Sprint Execution(I)

Dr. Elham Mahmoudzadeh

Isfahan University of Technology

mahmoudzadeh@iut.ac.ir

2022

Introduction

- Sprint execution is the work the Scrum team performs to meet the sprint goal.
- In sprint execution, all of the work necessary to deliver a potentially shippable product increment is performed.

کی این کار انجام میشه؟ زمانی که ما یک سایکل اسپرینت خواهد داشت مختص Sprint execution است
طول دیریشن توسط تیم مشخص میشه و دیریشن فیکس باشه در طول پروژه

Timing

- 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, sprint execution might account for about eight out of the ten days.

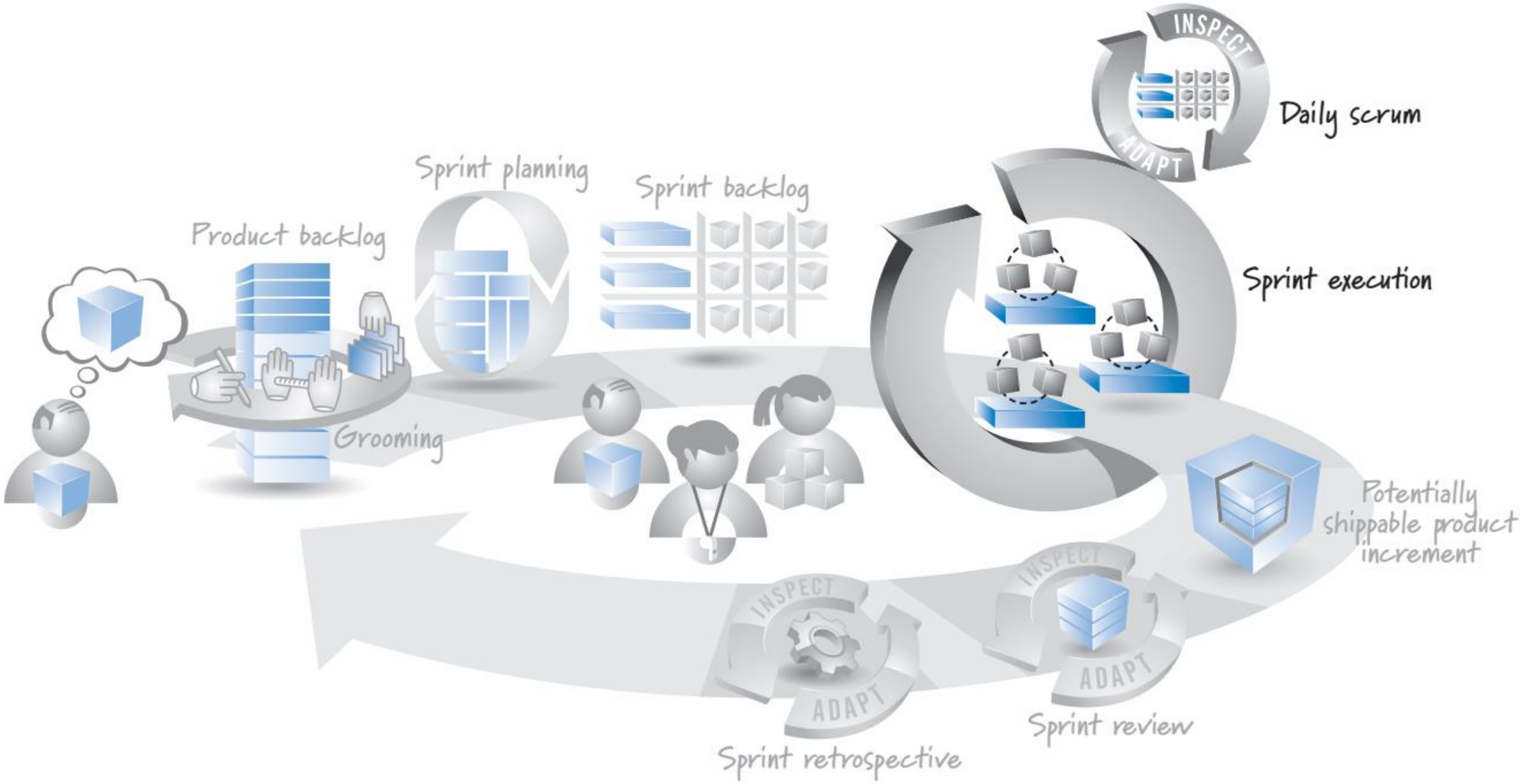


FIGURE 20.1 When sprint execution happens

Participants

- During sprint execution the development team members self-organize and determine the best way to meet the goal established during sprint planning.
- The ScrumMaster participates as the coach, facilitator, and impediment remover, doing whatever is possible to help the team be successful. The ScrumMaster doesn't assign work to the team or tell the team how to do the work. A self-organizing team figures these things out for itself.
- The product owner must be available during sprint execution to answer clarifying questions, to review intermediate work and provide feedback to the team, to discuss adjustments to the sprint goal if conditions warrant, and to verify that the acceptance criteria of product backlog items have been met.

چه افرادی توی Sprint execution هستن:
ادم های تیم کننده کار هستن
پروداکت اونر باید در دسترس باشه

Process

- The **inputs** to sprint execution are the sprint goal and the sprint backlog that were generated during sprint planning.
- The **output** from sprint execution is a potentially shippable product increment, which is a set of product backlog items completed to a high degree of confidence—where each item meets the Scrum team's agreed upon definition of done.
- **Activities** involves planning, managing, performing, and communicating the work necessary to create these working, tested features.

ورودی:
اسپرینت گل و بک لاگ اسپرینت است که توی اسپرینت پلنینگ اوکی شده و ست شده
و خروجی این potentially shippable product است که یک مجموعه ای از ایتم های
پروداکت بک لاگ است که توی اسپرینت پلینینگ توافق شده که اون ها روی این اسپرینت انجام
بشه

Process

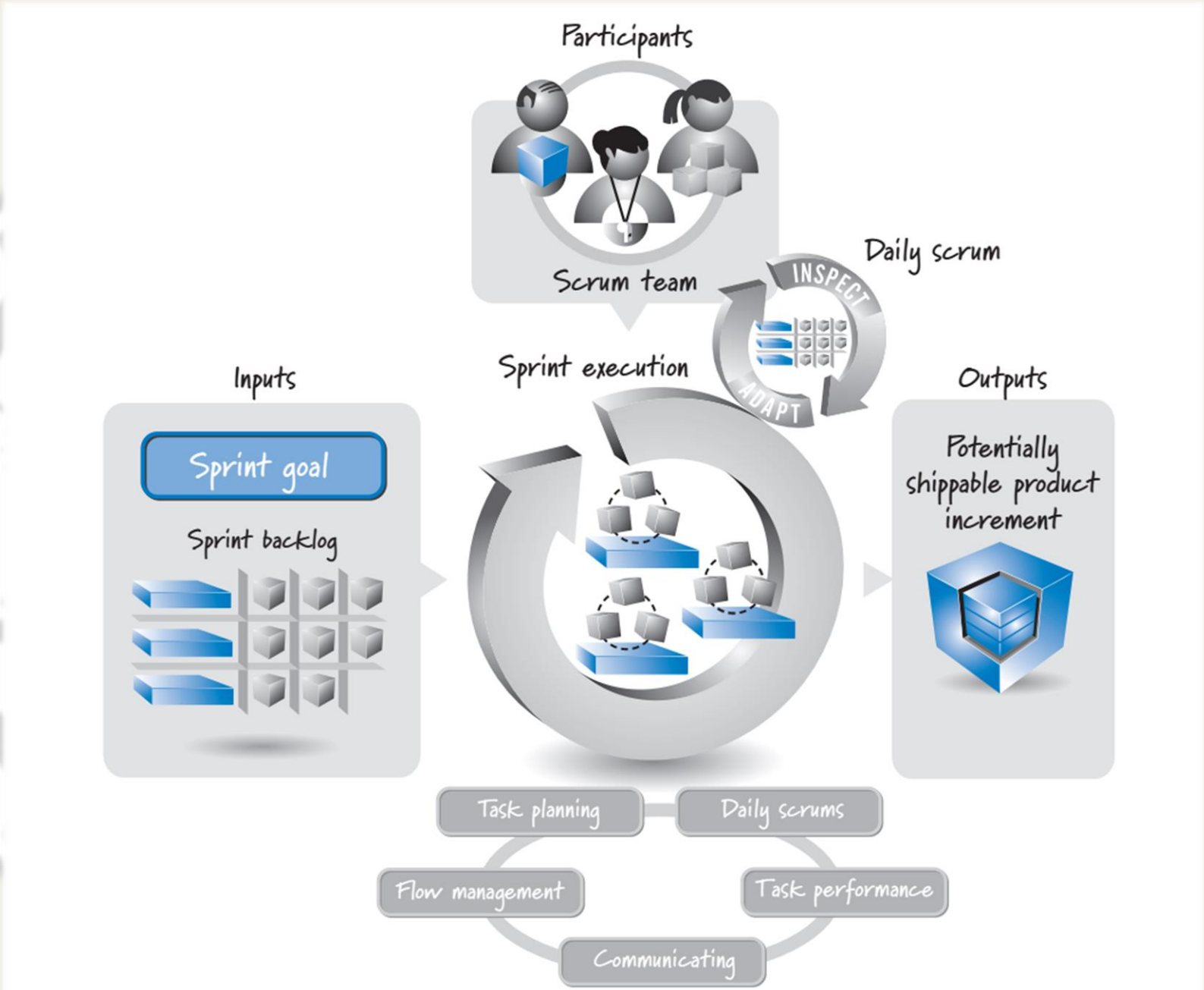


FIGURE 20.2 Sprint execution activity

اسپرینت گل و بک لاگ اسپرینت وارد exec.. میشه که بزرگترین سایکل اسپرینت است و افراد تیم همه در اون exec.. فعال هستن
خروجی E.. هست potentially shippable product ینی یک محصولی که پتانسیل ورودی به بازار رو داشته باشه

Sprint Execution Planning

- During sprint planning the team produces a *plan* for how to achieve the sprint goal. Most teams create a sprint backlog, which typically lists product backlog items and their associated tasks and estimated effort-hours.
- A massive influx of learning comes from actually building and testing something. This learning will disrupt even the best-conceived early plan.
- As a result, the team wastes valuable time putting a plan together, only to waste even more time changing it to reflect the reality of sprint execution.
- Some up-front planning is helpful for exposing important task-level dependencies

پلن e.. می شیم توی اسپرینت پلنینگ یک پلنی رو چیدیم که چجوری به گل اسپرینت برسیم و اکثرا در قالب اسپرینت بک لاگ این کار رو کردیم

خیلی توی اسکرام پلن مفصلی تعریف نمیشه چون اعتقاد بر اینه وقتی که میذاریم برای پلن اتلاف وقت میشه

افراد تیم خودشون بررسی می کنن به جای اینکه بیان

Sprint Execution Planning(Cnt'd)

- A good principle for sprint execution is to approach task-level planning opportunistically rather than trying to lay out up front a complete plan of how to do the work.
- Allow task planning to occur continuously during sprint execution as the team adapts to the evolving circumstances of the sprint.

Process

- The **inputs** to sprint execution are the sprint goal and the sprint backlog that were generated during sprint planning.
- The **output** from sprint execution is a potentially shippable product increment, which is a set of product backlog items completed to a high degree of confidence—where each item meets the Scrum team's agreed upon definition of done.
- **Activities** involves planning, managing, performing, and communicating the work necessary to create these working, tested features.

Flow Management

- It’s the team’s responsibility to manage the flow of work during sprint execution to meet the sprint goal.
- It must make decisions such as 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, and 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 her part of the solution.

Parallel Work and Swarming

- An important part of managing flow is determining how many product backlog items the team should work on in parallel to maximize the value delivered by the end of the sprint.
- Working on too many items at once contributes to team member multitasking, which in turn increases the time required to complete individual items and likely reduces their quality.

Cost of multitasking

Letters	Numbers	Roman numerals
a	1	i
b	2	ii
c	3	iii
d	4	iv
e	5	v
f	6	vi
g	7	vii
h	8	viii
i	9	ix
j	10	x

Row-at-a-time (multitasking)
Average time = 35 seconds

Letters	Numbers	Roman numerals
a	1	i
b	2	ii
c	3	iii
d	4	iv
e	5	v
f	6	vi
g	7	vii
h	8	viii
i	9	ix
j	10	x

Column-at-a-time (single tasking)
Average time = 16 seconds

Managing

- Working on too many items at the same time is wasteful.
- Working on too few items at a time is also wasteful. It leads to underutilization of team member skills and capacity, resulting in less work being completed and less value being delivered.
- To find the proper balance, I recommend that teams work on the number of items that leverages, but does not overburden, the T-shaped skills and available capacity of the team members.
- The goal is to reduce the time required to complete individual items while maximizing the total value delivered during the sprint (typically the number of items completed during the sprint).
- A term frequently used to describe this approach is swarming, where team members with available capacity gather to work on an item to finish what has already been started before moving ahead to start work on new items.

Managing(Cnt'd)

- Teams with a Musketeer attitude and some degree of T-shaped skills swarm.
- Teams that still think in terms of individual roles wind up with some members far ahead and others who are mired in unfinished work. a classic individual-role-focused thought is "The testers might still have 'their' work to finish up, but I'm finished coding this feature, so I'm off to start coding the next one."
- In a team that swarms, people would understand that it is typically better to stay focused and help get the testing done instead of running ahead to start working on new features.

Managing(Cnt'd)

- Some people mistakenly believe that swarming is a strategy to ensure that team members are 100% busy.
- This is not the goal of swarming. If we wanted to ensure that people were 100% busy, we would just start working on all product backlog items at the same time!
- Why don't we do that? Because the extensive multitasking required to make that happen would ultimately slow the flow of completed items. Swarming, on the other hand, helps the team remain goal focused instead of task focused, which means it gets more things done, faster.

Managing(Cnt'd)

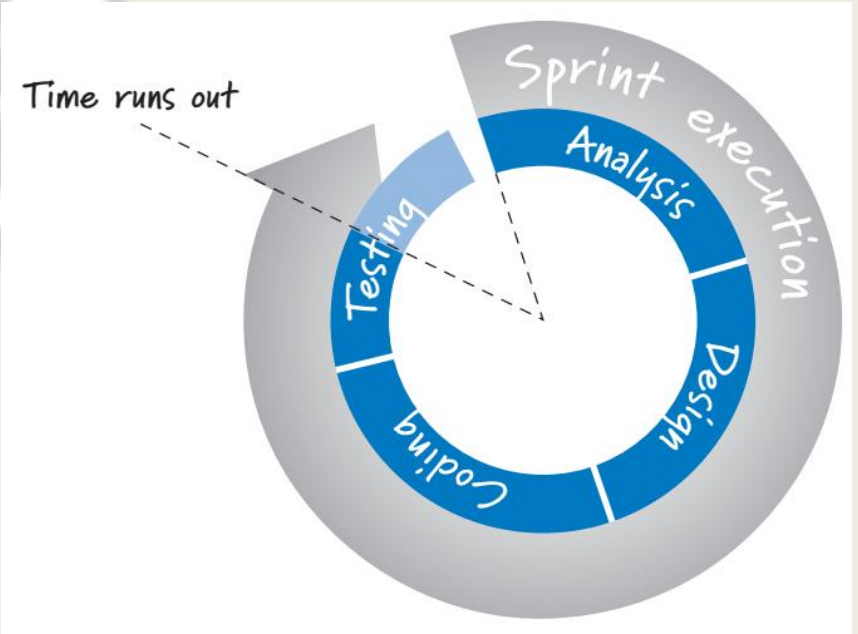
- While swarming favors working on fewer items concurrently, it doesn't necessarily mean working on only one product backlog item at a time.
- One item at a time might be correct in a given context, but just saying that all team members should collectively focus on a single item at a time is potentially dangerous.
- A different number of items might be appropriate when we consider the actual work that needs to be done, the skills of the team members, and other conditions that exist at the time a decision to start or not start working on another item needs to be made.

Managing(Cnt'd)

- Another dangerous approach would be to apply waterfall thinking at the sprint level and treat sprint execution like a mini waterfall project.
- First we would analyze all of the items to be worked

on this sprint, then design them all, then code them

all, and then, finally, test them all.



Managing(Cnt'd)

- Although this approach may seem logical, it is very risky. What if the team runs out of time and doesn't finish all of the testing?
- Do we have a potentially shippable product increment? No;
- a reasonable definition of done would never allow untested features to be called done.
- By using a mini waterfall strategy, we could end up with 90% of each feature complete, but no feature 100% done.
- The product owner gets no economic value from partially done work.

Reference

1- K. S. Rubin, “Essential Scrum, A Practical guide to the most popular agile process,” 2013.