



IT4305: Rapid Software Development

BIT – 2nd Year
Semester 4



Learning Outcome

After successful completion of this course students will be able to:

- Obtain a firm foundation on Agile concepts and methodologies.
- Acquire understanding of the practices and application of Agile practices Scrum and XP
- Learn how to apply the Agile framework in software Development Projects



Outline of Syllabus

1. Introduction to Agile Software Development
2. Agile Principles
3. Introduction to Scrum
4. Core Concepts in Scrum
5. Scrum Roles
6. Scrum Planning
7. Sprinting
8. Alternative Approaches to Agile Software Development



References

1. Essential Scrum Practical Guide to the Most Popular Agile Process by Kenneth S. Rubin.
2. The Art of Agile Development by James Shore and Shane Warden
3. Agile and Iterative Development: A Manager's Guide by Craig Larman, Agile Software development series, Alistair Cockburn and Jim Highsmith, Series Editors
4. <http://agilemanifesto.org>
5. <https://msdn.microsoft.com/en-us/library/hh533841.aspx>



IT4305: Rapid Software Development

Scrum Planning

Duration: 10 hours



Learning Outcome

- Understand the principals in Scrum Planning
- Describe how to apply scrum principles in planning.
- Understand principals behind Multi-level planning
- Discuss various Scrum planning activities and how they are interrelated with each other.
- Identify strategies for portfolio planning, grouped by scheduling, product inflow, and product outflow.
- Discuss how to determine whether or not more work should be invested in in-process products.
- Describe the envisioning approach used in Scrum principles.
- Discuss how release planning fits into the Scrum framework.
- Discuss how to perform release planning on both fixed-date and fixed-scope releases.



Detailed Syllabus

- 6.1 Scrum Planning Principles
- 6.2 Multilevel Planning
- 6.3 Portfolio Planning
- 6.4 Envisioning (Product Planning)
- 6.5 Release Planning (Longer-Term Planning)



6.1: SCRUM PLANNING PRINCIPLES



6.1: Scrum Planning Principles

- 6.1.1 Introduction
- 6.1.2 Don't Assume We Can Get the Plans Right Up Front
- 6.1.3 Up-Front Planning Should Be Helpful without Being Excessive
- 6.1.4 Keep Planning Options Open Until the Last Responsible Moment
- 6.1.5 Focus More on Adapting and Preplanning Than on Conforming to a Plan
- 6.1.6 Correctly Manage the Planning Inventory
- 6.1.7 Favor Smaller and More Frequent Releases
- 6.1.8 Plan to Learn Fast and Pivot When Necessary



SCRUM PLANNING

SCRUM Don't Plan !!!

Let's start the sprint and figure out what to do next!!!



INTRODUCTION TO PLANNING



Traditional Projects vs SCRUM Projects

- Traditional projects : Creates a detailed plan up front before development work begin.
 - ❖ get it right at the beginning so that rest can follow in an orderly fashion
- SCRUM : Does not believe it can be done in the very beginning
 - ❖ Can't produce all of the planning artifacts up front
 - ❖ Up-front and just-in-time planning



SCRUM Planning Principles

- Keep important options open until the last responsible moment
 - ❖ Wait till you have more information to take critical decisions
- SCRUM believes up-front plan is helpful , as long as you are willing to read and adapt to the changes
- Frequent re planning
- Use validated learning to continuously produce better, more useful plans.

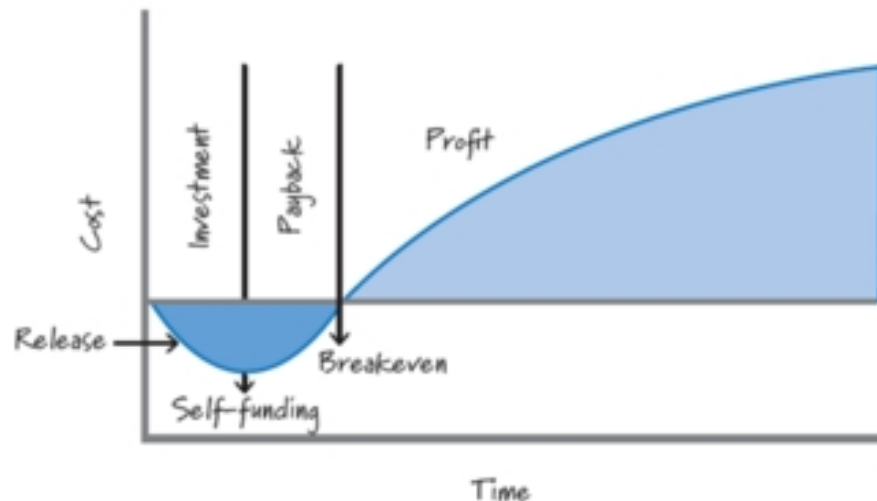


SCRUM Planning Principles Cont...

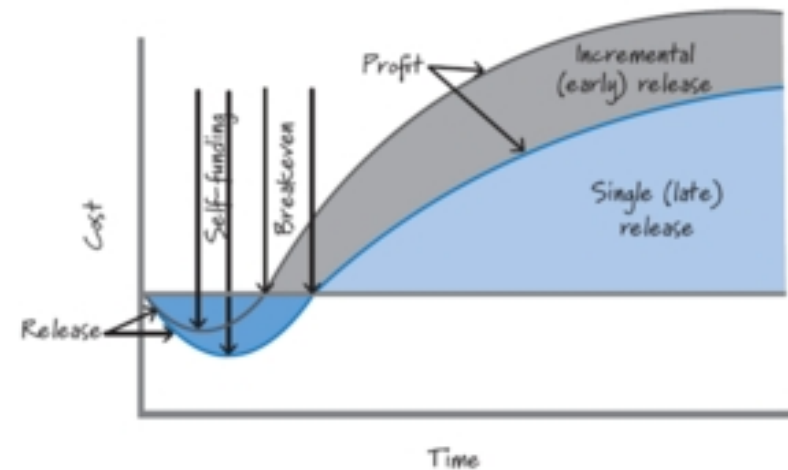
- Creating a large inventory of predictive, not-yet-validated planning artifacts might be wasteful
- Three forms of waste
 - ❖ wasted effort
 - ❖ waste of having to update the plan
 - ❖ wasted opportunity

SCRUM Planning Principles Cont...

- Favors smaller, more frequent releases
 - ❖ provide faster feedback
 - ❖ Improves ROI



Single-release economics



Multi-release economics



SCRUM Planning Principles Cont...

- Plan to learn fast and pivot when necessary
 - ❖ Doing something
 - ❖ Learning fast
 - ❖ Changing directions if needed



6.2: MULTILEVEL PLANNING



6.2: Multilevel Planning

- 6.2.1 Introduction
- 6.2.2 Portfolio Planning
- 6.2.3 Product Planning (Envisioning)
- 6.2.4 Release Planning
- 6.2.5 Sprint Planning
- 6.2.6 Daily Planning

Multilevel Planning

- Plan at multiple levels of detail and at multiple times throughout development.
 - Different levels of planning
 - ❖ Portfolio
 - ❖ Product
 - ❖ Release
 - ❖ Sprint
 - ❖ Daily
- } SCRUM formally define only these two



Portfolio Planning

- Determine which products to work on, in what order, and for how long.

Product Planning

- Product vision
 - ❖ provides a clear description of the areas in which the stakeholders get value.
- High-Level Product Backlog
- Product Roadmap

The outputs of product-level planning became inputs to portfolio planning



Release Planning

- Making scope, date, and budget trade-offs for incremental deliveries.
- To get an idea of what you can deliver by a fixed date -create and estimate a sufficient number of product backlog items
 - ❖ Draw a line through the product backlog
 - ❖ All of the items above the line are planned for the release
- Time dimension



Plan just in time and just enough



Sprint Planning

- Determine the specific product backlog items that the Scrum team will work on in the next sprint

Daily Planning

- The most detailed level of planning
- Team members get together and each person takes turns stating what he/she got done since the last daily scrum and the plan for today

"Today I am going to work on the stored procedure task, and I should have that done by lunch. Whoever is going to work on the business logic task, please keep in mind that the business logic task is on the critical path for getting our work done this sprint and you should be ready to work on it right after lunch."



6.3: PORTFOLIO PLANNING



6.3: Portfolio Planning

- 6.3.1 Introduction
- 6.3.2 Scheduling Strategies



Portfolio Planning

- An activity for determining
 - ❖ which portfolio backlog items to work on
 - ❖ in which order
 - ❖ for how long
- Backlog item can be a product, a product increment, or a project

Portfolio planning deals with a collection of products and is therefore larger in scope and higher level than individual product-level planning



Portfolio Planning Cont...

- Timing
 - ❖ A never-ending activity
 - ❖ the output of product-level is an important input to portfolio planning.
 - ❖ determines whether to fund the product and how to sequence it into the portfolio backlog
- Participants
 - ❖ portfolio planning focuses on both new products and in-process products
 - ❖ internal stakeholders
 - ❖ the product owners of individual products
 - ❖ senior architects and technical leads

Portfolio Planning Cont...

- Process
 - ❖ Portfolio planning has two outputs
 - ❖ portfolio backlog
 - ❖ set of active products
 - ❖ To arrive at these outputs, participants engage in four categories of activities
 - ❖ Scheduling
 - ❖ managing inflows
 - ❖ managing outflows
 - ❖ managing in-process products



Portfolio Planning : Scheduling Strategies

- Optimize for lifecycle profits.
- Calculate the cost of delay.
- Estimate for accuracy, not precision.



Portfolio Planning : Inflow Strategies

- Apply the Economic Filter
- Balance the Arrival Rate with the Departure Rate
- Quickly Embrace Emergent Opportunities
- Plan for Smaller, More Frequent Releases



Portfolio Planning : Outflow Strategies

- Focus on idle work, not idle workers.
- Establish a WIP limit. (Work In Progress)
- Wait for a complete team.



Portfolio Planning : In-Process Strategies

- Use Marginal Economics
 - ❖ Four options
 - ❖ Preserve—continue developing the product.
 - ❖ Deliver—stop working on the product and ship it.
 - ❖ Pivot—take what we have learned and change directions.
 - ❖ Terminate—stop working on the product and kill it.



6.4 ENVISIONING (PRODUCT PLANNING)

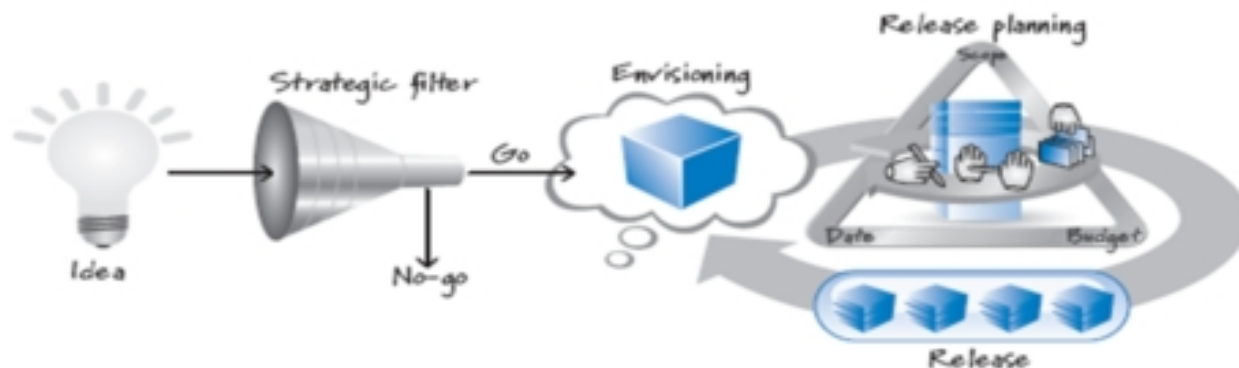


6.4: Envisioning (Product Planning)

- 6.4.1 Introduction
- 6.4.2 SR4U Example
- 6.4.3 Visioning
- 6.4.4 High-Level Product Backlog Creation
- 6.4.5 Product Roadmap Definition
- 6.4.6 Other Activities
- 6.4.7 Economically Sensible Envisioning
- 6.4.8 Closing

Product Planning (Envisioning)

- Used to create the initial product backlog
 - ❖ a product vision is needed to generate an initial product backlog
- The goal is to work upon an idea, describing the essence of the potential product and creating a rough plan for how to approach its creation.
- Can decide whether to fund the next level of more detailed development



Product Planning (Envisioning) Cont...

- Participants
 - ❖ The product owner
 - ❖ internal stakeholders
 - ❖ Specialists in various areas

Ideally, the Scrum Master and the development team should participate .
But in reality its not the case (Since companies waits till the initial envisioning is done to allocate the SCRUM team)

- Process
 - ❖ SR4U Example



Product Planning (Envisioning)) Cont...

- **Economically sensible envisioning**
 - ❖ Needs to be carried out in an economically sensible way.
 - ❖ It should be viewed as an investment
 - ❖ Too little envisioning
 - ❖ unprepared to do the first customer-value-creation sprint
 - ❖ Too much envisioning
 - ❖ will create a large inventory of product-planning artifacts that may have to be reworked or discarded

Product Planning (Envisioning)) Cont...

- **Guidelines for economically sensible envisioning**
 - ❖ Target a realistic confidence threshold
 - ❖ Focus on a short horizon
 - ❖ Act quickly
 - ❖ Pay for validated learning
 - ❖ Use incremental/provisional funding
 - ❖ Learn fast and pivot



6.5 RELEASE PLANNING (LONGER-TERM PLANNING)

Reference 01 page 307



6.5: Release Planning (Longer-Term Planning)

- 6.5.1 Introduction
- 6.5.2 Release Constraints
- 6.5.3 Grooming the Product Backlog
- 6.5.4 Refine Minimum Releasable Features (MRFs)
- 6.5.5 Sprint Mapping (PBI Slotting)
- 6.5.6 Fixed-Date Release Planning
- 6.5.7 Fixed-Scope Release Planning
- 6.5.8 Calculating Cost
- 6.5.9 Communicating



Release Planning

- Helps answer
 - ❖ When will we be done?
 - ❖ Which features can I get by the end of the year?
 - ❖ How much will this cost?
- Longer term planning
- Every organization must identify the proper sequence for releasing features
 - ❖ Different ways of releasing
 - ❖ Release after multiple sprints
 - ❖ Release every sprint
 - ❖ Release every feature

Release planning involves collaboration between the stakeholders and the full Scrum team



Release Planning Cont...

- The inputs to release planning include outputs from product planning,
 - ❖ the product vision
 - ❖ high-level product backlog
 - ❖ product roadmap
- Activities
 - ❖ Confirm the release constraints of scope, date, and budget
 - ❖ Product backlog grooming
 - ❖ creating, estimating, and prioritizing product backlog items



Release Planning Cont...

- MRFs (Minimum Releasable Features)
- Always review the MRFs
- Some organizations also create a sprint map during this time

Release Planning Cont...

- Release Constraints**

- ❖ The constraints of scope, date, and budget are important variables that affect the way project goals are achieved.
 - ❖ Eg: SR4U Release 1.0 had a fixed-date constraint
- ❖ Development Constraint Combinations

Project Type	Scope	Date	Budget
Fixed everything (not recommended)	Fixed	Fixed	Fixed
Fixed scope and date (not recommended)	Fixed	Fixed	Flexible
Fixed scope	Fixed	Flexible	Fixed (not really)
Fixed date	Flexible	Fixed	Fixed

at least one of these variables should be flexible



Release Planning Cont...

- Grooming the Product Backlog
 - ❖ The envisioning (product planning) stage creates a high-level product backlog
 - ❖ Used to define a set of minimum releasable features for each release
 - ❖ Many of these backlog items are too large to be useful during release planning . Therefore they need to be refined
 - ❖ Eg:
"As a typical user I want a simple, Google-like interface for requesting a review search so that I don't have to spend much time describing what I want."

can be refined into

"As a typical user I want to select a category of product or service so that I can help SR4U focus only on relevant reviews "



Release Planning Cont...

- Grooming the Product Backlog(contd..)
 - ❖ Once refined , the team would estimate them to determine rough cost.
 - ❖ Prioritize the estimated stories based on the release goal and constraints
- Sprint Mapping
 - ❖ In each sprint the team works on a set of product backlog items
 - ❖ Early mapping of near-term product backlog items into sprints is helpful
 - ❖ Example...



Release Planning Cont...

- Fixed-Date Release Planning
 - ❖ many organizations prefer to use fixed-date releases.
 1. calculate the number of sprints needed to perform in this release (Assume six)
 2. determine how much work the team can get done in six sprints
 3. determine what features can be completed during the sprints
 - ❖ SR4U example

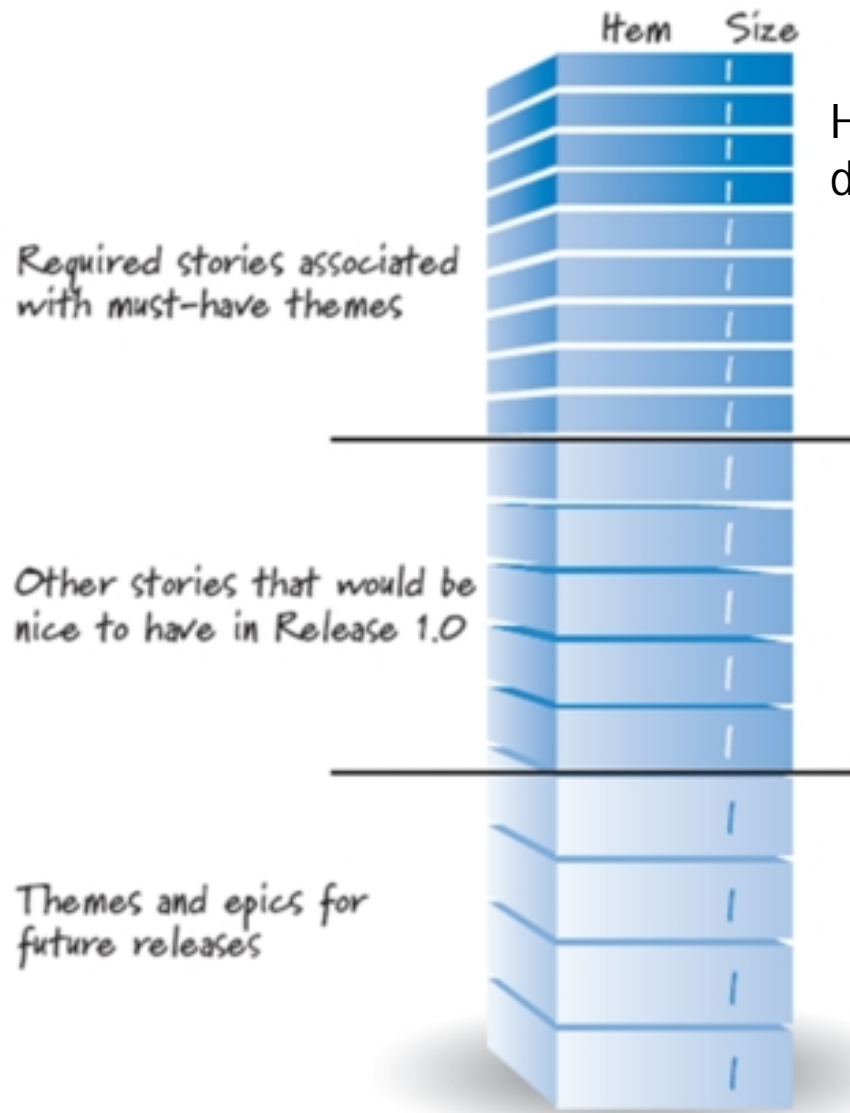




Release Planning Cont...

Steps for Performing Fixed-Date Release Planning

1. Determine how many sprints are in this release.
2. Groom the product backlog.
3. Measure or estimate the team's velocity.
4. Multiply the slower velocity by the number of sprints. (will-have line.)
5. Multiply the faster velocity by the number of sprints. (might-have line)



Helps answer , "What will I get by the release date?"

Early in the release it is difficult to give a precise answer to this. Thus we present a range. the broader the range, the less certain we are.



We will have our must-have features

We will have most of our must-have features, but we might or might not have all of them."

We won't have our must-have features.



Release Planning Cont...

- **Fixed-Scope Release Planning**

what if your product scope is more important than the date??

What if you have a large set of must have features in your minimum releasable features set ??

calculate the number of sprints needed to perform in this release (Assume six)

1. determine how much work the team can get done in six sprints
2. determine what features can be completed during the sprints

❖ SR4U example





Fixed-Scope Release Planning

Steps for Performing Fixed-Scope Release Planning

1. Groom the product backlog
2. Determine the total size of the PBI (Product Backlog Item)s
3. Estimate the team's velocity as a range.
4. Divide the total size of the PBIs by the faster velocity (round up)
5. Divide the total size of the PBIs by the slower velocity (round up)

Calculating Cost

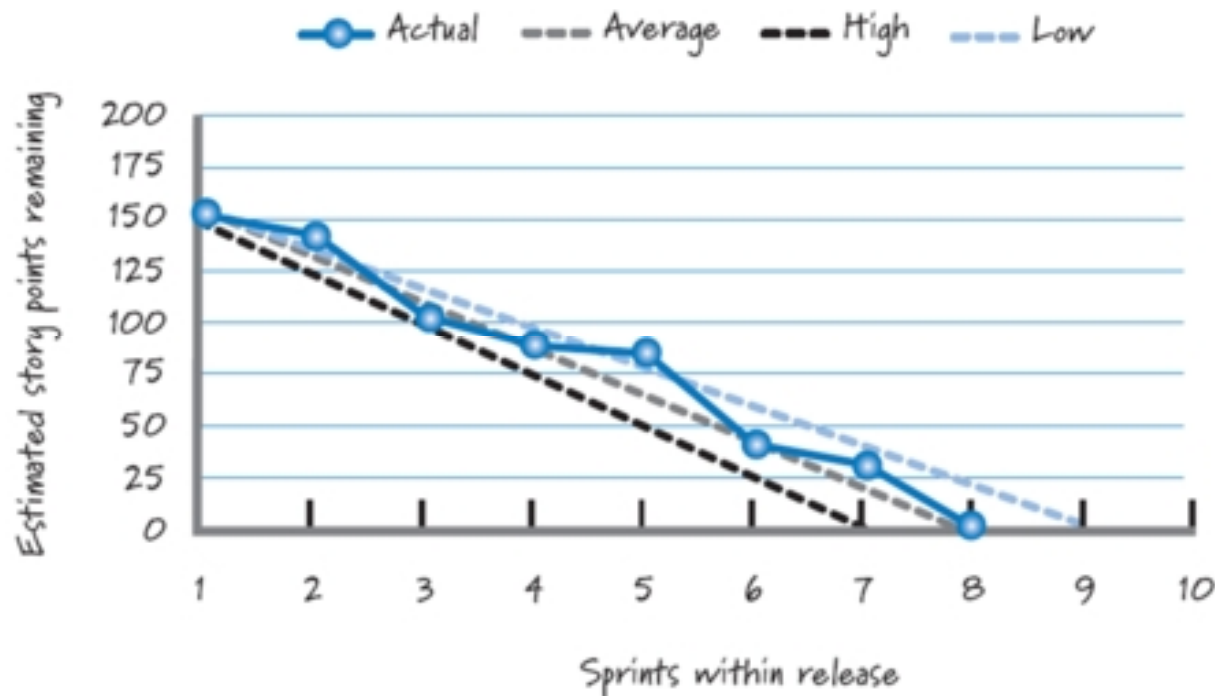
- Identify the team.
- Determine the sprint length.
- Based on team composition and sprint length, determine the personnel costs of running a sprint.
- For a fixed-date release
 - ❖ multiply the number of sprints in the release by the cost per sprint.
- For a fixed-scope release
 - ❖ multiply both the high and low number of sprints by the cost per sprint.

Communicating

- Communicating Progress on a Fixed-Scope Release
(Have an idea on the total scope to be achieved)
 - ❖ Fixed-Scope-Release Burn down Chart
 - ❖ shows the total amount of unfinished work
- Eg:
- ❖ 150 story points at the start
 - ❖ At the end of each sprint , the chart has to be updated to show the total amount of work remaining
 - ❖ Sprint velocity
 - ❖ Can shows projected outcomes

Communicating Progress on a Fixed-Scope Release : Fixed-Scope-Release Burn down Chart

- ❖ Three lines predicts, three release dates
- ❖ Highest velocity : 22 (Will finish within 7 weeks)
- ❖ Lowest velocity : 18 (Will take 9 weeks)





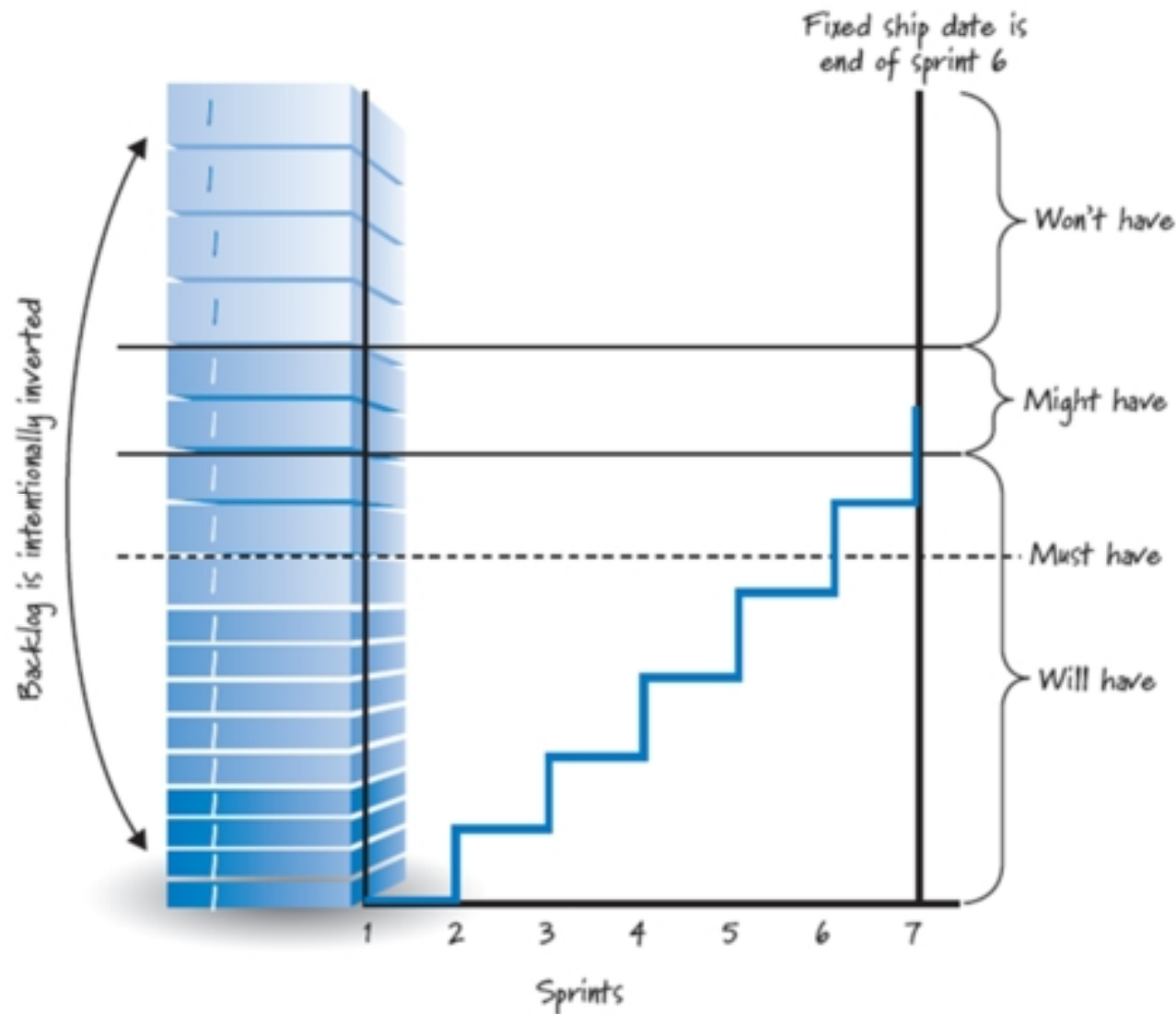
Communicating Progress on a Fixed-Scope Release : Fixed-Scope-Release Burn Up Chart

- Shows the total amount of work in a release as a goal and the progress of each sprint toward achieving that goal
- Similarly to the release burn down chart, this shows the same three predictive lines indicating the likely number of sprints to achieve the target.



Communicating Progress on a Fixed-Date Release

- Know the number of sprints in the release
- Goal is to communicate the range of features we expect to complete and sprint by sprint progress.
- Traditional burn down and burn up charts are ineffective
 - ❖ Why?
 - ❖ ...
- Need to update the chart created on fixed-date release plan at the end of each sprint
- **Can create a specialized burnup chart**



- ❖ The product backlog is inverted
- ❖ Highest-priority items at the bottom
- ❖ Lower-priority items at the top