

# **SCRUM**

**Scrum Intro**

**Scrum Framework**

**Scrum Drawbacks**

**Scrum in your Project**

# SCRUM INTRO

# SCRUM IN 100 WORDS

- Scrum is an agile process that allows us to focus on **delivering the highest business value in the shortest time.**
- It allows us to **rapidly and repeatedly inspect actual working software** (every two weeks to one month)
- The **business sets the priorities.** Teams self-organize to determine the best way to deliver the **highest priority features.**
- Every two weeks to a month anyone can see **real working software** and decide to release it as is or continue to enhance it for another sprint.

# WHAT IS SCRUM

- Agile method for software development
- One of the best known methods
- Mentioned first 1986 by Hirotaka Takeuchi and Ikujiro Nonaka for product development
- 10 years later enhanced for agile software development by Ken Schwaber and Jeff Sutherland
- Ken Schwaber and Mike Cohn Co-founded Scrum Alliance in 2002, initially within the Agile Alliance

# MAIN CHARACTERISTICS

- Self-organizing teams
- Product progresses in a series of 2-4 week long “sprints”
- Requirements are captured as items in a list of “product backlog”
- No specific engineering practices prescribed
- Uses generative rules to create an agile environment for delivering projects

# THE AGILE MANIFESTO

Individuals and interactions

over

Process and tools

Working software

over

Comprehensive documentation

Customer collaboration

over

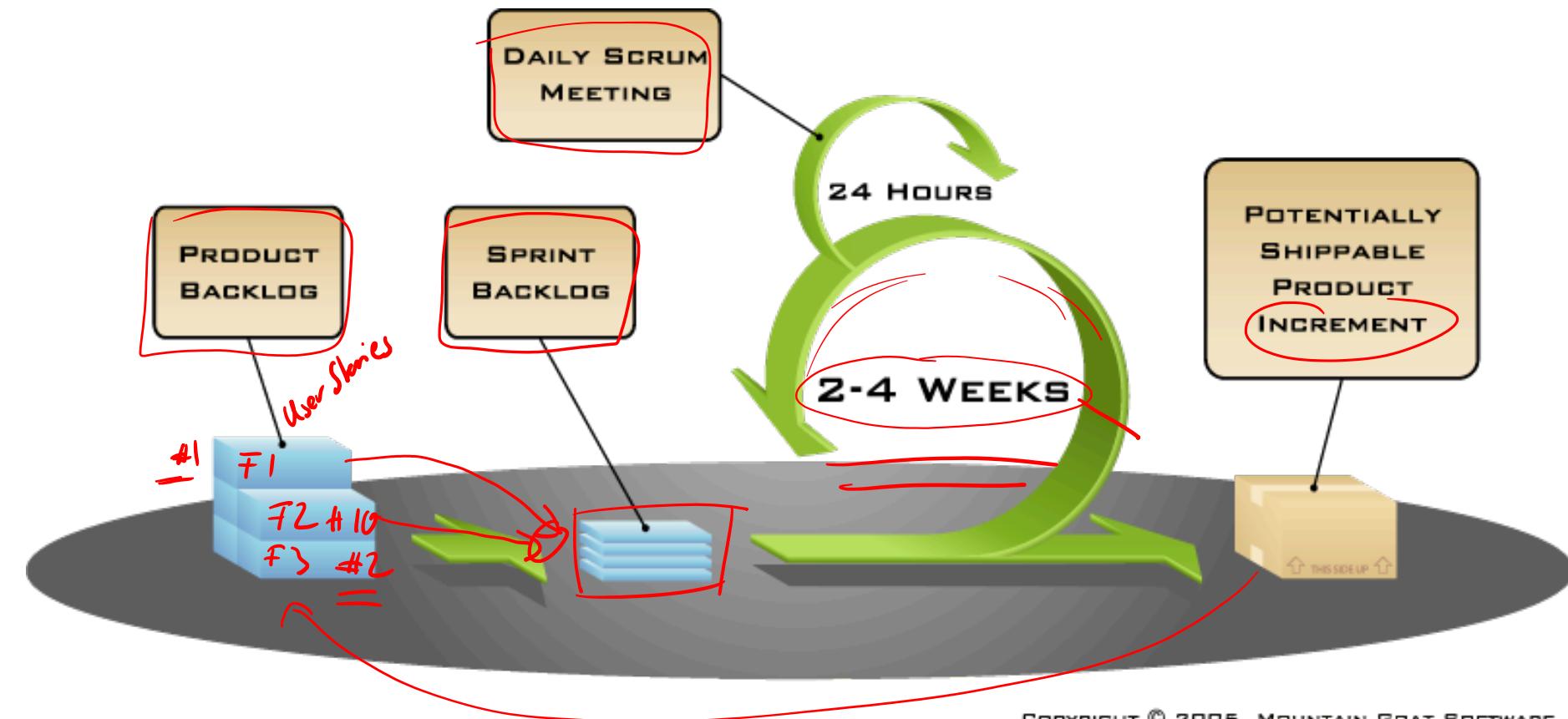
Contract negotiation

Responding to change

over

Following a plan

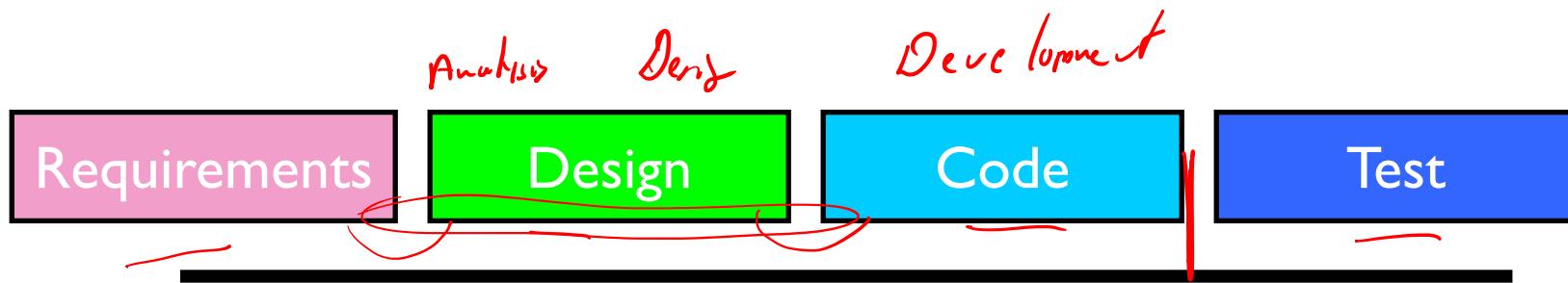
# PUTTING IT ALL TOGETHER



# SPRINTS

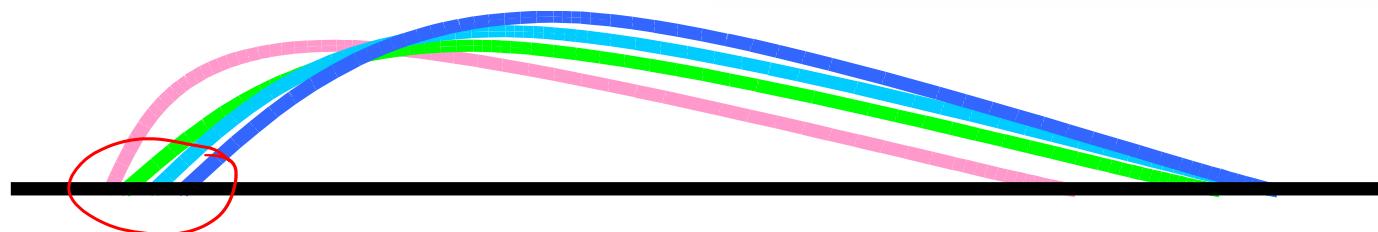
- Scrum projects make progress in a series of “sprints”
  - Analogous to Extreme Programming iterations
- Typical duration is 2–4 weeks or a calendar month at most
- A constant duration leads to a better rhythm
- Product is designed, coded, and tested during the sprint

# SEQUENTIAL VS. OVERLAPPING DEVELOPMENT

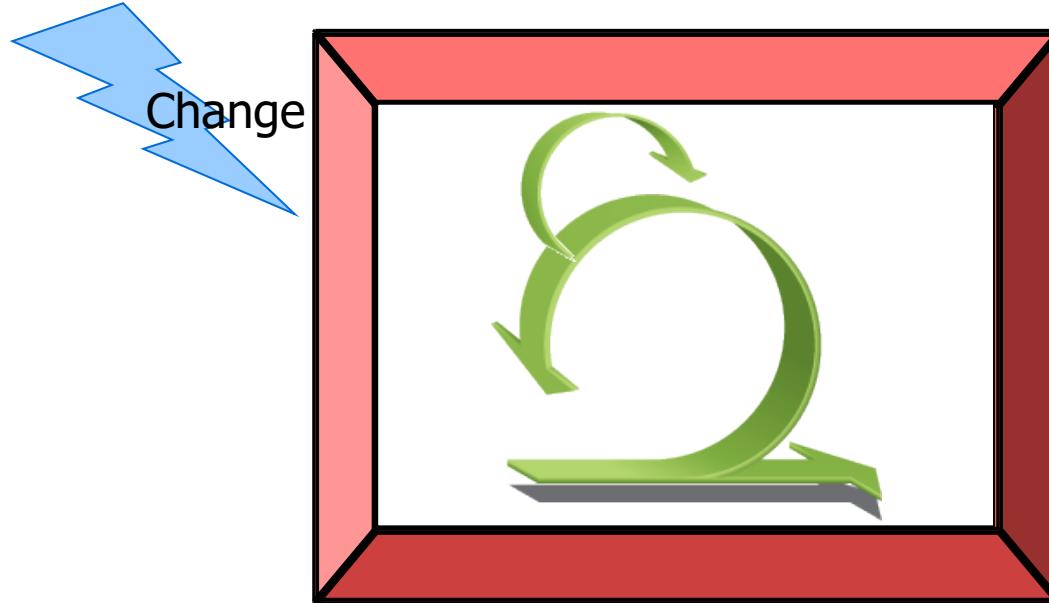


Rather than doing all of  
one thing at a time...

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



# NO CHANGES DURING A SPRINT



Plan sprint durations around how long you can commit to keeping  
change out of the sprint

# SUMMARY

- Scrum is an Agile method for software development
- Individuals, Working Software, Customer Collaboration and Responding to Change are Key features
- A Sprint includes: Requirements, Design, Coding and Testing
- Team needs to be self organized

# SCRUM FRAMEWORK

## Roles

Using Scrum (Ceremonies and Artifacts)

---

# SCRUM FRAMEWORK

## Roles

- Product owner
- ScrumMaster
- Team

## Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

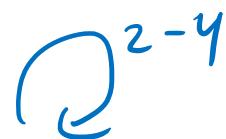
## Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

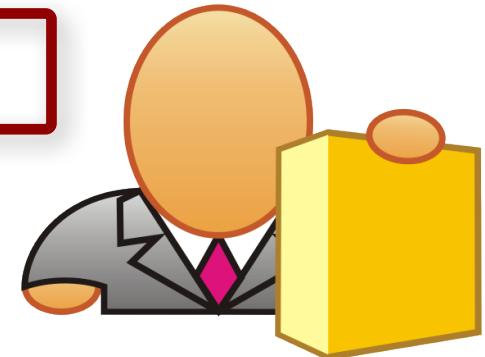


# PRODUCT OWNER

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



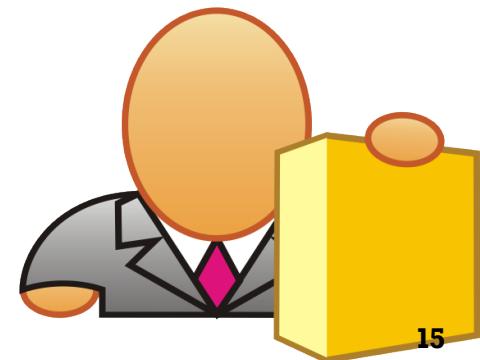
Is not part of the programming team



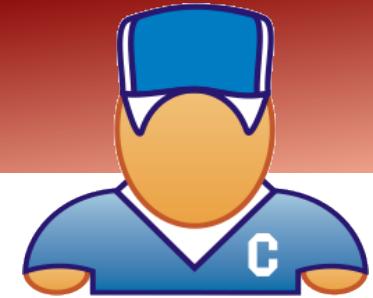
# A GOOD PRODUCT OWNER

- Is always available
- Needs to know about market, customers and users
- Communicative (talks with everybody)
- Needs to make decision
- Needs to have authority

(by Mike Cohn)



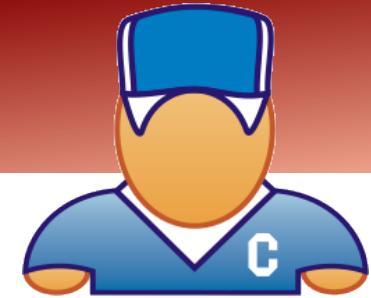
# THE SCRUMMASTER



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

Is not the boss of the team

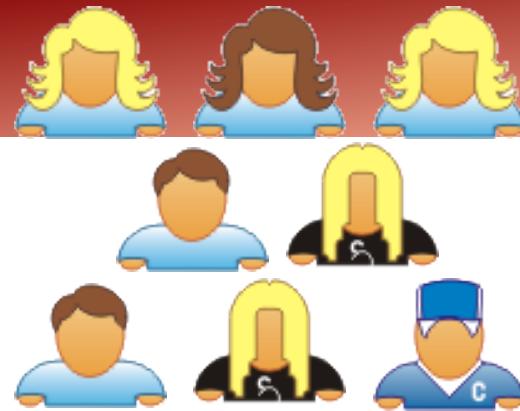
# THE SCRUMMASTER



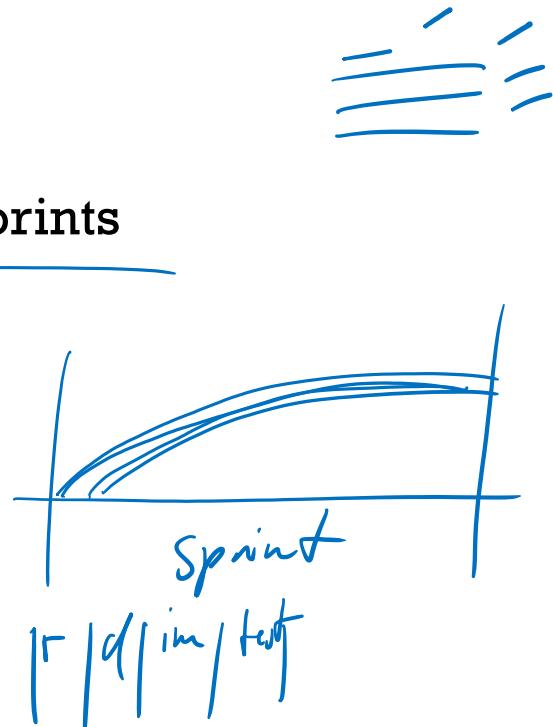
- **Responsible:** takes responsibility for team and team goals
- **Modest:** does not take credit only helps to achieve goals
- **Team spirit:** enhances team spirit
- **Dedicated:** tries to achieve goals the same way the team does
- **Position of influence:** Helps team to new ideas, but does not influence decision
- **Competent:** Technical and marketing knowledge

(by Mike Cohn)

# THE TEAM



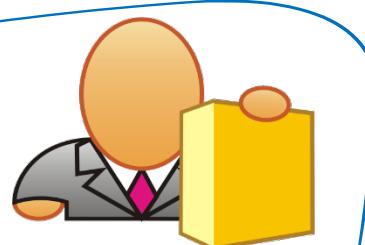
- Typically **5-9 people**
- Cross-functional:
  - Programmers, testers, user experience designers, etc.
- Members should be full-time
- Teams are self-organizing
- Membership should change only between sprints
- Responsible for creating functionality
- Decides how to implement functionality
- **Everyone in team is equal and responsible**



# SUMMARY

## Three main roles

- Product Owner

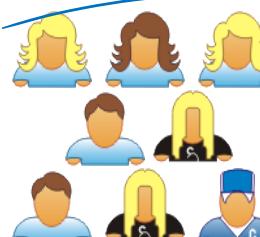


- Scrum Master



*not the boss*

- Development Team



# SCRUM FRAMEWORK

## Roles

- Product owner
- ScrumMaster
- Team

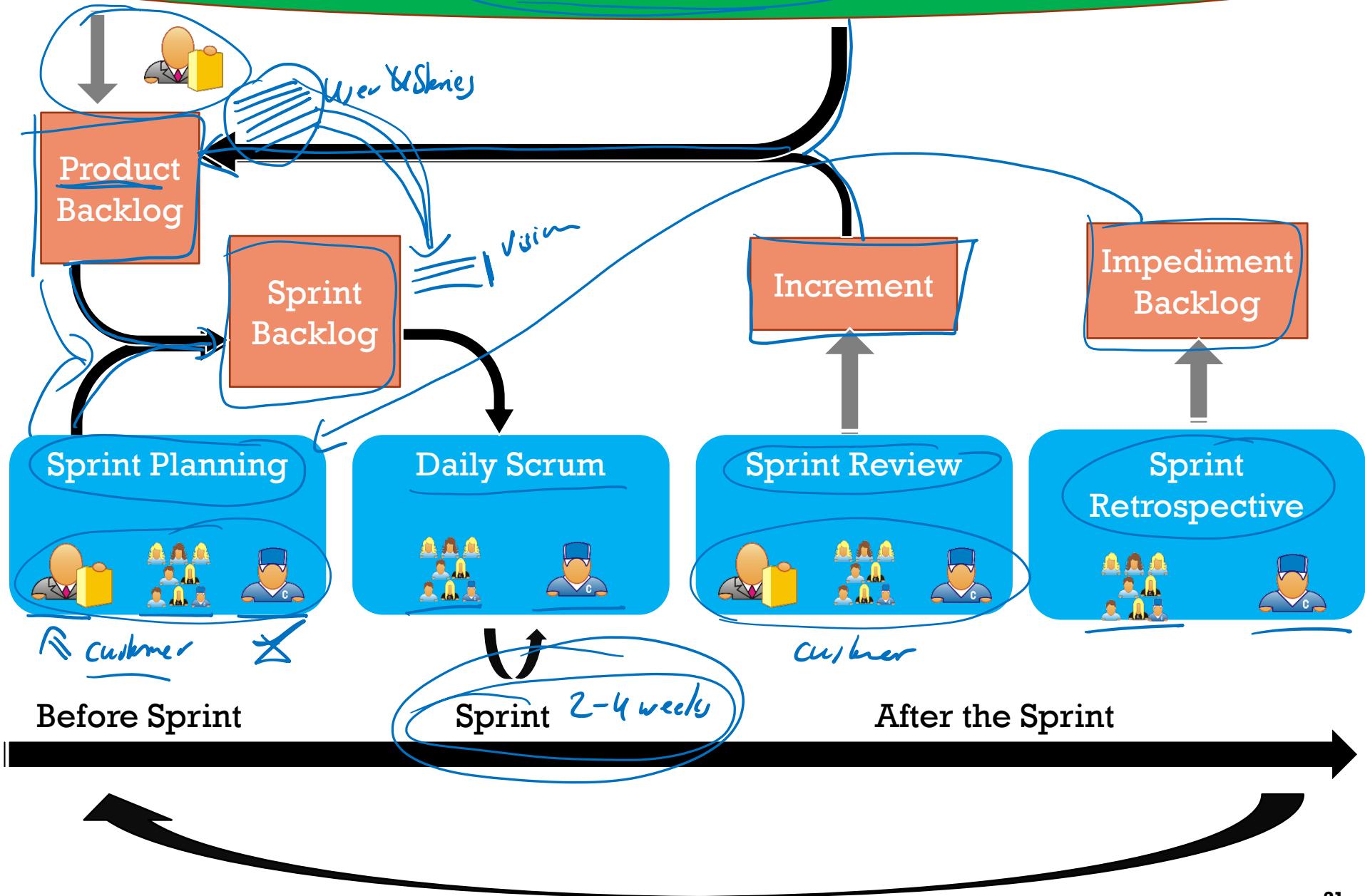
## Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

## Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

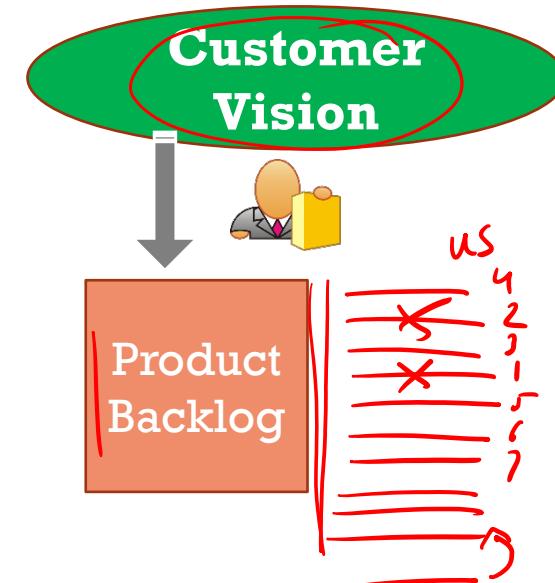
# Customer Vision



# VISION TO PRODUCT BACKLOG

# VISION AND PRODUCT BACKLOG

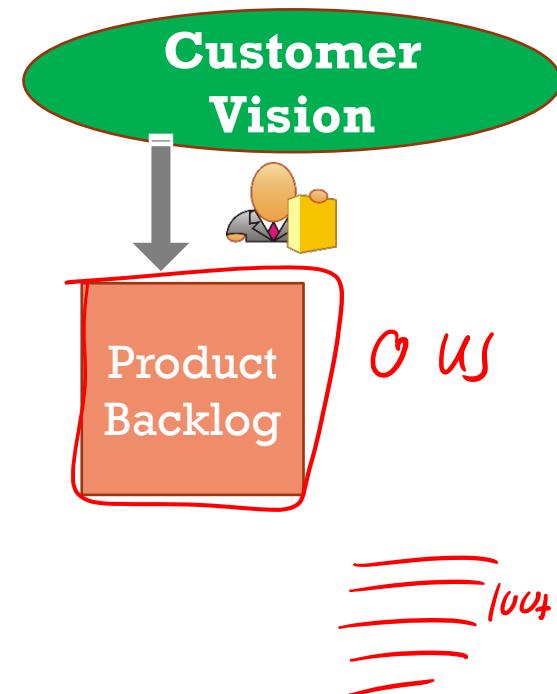
- Vision is the idea/wish for a project from customer
- This is shared with the Product Owner
- Results in Product Backlog
  - A list of all desired work (requirements) on the project
  - Requirements written as User Stories
  - Prioritized by the product owner
  - Reprioritized at the start of each sprint
  - Ideally expressed such that each item has value to the users or customers of the product



# PRODUCT BACKLOG

Prioritized queue of all work to be done on the product

- **List of User Stories**
- *“...represents everything that anyone interested in the product or process has thought is needed or would be a good idea in the product...”*
- As long as the product exists, the PB exists
- It is dynamic and sorted by priority at all times (gets evaluated again at end of Sprint/beginning of Sprint)
- The Product Owner owns the Product Backlog



# USER STORIES

# USER STORIES

- Capture the project requirements
- Traditionally written on 4 x 6 index cards
  - If your card is too crammed, consider breaking out detail into additional cards with sub-stories
- Focus on capturing a conversation
  - User Stories are meant to be torn up
  - “Give the user what they need, not what was written down”
- Broaden by identifying User Roles & Acceptance Tests
  - User Roles: Who is the conversation with?
    - What are their skill sets? Proficiencies with the system? etc.
  - Acceptance Tests: How do I know what done looks like?
    - User Stories are intentionally at a high-level
    - Tests allow developers to know when they are done

**“User Stories are a placeholder for a conversation”**

# ANATOMY OF A USER STORY

The screenshot shows the 'Edit User Story' page for 'US9: Credit card payments'. Several fields are highlighted with red boxes and arrows pointing to external descriptions:

- Story name:** Points to the 'Name' field containing 'Credit card payments'.
- Value statement:** Points to the 'Description' field containing the user story text: 'As a purchase on the website I want the ability to pay with a credit card So that I may immediately confirm my purchase.'
- Acceptance Criteria:** Points to the 'Acceptance Criteria' section listing:
  - Accept Discover, Visa, MC
  - Validate CC# when entered
  - Validate expiration date and CVV
  - Validate billing address
  - Generate success and failure messages after processing
- Definition of Done:** Points to the 'Definition of Done' section listing:
  - Passes all regression tests
  - Passes testing per acceptance criteria items
  - Approved by UI Team
  - Able to show feature in company demo
- Attachments:** Points to the 'Attachments' section showing a file named 'mockup.png'.
- Owner:** Points to the 'Owner' field set to 'Greg'.
- Schedule:** Points to the 'Schedule' section showing 'State: Defined', 'Iteration: Unscheduled', 'Plan Est: 8.0 Points', and 'To Do: 0.0 Hours'.
- Size (effort) estimate, in relative points:** Points to the 'Plan Est: 8.0 Points' field.

At the bottom, a large red box encloses the template text: 'As a <user> I want to <do something> so that <benefit>'.

<https://help.rallydev.com/writing-great-user-story>

# USER STORY BEST PRACTICES

How do you get the User Story?

➤ *Have the conversation with the customer*

PO  
Team

➤ Don't lead or constrain the user, or point toward an *a priori* solution. Let her/him tell a story!

- Active and goal-driven to ensure we know who it is for and why it is valuable
- Can be finished in one Sprint
- Have an Acceptance Test so we know what Done looks like

2-4 weeks

**User Story Format:**

➤ As a <user> I want to <do something>  
so that <benefit>

# THE INVEST HEURISTICS

- Characteristics of good User Stories (*INVEST [Wake 2003]*):
  - Independent: does not depend on other cards (User Stories)
  - Negotiable: Biz (Business) and Dev (Developer) will negotiate over these *(User)*
  - Valuable: To Users, Biz, or Dev. to somebody!
  - Estimate-able: Dev can assign IET (idealized estimated time) to it
  - Small: Large stories are complex and incomplete, hiding assumptions. Consider splitting if it is too large
  - Testable: Goal is to define acceptance tests

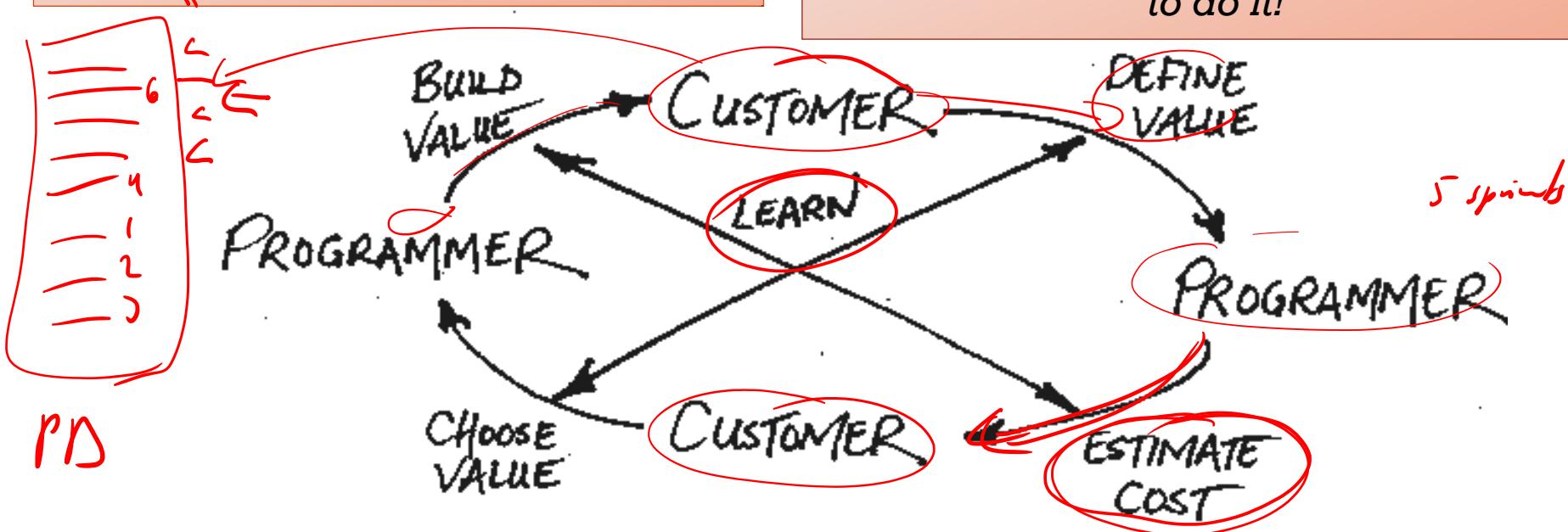


# USER STORIES AND THE AGILE RELATIONSHIP

Business People: Responsible for Scope, Priority, and the Composition of specific releases:  
✓ whether a release meets a threshold criteria

**Developers are empowered – they do estimates, determine tradeoffs, decide on processes.**

*They accept responsibility for a deliverable and are accountable but get to decide how to do it!*



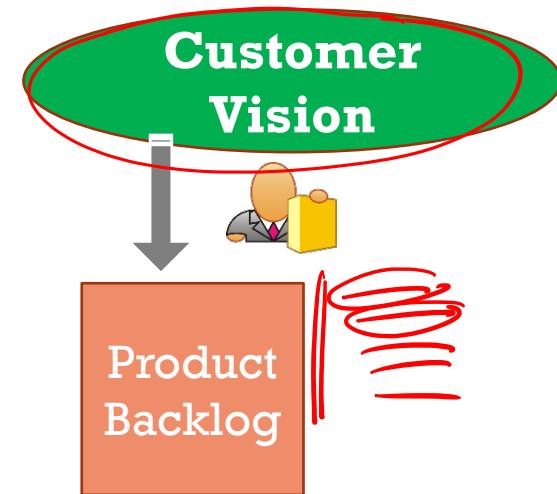
So how do the User Stories, once negotiated, fit into our SCRUM model?

# A SAMPLE PRODUCT BACKLOG

role	Backlog item	Estimate
	<p>As a <u>student</u>, I want to be able to get a list of all assignments due next week, so that I do not start too late to work on them.</p> <p><i>one thing</i></p>	3
	<p>As an <u>instructor</u>, I want be able to get a list of all students who have not submitted an assignment, so I can send them an email to remind them to submit.</p> <p><i>multiple</i></p>	5
	<p>As an <u>instructor</u>, I want to be able to upload a list of groups into Canvas, so that project groups are created on Canvas.</p> <p><i>3</i></p>	3
	<p>As a student, I want to be able to see my current grade, so that I know how I am doing.</p>	8
	...	...

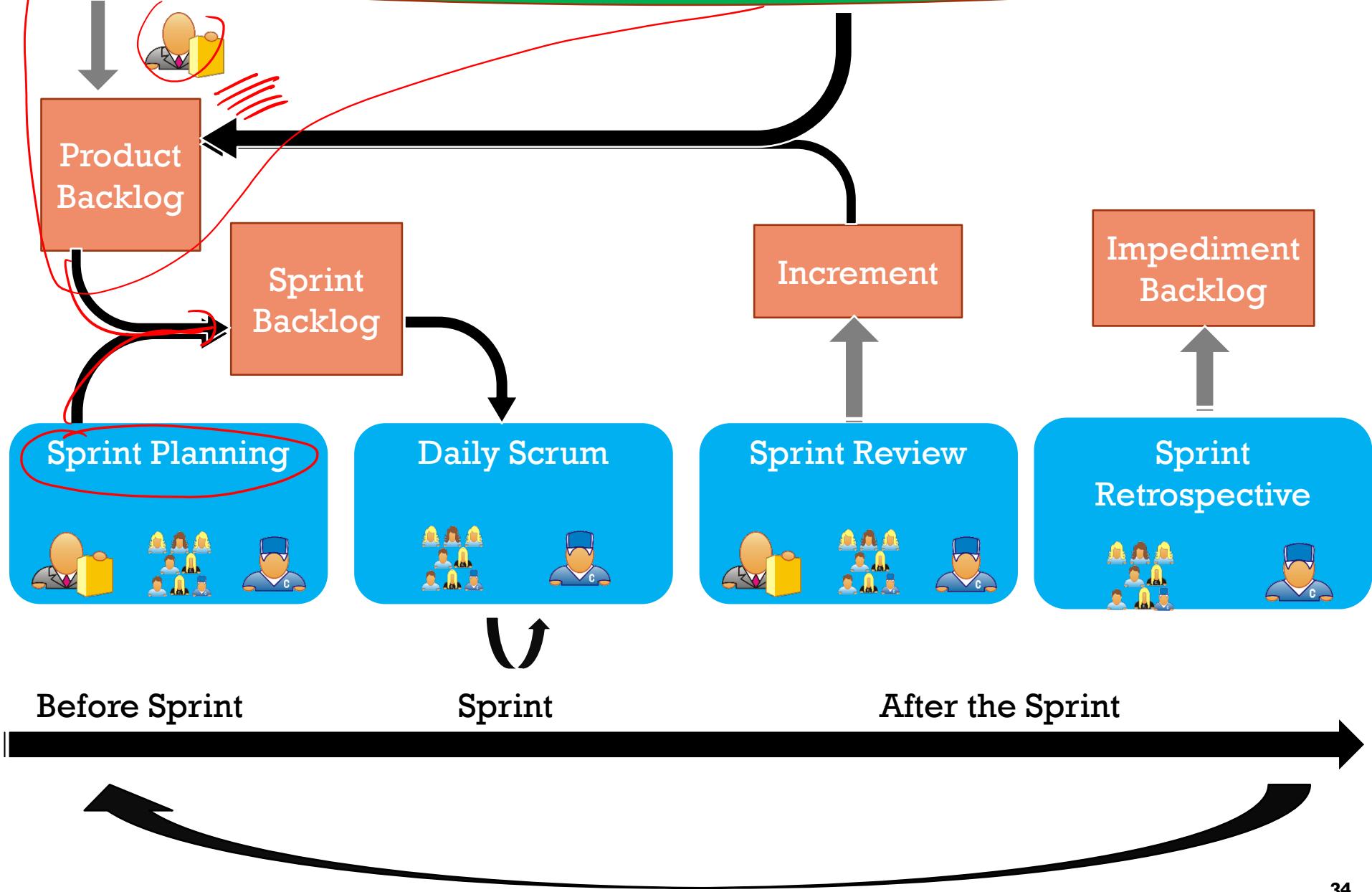
# SUMMARY

- **Customer Vision is the starting point**
- **Customer and Product Owner create Product backlog based on conversation**
  - Product Backlog is list of User Stories, which have value to customers and stakeholders
- **User Stories**
  - Are placeholders for a conversation
  - Are small enough to be finished in one Sprint
  - Are written as: “As a <user> I want to <do something> so that <benefit>”
  - Specify Acceptance Test so we know what Done looks like



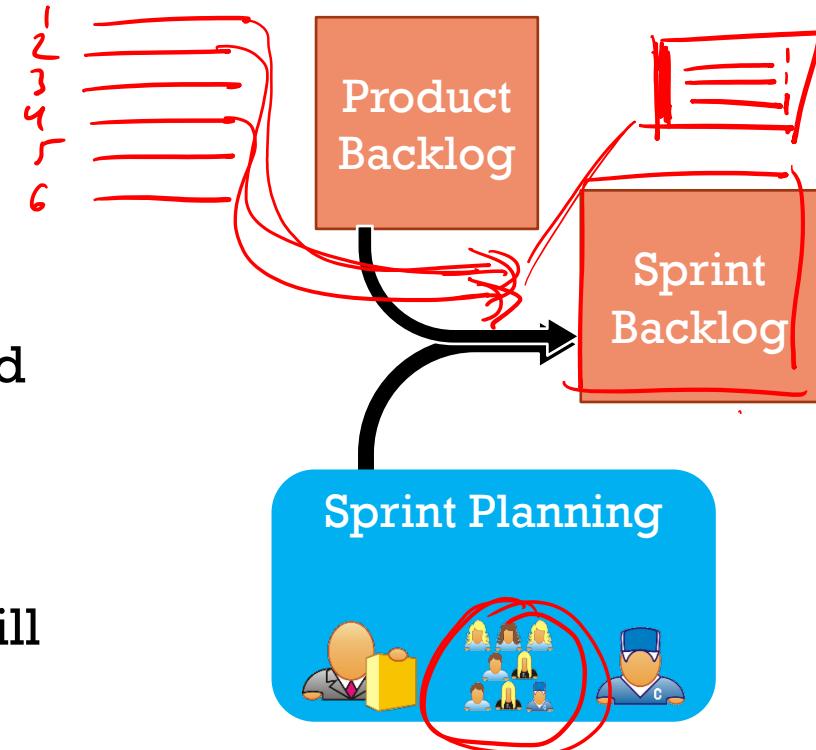
~~PRODUCT TO SPRINT BACKLOG  
VIA SPRINT PLANNING~~

# Customer Vision



# PRODUCT TO SPRINT BACKLOG VIA SPRINT PLANNING

- Created during the Sprint Planning Meeting (couple of hours)
- Subset of the Product Backlog assigned to the current Sprint
  - Valid for exactly one Sprint
  - “Contract” between the Customer and Team
  - Specifies which User Stories will be completed during the next Sprint
  - Sprint Goal defines what the Sprint will focus on
- The Scrum team owns the Sprint Backlog
- Scrum Master facilitates but does not own the Sprint Backlog



2-4 weeks

# SPRINT BACKLOG

# THE SPRINT GOAL

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

Database Application

Make the application run on SQL Server in addition to Oracle.

Life Sciences

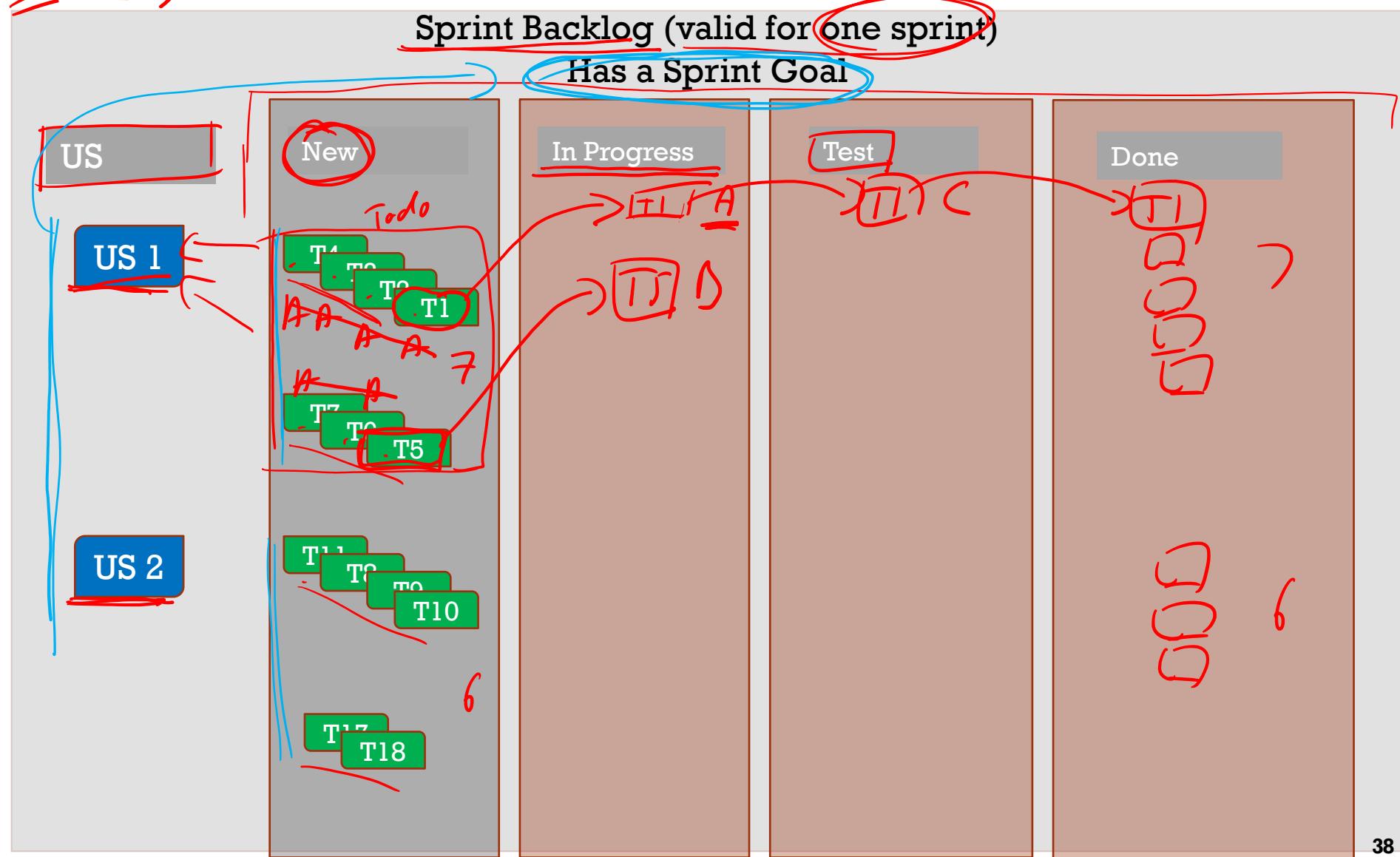
Support features necessary for population genetics studies.

Financial Services

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

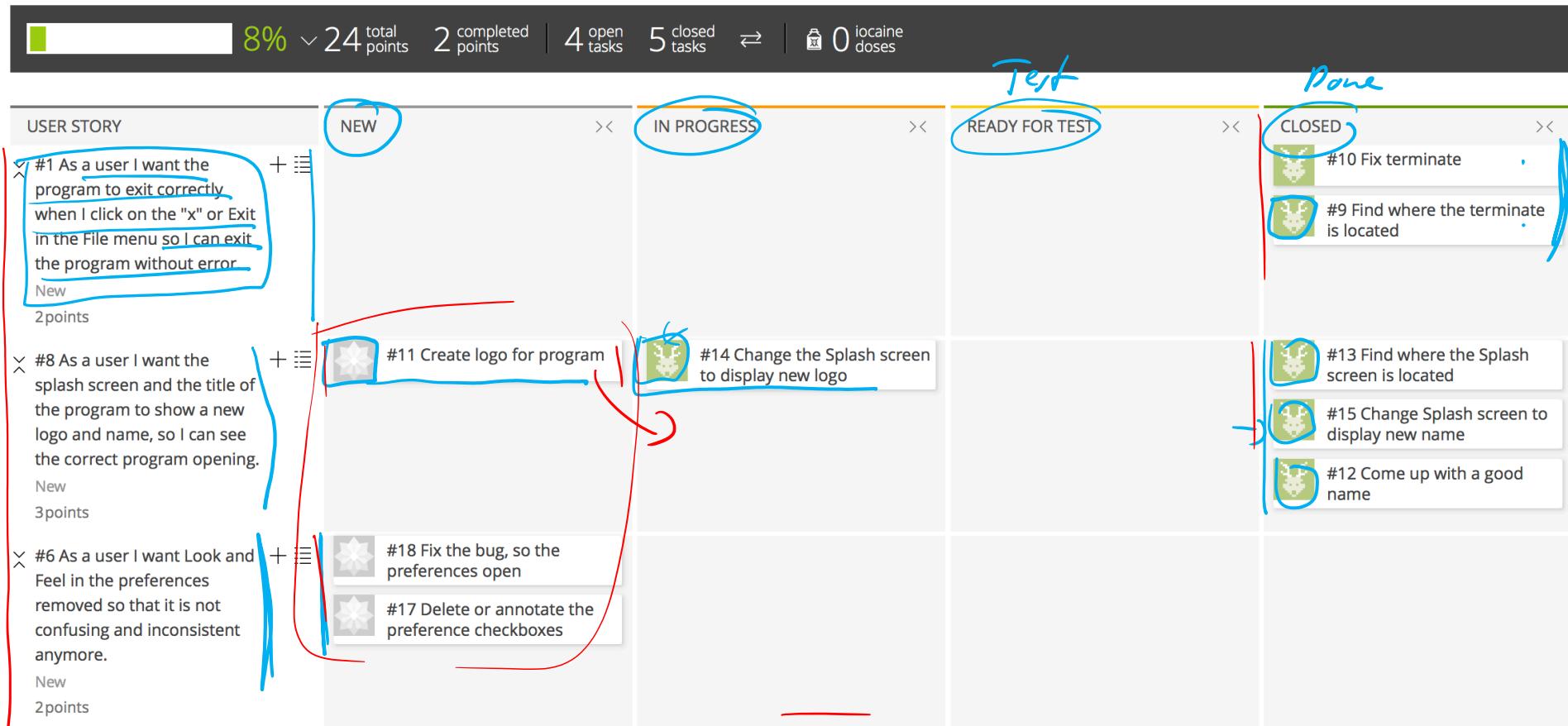
# ABSTRACT SPRINT BACKLOG

PB SD



# TAIGA SPRINT BACKLOG

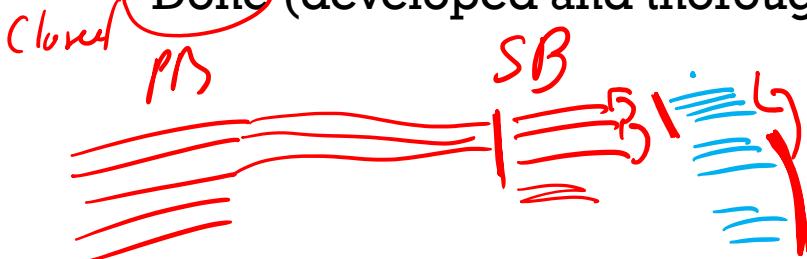
SER316 SPRING 2019 SPRINT 1 (UNDERSTANDING THE MAIN FEATURES OF MEMORANDA AND FIXING OBVIOUS BUGS) 23 JAN 2019-01



# TASKS

- Tasks are the things that need to be done to finish a User Story
  - A to-do list to finish the User Story
- Each US in the Sprint Backlog should have 1..\* tasks
- Tasks should be small (be able to finish them in one working session)
- Tasks are either:
  - New (not assigned)
  - In Progress (currently being worked on by a team member)
  - Ready for Test (currently being tested)
  - Done (developed and thoroughly tested)

full time student  $\frac{3}{16}$  8h  
2 - 4 h



# SPRINT BACKLOG SUMMARY

- Subset of User Stories from Product Backlog which will get done during the Sprint (contract negotiated during the Spring Planning meeting – stay tuned)
- Valid for one Sprint
- User Stories are split up into little to-dos (Tasks)
- Shows how “far along” each task and each User Story is
- Needs to be kept up to date at all times to be valuable

US                          Done  
=====

# SPRINT PLANNING

# SPRINT STEPS (IN REAL LIFE)

## Sprint Goal: declarative statement of Sprint objective

- The SG can be met to different degrees
- The team may achieve SG without completing all of the SB
- Team empowered to achieve SG however they deem best
  - This is the self-organizing, autonomous philosophy in action

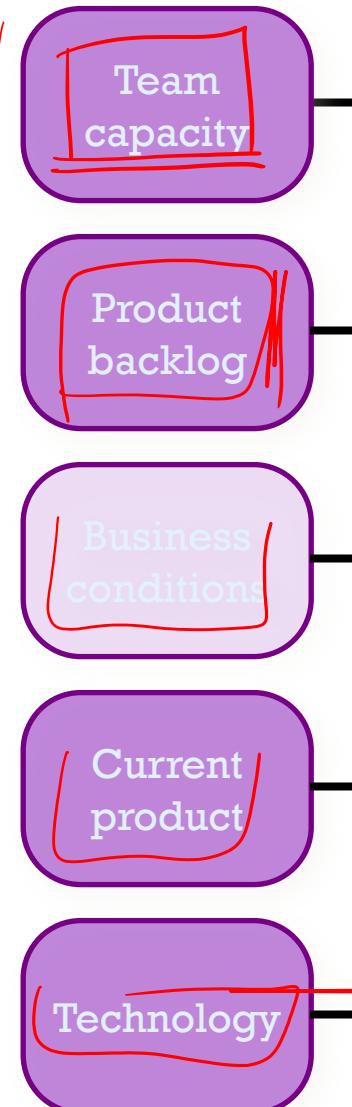
*Sprint backlog*

## Sprint Planning:

- Attended by the Scrum Master, Scrum Team, and any other parties that want to see how the Sprint will likely go.
- 1<sup>st</sup> half of the meeting (~ 4 hours): *real life*
  - Scrum Team and the Scrum Master agree on a Sprint Goal.
  - Scrum Team and the Scrum Master select items from Product Backlog that can be implemented in the next Sprint.
- 2<sup>nd</sup> half of the meeting (~ 12-20 hours): *-Tasks*
  - The Scrum Team creates the Sprint Backlog.
- Product Owner negotiates the Sprint Goal and Backlog to ensure the shippable increment adds business value.

*PO*

Skills



## Sprint planning meeting

### Sprint prioritization

- Analyze and evaluate product backlog
- Select sprint goal

### Sprint planning

- Decide how to achieve sprint goal (design)
- Create sprint backlog (tasks) from product backlog items (user stories / features)
- Estimate sprint backlog in hours

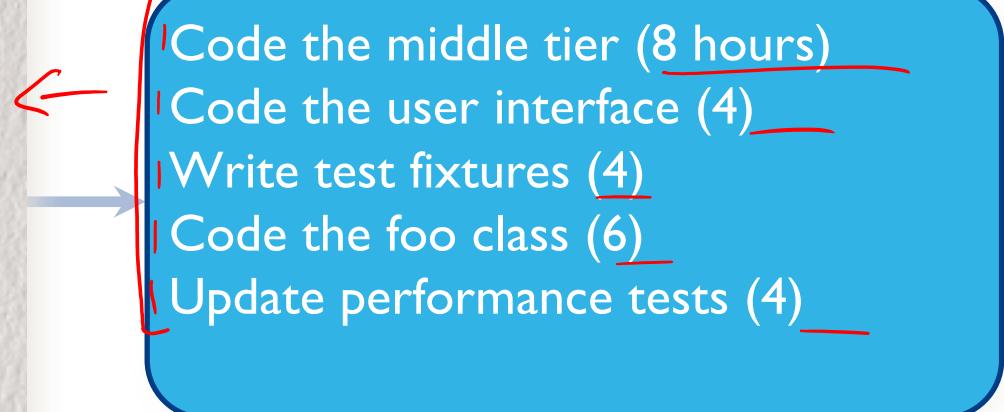
Sprint goal

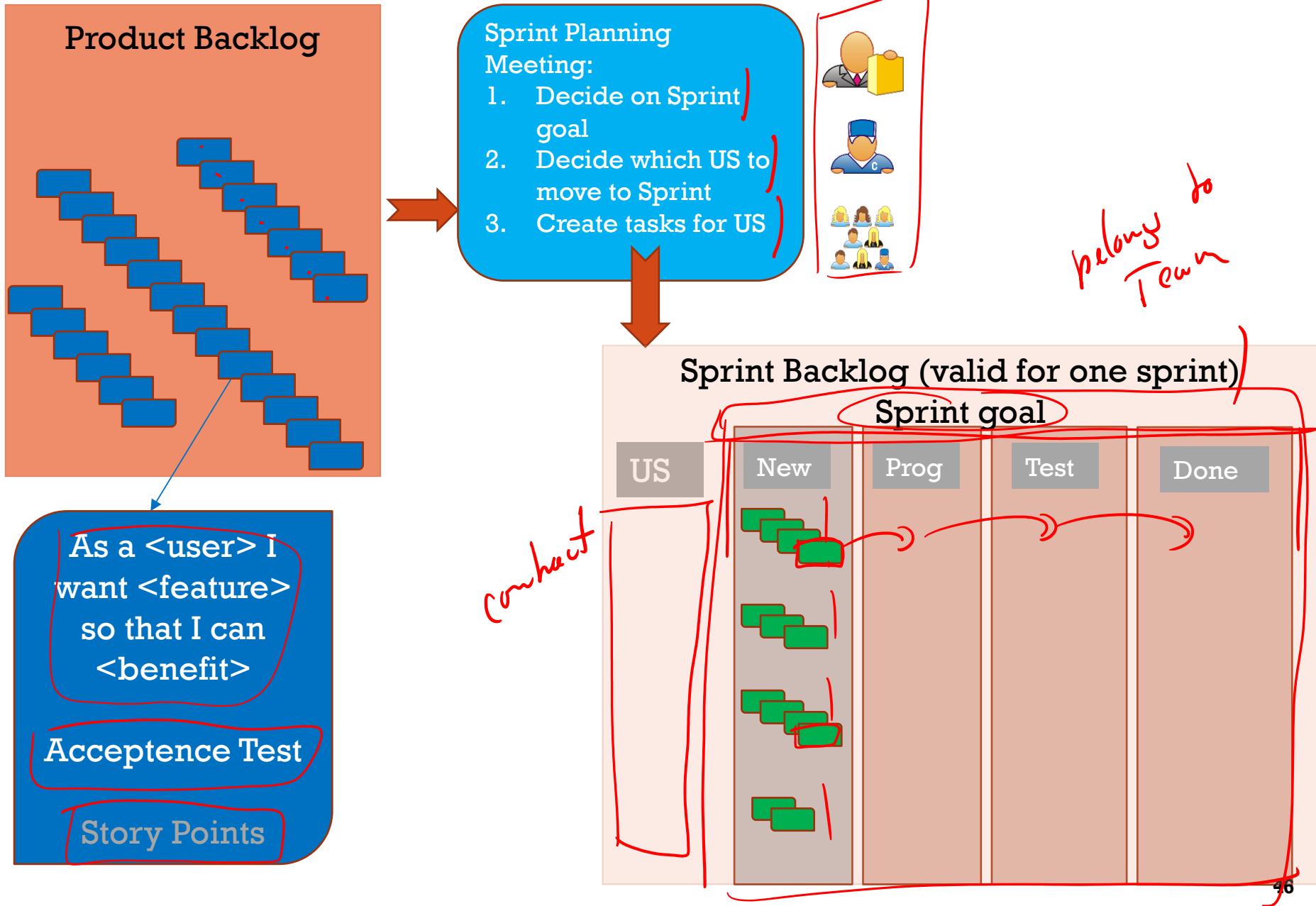
Sprint backlog

# 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
  - Collaborative, not done alone by the ScrumMaster
- High-level design is considered

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





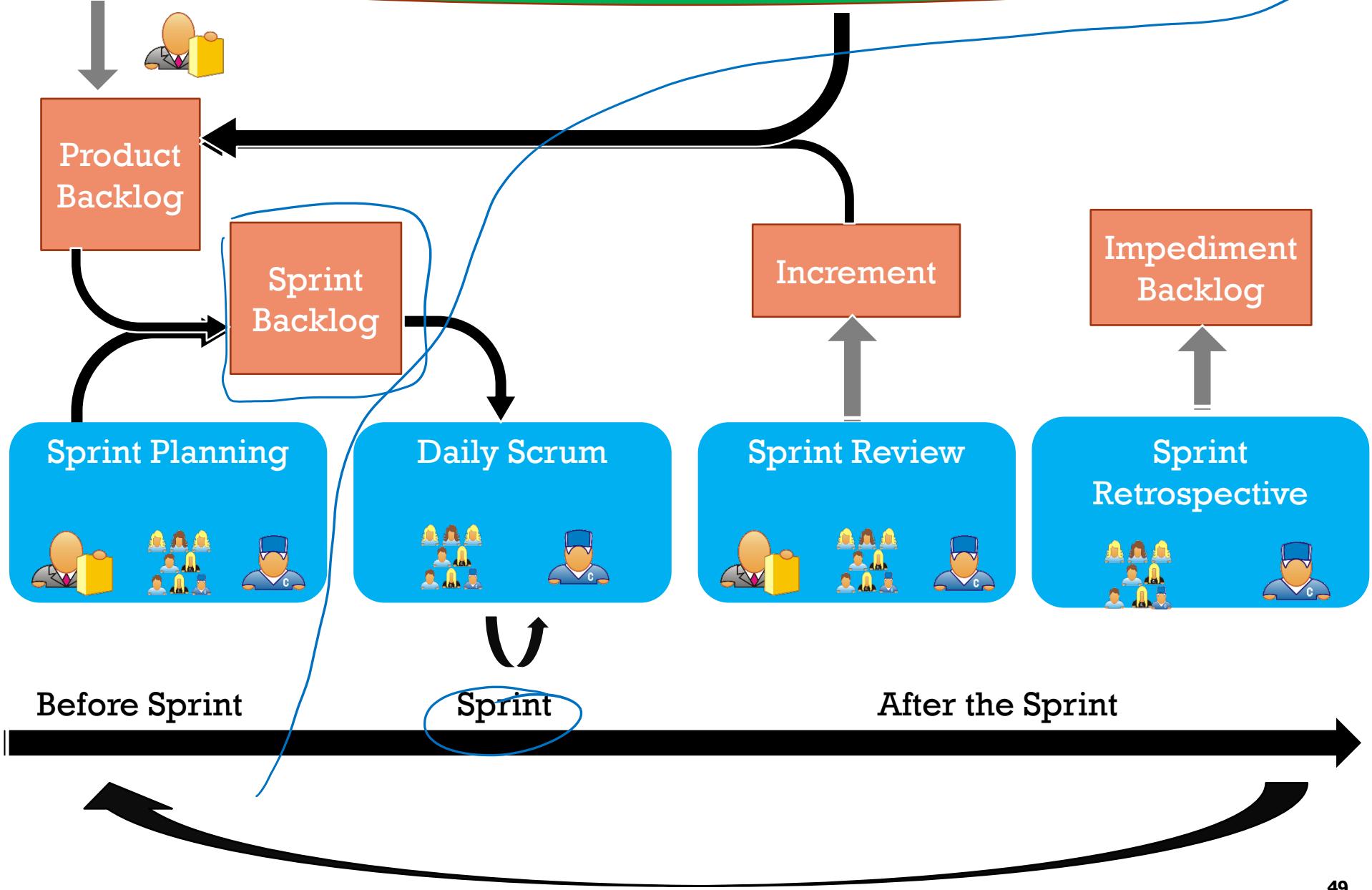
# SPRING PLANNING SUMMARY

PO, SM, Team Dev

- Meeting of whole team
  - Lessons learned from old Sprints are taken into account
  - Usually a couple of hours
- Decides what will get done in the next Sprint
- Results in ~~Product~~<sup>Sprint</sup> Backlog
  - Sprint Goal ~~fullfill~~
  - User Stories ~~- Contract~~
  - Tasks for User Stories (created by team)

# CONDUCTING THE SPRINT

# Customer Vision



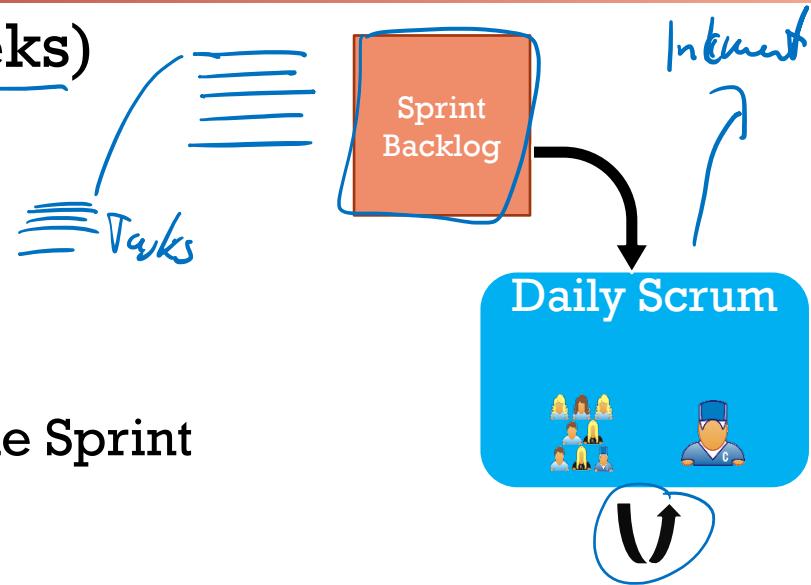
# CONDUCTING SPRINTS

- Sprints are short iterations (2-4 weeks) designed to:

- Produce working software
- Empower the development team

- Sprint mini how-to:

- Take responsibility for a Task on the Sprint Backlog
- Conduct daily standups
- Monitor the Scrumboard (*Burndown and CI/Test dashboards*)
- Large (Spike) and small (manipulate tasks) as needed



# #1: TAKING RESPONSIBILITY FOR A TASK

SER316 SPRING 2019 SPRINT 1 (UNDERSTANDING THE MAIN FEATURES OF MEMORANDA AND FIXING OBVIOUS BUGS) 23 JAN 2019-01

8% ✓ 24 total points | 2 completed points | 4 open tasks | 5 closed tasks | 0 cocaine doses

USER STORY	NEW	IN PROGRESS	READY FOR TEST	CLOSED
<p>✗ #1 As a user I want the program to exit correctly when I click on the "x" or Exit in the File menu so I can exit the program without error. New 2points</p> <p>✗ #8 As a user I want the splash screen and the title of the program to show a new logo and name, so I can see the correct program opening. New 3points</p> <p>✗ #6 As a user I want Look and Feel in the preferences removed so that it is not confusing and inconsistent anymore. New 2points</p>	<p>+  NEW</p> <p>+  #11 Create logo for program</p> <p>+  #18 Fix the bug, so the preferences open</p> <p>+  #17 Delete or annotate the preference checkboxes</p>	<p>&gt;&lt; IN PROGRESS</p> <p>+  #14 Change the Splash screen to display new logo</p> <p>IA 2-4h real 8h</p>	<p>&gt;&lt; READY FOR TEST</p>	<p>&gt;&lt; CLOSED</p> <p>+  #10 Fix terminate</p> <p>+  #9 Find where the terminate is located</p> <p>+  #13 Find where the Splash screen is located</p> <p>+  #15 Change Splash screen to display new name</p> <p>+  #12 Come up with a good name</p>

# TAKING RESPONSIBILITY FOR A TASK

1. Choose a tasks from New column
2. Move task to In Progress column
3. Assign your name to task (this task is now your responsibility)
4. When done, then move to Ready for Test
5. You or a team member should test
6. After test move to Done
7. Rinse and repeat

# #2: THE DAILY SCRUM

- Parameters
  - Daily *full time*
  - 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: 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 reports for the ScrumMaster
  - They are commitments in front of peers

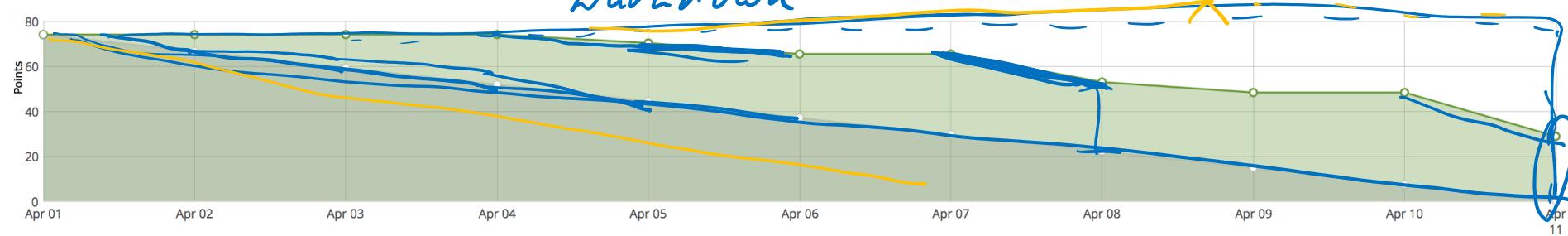
# #3: MONITOR SCRUM BOARD

*Spent Backlog*

- Always keep track of your Scrum board
- Percentages completed?
- Is work evenly distributed?
- Are you on the track of getting everything done?
- Are there any impediments?



*Downturn*

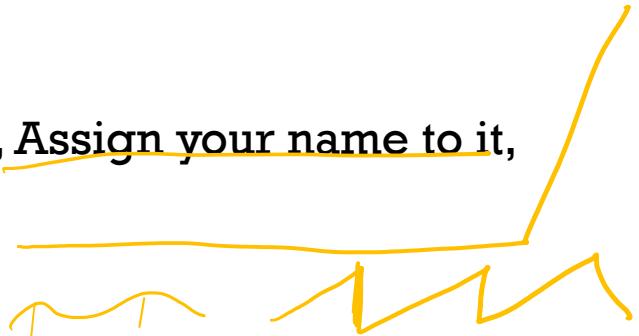


# #4: LARGE (SPIKE) AND SMALL (MANIPULATE TASKS) AS NEEDED

- **Spike:** Work as a team to overcome obstacles
- Split up tasks or merge them together, depending on how the team sees fit
- The Tasks in the Sprint Backlog belong to the Team
  - Team can add/change/delete tasks during the Sprint
- Team cannot add/change/delete User Stories (that would break the contract)

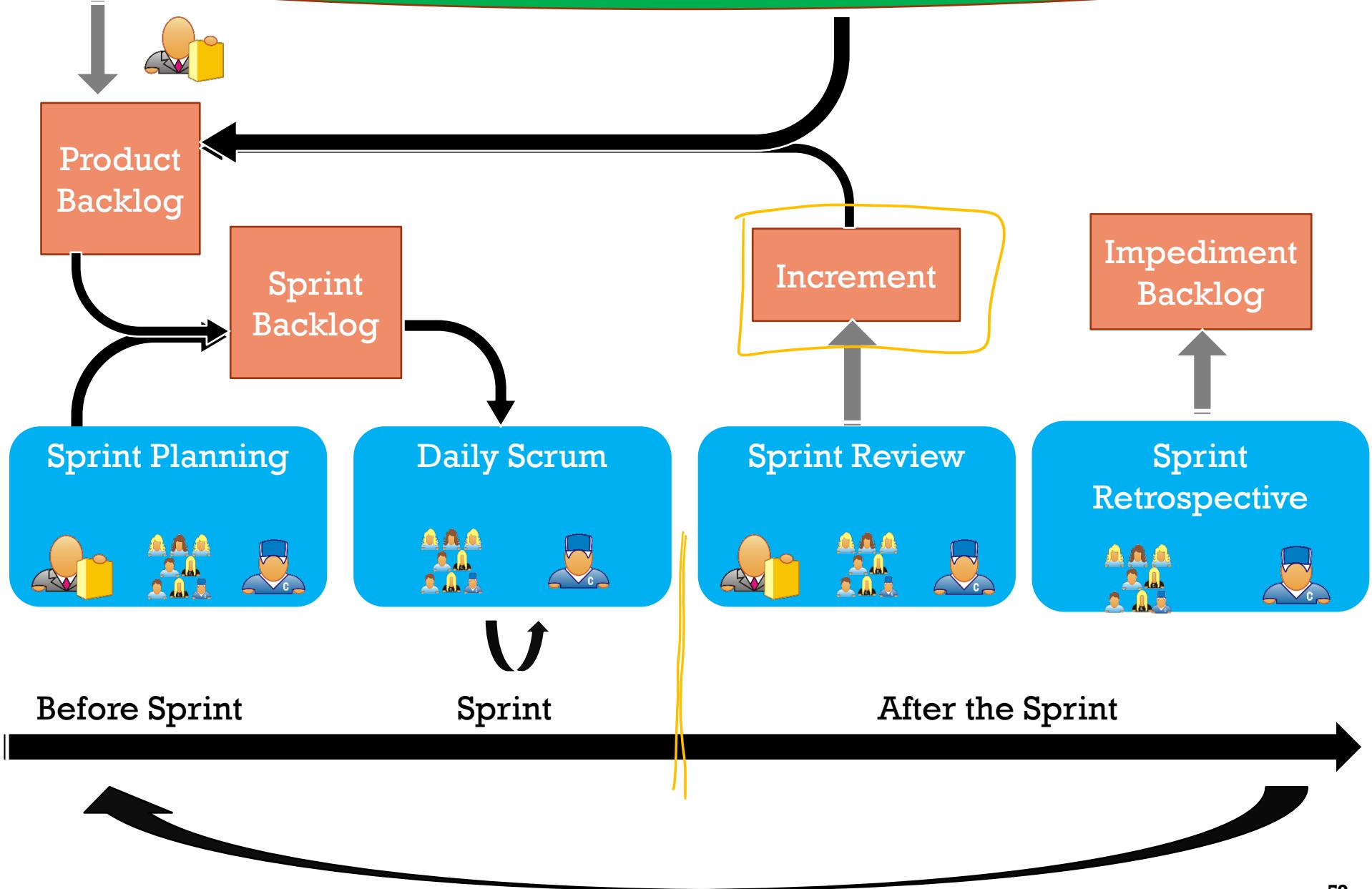
# SPRINT SUMMARY

- Take responsibility for a task
  - Choose a task from New, move to In Progress, Assign your name to it, start working on it
  - You should continuously work on something
- Daily Scrum
  - Talk with the team for 10-15 min:
    - What did you do, What will you do now, Is anything in your way?
- Monitor your Scrum Board
  - You should continuously work on reducing the percentages
- Work as a Team



# REVIEW

# Customer Vision



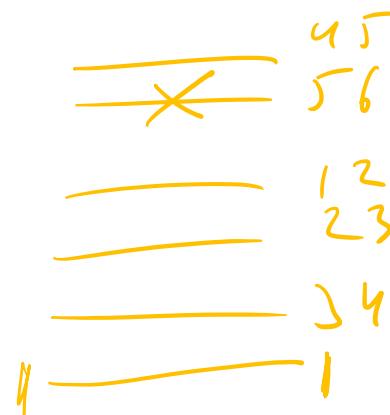
# SPRINT REVIEW

- 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



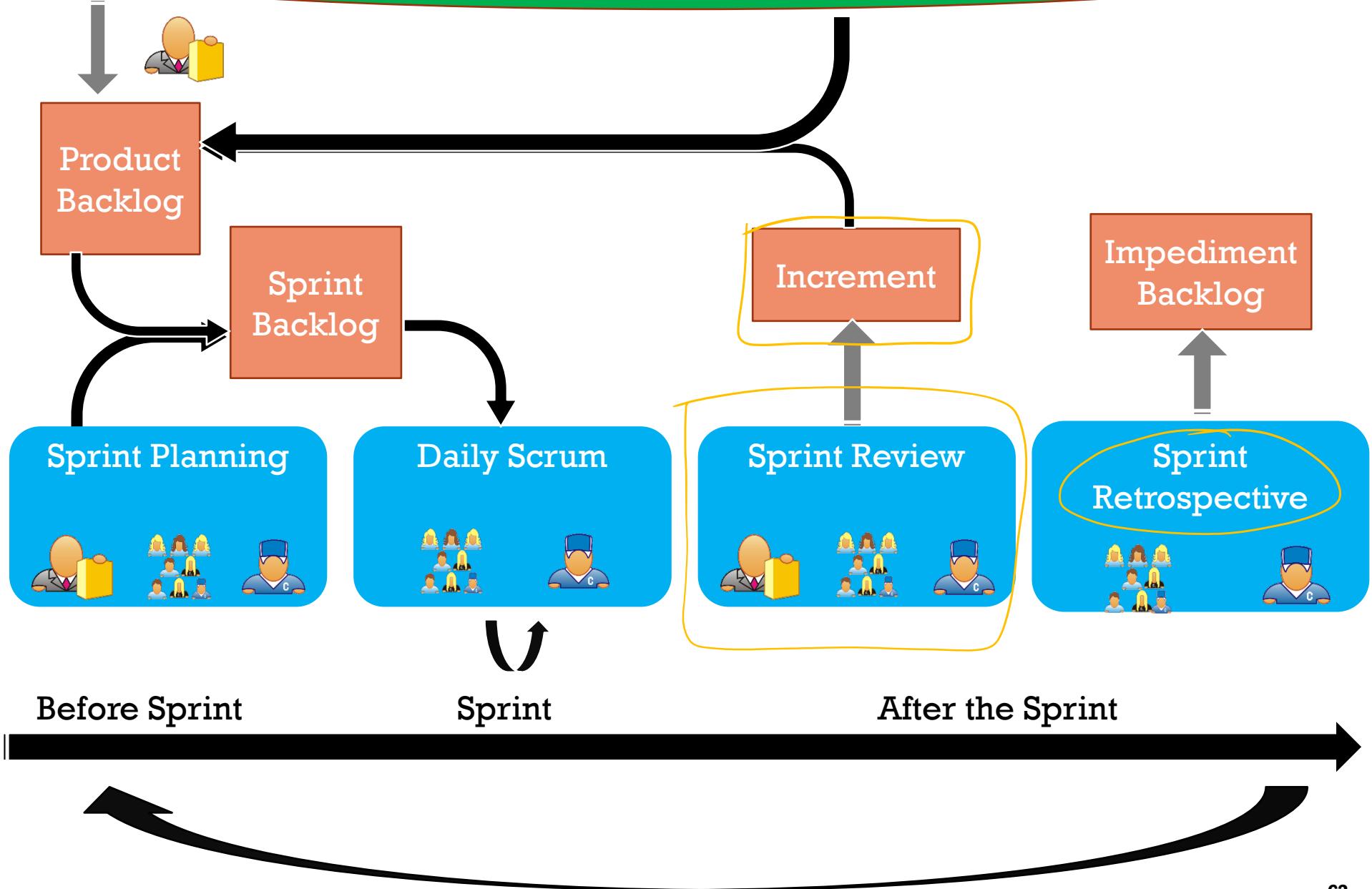
# SPRINT REVIEW

- Customer/Product Owner accept or reject next increment
- Product Backlog is revised/re-prioritized
- Customer needs to be able to provide feedback to the team



# RETROSPECTIVE

# Customer Vision



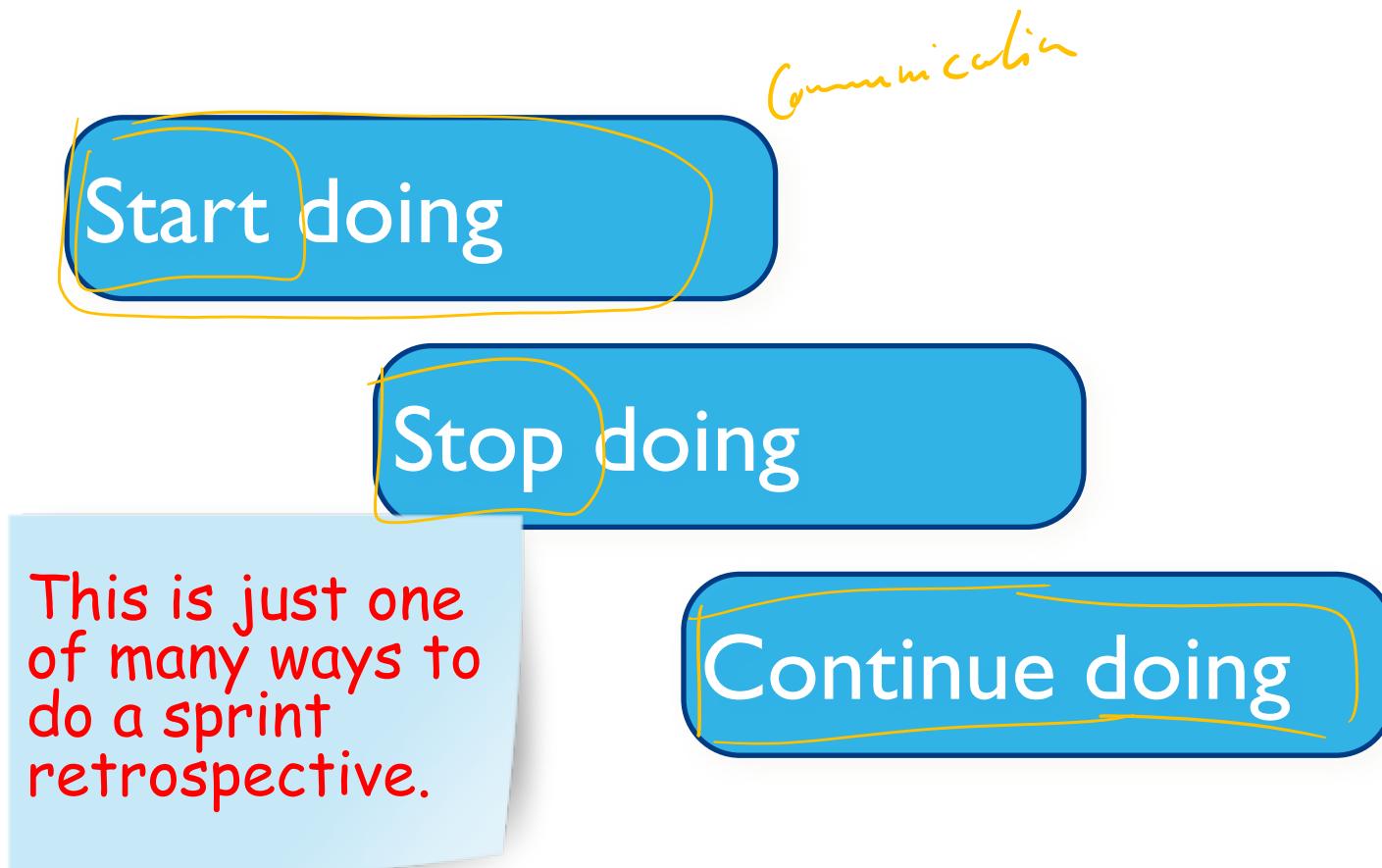
# SPRINT RETROSPECTIVE

- Periodically take a look at what is and is not working (in the team)
- Typically 15–30 minutes
- Done after every sprint
- Whole team can participate
  - Scrum Master
  - Product Owner
  - Team
  - Possibly customers and others



# START / STOP / CONTINUE

- Whole team gathers and discusses what they'd like to:



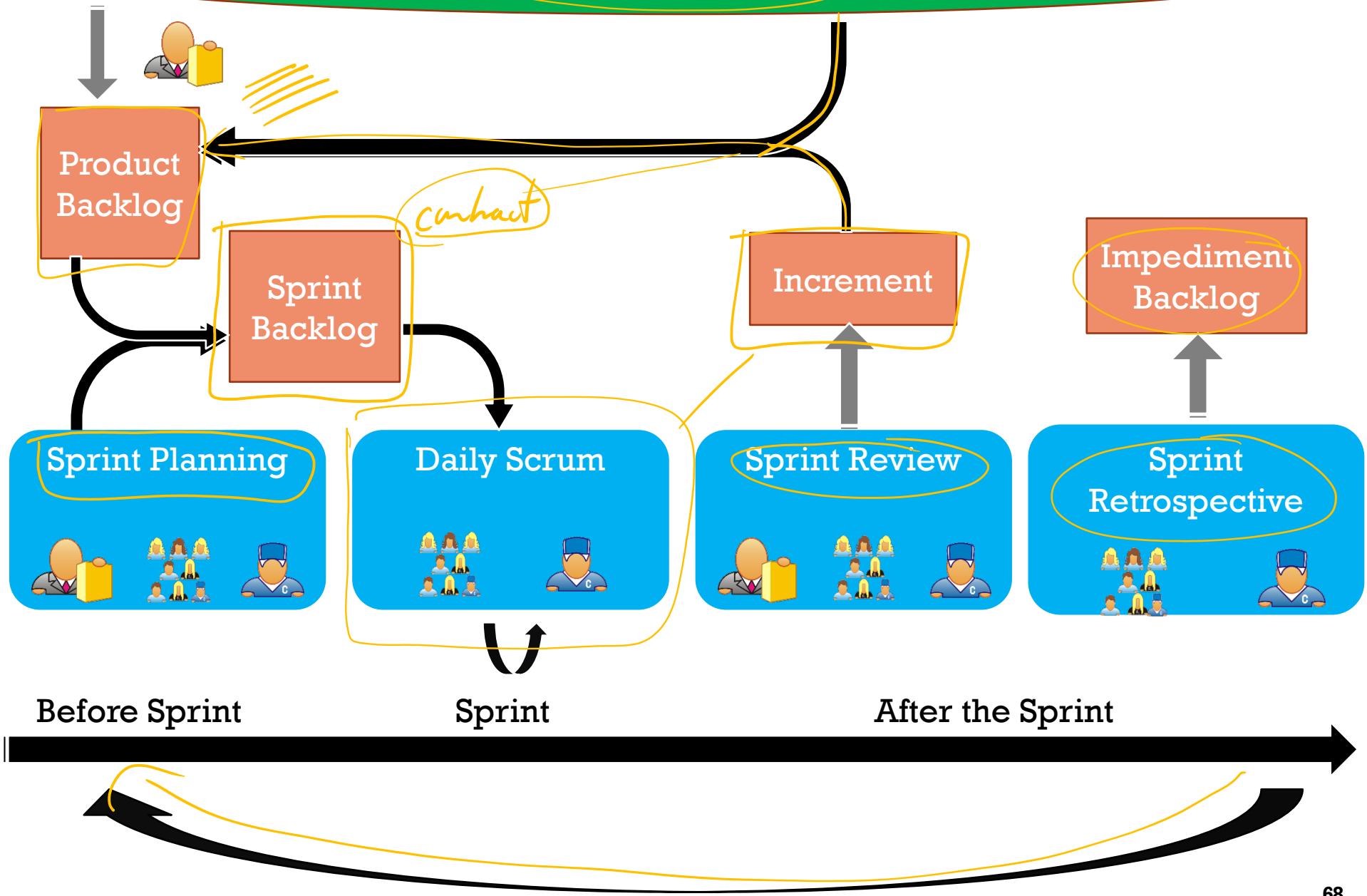
# REVIEW & RETROSPECTIVE SUMMARY

- To reflect on the last Sprint
- Review:
  - To get customer feedback
  - To revise the Product Backlog
- Retrospective
  - To reflect on team work
  - To reflect on the last Sprint planning and Sprint conduction

**The goal is to do better in the next Sprint and learn**

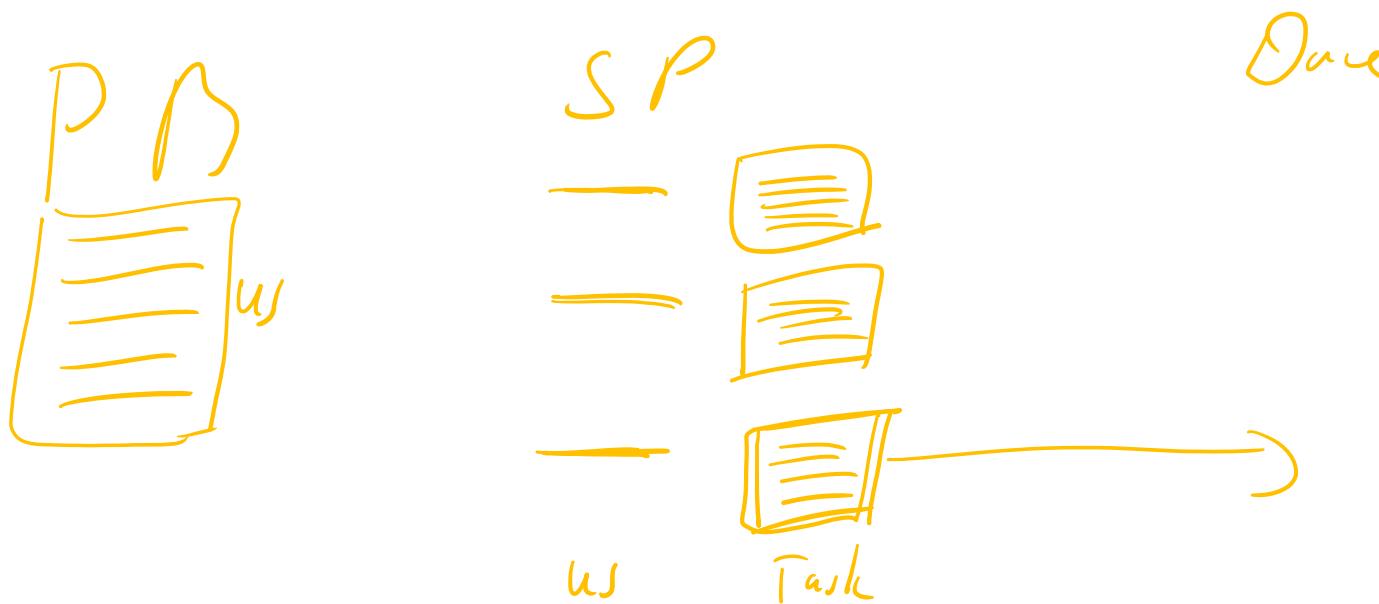
# SUMMARY

# Customer Vision



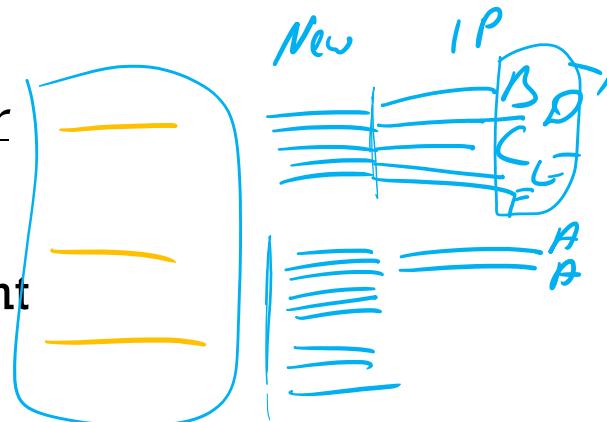
# WHEN ARE YOU DONE?

- All tasks of one User Story are done → User Story is done
- All User Stories from Sprint Backlog done → Sprint is done, goal fulfilled
- All User Stories from Product Backlog done → Product is done



# SPRINT STEPS: EXECUTING THE SPRINT

- Taking responsibility for a Release
  - The team is accountable to the Product Owner
- Taking responsibility for a Story
  - The Scrum Master is accountable for the Sprint
- Taking responsibility for a Task
  - Developers are accountable to the team
- Suggested Machinery
  - Developers should only be working on a small number of tasks at a time (some believe small = 1, we will do 1-2 at a time)
  - Minimize the number of developers working on a story
  - Change the tasks as you see fit
  - Done means DONE, all the way through testing

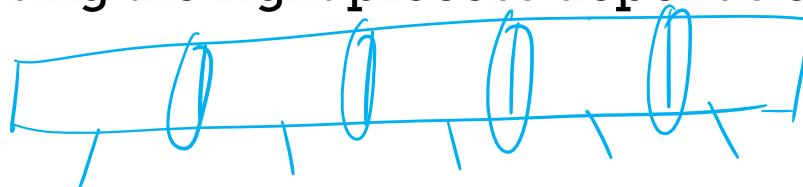


Dev teams are empowered – to estimate, determine tradeoffs, and decide on processes. They accept responsibility for a deliverable and are accountable but get to decide how to do it!

# SCRUM DRAWBACKS

# CLASSIC VS. AGILE

- Large projects often need a more structured process
- Many stakeholders also often need classical approach
- Small projects often better with Agile methods
- Parts of larger systems can be developed with Agile methods (Scrum of Scrums)
- Choosing the right process often decides if project succeeds or fails
- Choosing the right process depends on many factors



# PRACTICAL PROBLEMS WITH AGILE METHODS

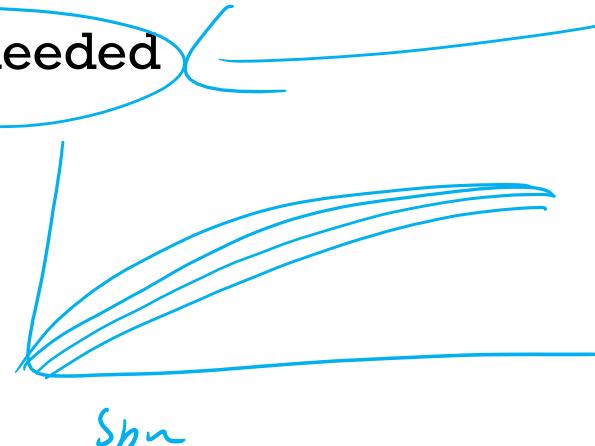
- Most appropriate for new projects rather than maintaining software
- Designed for small co-located teams yet much software development now involves distributed teams
- The informality of agile is incompatible with the legal approach to contract definition commonly used in industry
  - Most software contracts for custom systems are based on a spec, which defines what has to be implemented for the customer.
  - A contract that pays for developer time rather than functionality is often required, but this is not the way contracts are typically written.

# FURTHER NOTES

- Agile does not mean no design
  - There are still design phases – only shorter
  - Requirements are still needed – but they can be changed, adjusted, added – User Stories
- It is still a process and not just ad-hoc programming
- Team needs to work efficiently
  - Lots of communication needed
- Refactoring is periodically needed

refactor / redesign

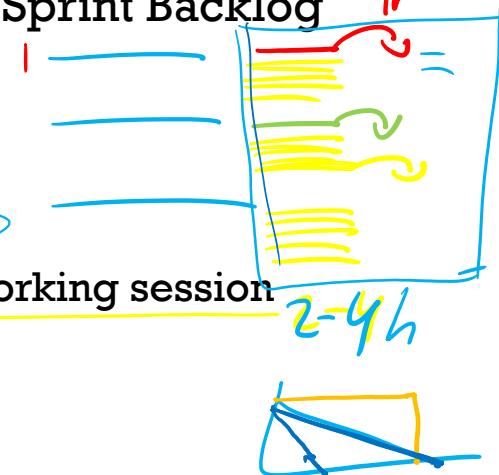
light weight



SCRUM IN YOUR  
PROJECT

# TAIGA: SCRUM BOARD

- Maintain your Product Backlog *You*
  - Write User Stories in the correct format, including Acceptance Tests
- Maintain your Sprint Backlog
  - Stories get moved to the Sprint Backlog in Sprint planning (without Product Owner approval no Stories should be taken on or off the Sprint Backlog *IP* during the Sprint)
  - Create tasks for each User Story
    - Usually more than one task per Story *1\**
    - All User Stories need to have tasks after the Sprint Planning
    - A task should be small enough so it can be finished in one working session
    - Tasks can be added/changed/deleted during the sprint
- Monitor your Scrum Board
- Wiki:
  - Put information about your meetings, discussions, review, planning, retrospective on your Wiki (more info in the kickoff document)
- **All your project work has to be documented on your Scrum Board**



# ROLES

- The instructor will be the Customer/Product Owner<sup>-us</sup>
- Your group will be the scrum team
  - You as a team will write the User Stories and prioritize them
- One of your team members will be the Scrum Master and you will take turns

You : Dev Team      Dev Team  
                                |  
                                not the boss

# PROCESS

- Conduct Sprint Planning
- Conduct the Sprint:
  - 3 Stand-up meetings a week
    - Answer the three questions (can be done distributed, you can create a specific channel for it)
  - Take tasks from Sprint Backlog and work on them
    - Make sure to move them correctly
  - Goal should be to finish all tasks and move them to DONE in a continuous fashion
- Review/Retrospective
  - Present your new working software to Customer/Product Owner (teaching team) (5min screencast and answering some questions – should not take you longer than 1 hour overall)
  - Do a retrospective on your process/team/work (should not be longer than 30-60 min)  
*15-30*
  - TIP: Meet to do the retrospective and do the planning right after that. You will have a 1-2 hour meeting and be done.

Mo Wed Sat

# WHICH TEAMS ARE SUCCESSFUL

- The ones communicating a lot
- The ones understanding the process
- The ones working consistently
- The ones where each team member feels responsible
- The ones using the process correctly

!!

ask