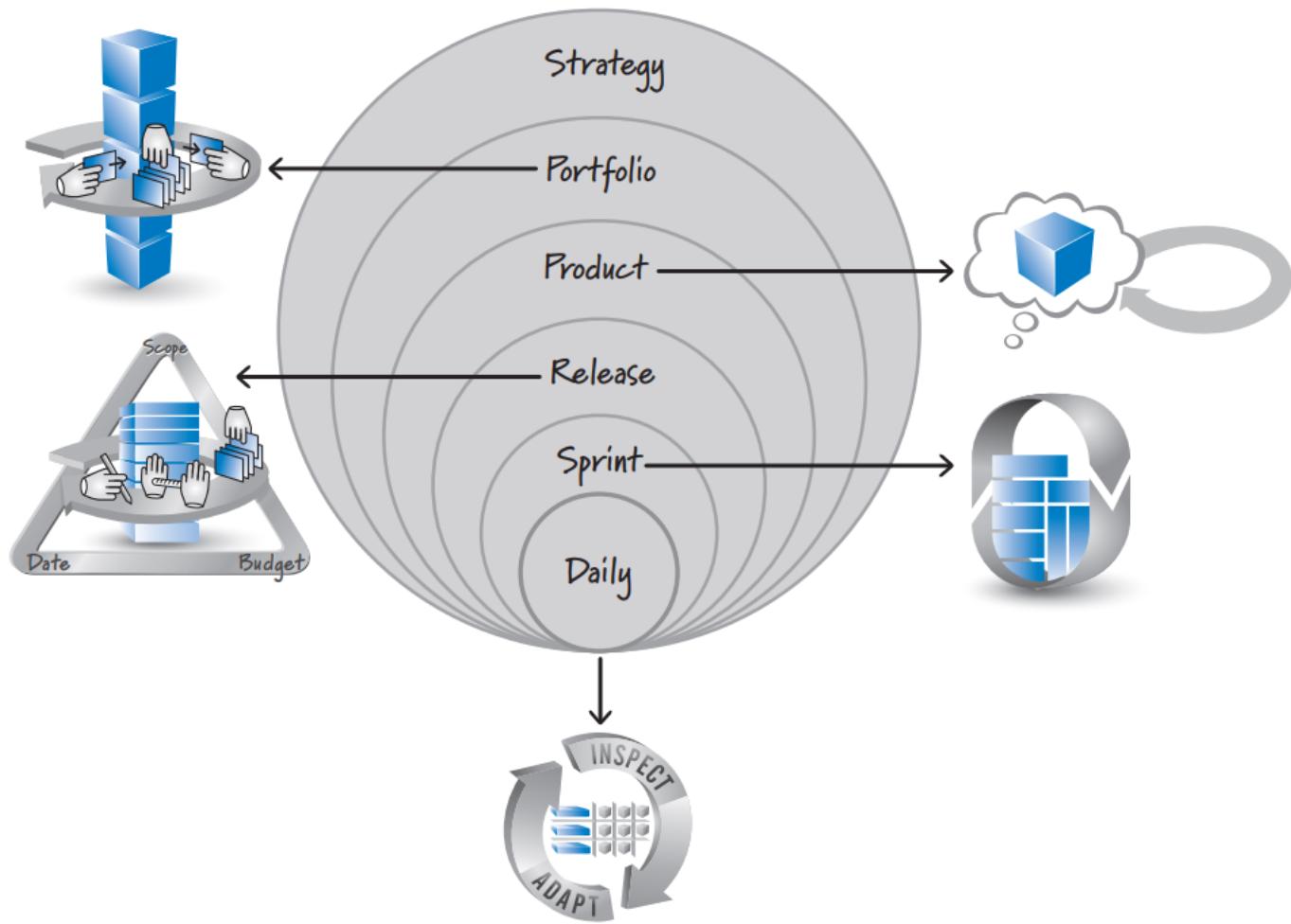


SCRUM PLANNING

Systems Analysis and Design
Sharif University of Technology
Fall 1400-1401

Different levels of planning

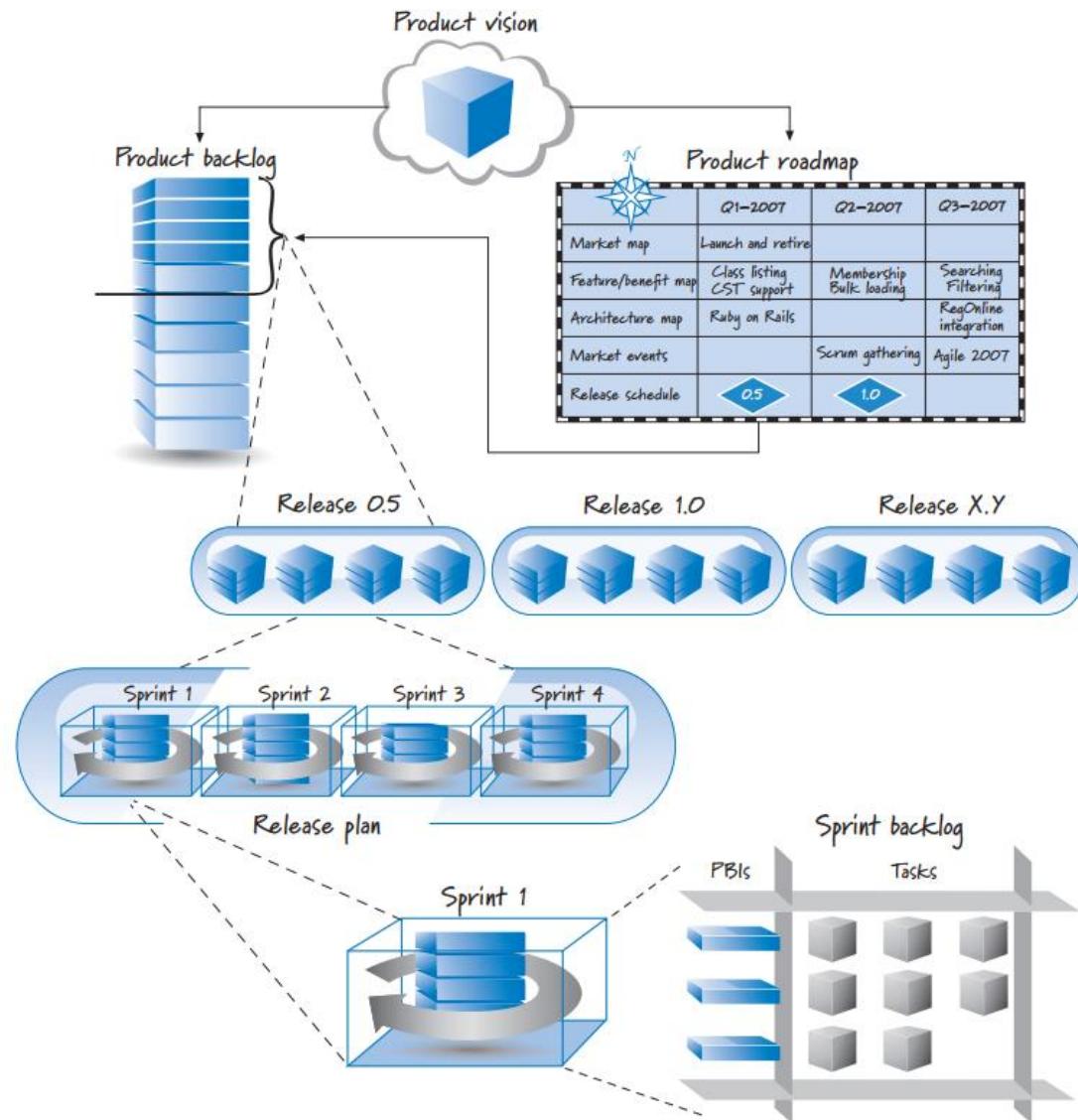


High-Level Planning

- **Strategic Planning:** An activity for determining sets (portfolios) of related software development projects that an organization needs to carry out.
- **Portfolio Planning/Management:** An activity for determining which products of a portfolio to work on, in what order, and for how long.
- **Product Planning (Envisioning):** An activity for capturing the essence of a potential product, and creating a rough plan for its creation.
 - Envisioning begins with the creation of a Vision, followed by the creation of a high-level Product Backlog, and frequently a Product Roadmap.
- **Release Planning:** An activity for making scope, date, and budget tradeoffs for incremental delivery of a product.

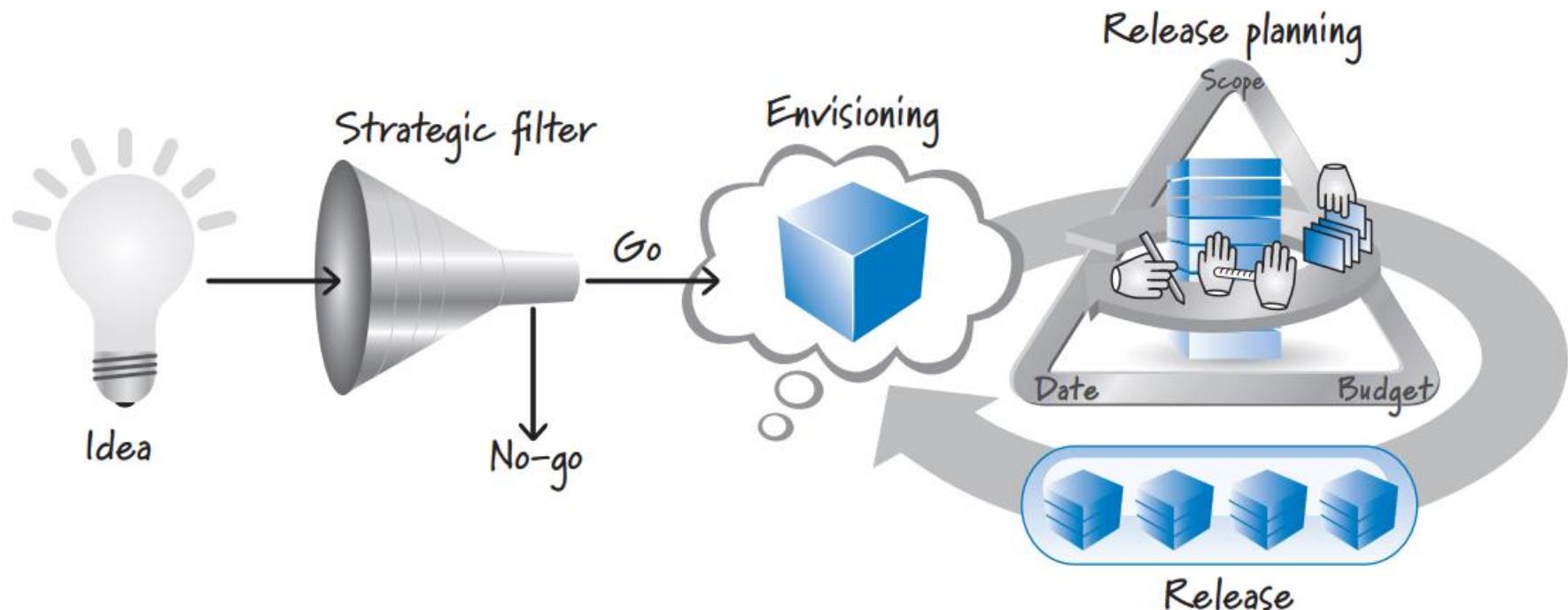
Level	Horizon	Who	Focus	Deliverables
Portfolio	Possibly a year or more	Stakeholders and product owners	Managing a portfolio of products	Portfolio backlog and collection of in-process products
Product (envisioning)	Up to many months or longer	Product owner, stakeholders	Vision and product evolution over time	Product vision, roadmap, and high-level features
Release	Three (or fewer) to nine months	Entire Scrum team, stakeholders	Continuously balance customer value and overall quality against the constraints of scope, schedule, and budget	Release plan
Sprint	Every iteration (one week to one calendar month)	Entire Scrum team	What features to deliver in the next sprint	Sprint goal and sprint backlog
Daily	Every day	ScrumMaster, development team	How to complete committed features	Inspection of current progress and adaptation of how best to organize the upcoming day's work

Hierarchical Scrum planning



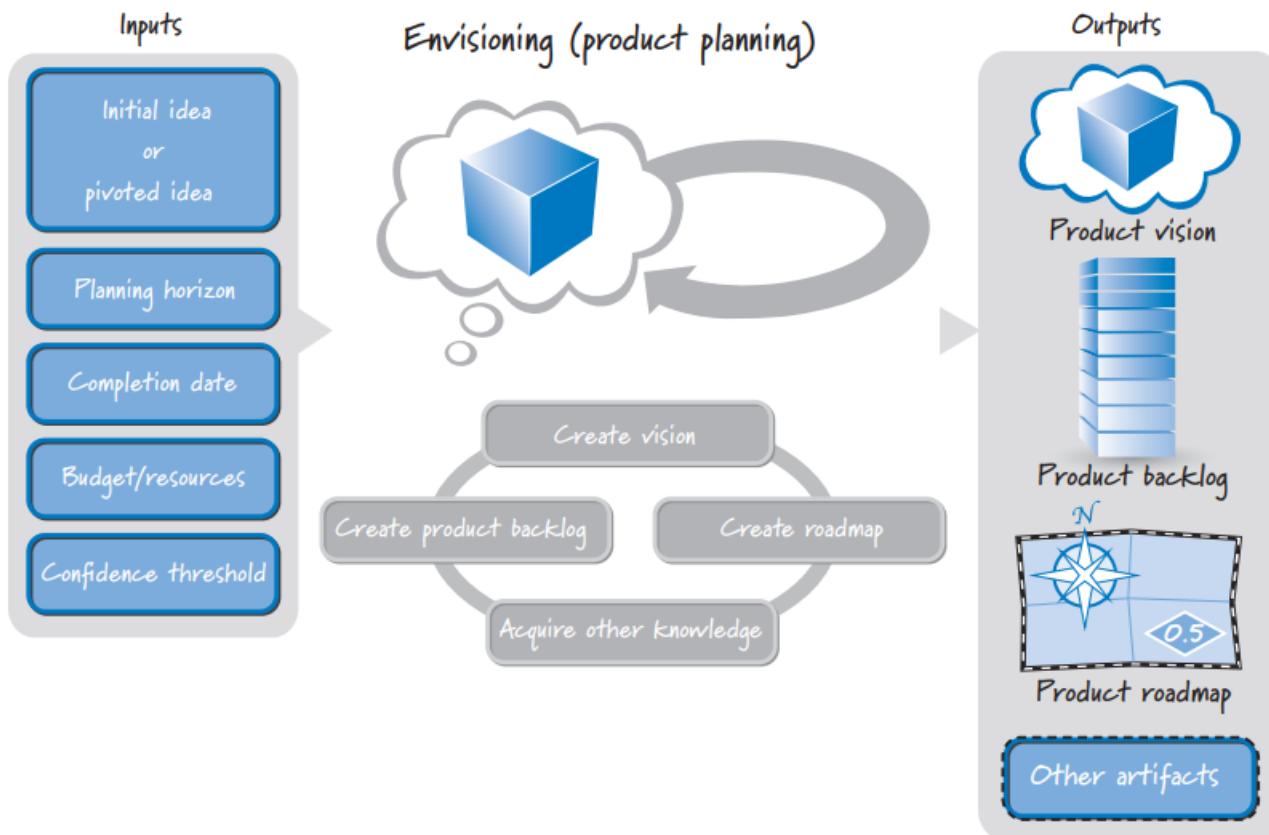
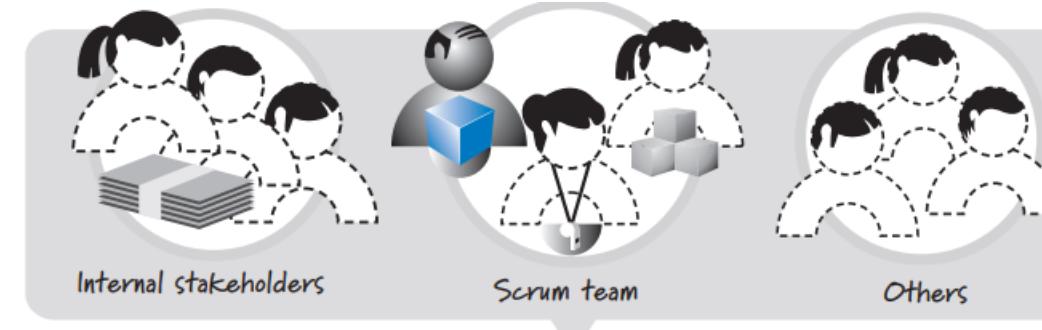
Product planning – Envisioning

Product Planning: timing



[Rubin 2012]

Product Planning: Process



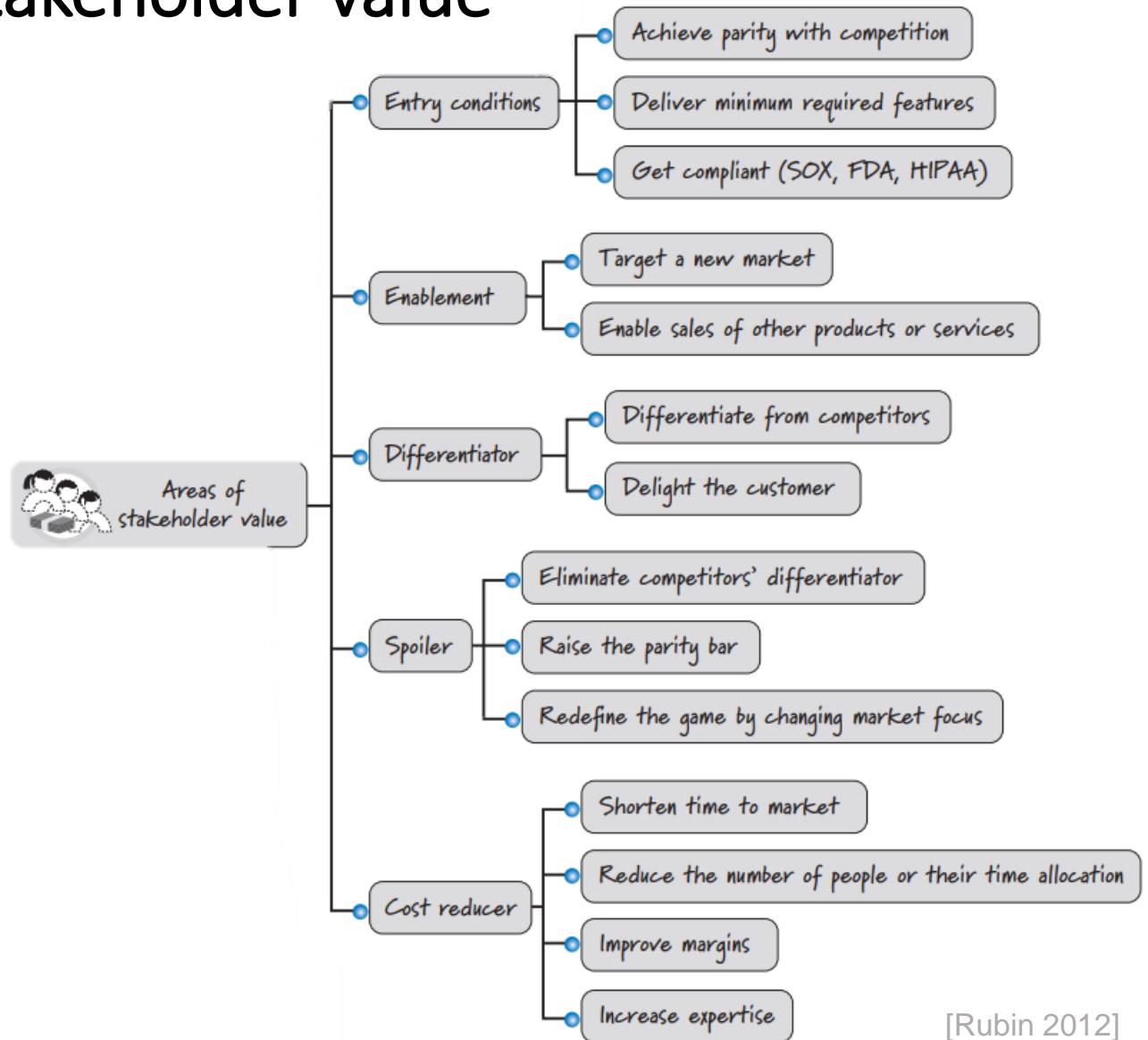
Product Planning: Process

- **Inputs:** Initial/Pivoted idea, Planning horizon, Completion date, Budget/Resources, and Confidence threshold.
- **Activities:** Create vision, create product backlog, create roadmap, and acquire other knowledge.
- **Output:** Product Vision, High-Level Product Backlog, Product Roadmap, and Other artifacts.
- **Participant:** The product owner is the only required participant (others are optional)

Visioning

- In Scrum, a vision is not an elaborate, several-hundred-page document even for complex systems. If we need this much space to describe our vision, we probably don't understand it.
- The vision is frequently expressed in terms of how the stakeholders get value (**areas of stakeholder value**).
- The **format** of the vision itself can be a variety of types. You should choose whatever format best suits your organization, envisioning group, and idea.

Areas of stakeholder value



[Rubin 2012]

Areas of stakeholder value – Example

Area	Description
Cost reducer/time savings	SR4U must save its users considerable time when searching for reviews.
Differentiator/delight the customer	SR4U must provide a “wow” experience for its users. Users must feel that the service performed an impressive task for them, helping them make an informed purchase.
Spoiler/raise the parity bar	SR4U should create substantial chaos for its competitors. Their current solutions should immediately look antiquated by comparison. SR4U will establish a new baseline for online review services that others will have to scramble to meet.

[Rubin 2012]

Popular Vision Formats

Format	Description
Elevator statement	Write a 30-second to one-minute quick pitch of the product vision. Imagine you have stepped into an elevator with a venture capitalist and you have to pitch him on your product vision. Could you do it in a short elevator ride?
Product datasheet	Write the product datasheet on the first day. Try to fit it on the front side of a one-page marketing piece.
Product vision box	Draw the box in which you want to put the product when it ships. Can you come up with three or four salient points to illustrate on the box? (Drafting 15 points is easier than drafting three or four.)
User conference slides	Create the two or three presentation slides that you would use to introduce the product at your user conference (or equivalent). Try to avoid any bullet points on your slides.
Press release	Write the press release you want to issue when the product becomes available. Good press releases clearly communicate what is newsworthy in one page or less.
Magazine review	Draft a fictitious magazine review bylined by the solution reviewer in your industry's most popular trade magazine.

Product Vision Example: Press Release Format

Review Everything, Inc., announced today the successful launch of its new SmartReview4You service. This service provides all online users with their own trainable agent to scour the Internet and identify unbiased, relevant product or service review information.

Remarked Doris Johnson, an avid user of online reviews, "I now have my very own personal assistant that mimics how I find and filter online reviews. It's amazing—I teach it what I like and don't like about reviews, then SmartReview4U tears across the Internet finding product or service reviews and automatically weeds out the biased or bogus ones. It does at lightning speed what used to take me forever. This service is a huge timesaver!"

C. J. Rollins, CEO of Review Everything, Inc., said, "We are pleased to offer the world's first truly smart review service. Since the inception of the Internet people have leveraged the wisdom of the online crowd. However, the crowd can get very noisy at times and it is hard to separate the wheat from the chaff. Our super-smart service does the laborious work of sifting through the huge volume of online review information, eliminating suspicious reviews and returning only relevant ones. You read only the reviews you'd choose to consider if you spent hours searching on your own."

The new SmartReview4You is available free of charge at the following website: www.smartreview4you.com.

Product Vision Example: Start point

For (target customer)

Who (statement of need or opportunity)

The (product name) **is a** (product category)

That (key benefit, reason to buy)

Unlike (primary competitive alternative)

Our product (statement of primary differentiation)

Online shopping Mall

For	people
Who	want an easy way to buy grocery items
The Bee Shopping Mall	is a web-based grocery mall
That	allows consumers to buy items from the web
Unlike	existing solutions that require consumers to buy from physical stores
Our product	will provide a totally web-based shopping experience

High-Level Product Backlog

- Consists of a set of initial epics, typically identified during a brainstorming session. Example:

As a typical user I want to teach SR4U what types of reviews to discard so that SR4U will know what characteristics to use when discarding reviews on my behalf.

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.

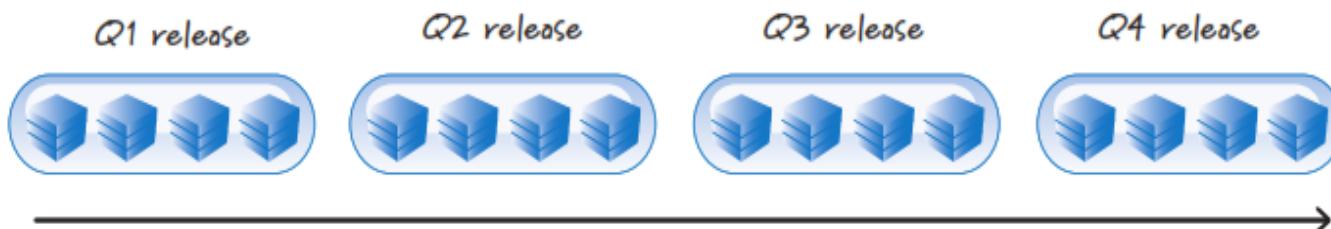
As a typical user I want to have SR4U monitor the Internet for new reviews on products or services of interest and automatically filter and report them to me so that I don't have to keep asking SR4U to do it for me.

As a sophisticated user I want to tell SR4U which sources to use when searching on my behalf so that I don't get back reviews from sites I don't like or trust.

As a product vendor I want to be able to show an SR4U-branded review summary for my product on my website so that people can immediately see what the marketplace thinks of my product as determined by a trusted source like SR4U.

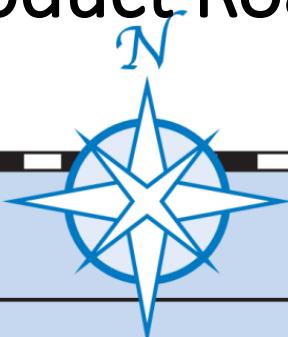
Product Roadmap

- Produced after the initial vision and the high-level product backlog.
- Consists of a series of releases for achieving some or all of the product vision.
- Focus each release on a small set of **minimum releasable features (MRFs)** around which the stakeholder community share a strong group consensus (also known as **MVP** or **MMF**).
- Some organizations use the strategy of fixed, periodic releases to simplify the product roadmap and establish a rhythm, or cadence.



[Rubin 2012]

Product Roadmap - Example



	Q3—Year 1	Q4—Year 1	Q1—Year 2
Market map	Initial launch	Better results More platforms	Sophisticated users
Feature/benefit map	Basic learning Basic filtering	Improved learning Complex queries	Define sources Learn by example
Architecture map	100K concurrent web users	iOS and Android	Web services interface
Market events	Social Media Expo	Review Everything User Conference	
Release schedule	1.0	2.0	3.0

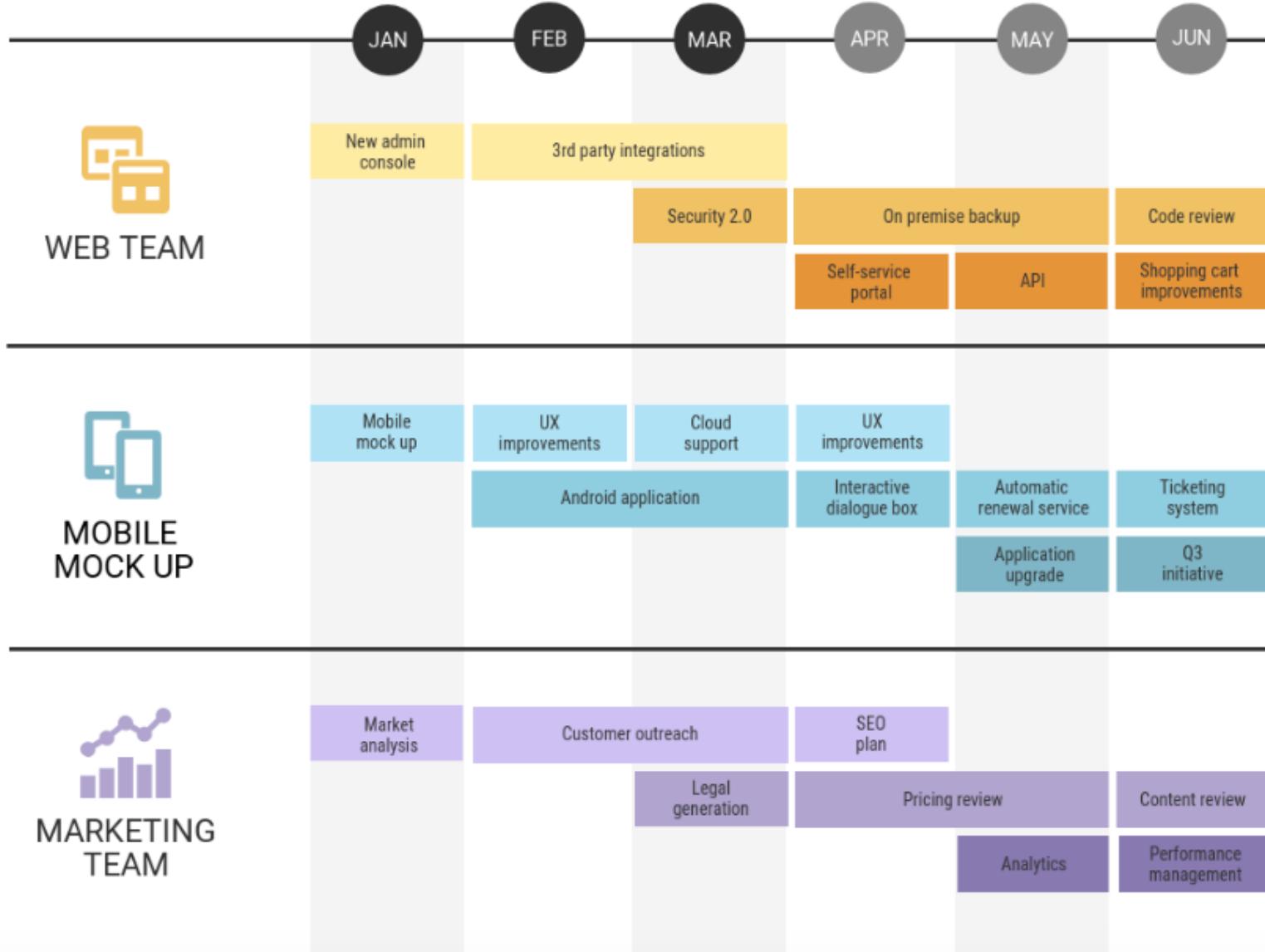
[Rubin 2012]

THE GO PRODUCT ROADMAP

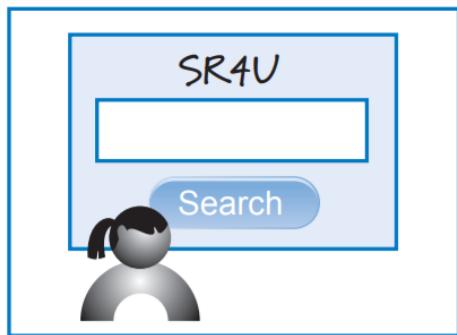
 DATE The release date or timeframe	<i>Date or timeframe</i>	<i>Date or timeframe</i>	<i>Date or timeframe</i>	<i>Date or timeframe</i>
 NAME The name of the new release	<i>Name/version</i>	<i>Name/version</i>	<i>Name/version</i>	<i>Name/version</i>
 GOAL The reason for creating the new release	<i>Goal</i>	<i>Goal</i>	<i>Goal</i>	<i>Goal</i>
 FEATURES The high-level features necessary to meet the goal	<i>Features</i>	<i>Features</i>	<i>Features</i>	<i>Features</i>
 METRICS The metrics to determine if the goal has been met	<i>Metrics</i>	<i>Metrics</i>	<i>Metrics</i>	<i>Metrics</i>

ADVENT CO. // ENVISION 6.0 // PRODUCT ROADMAP

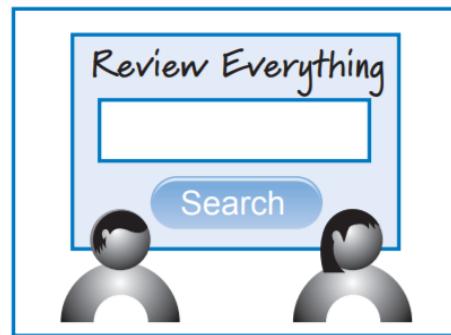
● Q1 ● Q2



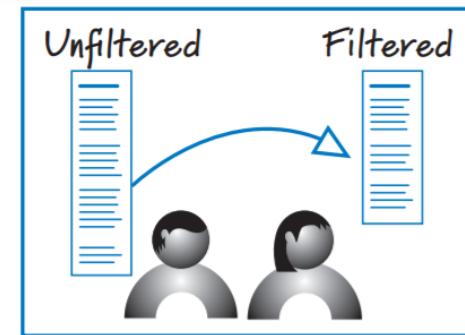
Other activates: Acquire Knowledge



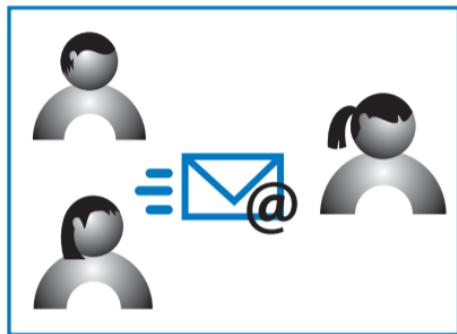
Customer submits query



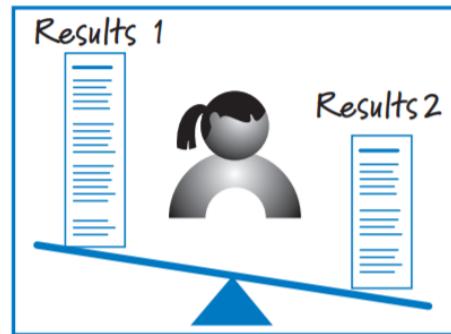
SMEs run existing product query



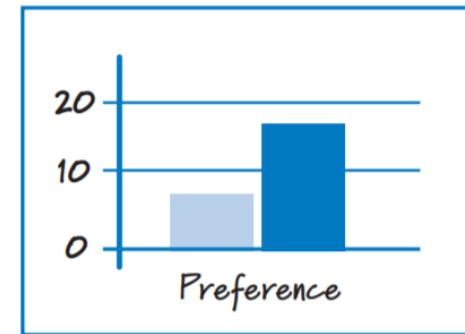
SMEs manually create filtered results



SMEs email both results to customer

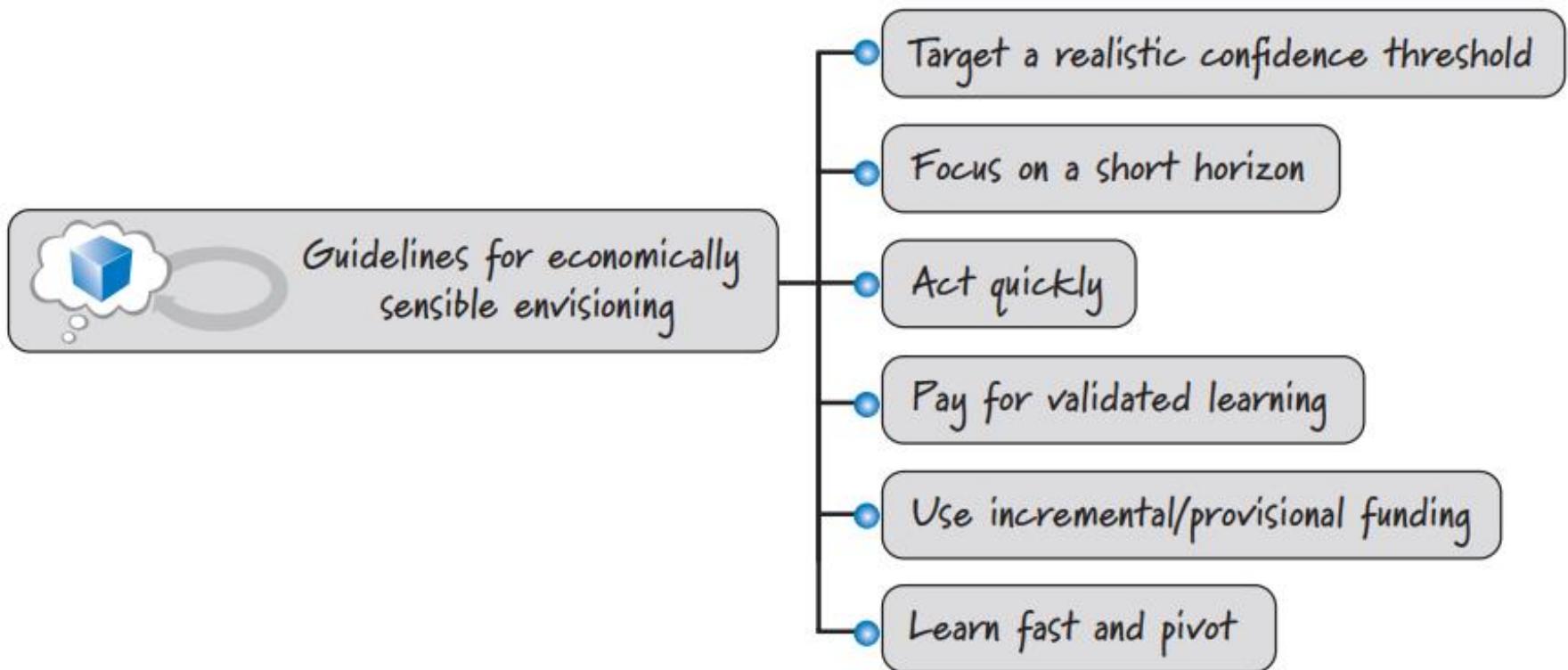


Customer evaluates results for preference



Roger and others analyze results and decide outcome

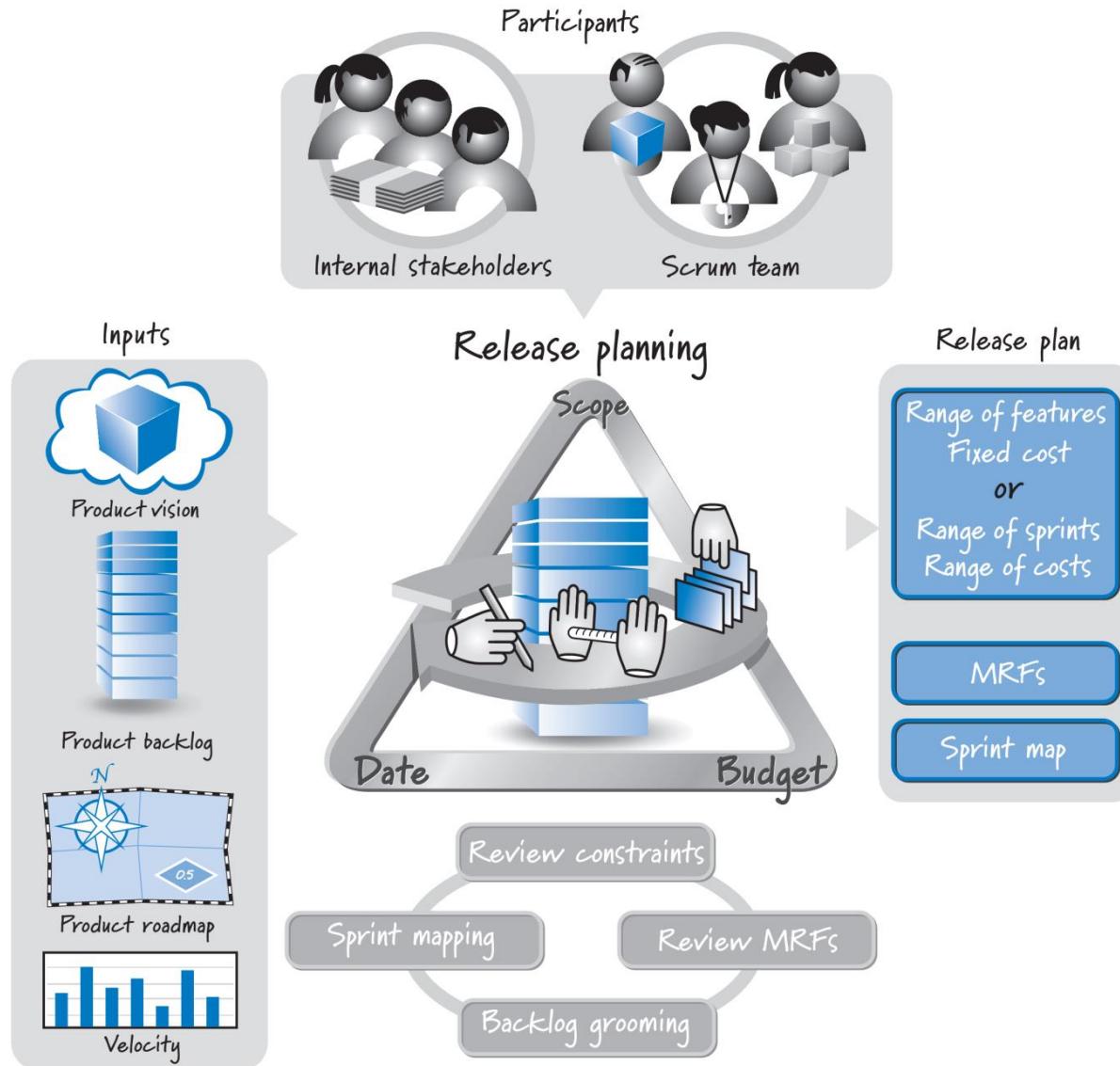
Product planning Guidelines



Release planning



Release Planning: Process



Release Planning: Process

- **Inputs:** Product Vision, High-Level Product Backlog, Product Roadmap, and Velocity (of the team or teams that will work on the release).
- **Activities:** Reviewing Release Constraints (scope, date, and budget), Reviewing Minimum Releasable Features (MRFs of the release), Product Backlog Grooming, and Sprint Mapping (indicating in which sprint some or many of the product backlog items might be created).
- **Output:** Release Plan (showing when we will **finish**, what **features** we will get, what the **cost** will be, what the desired **MRFs** are for the release, and how some of the product backlog items **map to sprints** within the release).

Development Constraints

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

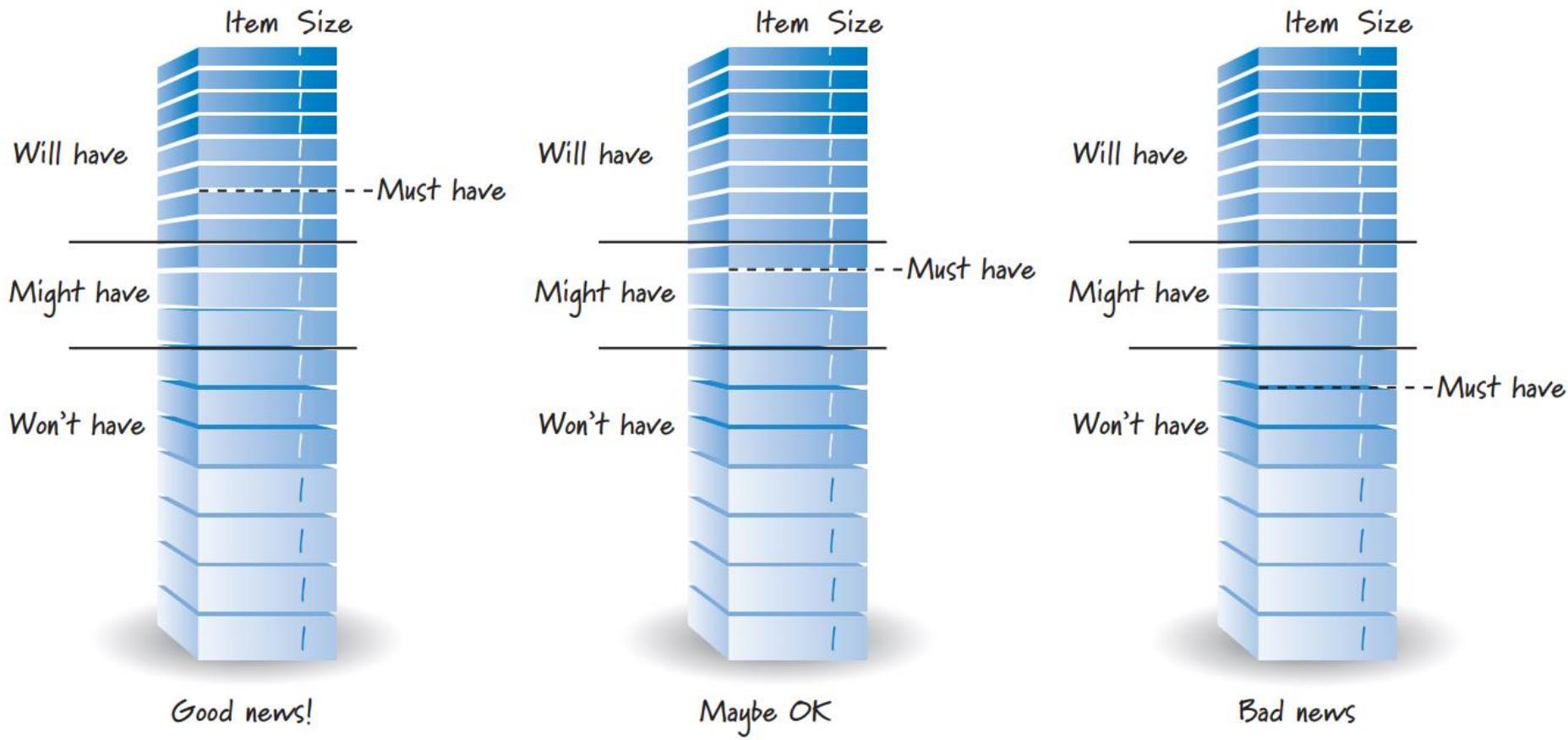
[Rubin 2012]

Fixed-Date Release Planning

Step	Description	Comments
1	Determine how many sprints are in this release.	If all sprint lengths are equal, this is simple calendar math because you know when the first sprint will start and you know the delivery date.
2	Groom the product backlog to a sufficient depth by creating, estimating the size of, and prioritizing product backlog items.	Because we are trying to determine which PBIs we can get by a fixed date, we need enough of them to plan out to that date.
3	Measure or estimate the team's velocity as a range.	Determine an average faster and an average slower velocity for the team
4	Multiply the slower velocity by the number of sprints. Count down that number of points into the product backlog and draw a line.	This is the "will-have" line.
5	Multiply the faster velocity by the number of sprints. Count down that number of points into the product backlog and draw a second line.	This is the "might-have" line.

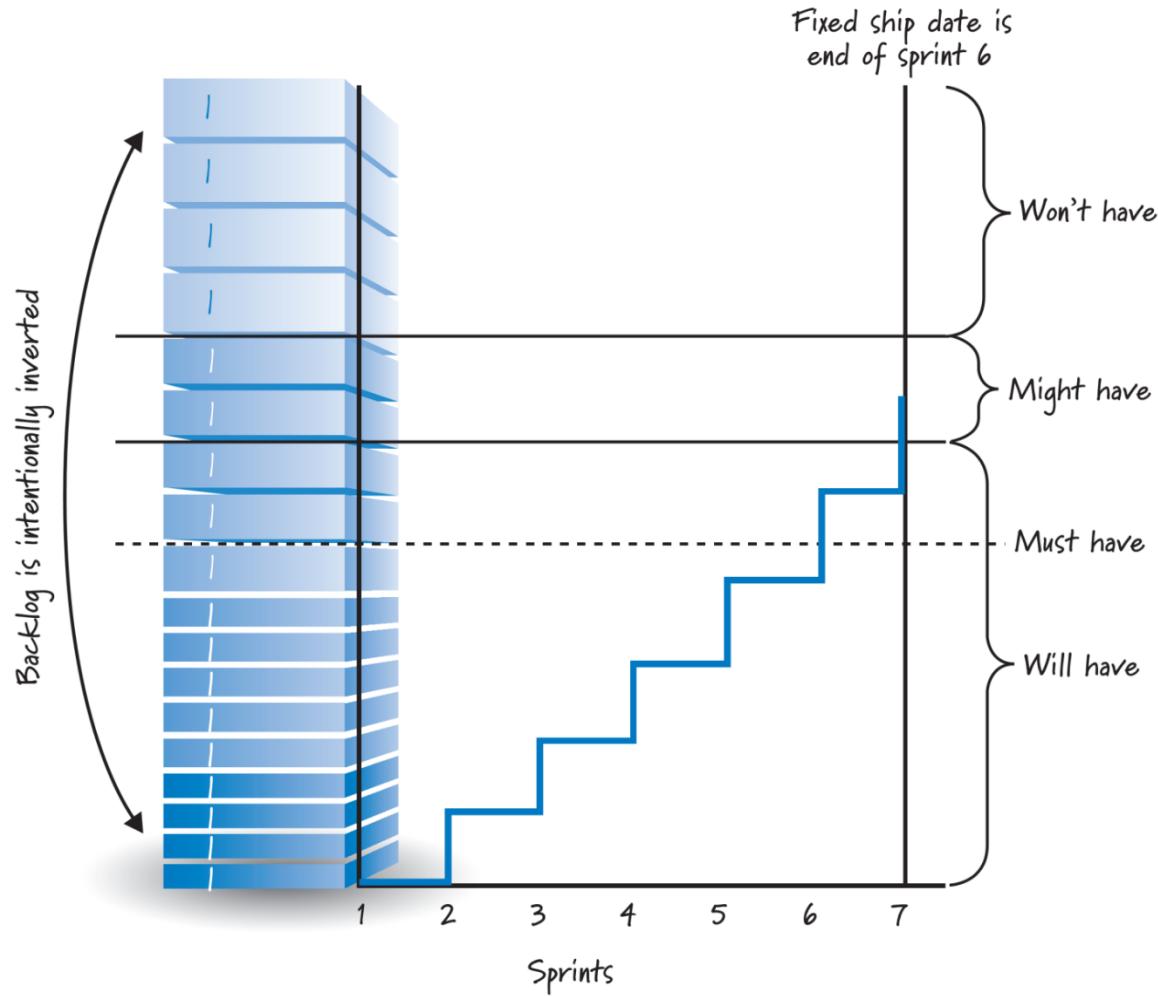
[Rubin 2012]

Fixed-Date Release Planning



[Rubin 2012]

Fixed-Date Burnup Chart



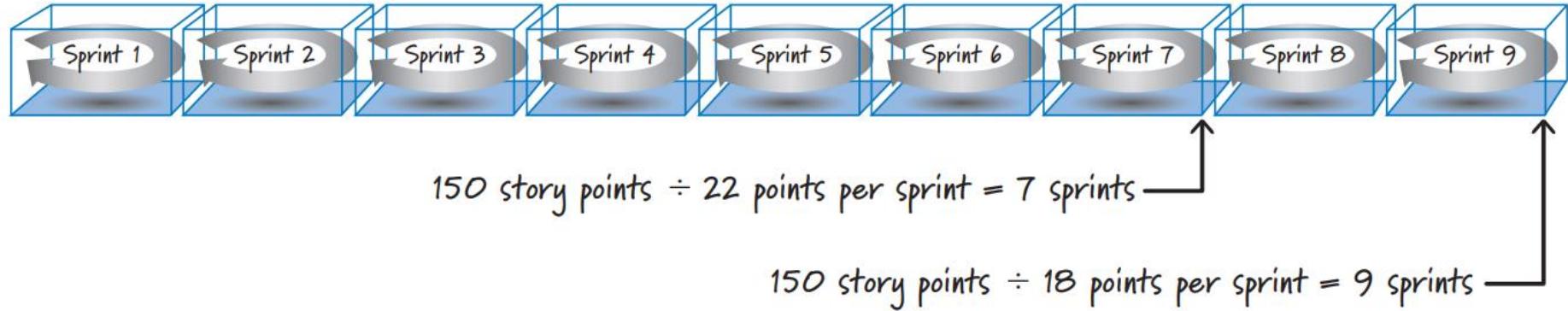
[Rubin 2012]

Fixed-Scope Release Planning

Step	Description	Comments
1	Groom the product backlog to include at least the PBIs we would like in this release by creating, estimating the size of, and prioritizing PBIs.	Because this is a fixed-scope release, we need to know which PBIs are in the fixed scope.
2	Determine the total size of the PBIs to be delivered in the release.	If we have a product backlog of estimated items, we simply sum the size estimates of all of the items we want in the release.
3	Measure or estimate the team's velocity as a range.	Determine an average faster and an average slower velocity for the team.
4	Divide the total size of the PBIs by the faster velocity and round up the answer to the next integer.	This will tell us the lowest number of sprints required to deliver the features.
5	Divide the total size of the PBIs by the slower velocity and round up the answer to the next integer.	This will tell us the highest number of sprints required to deliver the features.

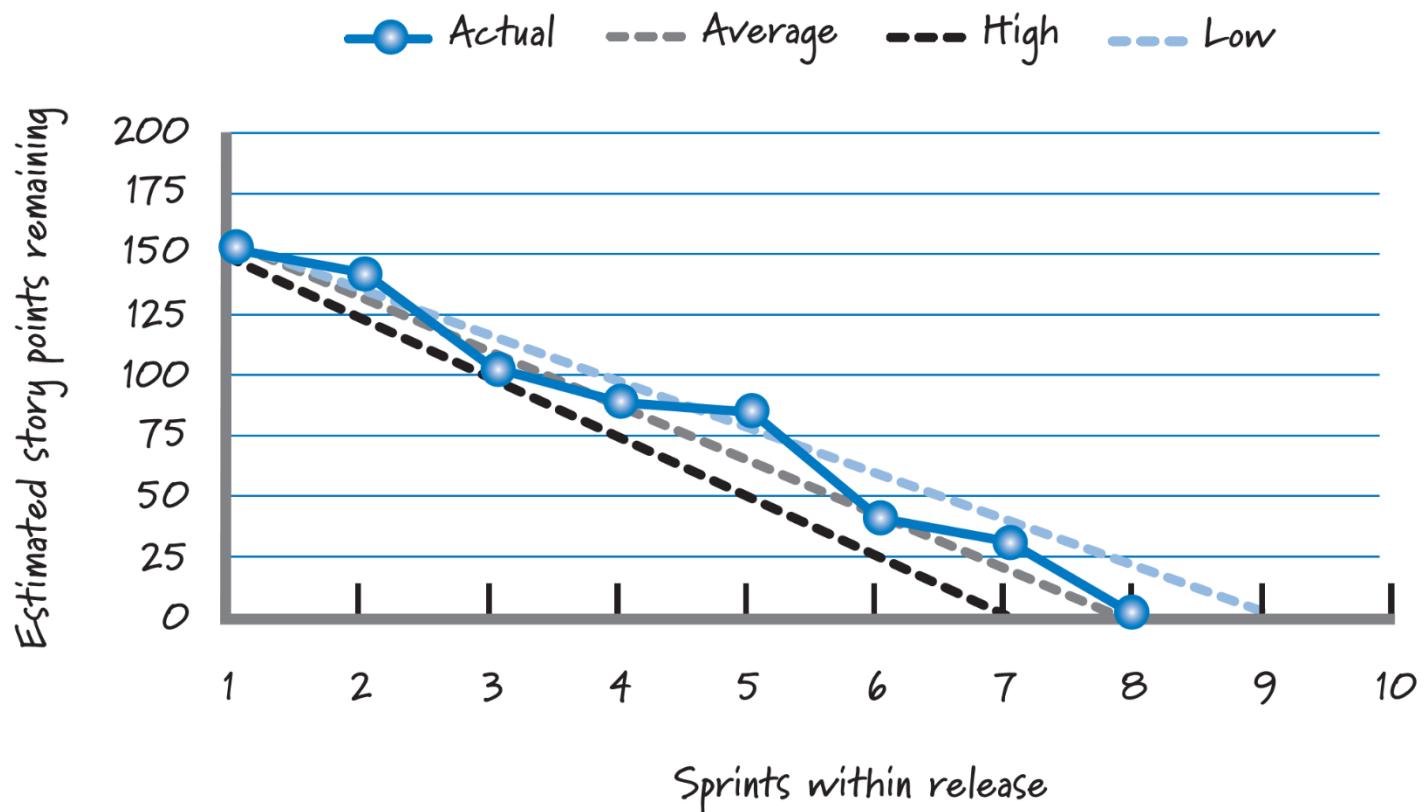
[Rubin 2012]

Fixed-Scope Release Planning



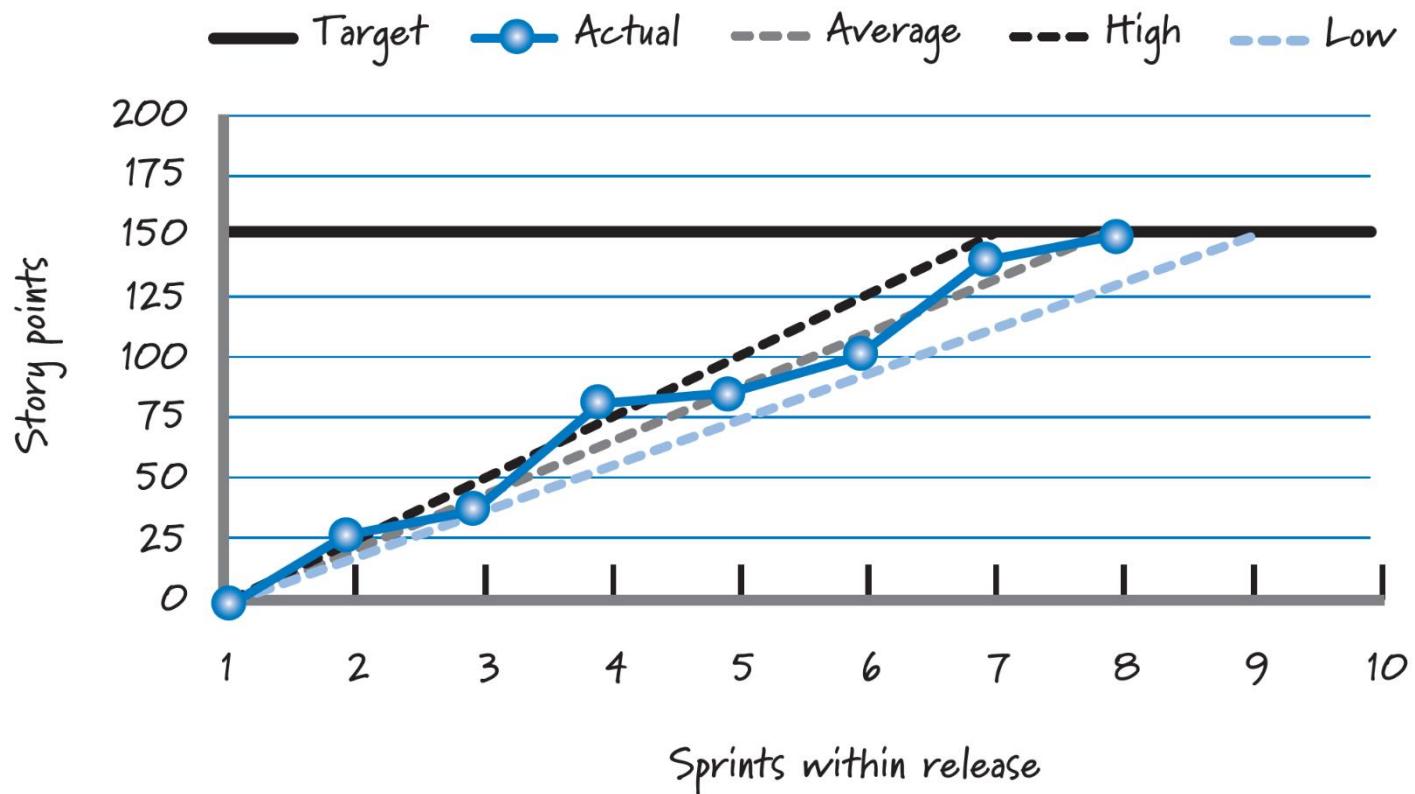
[Rubin 2012]

Fixed-Scope Burndown Chart



[Rubin 2012]

Fixed-Scope Burnup Chart



[Rubin 2012]

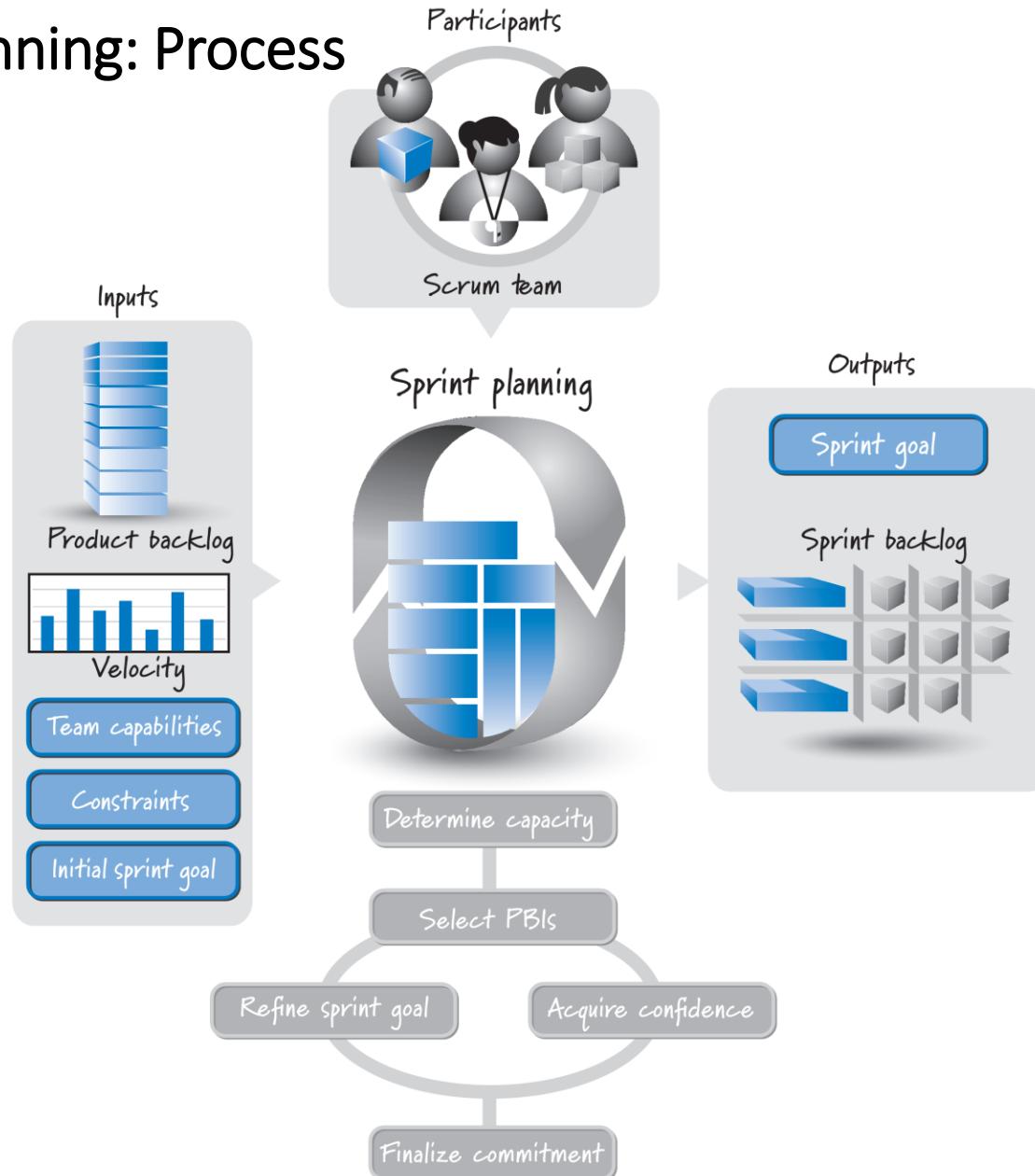
Sprint planning



Sprint Planning

- **When?**
 - At the beginning of each sprint.
- **How long?**
 - Four to eight hours (for a two-week to month-long sprint).
- **By whom?**
 - The full Scrum team:
 - The **product owner** shares the **initial sprint goal**, presents the **prioritized product backlog**, and **answers** any questions the team might have on PBIs.
 - The **development team** works diligently to determine what it can deliver and then makes a realistic commitment (forecast) at the end of sprint planning.
 - The **Scrum Master**, acting as the coach, observes the planning activity, asks probing questions, and facilitates to help ensure a successful result.
 - Also, **challenges the team's commitment** to ensure that it is realistic.

Sprint Planning: Process

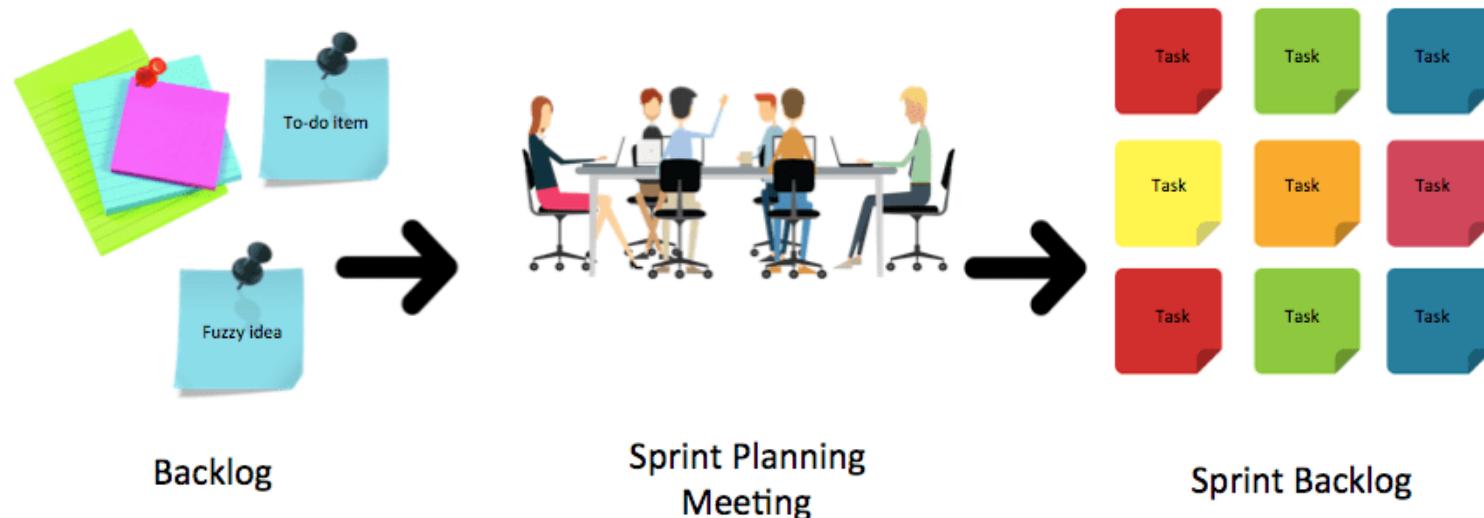


Sprint Planning: Process

- **Inputs:** Product backlog, Team velocity, Constraints, Team capabilities, Initial sprint goal.
 - **Initial sprint goal** is presented by the product owner:
 - It might be a specific **set of high-priority** product backlog items: “I’d really like to get the top five product backlog items done this sprint,” or
 - It might be in the form of a more **general notion**: “At the end of this sprint I want a typical user to be able to submit a simple keyword query.”
- **Outputs:** Sprint goal and Sprint backlog, which collectively form the commitment.

Sprint Planning: Process

- **Activities:** Determine capacity, Select PBIs, Refine sprint goal, Acquire confidence, Finalize commitment.
 - A **realistic commitment** is achieved through **collaboration** (and negotiation) between the product owner and the development team.
 - To acquire confidence in what it can accomplish, the development team will create a plan by **breaking** down each targeted PBI into a set of **estimated tasks**.
 - Teams typically follow a helpful rule of breaking down tasks so that no one task is more than **eight** hours of effort, although some might be a bit larger.

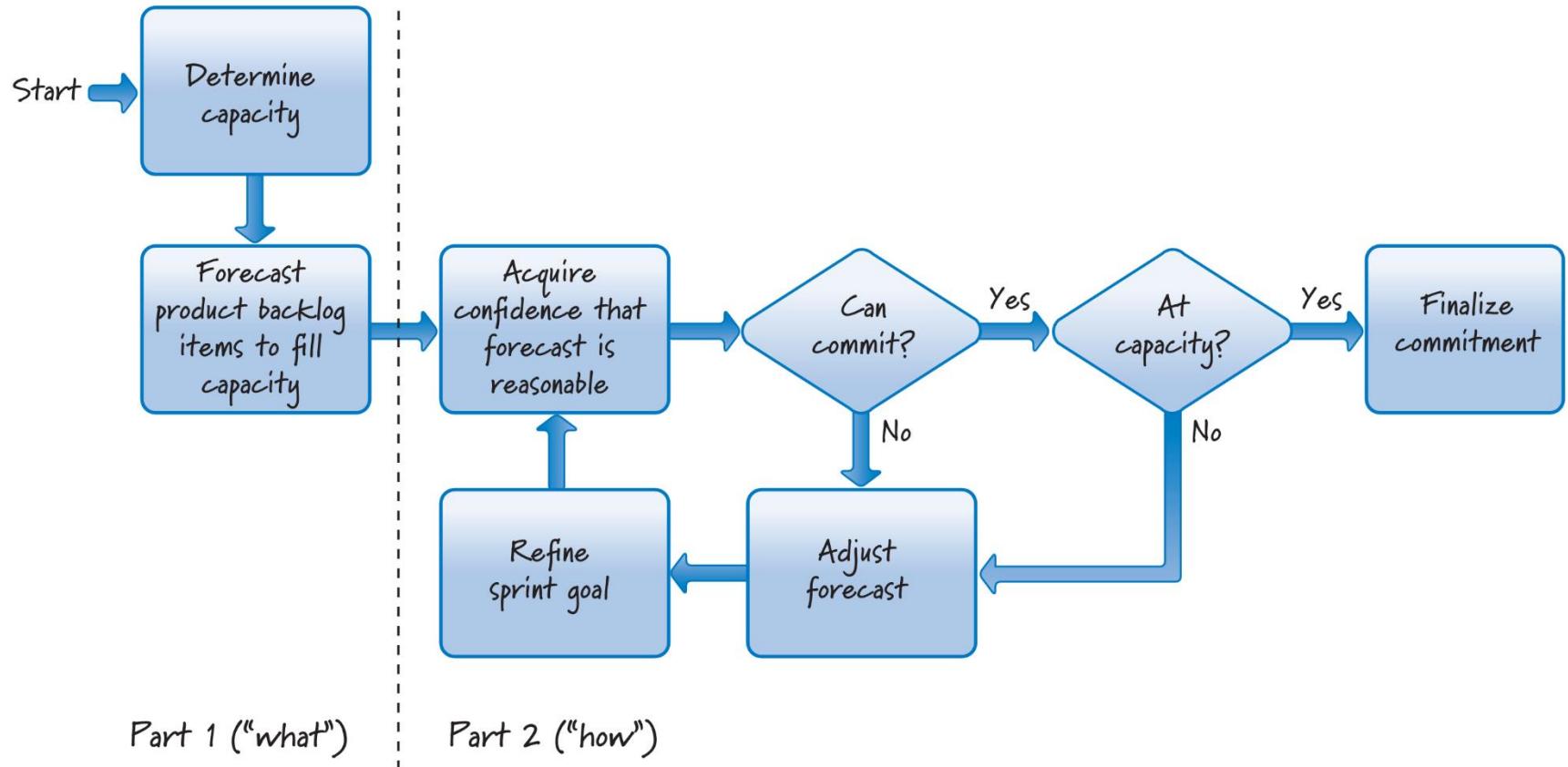


Sprint Planning: Inputs

Input	Description
Product backlog	Prior to sprint planning, the topmost product backlog items have been groomed into a <i>ready</i> state.
Team velocity	The team's historical velocity is an indicator of how much work is practical for the team to complete in a sprint.
Constraints	Business or technical constraints that could materially affect what the team can deliver are identified.
Team capabilities	Capabilities take into account which people are on the team, what skills each team member has, and how available each person will be in the upcoming sprint.
Initial sprint goal	This is the business goal the product owner would like to see accomplished during the sprint.

[Rubin 2012]

Two-Part Sprint Planning

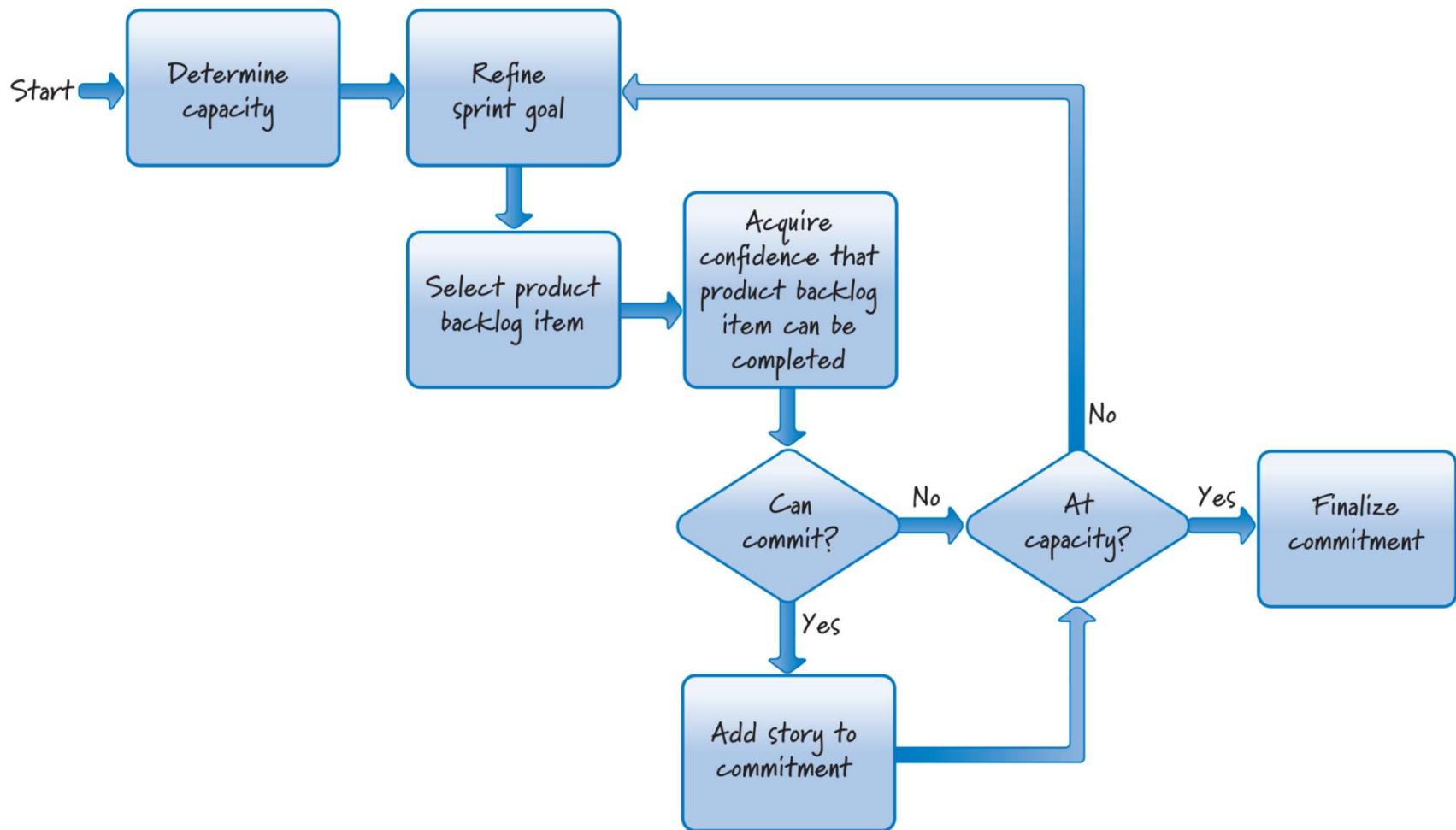


[Rubin 2012]

Approaches: Two-Part Sprint Planning

1. **Part 1 (the “what” part):** The team **determines its capacity** to complete work and then **forecasts the PBIs** that it believes it can deliver in the sprint.
 - So if the team believes it can accomplish 40 story points, it will select about 40 story points’ worth of work.
2. **Part 2 (the “how” part):** The team **acquires confidence** in its ability to complete the items that it forecast in part 1 by creating a plan.
 - Teams create this plan by **breaking** the product backlog items into a set of **tasks** and then estimating (in hours) the effort required to complete each task.
 - The team then **compares the estimate of task hours against its capacity**, in terms of hours, to see if its initial commitment was realistic.
 - If the team finds it has selected too much or too little, or has selected items that cannot be developed in the same sprint given one or more constraints, then:
 - The forecast is **adjusted**, and the **sprint goal is refined** (if necessary), to fit the available capacity and constraints.
 - When the team’s **forecast** is comfortably **within its capacity** range and constraints, it finalizes its commitment and **sprint planning is over**.

One-Part Sprint Planning



[Rubin 2012]

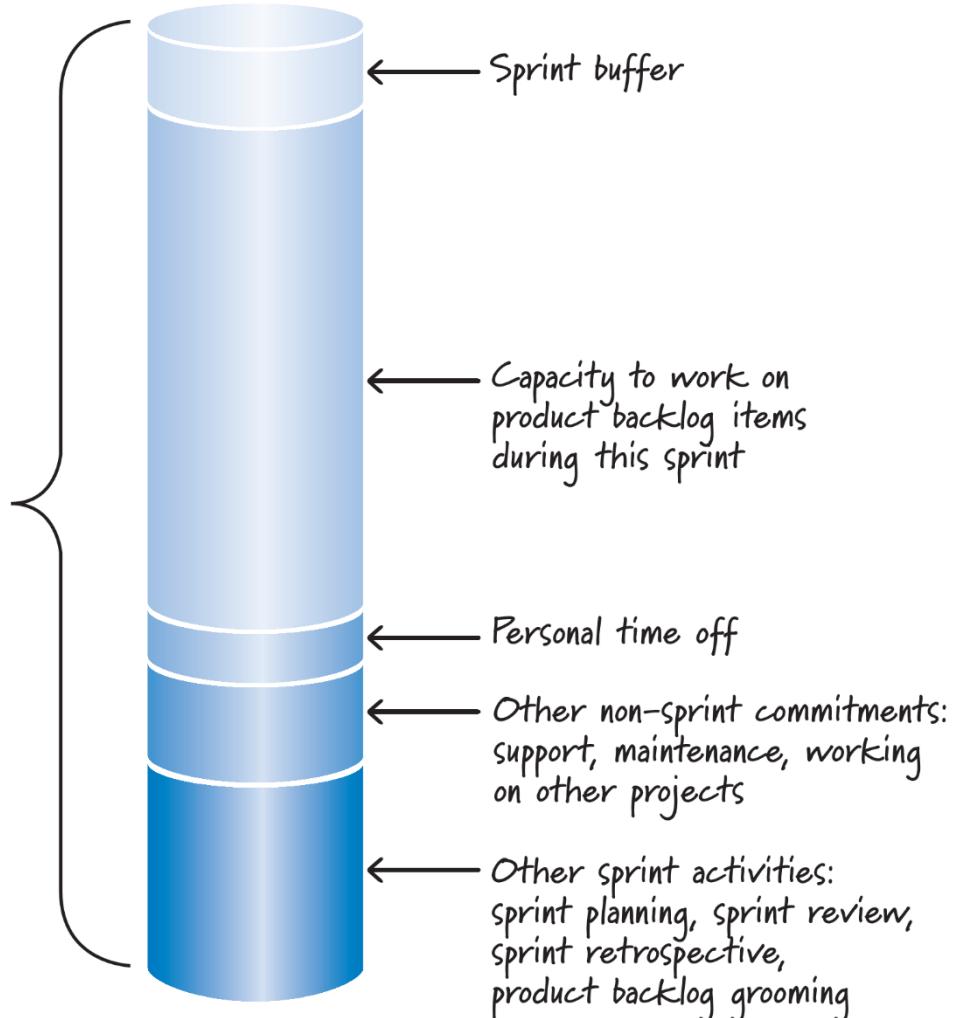
Approaches: One-Part Sprint Planning

- The one-part approach **interleaves** selecting a PBI and acquiring confidence that it can be delivered.
 1. The development team begins by **determining its capacity** to complete work.
 2. Based on available capacity, the **sprint goal is refined** if necessary.
 3. The team **selects** a PBI and then **acquires confidence** that the selected item will reasonably fit within the sprint.
 - Consideration should be given to other items already included in the team's evolving commitment.
 4. Steps 2 and 3 are repeated until the team is out of capacity to do any more work.

Determining Capacity

- Several factors reduce a team's capacity to work on PBIs during a sprint.

Total sprint capacity



Determining Capacity

- After considering the factors, what remains is the team's capacity.
- For instance, in a two-week (ten-day) sprint, we need to reserve:
 - One day collectively for sprint planning, sprint review, and sprint retrospective;
 - up to 10% of the time to assisting the product owner with PB grooming;
 - time for work outside the sprint, such as supporting the current product, maintaining another product, etc.;
 - time for the team members' organizational responsibilities, such as attending meetings, responding to emails, interruptions, etc.;
 - time off previously scheduled by team members;
 - some buffer (based on past experience) against things not going quite as planned.

Units of Measure for Capacity

- We can use either of the following as units of measure for capacity:
 - The same unit as the PBIs (typically **story points** or **ideal days**); or
 - the same unit as the sprint backlog tasks (**effort-hours**).



Units of Measure for Capacity: Story Points

- We usually express capacity in story points:
 - Therefore, determining capacity is the same as predicting our team's **target velocity** for the upcoming sprint:
 - Start with the team's long-term average velocity or the previous sprint's velocity as an **initial estimate** of its capacity/velocity for the upcoming sprint.
 - Consider whether the **upcoming sprint might differ** from typical or previous sprints, and adjust the predicted velocity/capacity accordingly.

Units of Measure for Capacity: Effort Hours

- An alternative way to express capacity is in **effort-hours**.
- The calculation of a team's effort-hour capacity to perform task-level work for a sprint is typically carried out as follows:
 1. Team members express **how many days** they have available to work on the upcoming sprint (personal time off is deducted).
 2. Team members determine how much time (in days) to **reserve for other Scrum activities**.
 3. Team members determine how **many hours** per day they could **dedicate to work** in this sprint by deducting the overhead for their other non-sprint commitments.
 4. A capacity is calculated for **each of the team members** based on the above figures; the sum of all capacities yields the total team capacity in effort-hours.
 5. The team should **deduct a reasonable buffer** from the calculated capacity to obtain a more reasonable figure, thus preparing against mishaps.

Determining Effort-Hour Capacity: Example

Person	Days Available (Less Personal Time)	Days for Other Scrum Activities	Hours per Day	Available Effort-Hours
Jorge	10	2	4–7	32–56
Betty	8	2	5–6	30–36
Rajesh	8	2	4–6	24–36
Simon	9	2	2–3	14–21
Heidi	10	2	5–6	40–48
Total				140–197

[Rubin 2012]

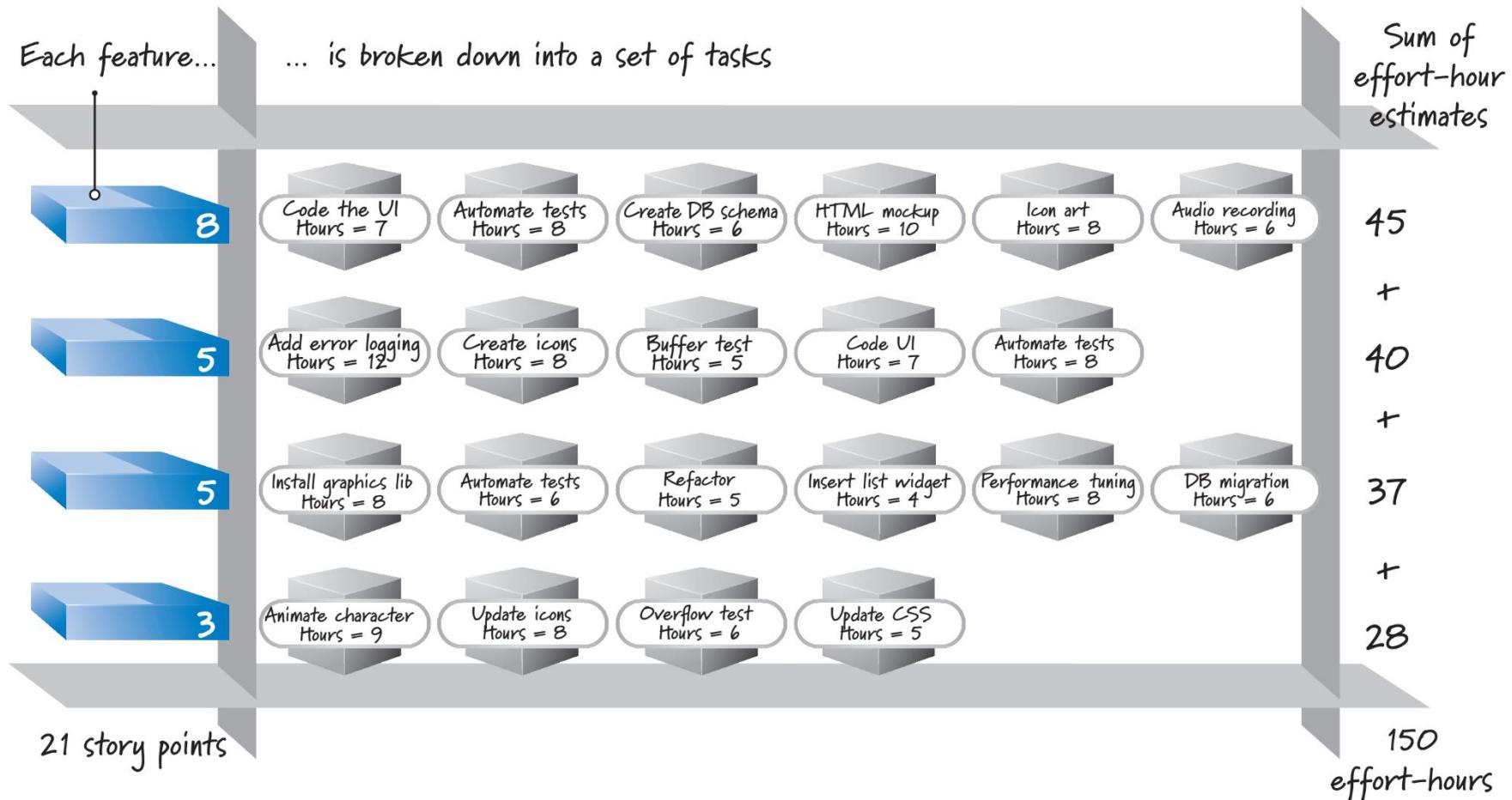
Selection of PBIs

- Selection of PBIs can be done in several ways:
 - If we have a **sprint goal**, we would select PBIs that **align with that goal**.
 - The goal, however, can be refined during sprint planning as the participants work together to determine what can realistically be delivered.
 - If there is no formal sprint goal, our default is to select items from the **top of the product backlog**.
 - If the team were not able to commit to the next-highest-priority item (perhaps due to a skills capacity issue), it would select the next appropriate higher-priority PBI that looks as if it can be completed within the constraints.
 - **Never start what you cannot finish**; If the next PBI is too big to complete in the sprint (given the other PBIs that we have already agreed to complete):
 - Try to break down the next PBI into two or more smaller items, each of which would be **valuable** to our customers, or
 - consider working on another item that *can* be completed.

Acquiring Confidence: Sprint Backlog

- One way to acquire confidence is to **use predicted velocity** to see if the commitment is realistic; but this is usually not enough.
- The necessary level of confidence is usually obtained **by breaking down** the PBIs into the tasks that are required to complete them.
 - The result is the **sprint backlog**.
- In Scrum, team members are typically not assigned to tasks during sprint planning (this is decided during sprint execution).
 - However, we need to at least consider our **skills capacity** for completing tasks which require special skills, or we could make a bad commitment.
 - For this reason, you can note on each task who is **the person most likely to work** on that task.

Sprint Backlog



[Rubin 2012]