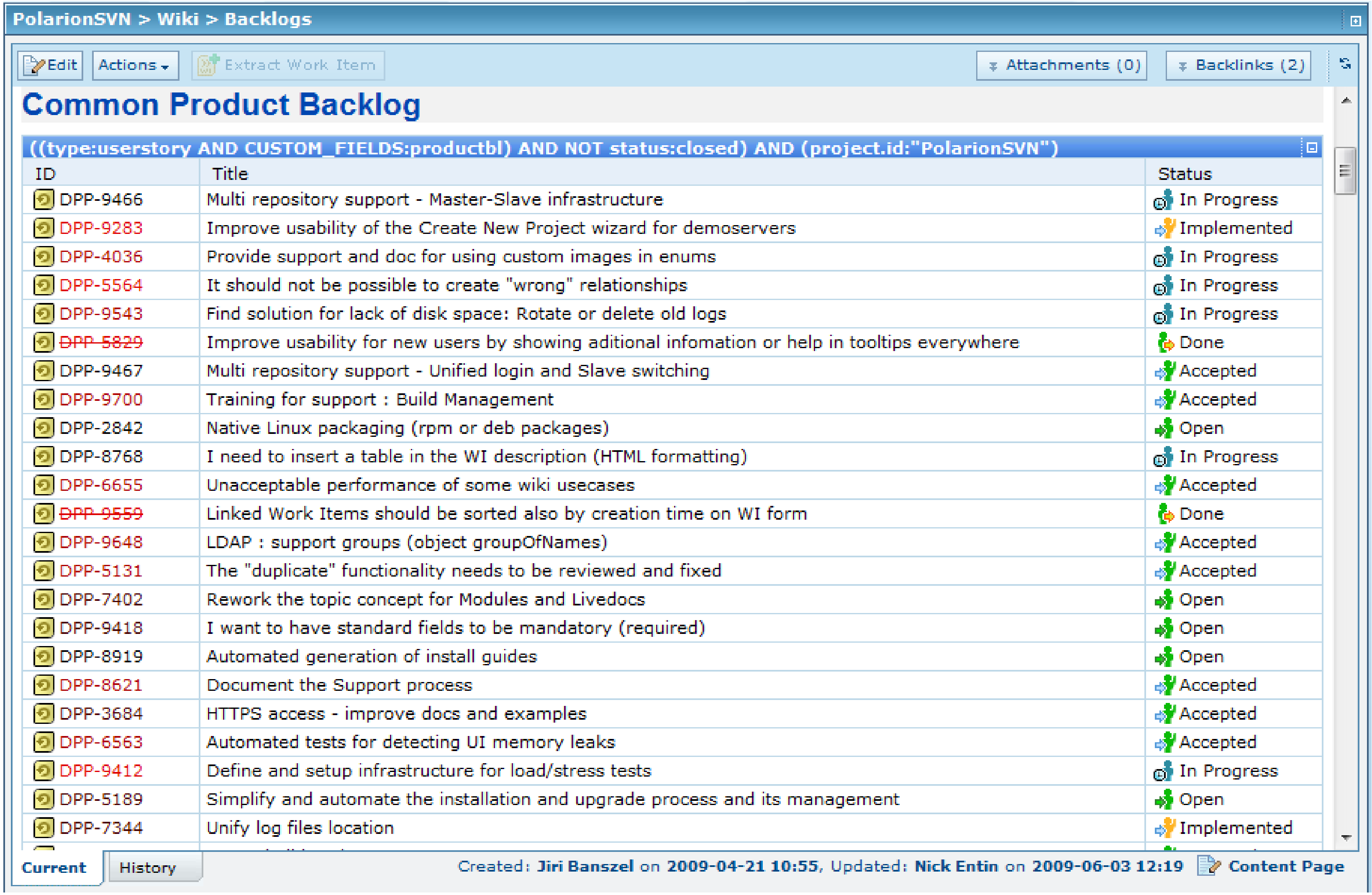
## Events

- Sprint
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

## Artifacts

- Project Charter
- Product Backlog
- Sprint Backlog
- Burn down chart

# Product Backlog



## Common Product Backlog

((type:userstory AND CUSTOM_FIELDS:productbl) AND NOT status:closed) AND (project.id:"PolarionSVN")

| ID | Title | Status |
|---|---|---|
| DPP-9466 | Multi repository support - Master-Slave infrastructure | In Progress |
| DPP-9283 | Improve usability of the Create New Project wizard for demoservers | Implemented |
| DPP-4036 | Provide support and doc for using custom images in enums | In Progress |
| DPP-5564 | It should not be possible to create "wrong" relationships | In Progress |
| DPP-9543 | Find solution for lack of disk space: Rotate or delete old logs | In Progress |
| ~~DPP-5829~~ | Improve usability for new users by showing aditional infomation or help in tooltips everywhere | Done |
| DPP-9467 | Multi repository support - Unified login and Slave switching | Accepted |
| DPP-9700 | Training for support : Build Management | Accepted |
| DPP-2842 | Native Linux packaging (rpm or deb packages) | Open |
| DPP-8768 | I need to insert a table in the WI description (HTML formatting) | In Progress |
| DPP-6655 | Unacceptable performance of some wiki usecases | Accepted |
| ~~DPP-9559~~ | Linked Work Items should be sorted also by creation time on WI form | Done |
| DPP-9648 | LDAP : support groups (object groupOfNames) | Accepted |
| DPP-5131 | The "duplicate" functionality needs to be reviewed and fixed | Accepted |
| DPP-7402 | Rework the topic concept for Modules and Livedocs | Open |
| DPP-9418 | I want to have standard fields to be mandatory (required) | Open |
| DPP-8919 | Automated generation of install guides | Open |
| DPP-8621 | Document the Support process | Accepted |
| DPP-3684 | HTTPS access - improve docs and examples | Accepted |
| DPP-6563 | Automated tests for detecting UI memory leaks | Accepted |
| DPP-9412 | Define and setup infrastructure for load/stress tests | In Progress |
| DPP-5189 | Simplify and automate the installation and upgrade process and its management | Open |
| DPP-7344 | Unify log files location | Implemented |

Created: Jiri Banszel on 2009-04-21 10:55, Updated: Nick Entin on 2009-06-03 12:19

# Roles

## 1. Product Owner

- Defines the features of the product
- Decides on release date and content
- Prioritizes features according to market value. Adjusts features and priority every iteration, as needed
- Accepts or rejects work results

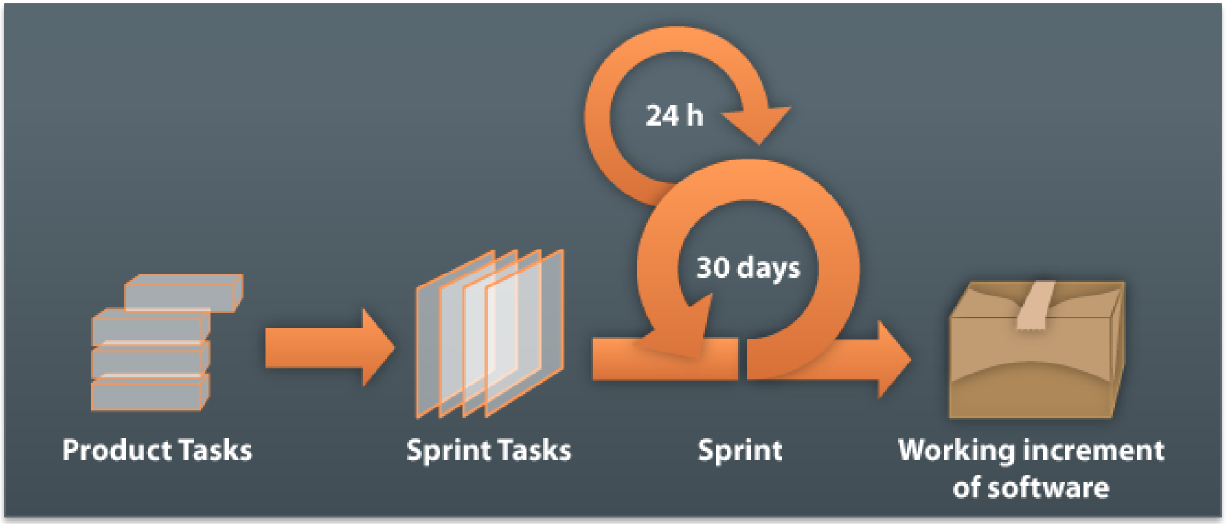
## 2. Scrum Master

- Represents management to the project
- Responsible for endorsing Scrum values and practices
- Removes obstacles
- Ensures that the team is fully functional
- Enables close cooperation across all roles and functions
- Shields the team from external interferences

# Roles (cont.)

3. **Development Team**
   - Members are cross-functional, with all of the skills as a team necessary to create a product
     - Programmers, user interface designers, testers, business analysts, etc.
   - Members have no titles other than “Developer”, regardless of the work being performed by the person
   - Members may have specialized skills and areas of focus, but accountability belongs to the team as a whole
   - Teams do not contain sub-teams dedicated to particular domains like testing or  business analysis.
   - Optimal team size: 4-6 members

# Scrum Framework



## Roles

- Product Owner
- Scrum Master
- Development Team

## Events

- Sprint
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

## Artifacts

- Product Backlog
- Sprint Backlog

# Events - Sprint

- Considered the "heart" of Scrum

- A sprint is a time-box of one month or less (usually between 2-4 weeks) during which a "Done," useable, and potentially-releasable product increment is created.

- Sprints have consistent durations throughout a development effort.

- A new Sprint starts immediately after the conclusion of the previous Sprint.

- During the Sprint:
  - No changes are made that will affect the Sprint Goal
  - Development Team structure remains constant
  - Software Quality attributes do not change
  - Scope may be clarified and re-negotiated between the Product Owner and Development Team as more is learned

# Events - Sprint Planning Meeting

- A Sprint Planning Meeting is a meeting where the work to be performed in the sprint is planned.

- Usually time-boxed to eight hours for a one-month sprint. For shorter sprints, the event is proportionately shorter.

- There are two parts to the meeting:
  - What will be delivered in this Sprint?
  - How will the work get done?

# What will be delivered in this sprint?

- The Product Owner presents to the Development Team an ordered list of all items needed to be completed.

- The entire Scrum Team collaborates on understanding the work to be done in the Sprint.

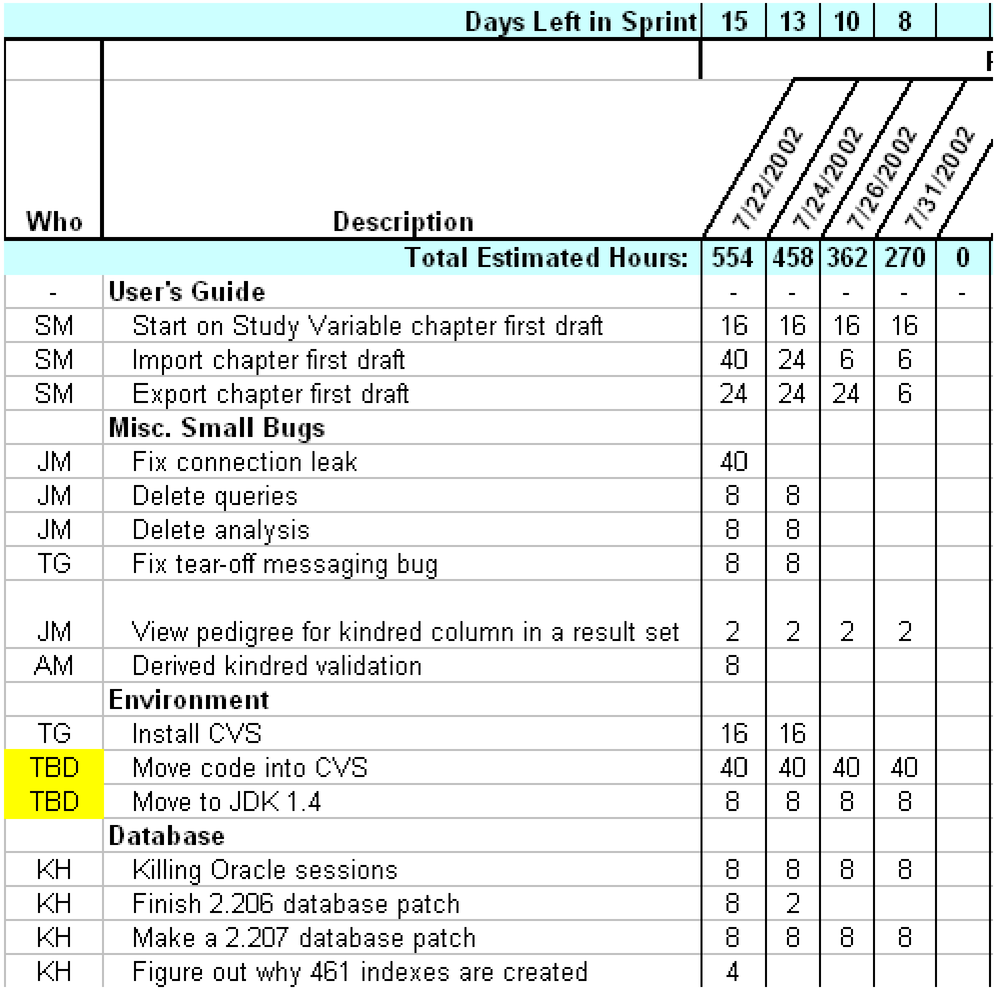
- Other items evaluated before making a decision:
  - Product Backlog
  - The latest product Increment (or Iteration)
  - Projected capacity of the Development Team for the upcoming Sprint
  - Past performance of the Development Team

- A **Sprint Goal**, or an objective that will be met within the Sprint, is defined during this meeting.

# How will the work get done?

- The Development Team decides how it will build this functionality into a “Done” product Increment during the Sprint
- The Product Backlog items selected for this Sprint plus the plan for delivering them is called the Sprint Backlog.
- The Development Team self-organizes the Sprint Backlog by estimating the time it will take to complete each item and assigning tasks to individual team members

# Sprint Backlog



| | Days Left in Sprint | 15 | 13 | 10 | 8 | |
|---|---|---|---|---|---|---|
| Who | Description | 7/22/2002 | 7/24/2002 | 7/26/2002 | 7/31/2002 | |
| | Total Estimated Hours: | 554 | 458 | 362 | 270 | 0 |
| - | **User's Guide** | - | - | - | - | - |
| SM | Start on Study Variable chapter first draft | 16 | 16 | 16 | 16 | |
| SM | Import chapter first draft | 40 | 24 | 6 | 6 | |
| SM | Export chapter first draft | 24 | 24 | 24 | 6 | |
| | **Misc. Small Bugs** | | | | | |
| JM | Fix connection leak | 40 | | | | |
| JM | Delete queries | 8 | 8 | | | |
| JM | Delete analysis | 8 | 8 | | | |
| TG | Fix tear-off messaging bug | 8 | 8 | | | |
| JM | View pedigree for kindred column in a result set | 2 | 2 | 2 | 2 | |
| AM | Derived kindred validation | 8 | | | | |
| | **Environment** | | | | | |
| TG | Install CVS | 16 | 16 | | | |
| TBD | Move code into CVS | 40 | 40 | 40 | 40 | |
| TBD | Move to JDK 1.4 | 8 | 8 | 8 | 8 | |
| | **Database** | | | | | |
| KH | Killing Oracle sessions | 8 | 8 | 8 | 8 | |
| KH | Finish 2.206 database patch | 8 | 2 | | | |
| KH | Make a 2.207 database patch | 8 | 8 | 8 | 8 | |
| KH | Figure out why 461 indexes are created | 4 | | | | |

# Events - Daily Scrum Meeting

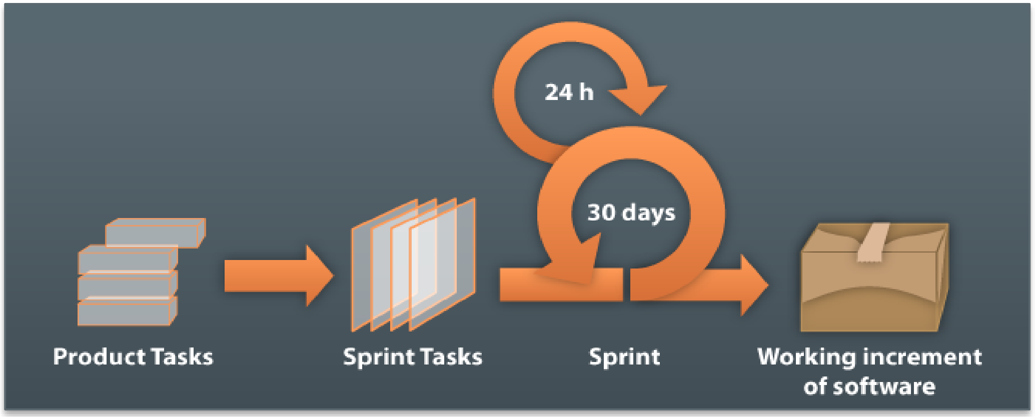
- The Daily Scrum (or “Stand-up Meeting”) is a 15-minute time-boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours.

- The Daily Scrum is held at the same time and place each day to reduce complexity.

- Each team member answers three questions:
  - What has been accomplished since the last meeting?
  - What will be done before the next meeting?
  - What obstacles are in the way?

# Events - Sprint Review Meeting

- The Sprint Review Meeting is the time when the Development Team presents what it accomplished during the sprint
- Takes the form of a demo of new features or concrete progress
- Informal and requires little prep time
- The entire team participates



# Events - Sprint Retrospective Meeting

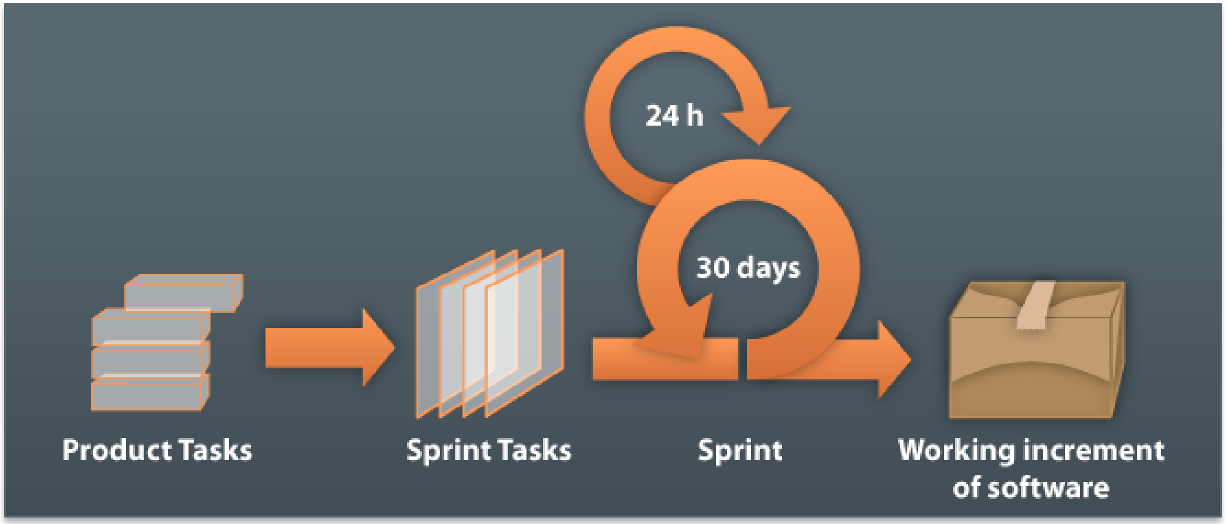
- The Sprint Retrospective is an opportunity for the Scrum Team to reflect and create a plan for improvements to be enacted during the next Sprint.

- The purpose:
  - Think about  how the last Sprint went with regards to people, relationships, process, and tools
  - Identify the major items that went well and potential improvements
  - Create a plan for implementing improvements to the way the Scrum Team does its work

- All team members have the opportunity to answer:
  - What went well during the last sprint?
  - What didn’t go well during the last sprint?
  - How should the team improve for the next sprint?

⇨

# Scrum Framework



## Roles

- Product Owner
- Scrum Master
- Development Team

## Events

- Sprint
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

## Artifacts

- Project Charter
- Product Backlog
- Sprint Backlog
- Burn down chart

# Artifacts - Project Charter

- Problem Statement:
  - Short and succinct (one or two sentences)
- Project Objectives:
  - What the project will achieve
- Stakeholders:
  - Persons who will be actively involved with the project (e.g. project sponsor, types of users, etc.)
- Project Deliverables:
  - The major results or services that will be produced, what are the specific things the software will do

# Artifacts - Product Backlog

- The **Product Backlog** is an ordered list of everything that might be needed in the product and is the single source of **Requirements** for any changes to be made to the product.
- The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering.
- A Product Backlog is never complete.
- Lists all features, functions, requirements, enhancements, and fixes that constitute the changes to be made to the product in future releases.

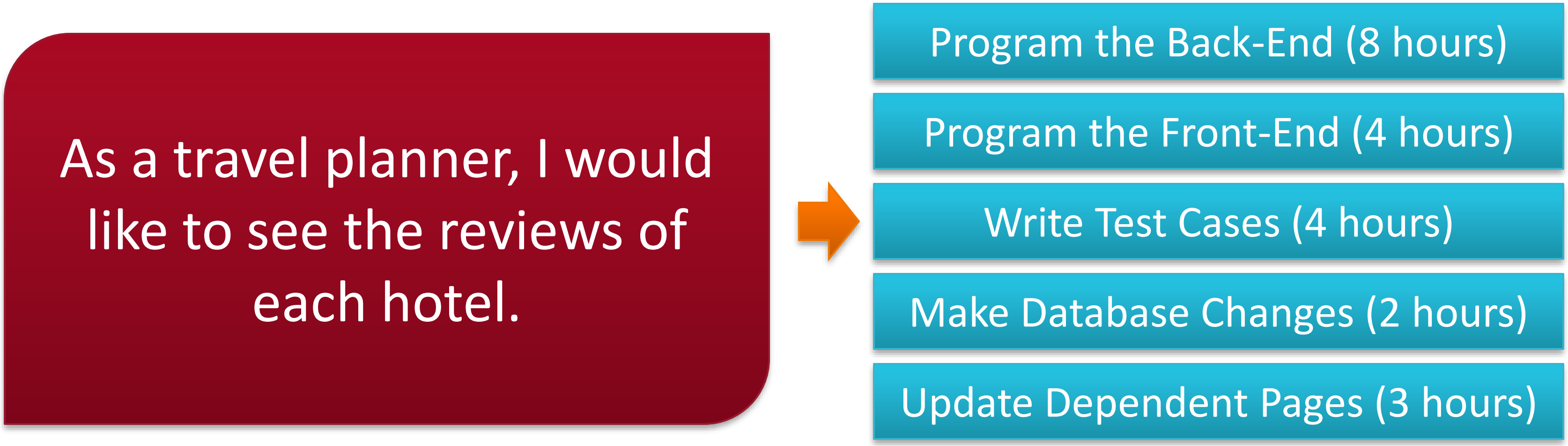
# Product Backlog Sample

| Backlog Item | Priority |
|---|---|
| As a guest, I want to make a reservation. | 3 |
| As a guest, I want to cancel a reservation. | 5 |
| As a guest, I want to change the dates of a reservation. | 3 |
| As an admin, I want to change the availability of dates at my hotel. | 8 |
| As a developer, I want to improve exception handling. | 8 |

- Backlog items are usually in the form of:
  - As a _______, I want to __________ (so that I can ________).
- Product Backlog items are sometimes called "**user stories**"

# Artifacts - Sprint Backlog

- In a Sprint Backlog, the Development Team selects items from the product backlog they can commit to completing in a given sprint.
- The team then identifies tasks to complete these and each is estimated (how many hours to complete).



As a travel planner, I would like to see the reviews of each hotel.

Program the Back-End (8 hours)

Program the Front-End (4 hours)

Write Test Cases (4 hours)

Make Database Changes (2 hours)

Update Dependent Pages (3 hours)

User Story (from Product Backlog)

Sprint Backlog Items

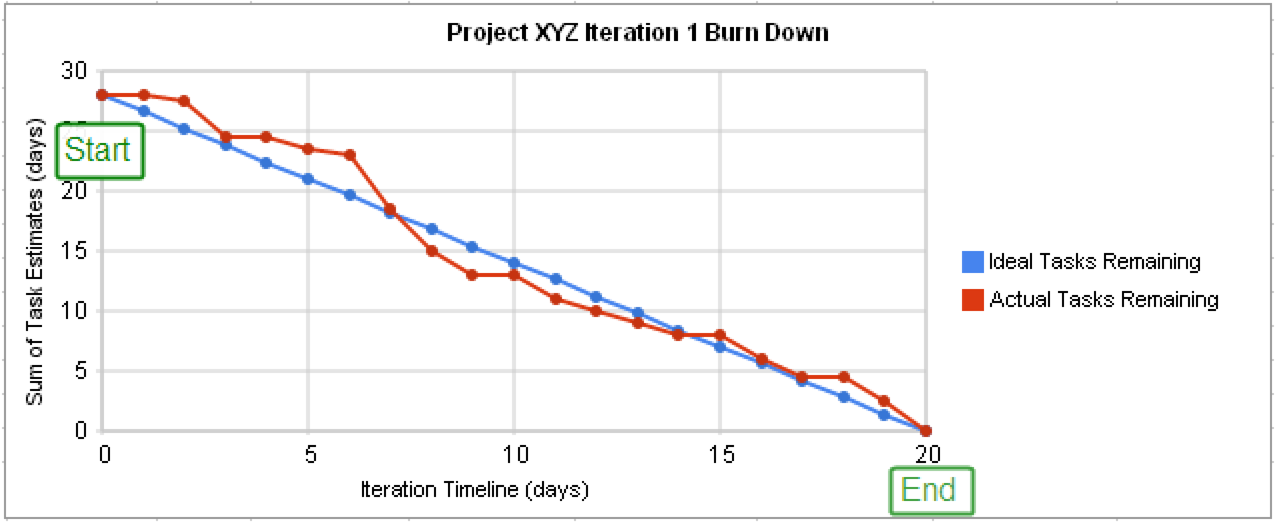
# Managing the Sprint Backlog

- Individuals sign up for work of their own choosing
- Work is never assigned
- Estimated work remaining is updated daily
- Any team member can add to, delete from, or modify the 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

# Sprint Backlog Sample

| Backlog Tasks | Mon | Tue | Wed | Thur | Fri |
|---|---|---|---|---|---|
| Program the Back-End | 3 | 4 | 1 | | |
| Program the Front-End | | | 2 | 2 | |
| Write Test Cases | 3 | | 1 | | |
| Make Database Changes | | 2 | | | |
| Update Dependent Pages | | | | | 3 |

# Burn down chart



- Graphical representation of work left to do vs time left in which to do it
- Time is x axis
- Outstanding work (backlog) is y axis