



Scrum: Sprint Execution

Lecturer: Adel Vahdati



Sprint Execution

■ When?

- Sprint execution accounts for the majority of time during a sprint. It begins after sprint planning and ends when the sprint review starts.
- On a two-week-long sprint, execution might account for eight of the ten days.

■ By whom?

- The full Scrum team:
- Development team members self-organize and determine the best way to meet the goal established during sprint planning.
- The Scrum Master acts as the coach, facilitator, and impediment remover.
- The product owner is available during sprint execution to
 - answer clarifying questions,
 - review intermediate work and provide feedback to the team,
 - discuss adjustments to the sprint goal if conditions warrant, and
 - verify that the acceptance criteria of PBIs have been met.



Sprint Execution: Process

- **Inputs:** Sprint goal and Sprint backlog, which collectively form the commitment.
- **Activities:** Planning, Flow management, Performing tasks, and Communicating.
- **Outputs:** Potentially shippable product increment, which is a set of PBIs completed to a high degree of confidence according to the definition of done.



Planning



- Some up-front planning is typically done for exposing important task-level dependencies.
 - Preparing a Gantt chart, however, is typically not worth the effort.
- A good principle for sprint execution is to approach task-level planning in an opportunistic, flexible and ongoing manner.
 - Allow task planning to occur continuously during sprint execution as the team adapts to the evolving circumstances of the sprint.



Flow Management



- It's the team's responsibility to manage the flow of work during sprint execution to meet the sprint goal.
- The team must make decisions on:
 - How much work the team should do in parallel.
 - When work should begin on a specific item.
 - How the task-level work should be organized.
 - What work needs to be done.
 - Who should do the work.
- When answering these questions, teams should discard old behaviors, such as
 - trying to keep everyone 100% busy,
 - believing that work must be done sequentially, and
 - having each person focus on just their part of the solution.



Flow Management: Parallel Work

- An important part of managing flow is determining how many PBIs the team should work on in parallel to maximize delivered value.
- Working on too many items at once leads to multitasking, which increases the time required to complete individual items, and reduces quality.
- Working on too few items at a time leads to underutilization of member skills and capacity, resulting in less work done and less value delivered.
- To find the proper balance, teams work on the number of items that leverages, but does not overburden, their skills and available capacity.

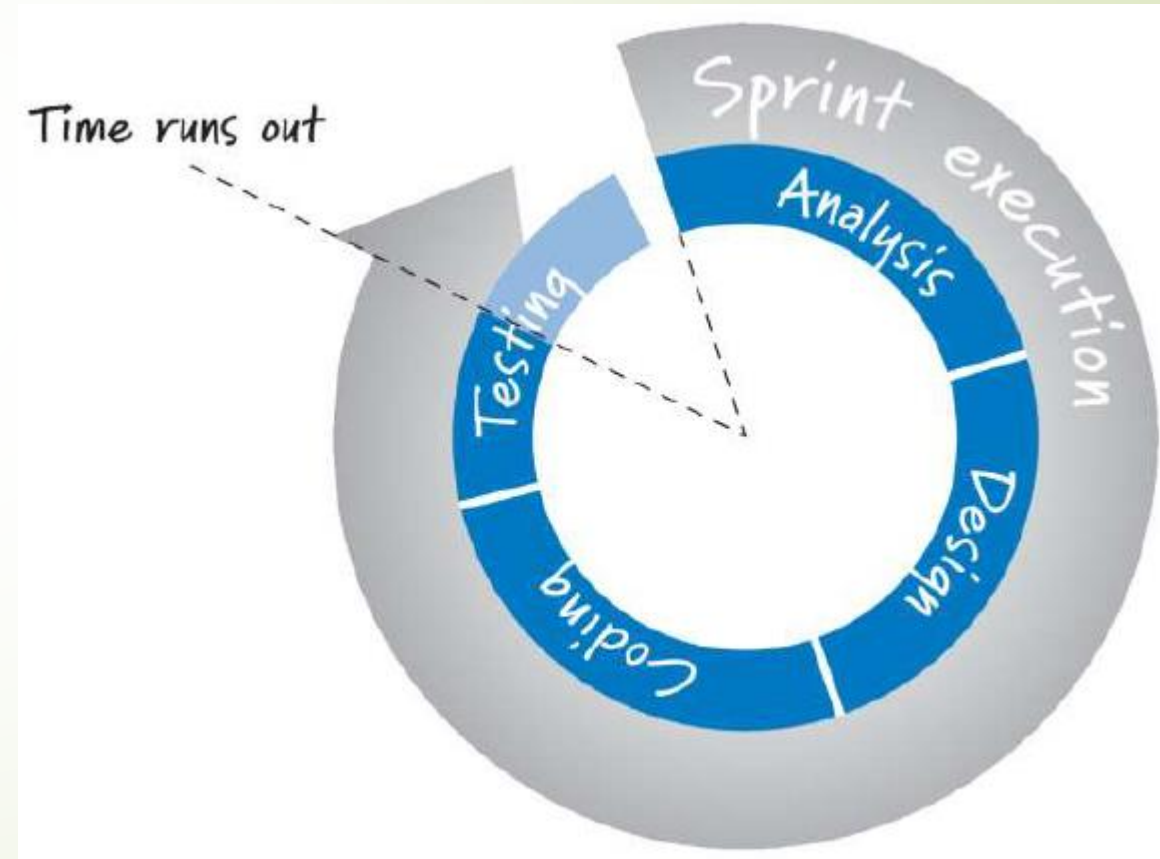


Flow Management: Swarming

- ▶ Team members with available capacity gather to work on an item to finish what has already been started before working on new items.
- ▶ Teams with a **Musketeer attitude** and some degree of **T-shaped skills** swarm.
 - ▶ **Musketeer attitude:** “All for one and one for all.” Team members collectively own the responsibility of getting the job done.
 - ▶ **T-shaped skills:** Having deep skills in a preferred functional area, discipline, or specialty, but also able to work outside the specialty area.
- ▶ Swarming is not a strategy to ensure that team members are 100% busy.
- ▶ Swarming does not necessarily mean working on only one PBI at a time.
- ▶ Sprint execution should not be treated like a **mini-waterfall** project.

Risks of Mini-Waterfall Approach to Sprint Execution

- In this approach, we work on all PBIs at the same time
- We first analyze all the items, then design them all, then code them all, and then test them all.
- This approach is very risky
- If the team does not finish all the testing, we could end up with 90% of each feature complete, but no feature 100% done.





Flow Management: Important Concerns

- **Which PBI to Start:** The simplest way is to select the next highest-priority item as specified by the product owner.
 - However, technical dependencies or skills capacity constraints might dictate that items be selected in a different order.
- **How to Organize Task Work in a PBI:** Value-delivery-focused method.
 - Team members opportunistically organize the tasks and who will work on them, and work is highly interleaved. Swarming is encouraged.
- **What Task-Level Work Should Be Done:** Ultimately, the team decides; product-owners/managers empower the team, but can affect their work by:
 - Defining the scope of a feature and its acceptance criteria.
 - Providing business-facing requirements for the definition of done.
 - Working with the team to ensure that their technical or feature-specific decisions are made in an economically sensible way.

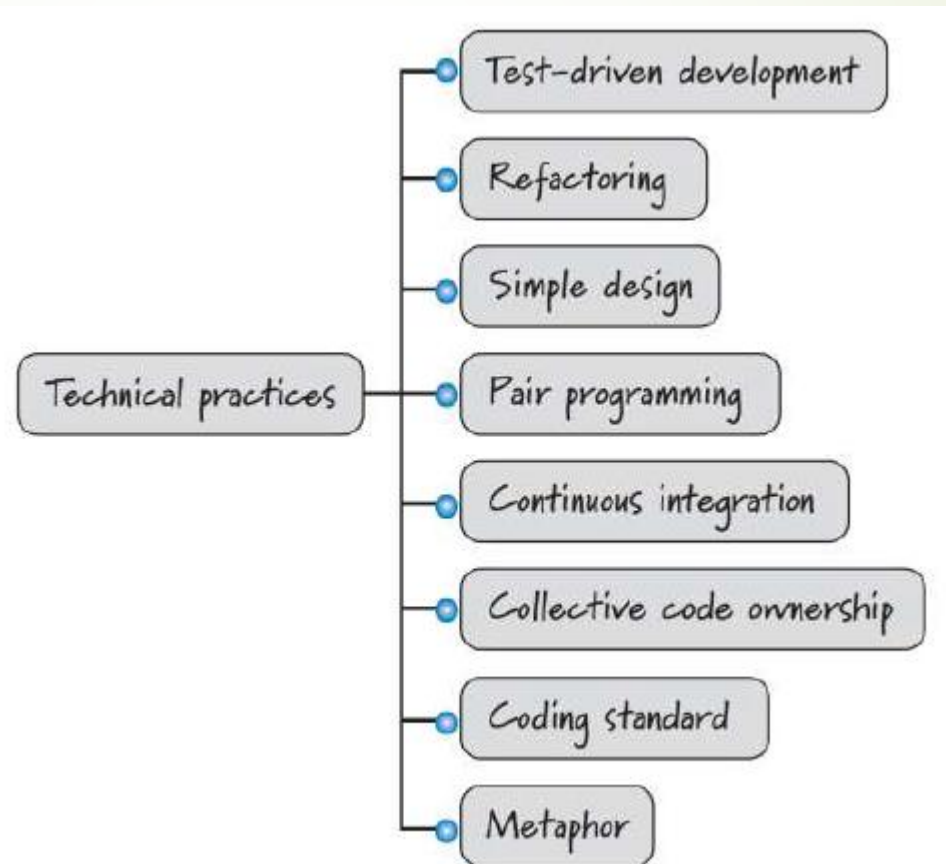


Flow Management: Daily Scrum

- The daily scrum is a critical, inspect-and-adapt activity.
- A 15-minute, time-boxed activity that takes place once every 24 hours.
- It serves as an inspection, synchronization, and daily adaptive planning activity that helps a self-organizing team do its job better.
- Scrum team convenes to share the big picture of what is happening so that they can collectively understand
 - how much to work on,
 - which items to start working on, and
 - how to best organize the work among the team members.
- The daily scrum helps avoid waiting: If there is an issue that is blocking
- flow, the team would never have to wait more than a day to discuss it.

Performing Tasks: Technical Practices

- Team members should be skilled in agile technical practices (such as automated testing); most of these are attributed to XP.

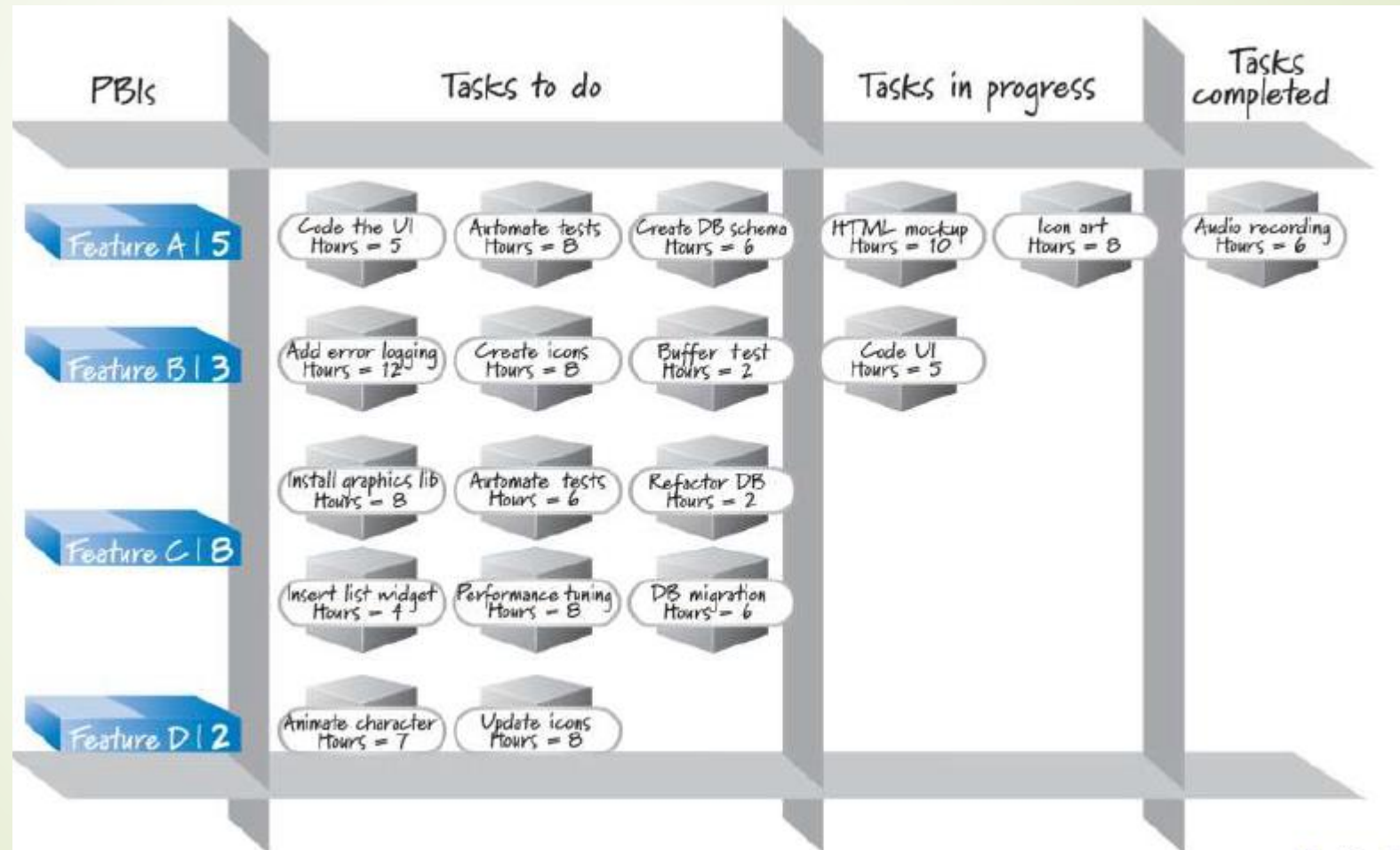




Communicating: Task Board

- In Scrum, communicating progress is done by using simple charts as their principal **Information Radiators**:
- Although any highly visible way of communicating progress can be used, most teams use a task board along with a burndown chart and/or burnup chart.
- **Task Board**: Shows the evolving state of the sprint backlog over time.
- Each product backlog item planned to be worked on during the sprint is shown with the set of tasks necessary to get the item done.
 - All tasks initially start off in the **“to do”** column.
 - As the team starts to work on the tasks of a PBI, these tasks are moved from the “to do” column to the **“in progress”** column.
 - When a task is completed, it is moved to the **“completed”** column.
- A team may choose to put other columns on its task board if it thinks that visualizing the flow of work through other states is helpful.

Communicating: Task Board





Communicating: Progress Charts

- Each day during sprint execution, team members **update the estimate** of how much effort remains (in hours) for each task.
- A table can be used to visualize this data.
- The number of hours remaining for each task follows the general trend of being smaller each day during the sprint.
- If a task has not yet been started yet, the size of the task might appear the same from day to day until the task is started.
 - If a task turns out to be larger than expected, its size may increase day over day, or remain the same even after the team has started working on it.
- New tasks related to the committed PBIs can also be added to the sprint backlog at any time, and will be reflected in the corresponding table.

Communicating: Task Progress Table

Tasks	D1	D2	D3	D4	D5	D6	D7	D8	D9	...	D15
Task 1	8	4	4	2							
Task 2	12	8	16	14	9	6	2				
Task 3	5	5	3	3	1						
Task 4	7	7	7	5	10	6	3	1			
Task 5	3	3	3	3	3	3	3				
Task 6	14	14	14	14	14	14	14	8	4		
Task 7						8	6	4	2		
Tasks 8–30	151	139	143	134	118	99	89	101	84		0
Total	200	180	190	175	155	130	115	113	90		0



Communicating: Sprint Burndown Chart

- **Sprint Burndown Chart:** The result of plotting the “Total” row, which is the sum of the remaining effort-hours across all tasks on a given day, on a graph.
 - Vertical axis numbers are the estimated effort-hours remaining, and horizontal axis numbers are days within a sprint.
 - Each day we update this chart to show the total estimated effort remaining across all of the uncompleted tasks.
- Sprint burndown charts are useful for tracking progress and can also be used as a leading indicator to predict when work will be completed.



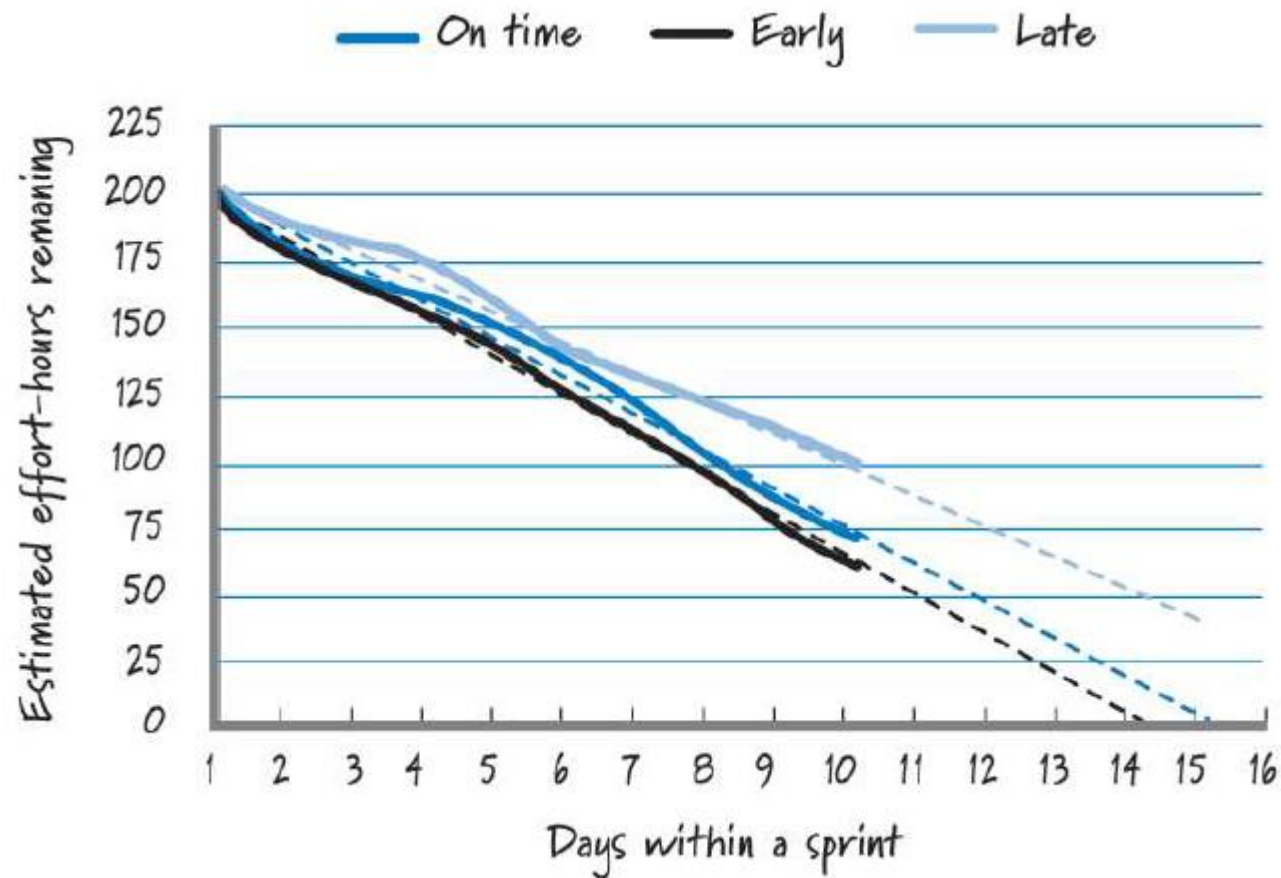
Communicating: Sprint Burndown Chart

- At any point in time, we can compute a trend line based on historical data and use it to see when we are likely to finish if the current pace and scope remain constant.
 - When the trend line intersects the horizontal axis close to the end of the sprint duration, we can infer that we're in reasonable shape (**"On time"**).
 - When it lands significantly to the left, we should probably take a look to see if we can safely take on additional work (**"Early"**).
 - When it lands significantly to the right (**"Late"**), it warns us that we're not proceeding at the expected pace or that we've taken on too much work.

Communicating: Sprint Burndown Chart



Communicating: Sprint Burndown Chart (with Trend Lines)

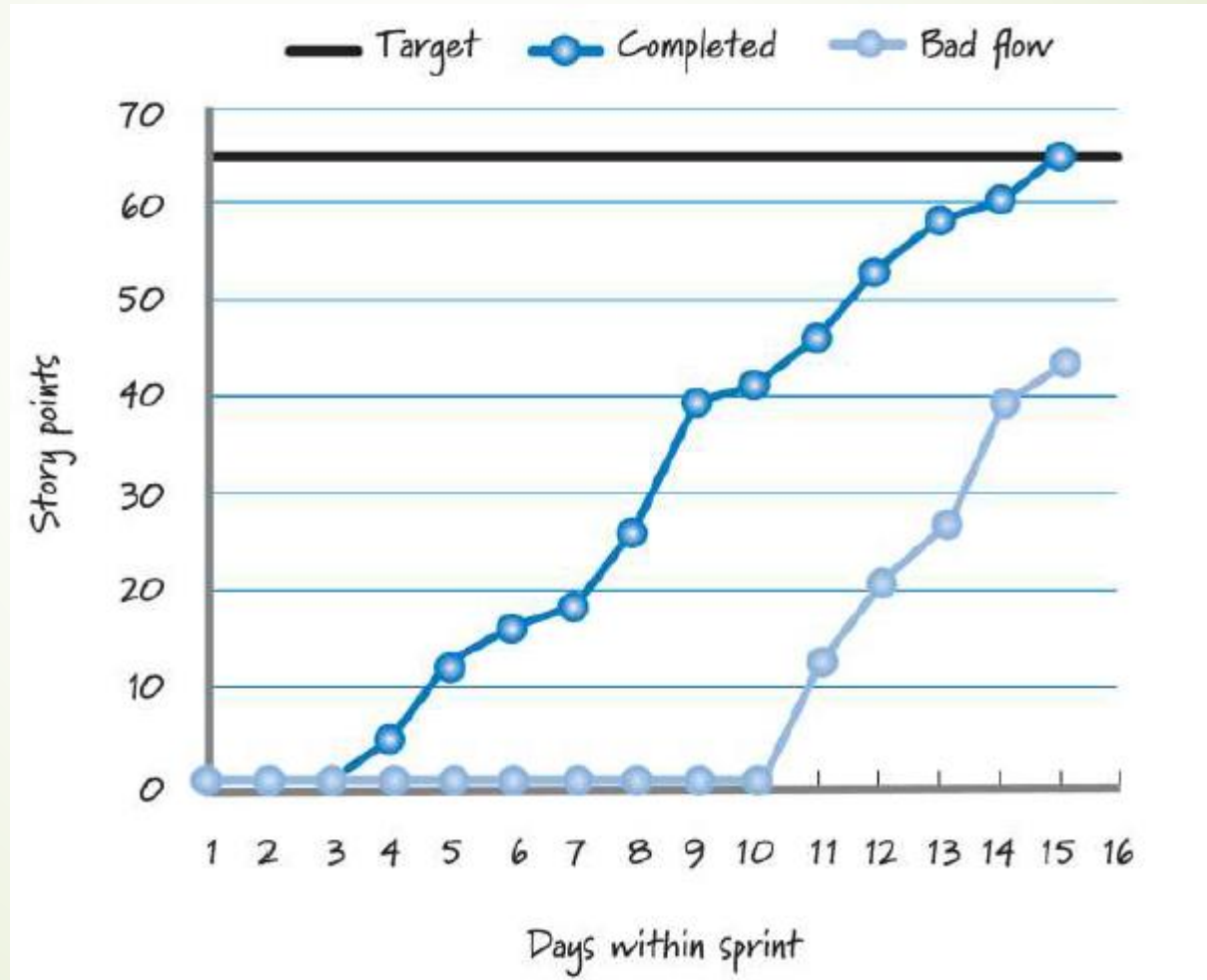




Communicating: Sprint Burnup Chart

- **Sprint Burnup Chart:** Represents the amount of work completed toward achieving the sprint goal.
- In sprint burnup charts, work can be represented in either effort-hours (as in the burndown chart) or in story points; story points are preferred because:
 - 1. At the end of the sprint, the only thing that really matters to the Scrum team is business-valuable work that was completed.
 - 2. At a glance, we can get a good feel for how the work is flowing and how the team is completing PBIs through the sprint.

Communicating: Sprint Burnup Chart





References



- Rubin, K.S., Essential Scrum: A Practical Guide to the Most Popular Agile Process, Addison-Wesley, 2012.
- Schwaber, K., Sutherland, J., The Scrum Guide, Published online at: <http://www.scrumguides.org/>, July 2013 (last visited on: 7 November 2014).
- Ramsin, Raman. "Home." Department of Computer Science and Engineering, Sharif University of Technology. Accessed February 15, 2025. <https://sharif.edu/~ramsin/index.htm>.