

Scrum

COMPSCI5059 - Software Engineering

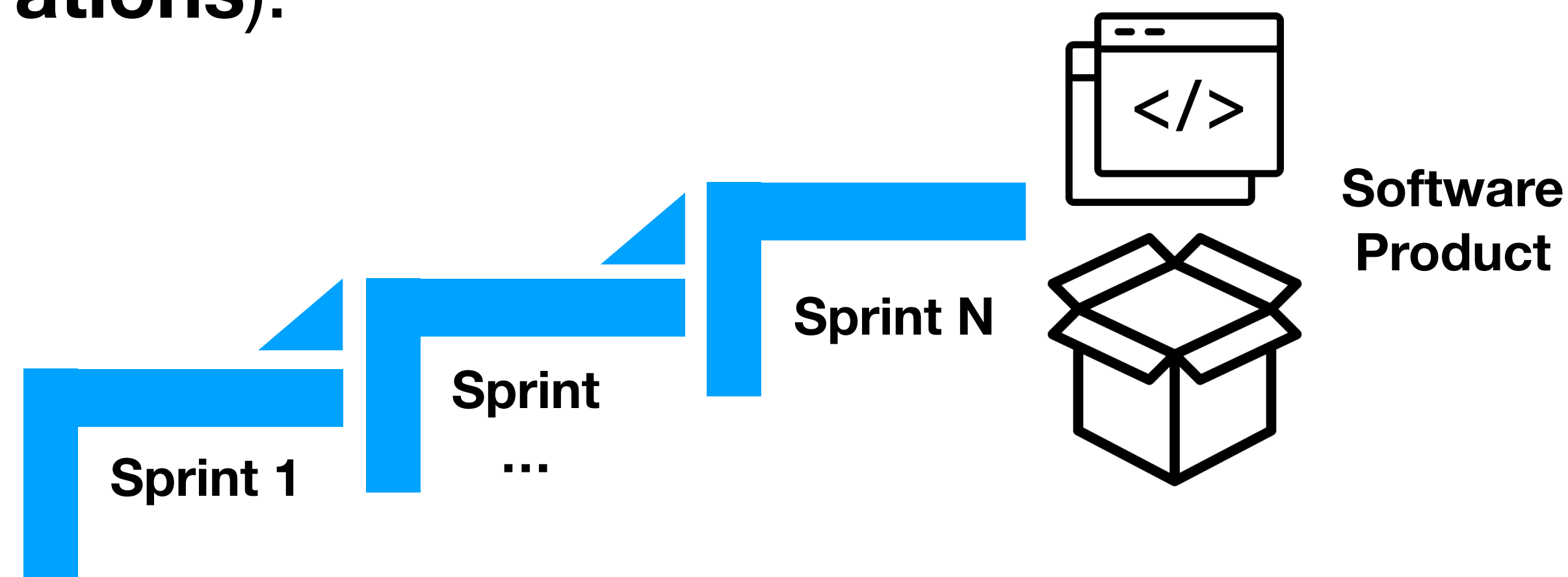
H. Gül Calikli, Ph.D.

Overview

- ☐ Scrum - Introduction
- ☐ The Scrum Team
- ☐ Scrum Meetings
- ☐ Recording Progress and Work To Do

Scrum - Introduction

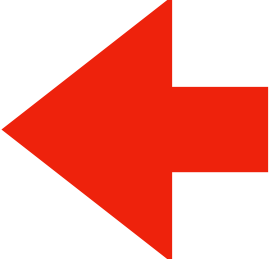
- Scrum is a framework for the management of iterative and incremental product development.
- Scrum was applied to software projects in the early 1990s.
- Scrum is one of the most popular ways to implement Agile Software Development.
- The Scrum approach is to divide the project into smaller logical chunks and execute those chunks in short iterations of ideally 1-4 weeks, which we call **sprints** (i.e., **iterations**).



Sprints (Iterations)

- A **sprint** (i.e., **iteration**) is between 1- 4 weeks, typically 2 weeks.
- Sprints are time-boxed (i.e., must end on time).
- If a story can't be finished it is returned to the product backlog.
- Problems that can't be fixed during a sprint generate stories to be added to the product backlog.
- An entire sprint can be dedicated to fixing problems.

Overview

- ☒ Scrum - Introduction
- ☐ The Scrum Team 
- ☐ Scrum Meetings
- ☐ Recording Progress and Work To Do

The Scrum Team

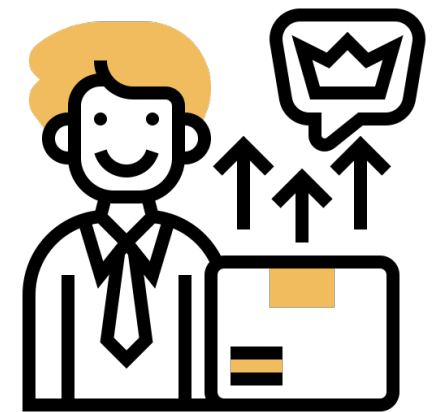


Product Owner: responsible for the business value of the product (customer). Selects what gets done and when.

The Scrum Team



Product Owner: responsible for the business value of the product (customer). Selects what gets done and when



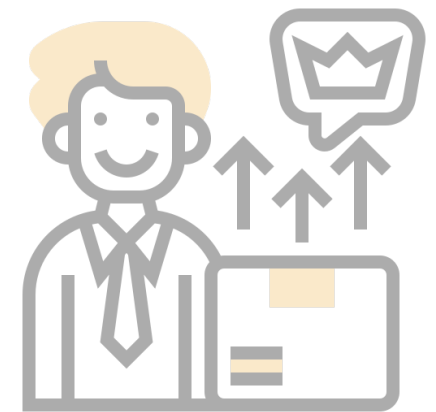
Scrum Master: ensures the team is motivated and productive.

- Removes obstacles.
- Ensures proper processes are followed.

The Scrum Team

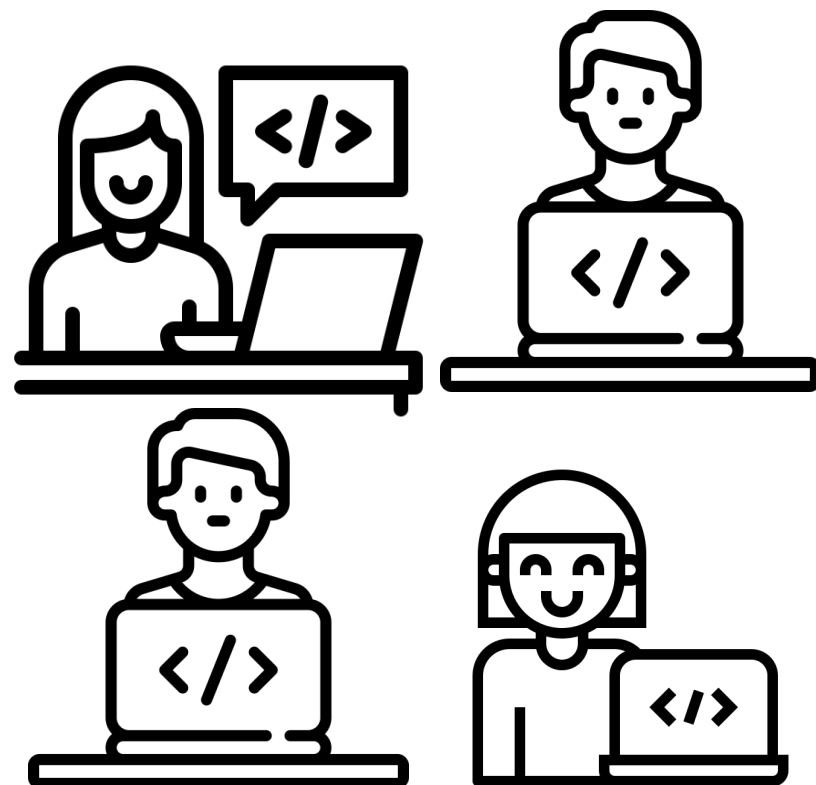


Product Owner: responsible for the business value of the product (customer). Selects what gets done and when



Scrum Master: ensures the team is motivated and productive.

- Removes obstacles.
- Ensures proper processes are followed.



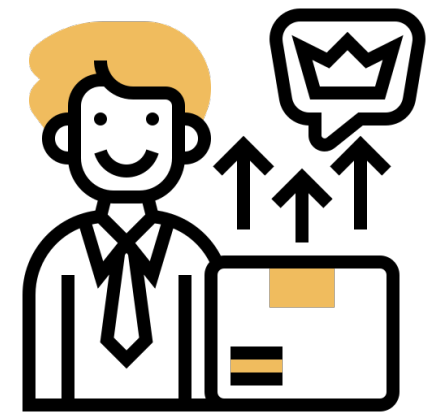
The Scrum Team: Between 3 and 9 people in total, including **product owner** and **scrum master**.

- Developers and Testers.
- User Experience analyst, if appropriate.

The Scrum Team

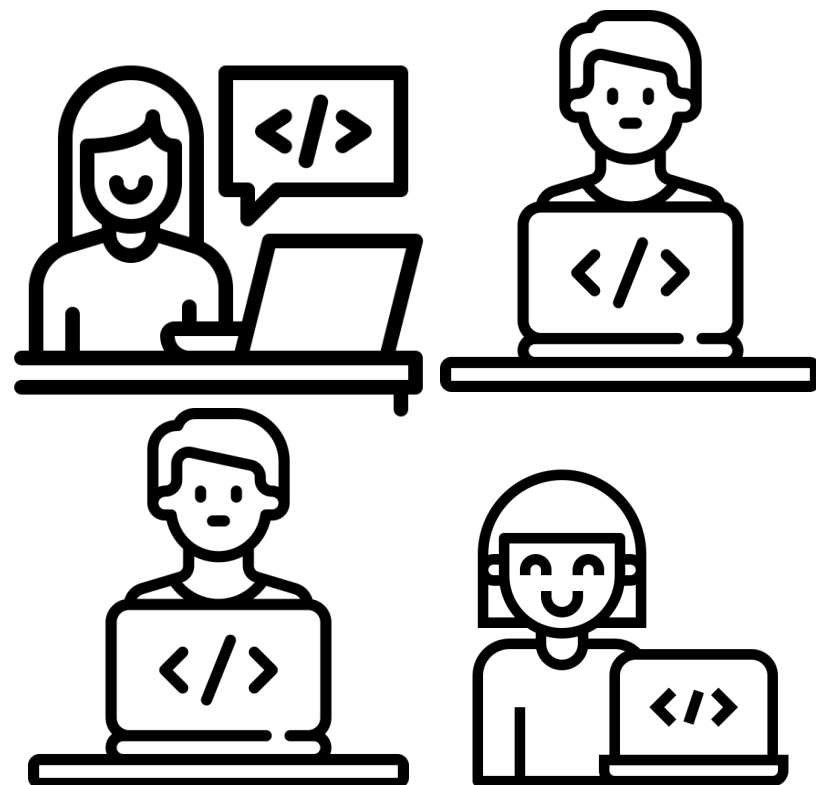


Product Owner: responsible for the business value of the product (customer). Selects what gets done and when



Scrum Master: ensures the team is motivated and productive.

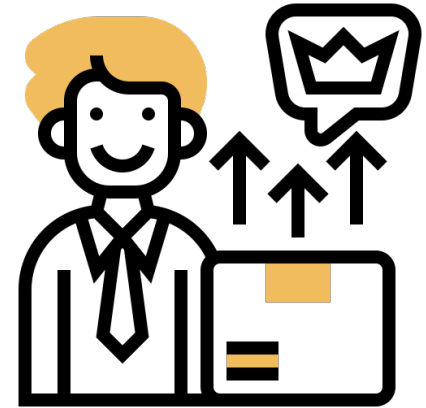
- Removes obstacles.
- Ensures proper processes are followed.



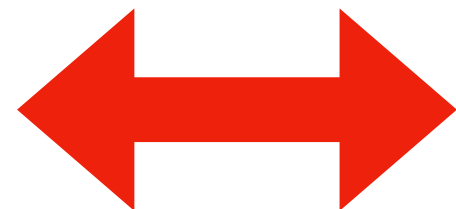
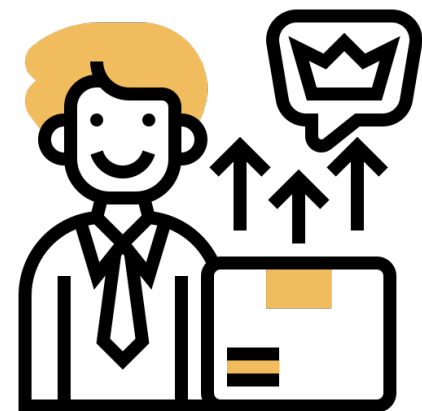
The Scrum Team: Between 3 and 9 people in total, including **product owner** and **scrum master**.

- Developers and Testers.
- User Experience analyst, if appropriate.

The Scrum Master

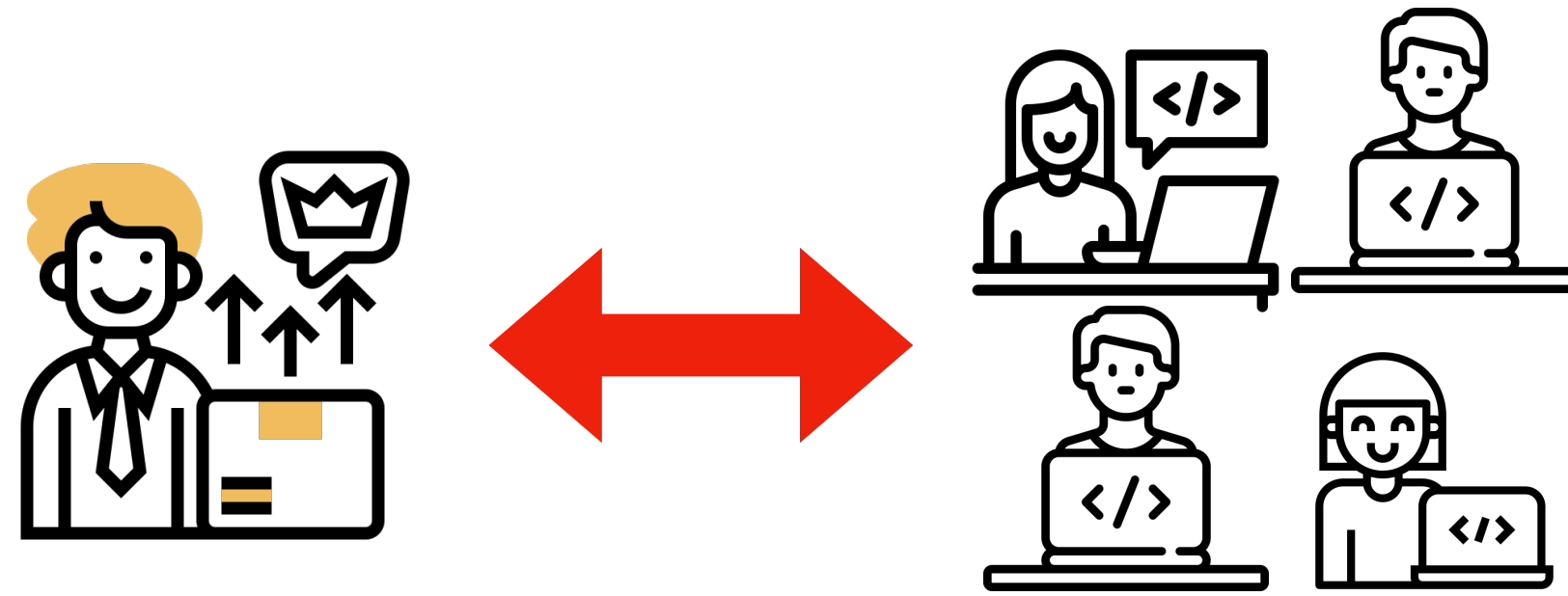


- **Not** a team leader! The team is self-organising!

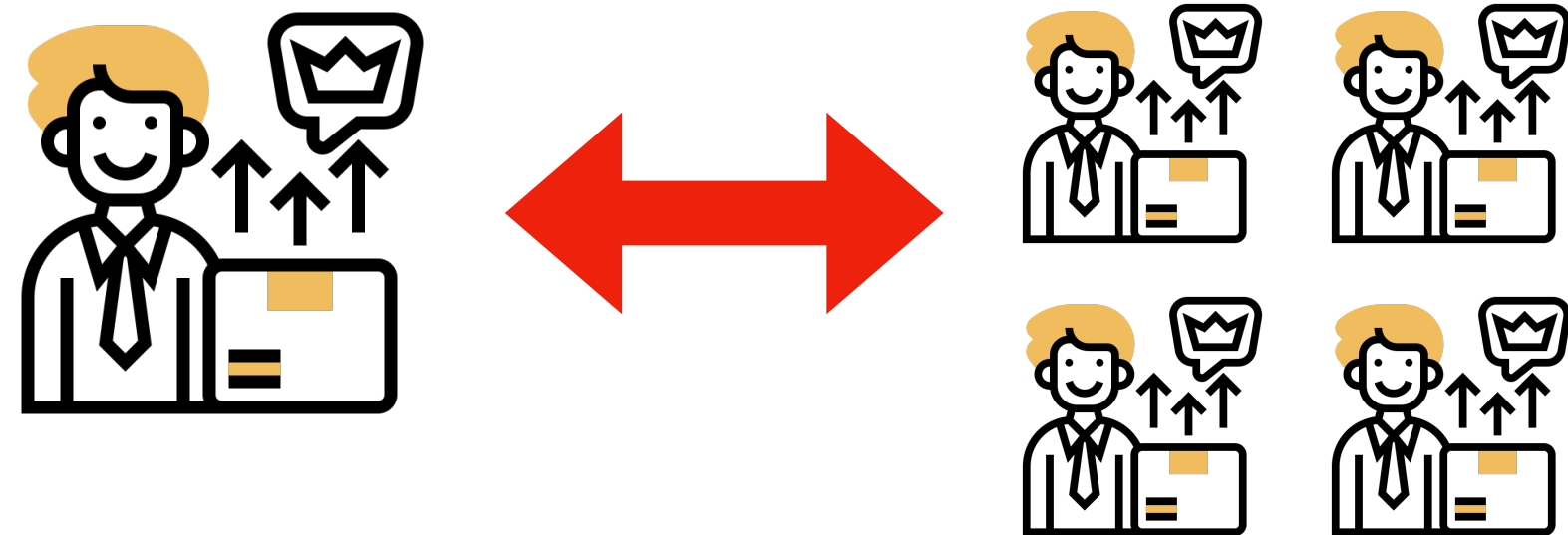


- Works with the **Product Owner**
 - Helps them define the product backlog (stories).
 - Translates the customers' and stakeholders' language into words the team will understand.

The Scrum Master



- Works with the **team members**
 - Facilitates the events
 - Coaching and teaching.
 - Removing impediments.

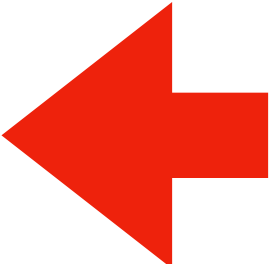


- Works with the rest of the **organisation**.
 - Liases with other scrum masters.

Overview

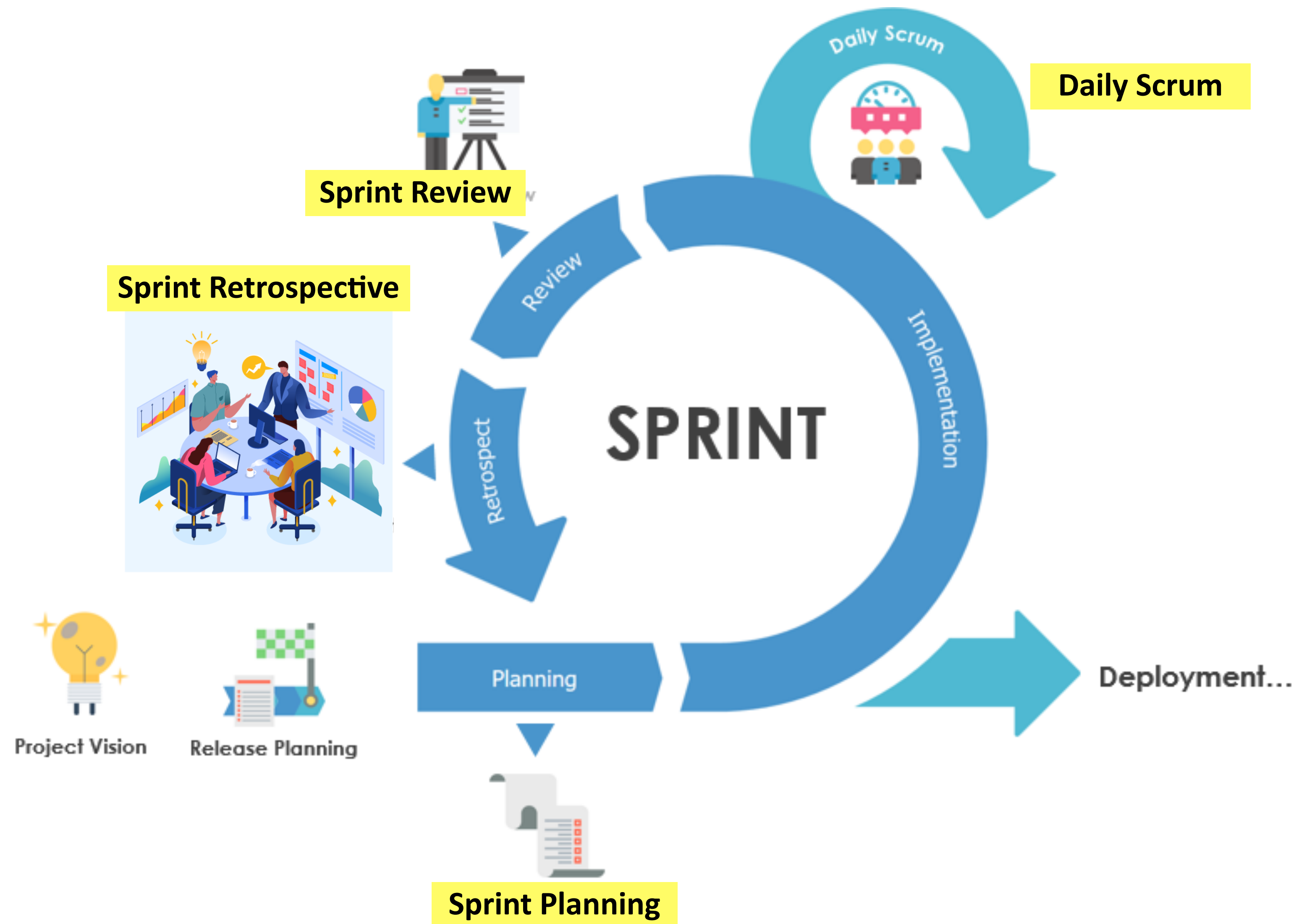
☒ Scrum - Introduction

☒ The Scrum Team

☐ Scrum Meetings 

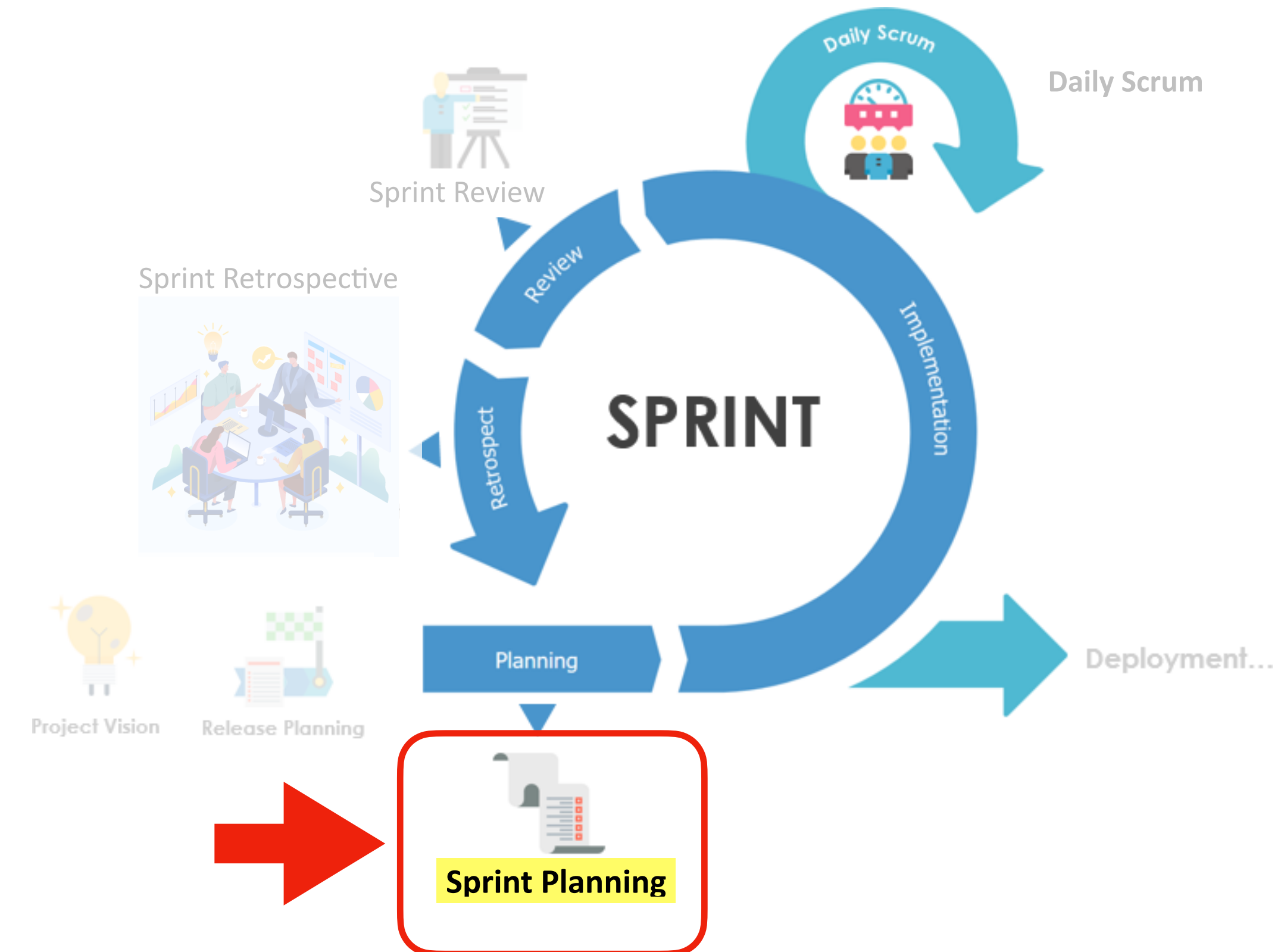
☐ Recording Progress and Work To Do

The Scrum Meetings



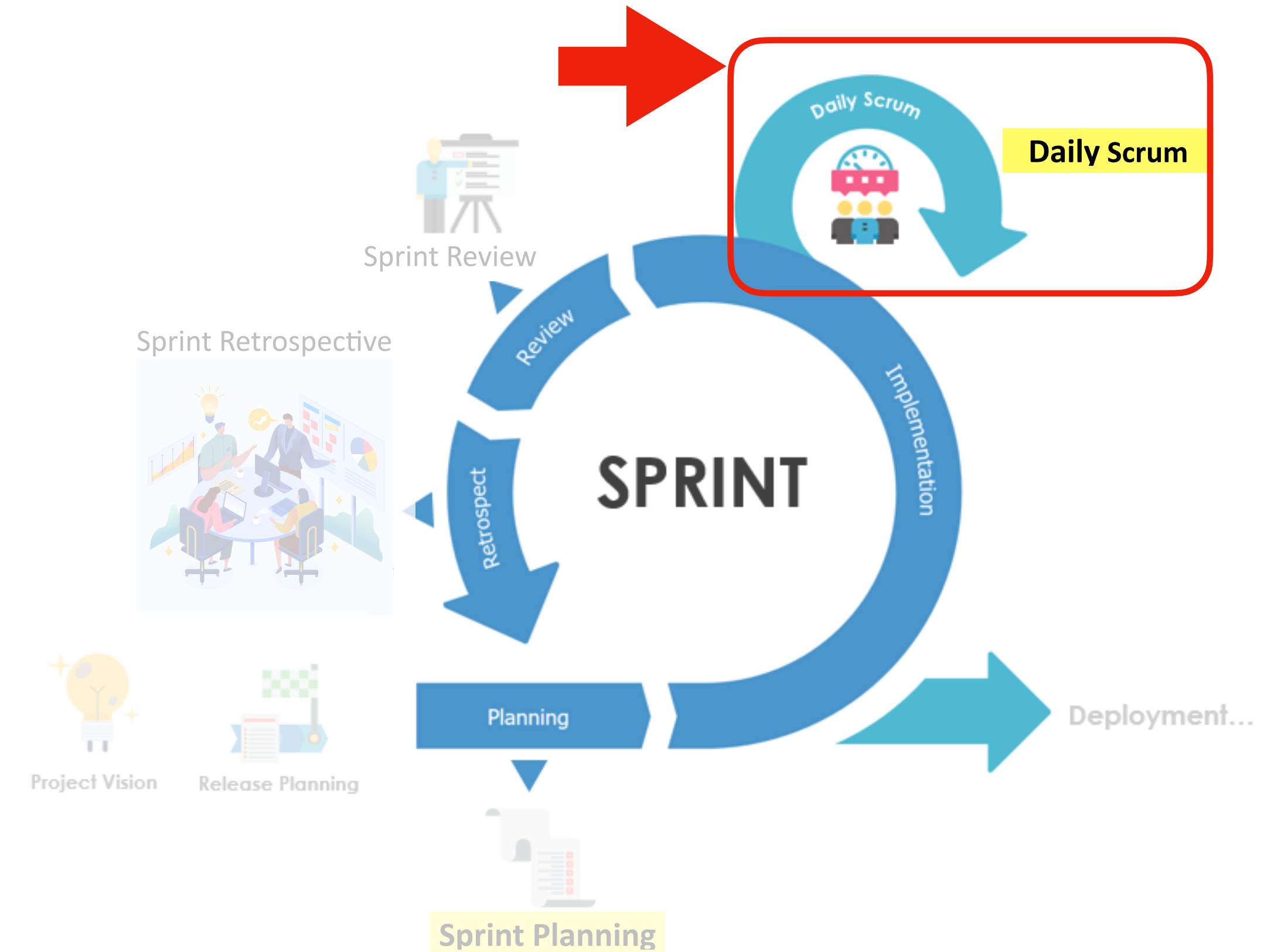
Sprint (Iteration) Planning

- Sprint (iteration) planning at the start:
 - The team and Product Owner agree on what to do in the sprint.
 - The meeting is time-boxed for 4 hours (for a 2 week sprint).
 - The stories to be included are chosen.
 - The product owner then leaves and the work is divided up.



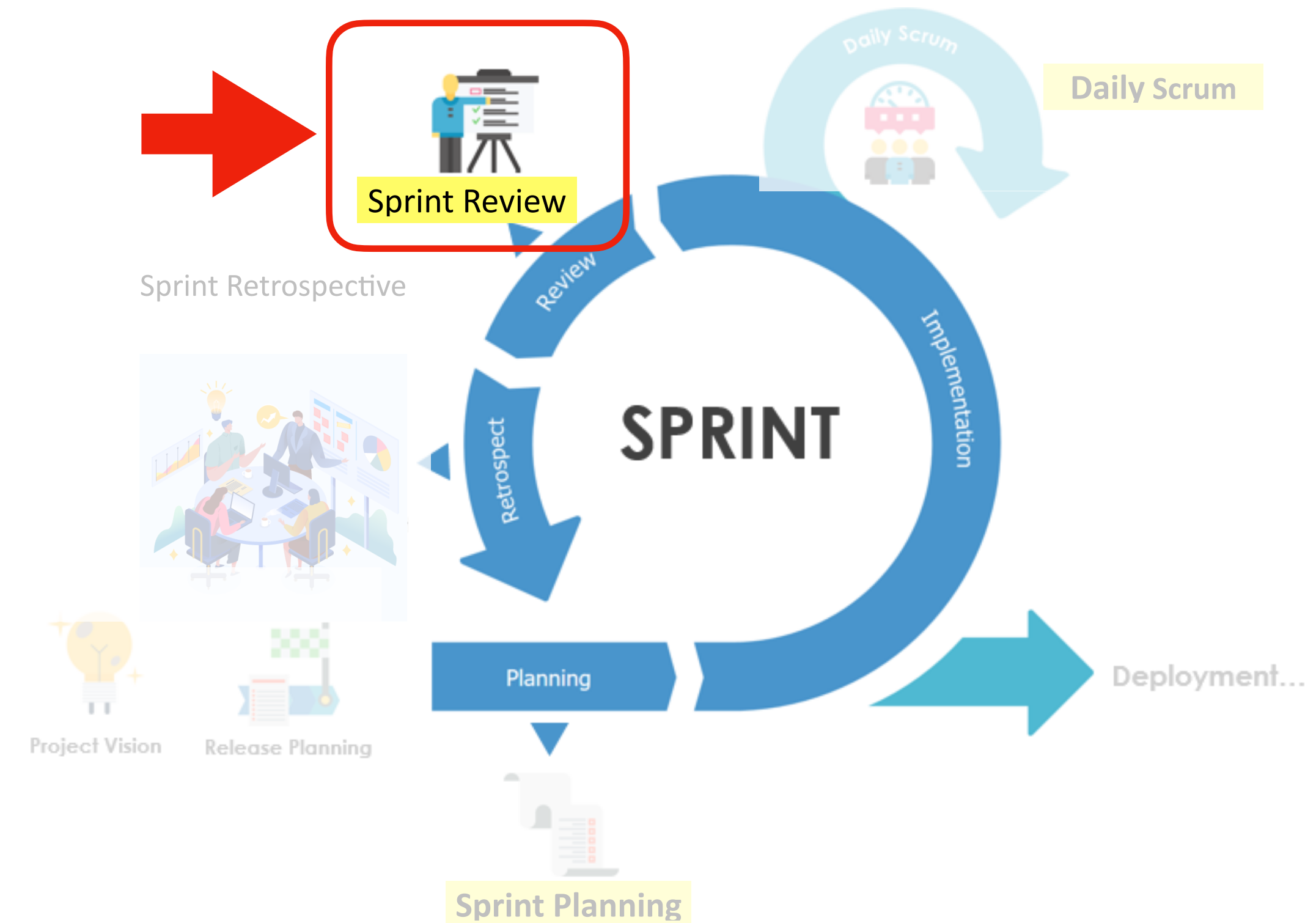
Daily Scrum

- Team stand up meeting at the start of each day.
 - Time-boxed at 15 minutes. Only the team members speak.
 - What did you do yesterday
 - What will you do today.
 - Are there any impediments standing in your way. The scrum master will try and remove them.



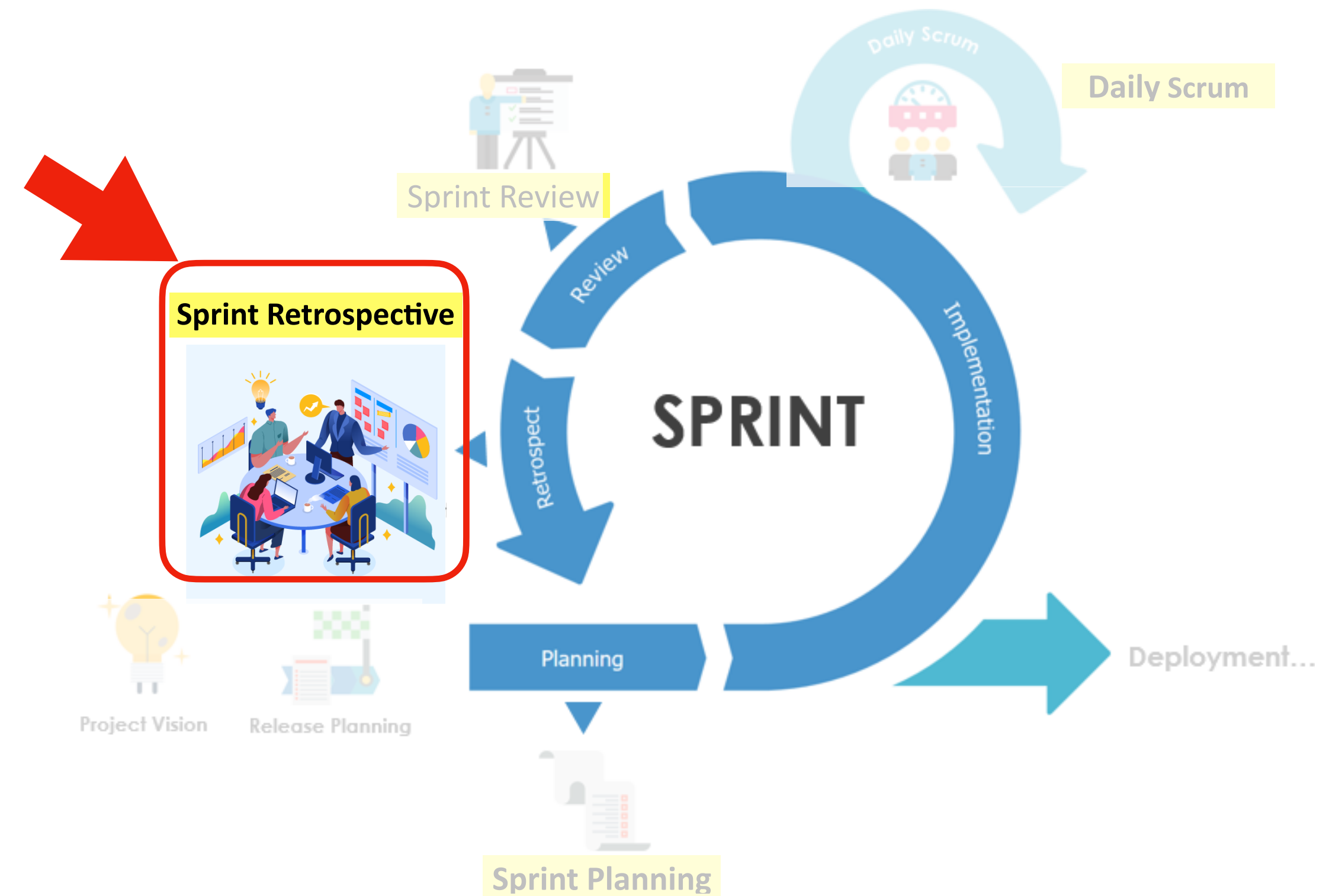
Sprint Review

- Sprint review is at end of the sprint.
 - Time-boxed to 2 hours (for 2 week sprint)
 - Product owner identifies what has been done.
 - Team discuss what went well, what not so well.
 - Team demonstrate the work they have done.
 - Product owner discusses the remaining product backlog and likely completion date.
 - Discussion on what to do for the next sprint.

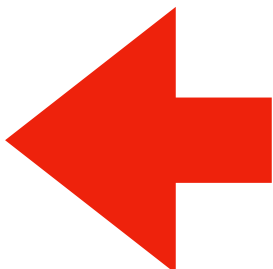


Sprint Retrospective

- Sprint Retrospective: Look for possible improvements.
 - Time-boxed to 1.5 hours (for 2 week sprint)
 - How the sprint went: people, relationships, processes and tools.
 - What went well.
 - Develop a plan to implement improvements.



Overview

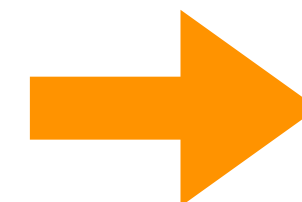
- ☒ Scrum - Introduction
- ☒ The Scrum Team
- ☒ Scrum Meetings
- ☐ Recording Progress and Work To Do 

Iteration Plan

- An iteration will consist of small number of user stories.
- The team discusses the stories in priority order.
 - The customer (product owner) must be there to answer questions.

A small project with 12 user stories

User Story Name	Priority	Effort (SPs)
A	Must have	4
B	Must have	2
C	Must have	2
D	Must have	1
E	Must have	8
F	Should have	4
G	Should have	4
H	Should have	2
J	Could have	1
K	Could have	2
L	Would like to have	4
M	Would like to have	2



Iteration	User Stories	Total Effort (SPs)
1	C, E	10
2	A, B, D, H	9
3	F, G, K	10
4	J, L, M	7

Iteration Plan

- An iteration will consist of small number of user stories.
- The team discusses the stories in priority order.
 - The customer (product owner) must be there to answer questions.
- **The MoSCoW value will be on the story card.**

Title: "Notification to Prevent Overspending"	Priority: Should have	Estimate: 2 SP
User Story: As a credit card holder I want to receive a notification when I have less than £1000 in my account after making a big deposit so that I don't overspend.		
Acceptance Criteria: Given the user has opened an account and received a credit card When the user has less than £1000 after a bigger deposit Then the system notifies the user of their balance.		

Planning an Iteration

- Each **user story** is split into individual **tasks**.
- The developers pick the tasks that they want to work on.
- They estimate the time needed for each of their tasks.
 - Usually in **ideal hours**.

1 story point = 1 ideal day = 8 ideal hours.

Splitting User Stories to Tasks - Example

- Given the following **user story**:
 - *“As a user I want to see on the website all sold trips to a specific destination so that ...”*

Splitting User Stories to Tasks - Example

- Given the following **user story**:
 - *“As a user I want to see on the website all sold trips to a specific destination so that ...”*
- The **tasks** can be as follows:
 - Task 1: Create trips API where we can send data from the Agency Database
 - Task 2: Create web page that will display all the content of trips API with sorting and paging
 - Task 3: Add ‘search by destination’ functionality to the trips web page we created.
 - ...

Iteration Length and Velocity

- All iterations (i.e., sprints) are the same length (e.g., 2 weeks).
- The iteration **velocity** will then be the expected number of story points (i.e., ideal days) that can be **completed** in **each iteration**.
- Velocity is the same for all iterations.

Estimated Project Length = (Total # of SPs) / velocity

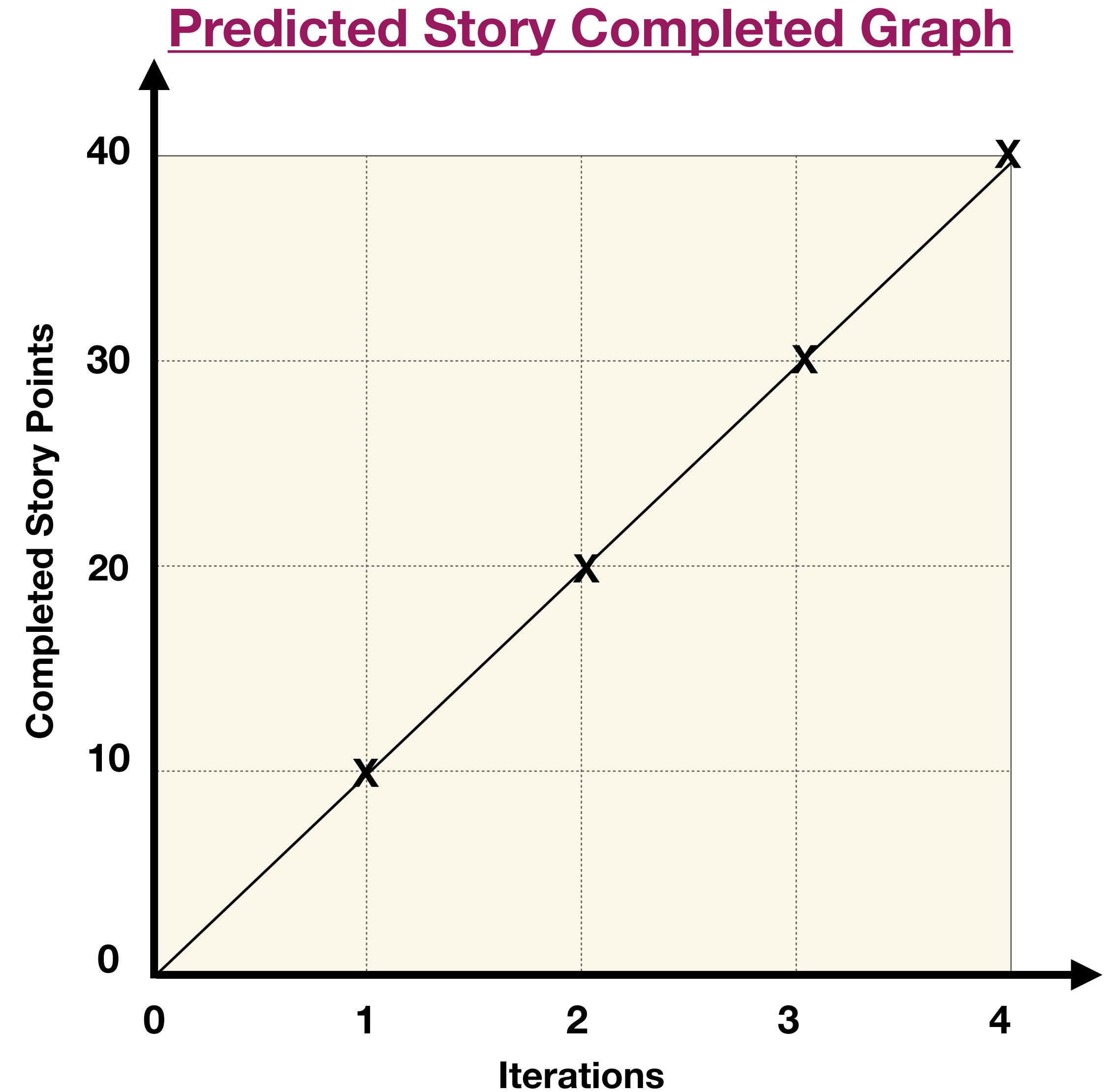
Total number of SPs for
all user stories

Monitoring Velocity

- We can monitor velocity using:
 - Story Points Completed Graph
 - Iteration Burndown Chart

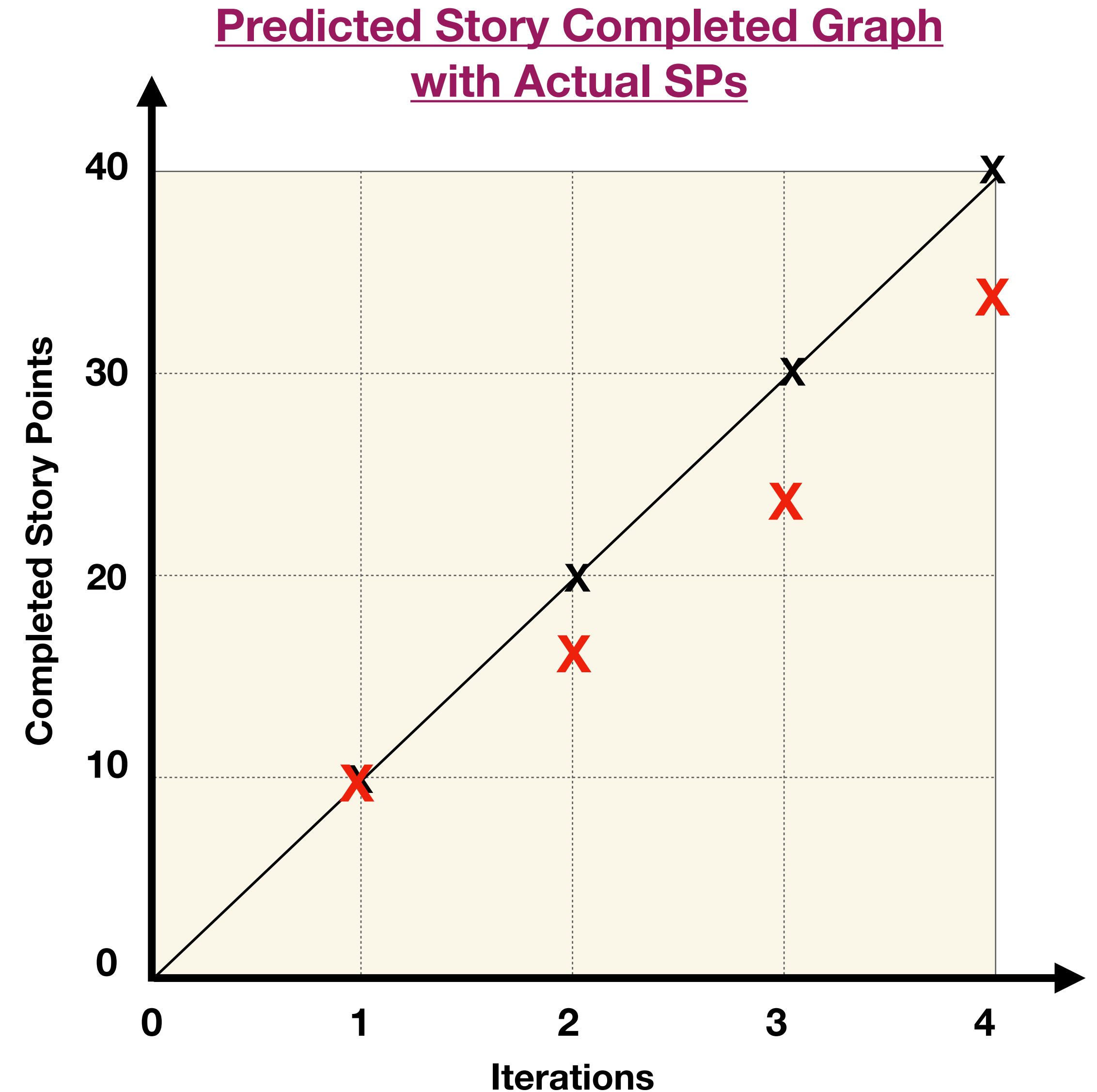
Story Completed Graph (Example)

- Our target is 40 Story Points (SPs) in 4 iterations.
- Velocity = 10 = 40 SPs / 4 iterations



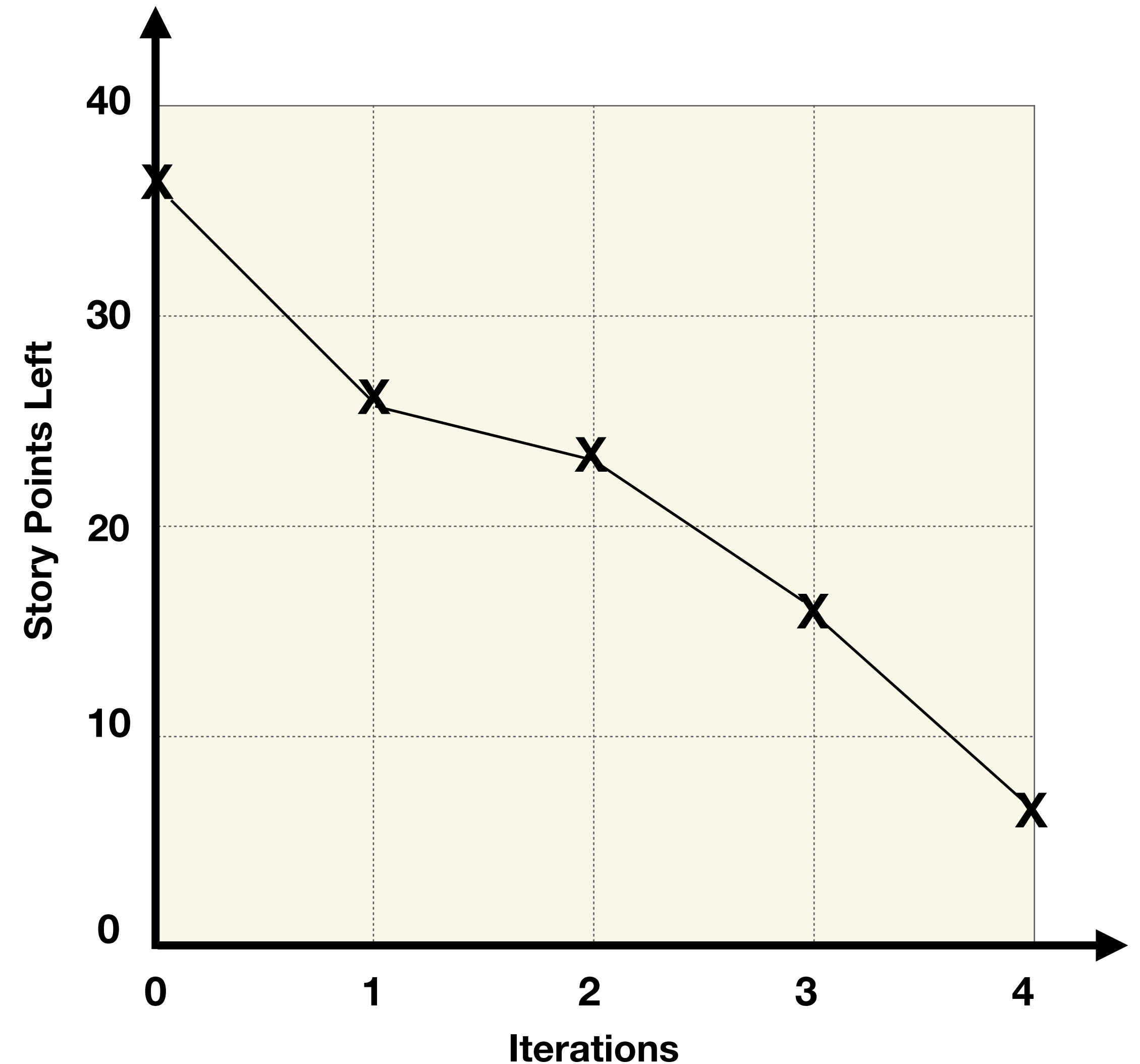
Story Completed Graph (Example)

- Actual points and number of iterations for the predicted and actual charts can be different.
- Additional stories and story points can be added during the project.
- Additional stories don't appear on the chart.



Burndown Chart

- It measures the work still to be done rather than the work already done.
- We start with an initial estimate of total story points (SPs) and make adjustments by:
 - subtracting the completed SPs.
 - adjusting for the changed estimates of existing user stories that have not started yet.
 - adding SPs for the newly created user stories.



Iteration Hit Rate

- Iteration Hit Rate is the fraction of story points completed in each iteration.
- It is expressed as percentage.

$$\text{Iteration Hit Rate} = (\# \text{ of SPs completed}) / (\# \text{ of planned SPs}) \times 100$$

Release Planning

- Not every iteration results in a new release.
- Each release will consist of a number of iterations.

Iteration	User Stories	Total Effort (SPs)	
1	C, E	10	
2	A, B, D, H	9	→ Release #1
3	F, G, K	10	
4	J, L, M	7	→ Release #2

Release Planning

- We might fix the release date in advance.
 - This will give us a limit on the number of story points that can be achieved.
 - We can then decide what goes into a release, based on this constraint.
- We might want a given functionality for the release.
 - Decide which user stories are needed to achieve this functionality.
 - Add up the story points and decide how many iterations are needed.

Iteration	User Stories	Total Effort (SPs)
1	C, E	10
2	A, B, D, H	9
3	F, G, K	10
4	J, L, M	7

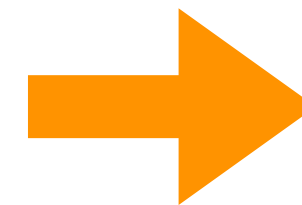
➡ Release #1

➡ Release #2

The Release Plan

- The high priority user stories will be in an early release.

User Story Name	Priority	Effort (SPs)
A	Must have	4
B	Must have	2
C	Must have	2
D	Must have	1
E	Must have	8
F	Should have	4
G	Should have	4
H	Should have	2
J	Could have	1
K	Could have	2
L	Would like to have	4
M	Would like to have	2



Iteration	User Stories	Total Effort (SPs)
1	C, E	10
2	A, B, D, H	9
3	F, G, K	10
4	J, L, M	7

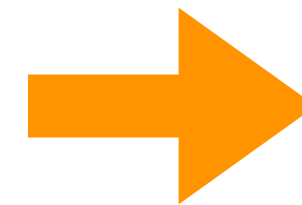
Release #1

The Release Plan

- Make sure that each iteration has *roughly* the same number of story points.
- SPs on the user story cards may not add up to the iteration velocity.

User Story Name	Priority	Effort (SPs)
A	Must have	4
B	Must have	2
C	Must have	2
D	Must have	1
E	Must have	8
F	Should have	4
G	Should have	4
H	Should have	2
J	Could have	1
K	Could have	2
L	Would like to have	4
M	Would like to have	2

- Velocity = 10
- Story Points (SPs): 1, 2, 2, 4
- Total = 9 SPs
- None of the next high priority user stories have a length of 1 SP



Iteration	User Stories	Total Effort (SPs)
1	C, E	10
2	A, B, D, H	9
3	F, G, K	10
4	J, L, M	7

Example

- Let's have a look at this small project with 12 user stories in more detail.

User Story Name	Priority	Effort (SPs)
A	Must have	4
B	Must have	2
C	Must have	2
D	Must have	1
E	Must have	8
F	Should have	4
G	Should have	4
H	Should have	2
J	Could have	1
K	Could have	2
L	Would like to have	4
M	Would like to have	2

Example

- We estimate: velocity = 10 SPs
 - 2 developers, each with 2.5 ideal days (SPs) per week
 - Iteration length = 2 weeks

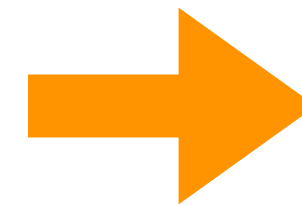
Velocity = 2 developers x 2.5 SPs/week x 2 weeks = 10 SPs

User Story Name	Priority	Effort (SPs)
A	Must have	4
B	Must have	2
C	Must have	2
D	Must have	1
E	Must have	8
F	Should have	4
G	Should have	4
H	Should have	2
J	Could have	1
K	Could have	2
L	Would like to have	4
M	Would like to have	2

Example

- We will schedule 4 iterations
- There will be an initial release after the second iteration and a final release at the end.

User Story Name	Priority	Effort (SPs)
A	Must have	4
B	Must have	2
C	Must have	2
D	Must have	1
E	Must have	8
F	Should have	4
G	Should have	4
H	Should have	2
J	Could have	1
K	Could have	2
L	Would like to have	4
M	Would like to have	2



The Iteration Schedule

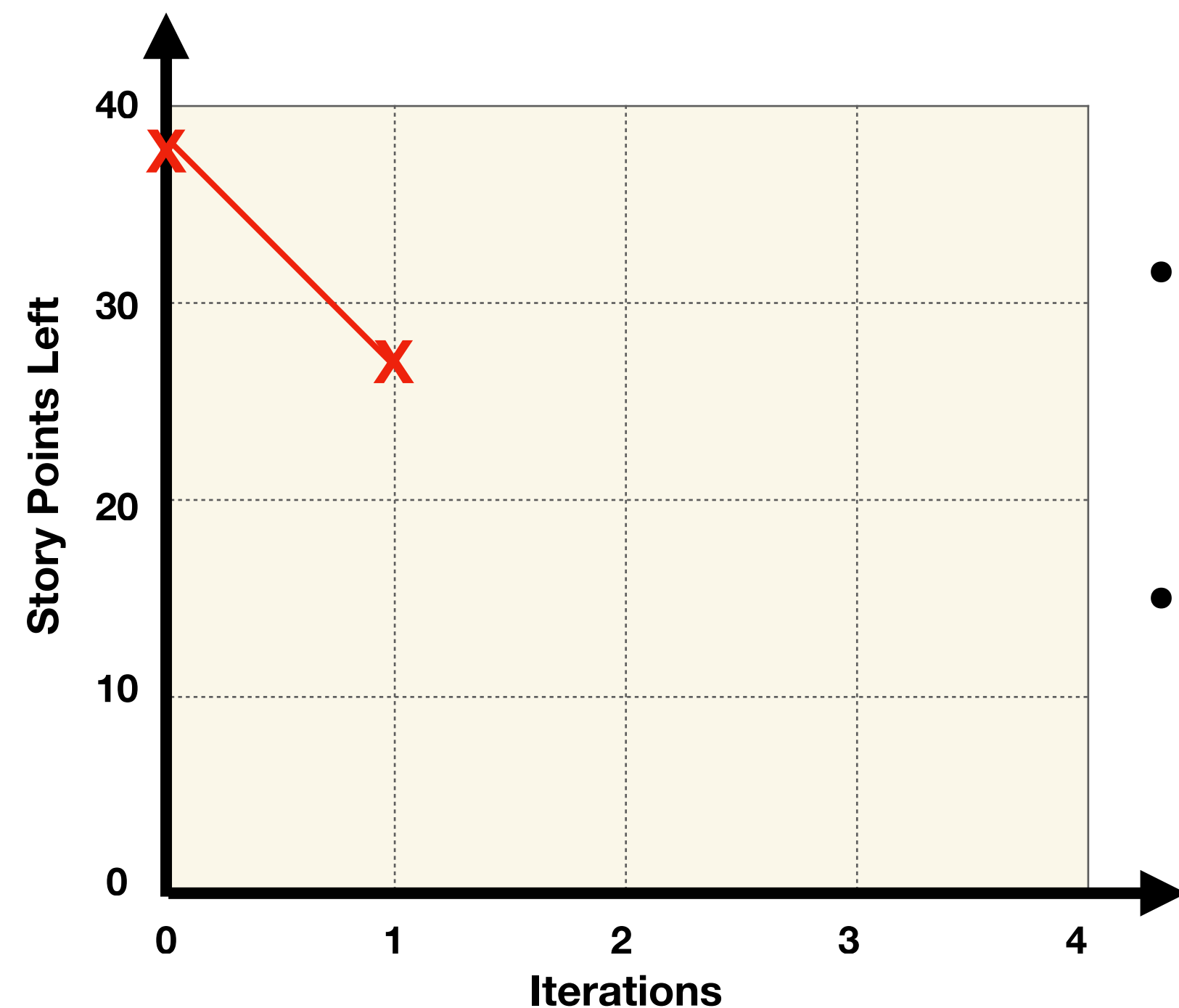
Iteration	User Stories	Total Effort (SPs)
1	C, E	10
2	A, B, D, H	9
3	F, G, K	10
4	J, L, M	7

Example - Iteration 1

- We find the tasks for user stories C and E are easier than expected and finish in 6 ideal days.

Iteration	User Stories	Total Effort (SPs)
1	C, E	10

- We use the initial estimate although we have only expanded 6 ideal days of actual work.



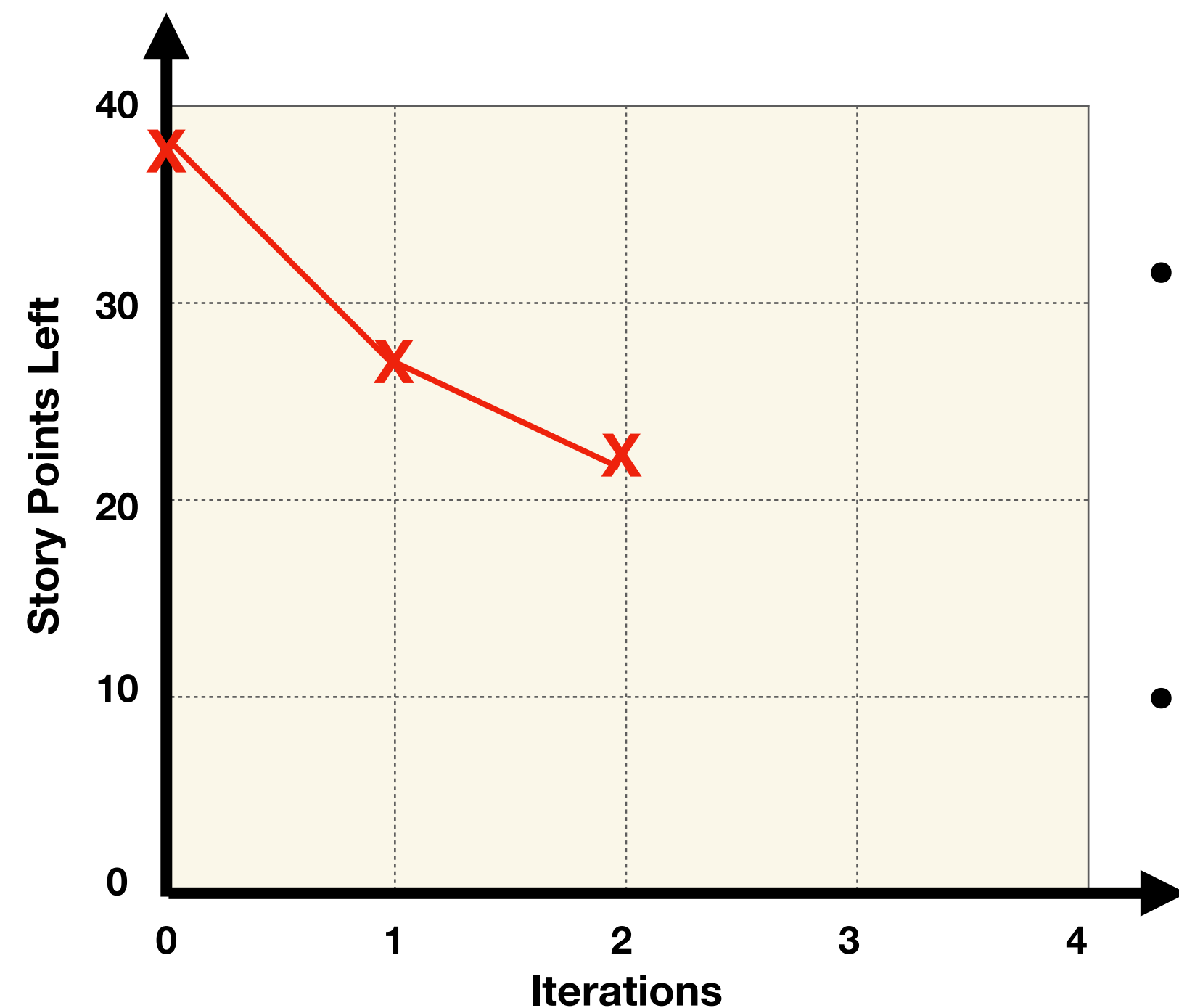
- Burndown chart goes from **36 SPs** to **26 SPs** since we completed 10 SPs (i.e., $36 \text{ SPs} - 10 \text{ SPs} = 26 \text{ SPs}$).
- Iteration Hit Rate = $10 / 10 \times 100 = 100 \%$

Example - Iteration 2

- B (2 SPs) turns out to be a lot harder than we thought.
- We discover that B is two user stories: B1 (2 SPs), B2 (4 SPs)

Iteration	User Stories	Total Effort (SPs)
2	A, B , D, H	9

- We complete 7 SPs and could not complete B (2 SPs) [We always use the initial estimate.]

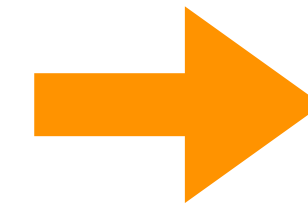


- Burndown chart goes from **26 SPs** to **23 SPs** since we completed 7 SPs and added 4 SPs (i.e., $26 \text{ SPs} - 7 \text{ SPs} + 4 \text{ SPs} = 23 \text{ SPs}$).
- Iteration Hit Rate = $7 / 9 \times 100 = 78 \%$

Example - Iteration 2

- The release is postponed to iteration 3 since we could not finish B.
- Iteration 3 will now do B1, B2 and F (G and K postponed to iteration 4)
- Iteration 4 will now do G, K, J and M, but not L.

Iteration	User Stories	Total Effort (SPs)
3	F, G, K	10
4	J, L, M	7

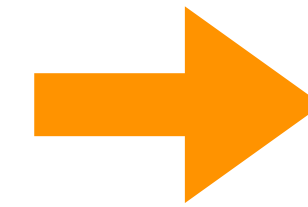


Iteration	User Stories	Total Effort (SPs)
3	B1, B2, F	10
4	G, K, J, M	9

Example - Iteration 3

- B still continues to cause problems: B1 takes 3 SPs (ideal days) and B2 takes 5 SPs.
- F is pushed back to iterations 4.
- Iteration 4 no longer plans to do J and M.

Iteration	User Stories	Total Effort (SPs)
3	B1, B2, F	10
4	G, K, J, M	9



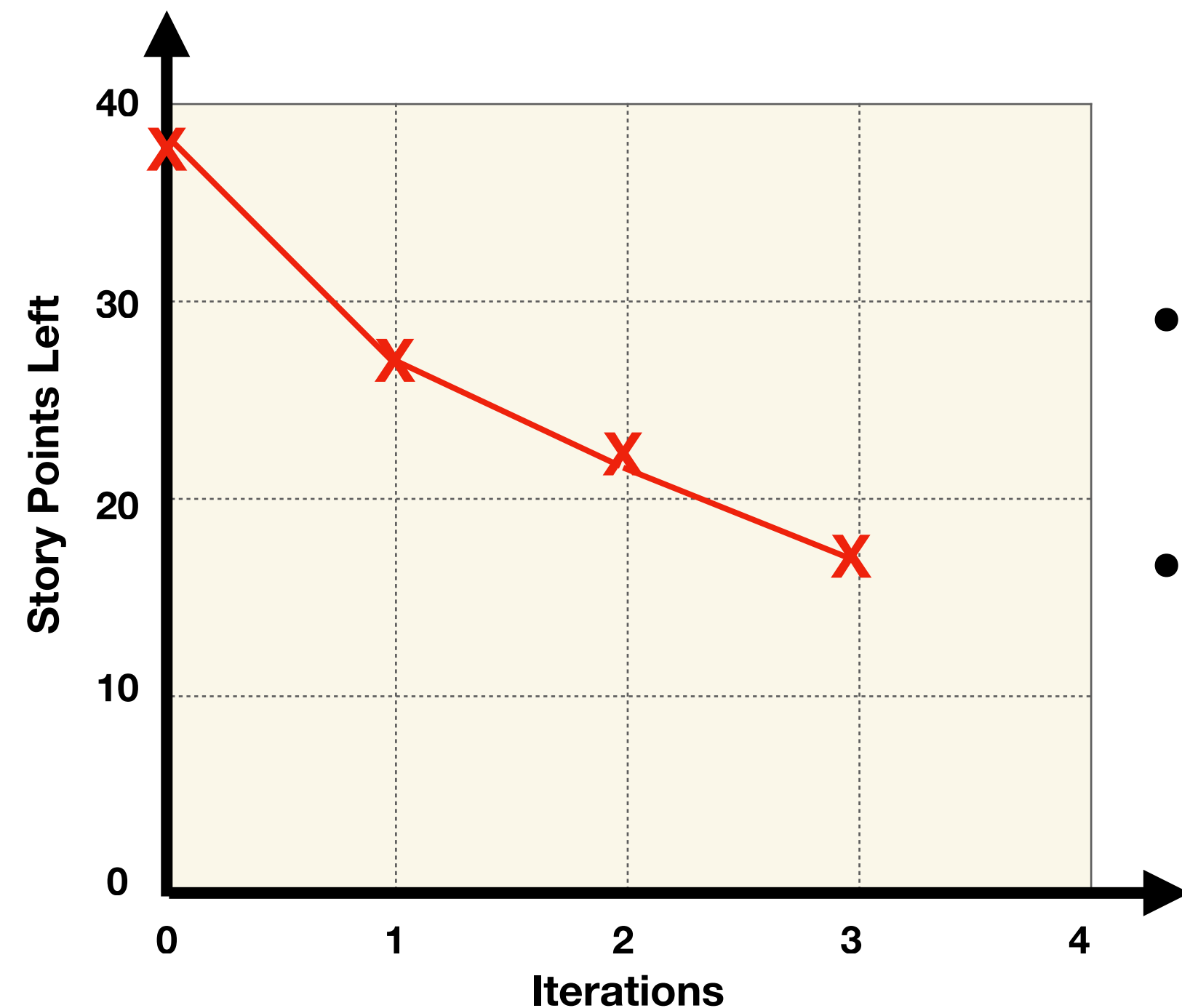
Iteration	User Stories	Total Effort (SPs)
4	F , G, K	10

Example - Iteration 3

- B still continues to cause problems: B1 takes 3 SPs (ideal days) and B2 takes 5 SPs.
- F is pushed back to iterations 4.

Iteration	User Stories	Total Effort (SPs)
3	B1, B2, F	10
4	G, K, J, M	9

- We complete 6 SPs and could not complete F (4 SPs) [We always use the estimate at the end of the previous iteration.]



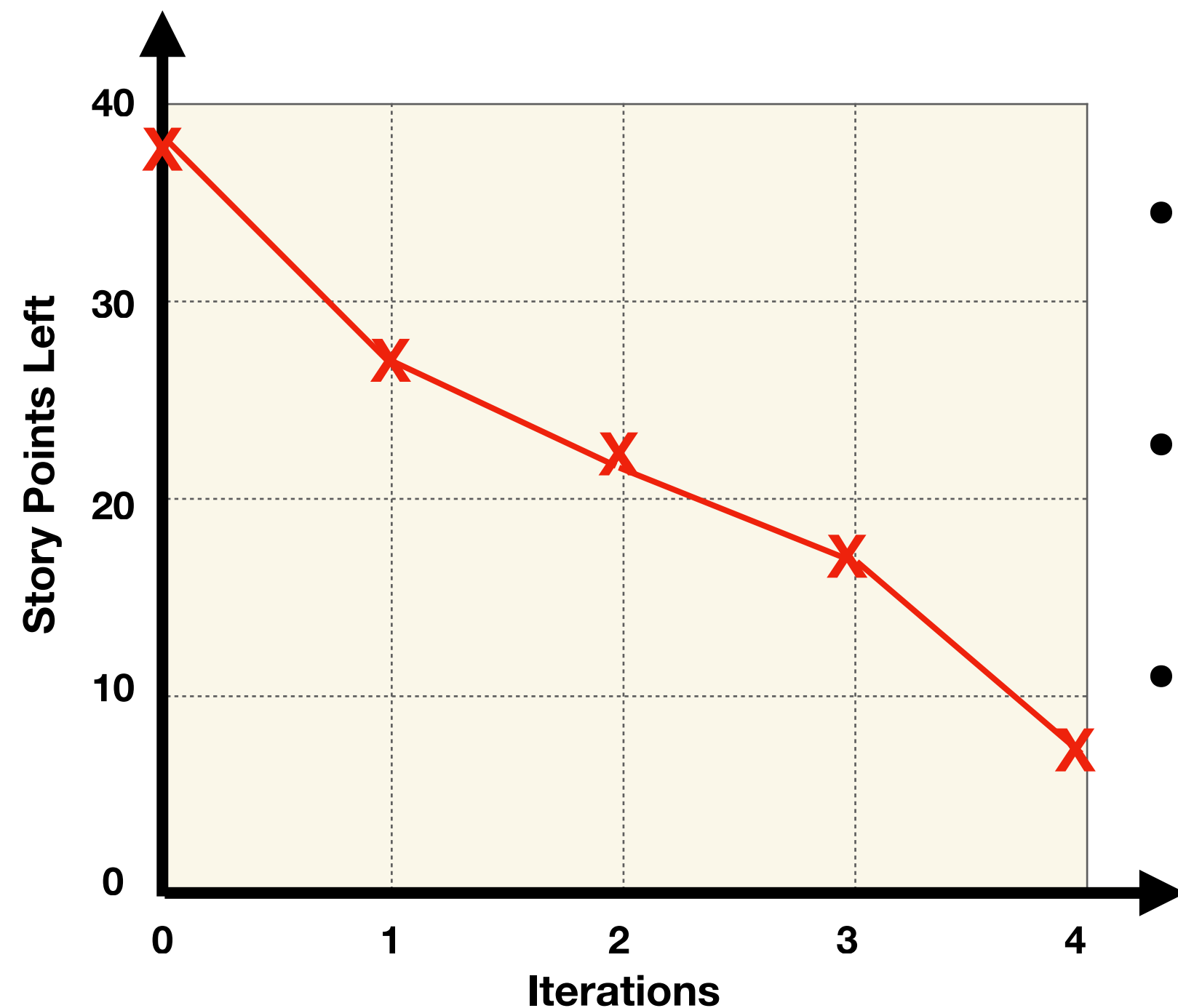
- Burndown chart goes from **23 SPs** to **17 SPs** since we completed 6 SPs (i.e., 23 SPs - 6 SPs = 17 SPs).
- Iteration Hit Rate = $6 / 10 \times 100 = 60 \%$

Example - Iteration 4

- We finish all user stories in iteration 4.

Iteration	User Stories	Total Effort (SPs)
4	F, G, K	10

- Iteration 4 finishes normally, so 10 SPs are completed.



- Burndown chart goes from **17 SPs** to **7 SPs** since we completed 10 SPs (i.e., $17 \text{ SPs} - 10 \text{ SPs} = 7 \text{ SPs}$).
- We could not finish J, L and M, hence $1 \text{ SP} + 4 \text{ SPs} + 2 \text{ SPs} = 7 \text{ SPs}$ are left.
- However, J, L, M were not planned for iteration 4 any more. Hence, Iteration Hit Rate = $10 / 10 \times 100 = 100 \%$

Interactive Exercise

- Use your computer or smartphone to access the exercise
- If you use your **computer** or **smartphone**: Open a web browser, enter www.menti.com and enter code **2184 2013**
- With your smartphone you can also use the following QR code



Questions

(Please work on the following questions in your free time. Answers will be posted on Moodle next week)

1. What does the Product Owner do? How much are they involved in the day to day development?
2. What does the Scrum Master do? How do they keep track of what developers do?
3. How are the stories for each iteration chosen?
4. How is work allocated in Scrum?
5. How is progress tracked?
6. How are unexpected difficulties dealt with?

Case Study

- Go to Moodle website of this course and download the Case Study
- To work on this Case Study, you will be assigned to Breakout Rooms (on Zoom)
- You will be working on this Case Study in groups in the designated Breakout Rooms.
- You should be able to share screens in the breakout rooms

The screenshot shows a Moodle course page for 'COMPSCI5059 Software Engineering M - 2022-23'. The breadcrumb trail is: Dashboard > My courses > College of Science and Engineering > Schools > Computing Science > Welcome to > MSc(IT+) and MSc(CS+) > Compulsory courses (IT+ only) > COMPSCI5059 Software Engineering M - 2022-23 > Semester 1: Lecture Materials and Supporting Documents > Week 2.

Under the 'Week 2' heading, there is a folder icon containing two PDF files: 'Case Study.pdf' and 'Scrum_Supplementary-Material.pdf'. The 'Case Study.pdf' file is highlighted with a red box and a red arrow pointing to it, with the text 'Download the Case Study' next to it. The 'Scrum_Supplementary-Material.pdf' file is highlighted with a blue box and a blue arrow pointing to it, with the text 'You can use the supplementary material to refer to today's lecture' next to it. Below the files are two buttons: 'Download folder' and 'Edit'.