

# Estimation and Velocity

CEN 4010 Intro to Software Engineering

Professor Alex Roque

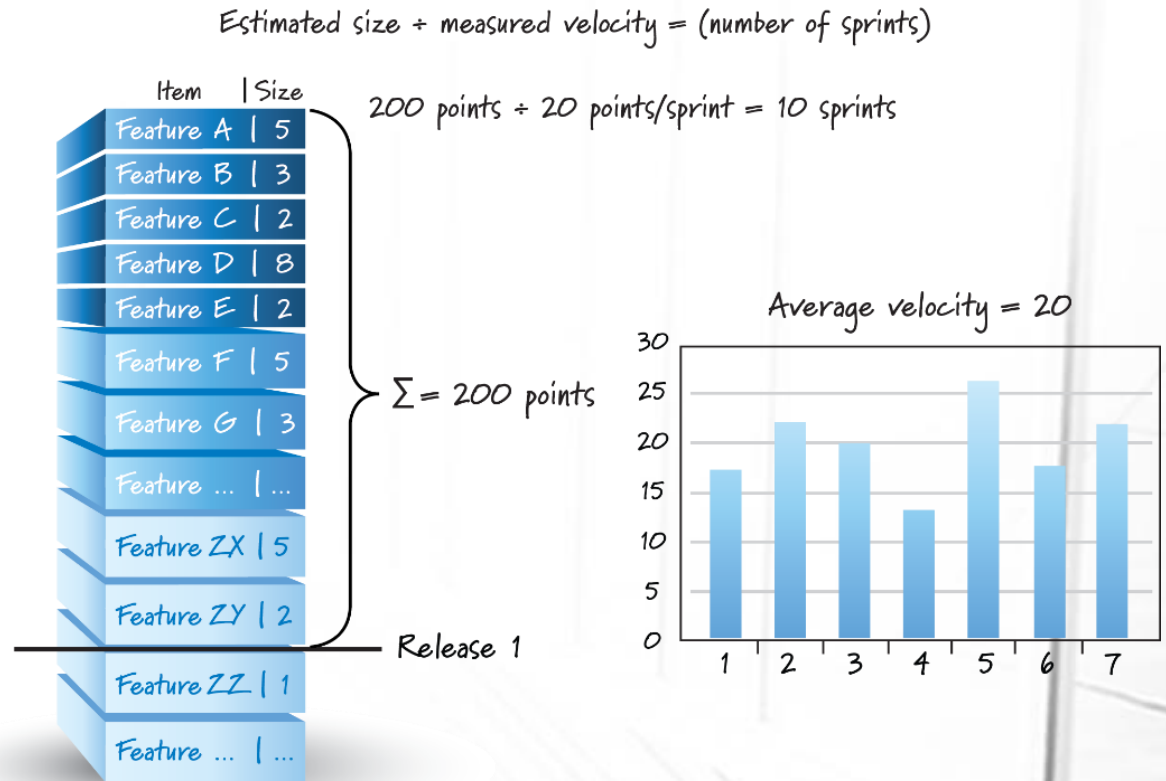
# Overview: Estimation and Velocity

- Common Development Project Questions
  - How many features will be completed?
  - When will we be done?
  - How much will this cost?
- Scrum's answer involves:
  - **Estimating the size** of what is being built
  - Measuring the **velocity** (rate) the team can get work done
- With the **size** and **velocity** information we can derive the likely development **duration** and **cost**



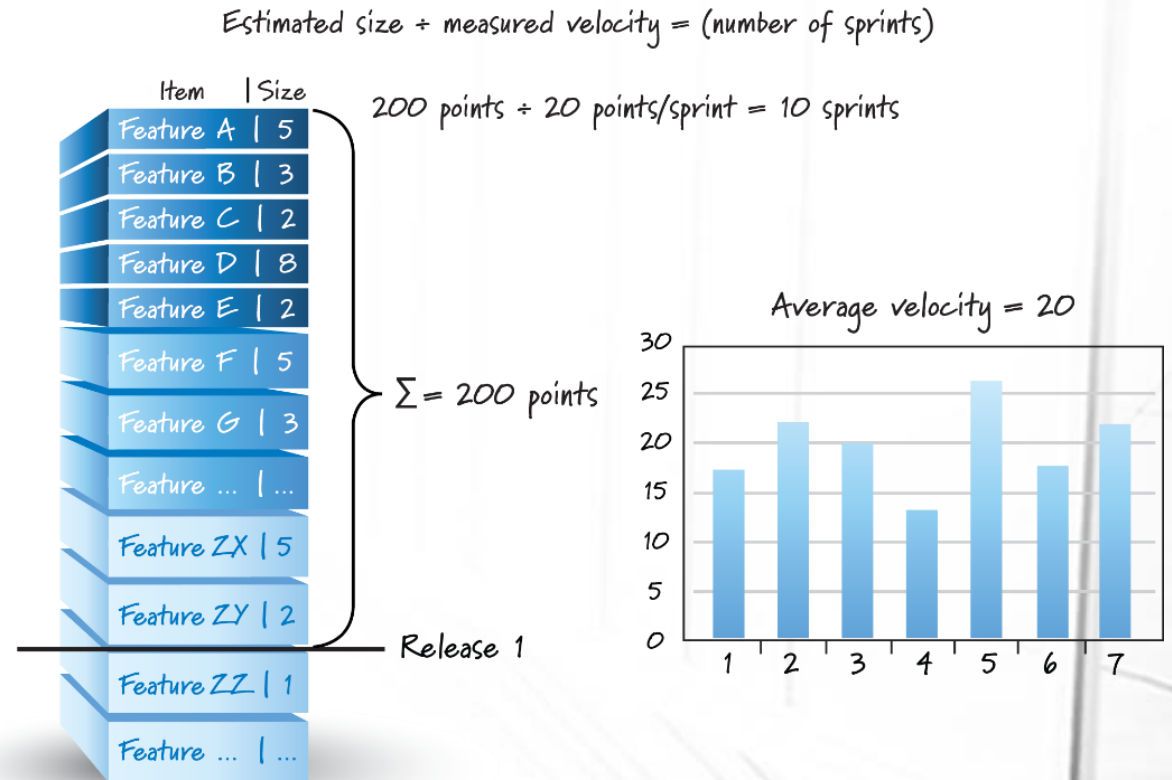
# Overview – Estimation and Velocity

- Once individual PBI size is **estimated**, we can add up the total PBI for the release (200 story points) which includes **several sprints**
- **Velocity** is a measure of how much work the team can accomplish in a **sprint**
- At the end of a sprint we add up the points for the PBIs that were **completed** (done)



# Overview – Estimation and Velocity

- This point total becomes the **velocity** for the sprint
- Velocity will fluctuate across sprints yielding an **Average Velocity** (20 points)
- **Duration** can now be calculated by dividing the **size** by the **velocity** (10 sprints)



# What and When We Estimate

- Story points are one technique for doing PBI estimates
- We also need to estimate at varying levels of granularity throughout planning & grooming
  - Portfolio Backlog estimates
  - Product Backlog estimates
  - Sprint Backlog (PBI) estimates
    - **Ideal/Effort Hours** is not necessarily same as **elapsed**
    - “**Ideal**” is usually to describe an effort were you are not interrupted
    - Sometimes, **optional** if team is “good/seasoned”
    - If optional, then **Velocity** is **number of PBIs done**, not the number of story points done

# What and When We Estimate

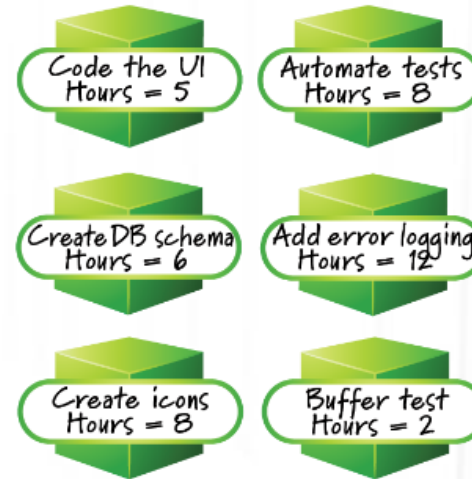
Portfolio backlog



Product backlog



Sprint backlog tasks

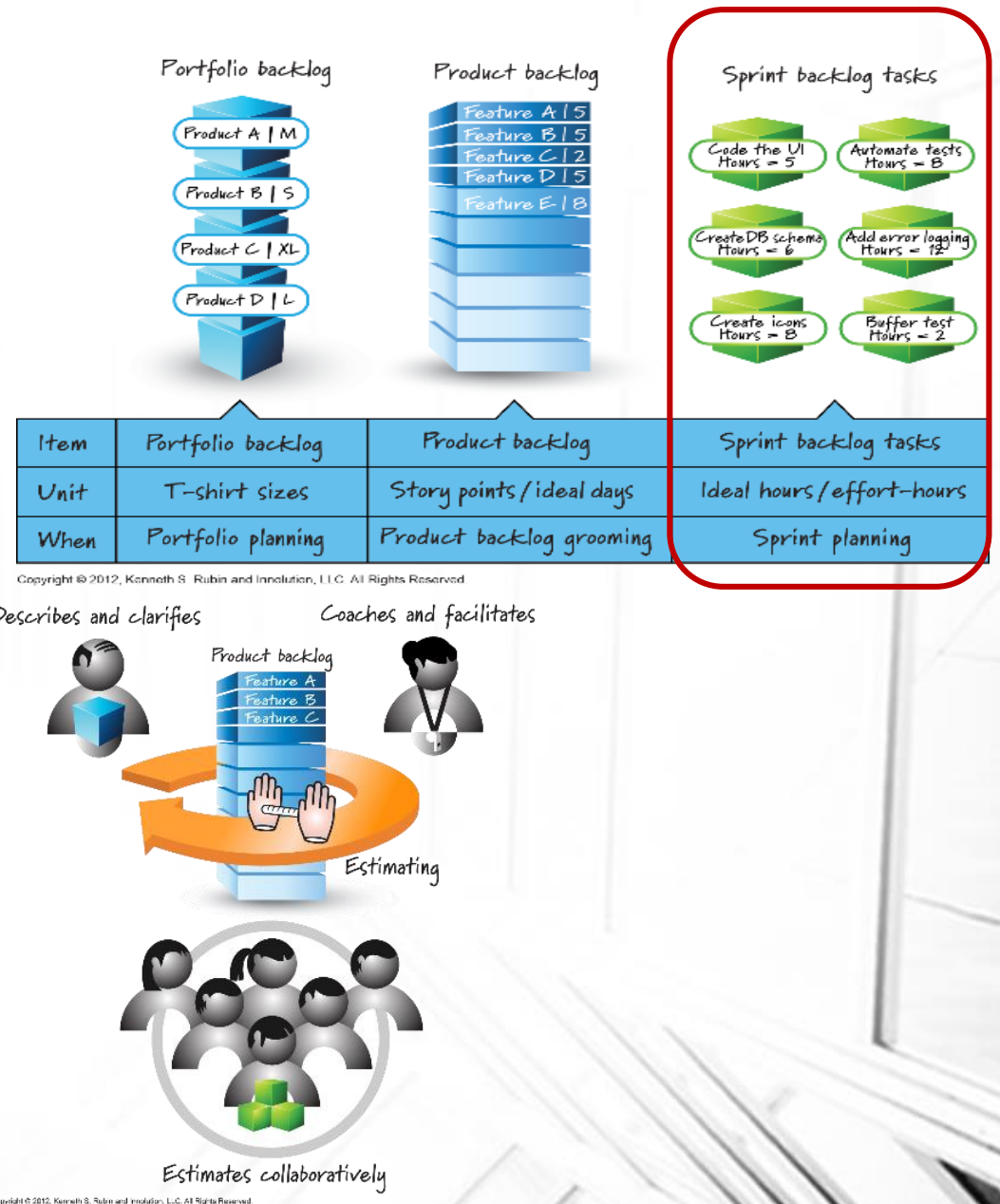


Item	Portfolio backlog	Product backlog	Sprint backlog tasks
Unit	T-shirt sizes	Story points/ideal days	Ideal hours/effort-hours
When	Portfolio planning	Product backlog grooming	Sprint planning



# PBI Estimation Concepts

- Estimate as a Team
  - Traditional projects usually have Proj/Prod manager, architect, or lead developer do initial size estimation
  - Scrum Principle – The people who do the work (Dev. Team) collectively provide the estimates
    - PO participates by describing PBIs and answering questions
    - ScrumMaster's role is to help coach & facilitate



# Estimating as a team

- This is a central concept to agile, but why is it so important?
- What are the dangers of having one person do the estimating, regardless of their role?
- What should your team be doing when estimating the work to be done?





# PBI Estimation Concepts

- Estimates Are Not Commitments
  - Sounds “crazy”, but it is important that we do not treat them as such
  - Commitments promote a “**sandbag**” or “**over-zealous**” mindset for team members
  - We want **realistic** measures, not inflated or understated to impress



PBI estimating concepts

Estimate as a team

Estimates are not commitments

Focus on accuracy, not precision

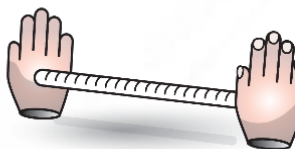
Use relative versus absolute sizes

Copyright © 2012, Kenneth S. Rubin and Innolution, LLC. All Rights Reserved.

Original size estimate



Revised estimate based on commitment



Copyright © 2012, Kenneth S. Rubin and Innolution, LLC. All Rights Reserved.

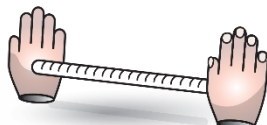
# PBI Estimation Concepts

- Focus on Accuracy, Not Precision
  - **Precise:** 10,275 hours;  
\$132,865.87 = **wasteful**
  - **Time sink** estimating something we don't fully understand
- Invest enough effort to get a **good-enough, roughly right estimate**
- There exists a **point of diminishing return**

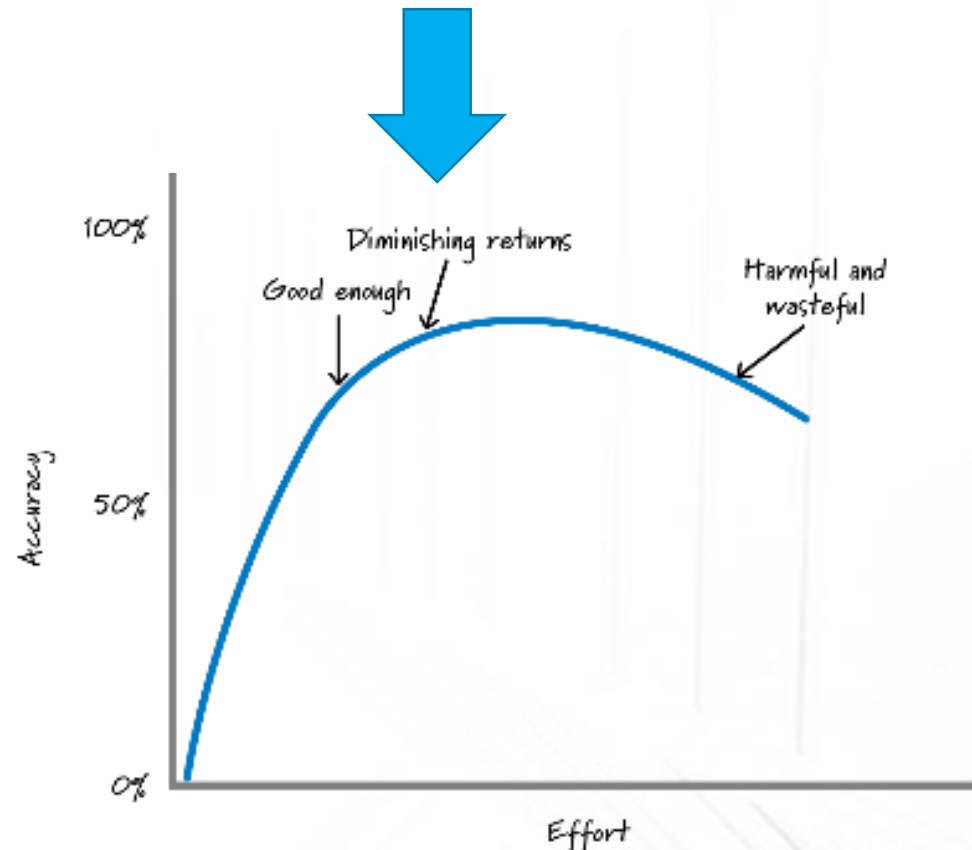
Original size estimate



Revised estimate based on commitment



Copyright © 2012, Kenneth S. Rubin and Innovation, LLC. All Rights Reserved



Copyright © 2012, Kenneth S. Rubin and Innovation, LLC. All Rights Reserved

# PBI Estimation Concepts

- Use Relative Versus Absolute Sizes
  - **Size** is determined by comparing a PBI to other PBIs
  - Most of us are better at relative size estimation than absolute
- PBI Estimation Units (no standard unit)
  - **Story Points (70%)**
    - Measures the **Bigness** or **Magnitude** of a PBI
      - **Complexity & Size** factors considered
      - Comparing User Stories (PBIs) with each other
  - **Ideal (effort/person) Days (30%)**
    - Not the same as elapsed days (example: any sport)
    - Risk of misinterpretation by stakeholders – two ideal days is not necessarily the same as two calendar days

It's just  
Root  
Beer!

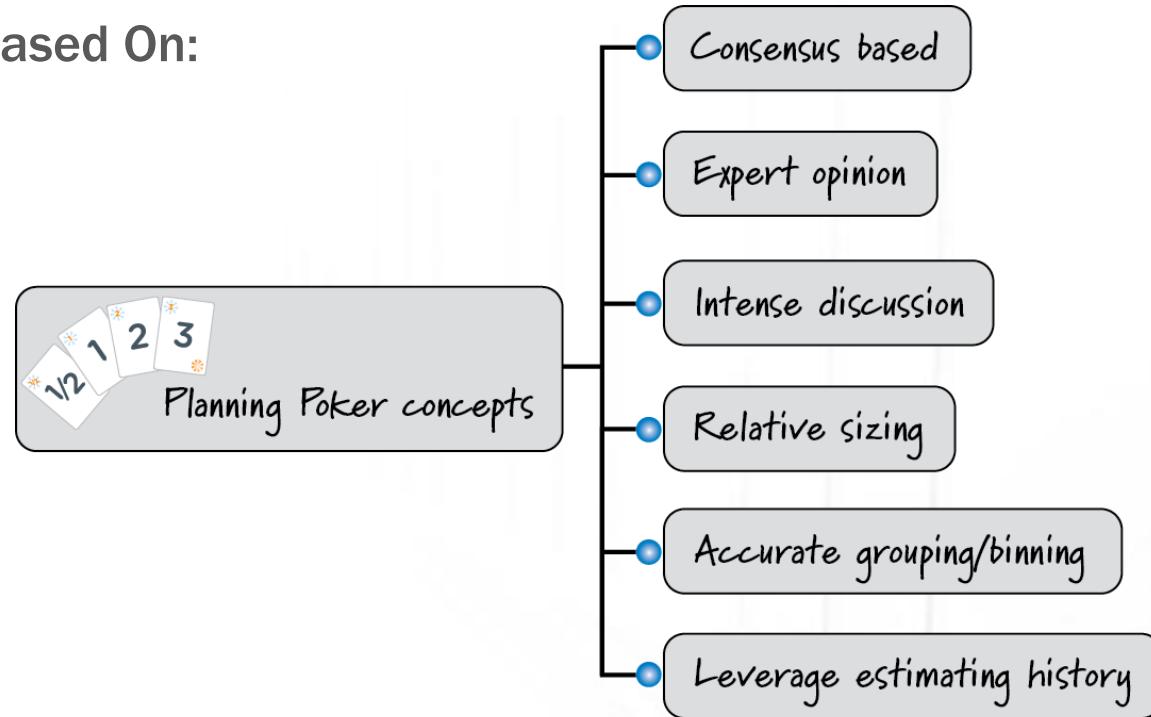


Copyright © 2012, Kenneth S. Rubin and Associates, LLC. All Rights Reserved.

# Planning Poker

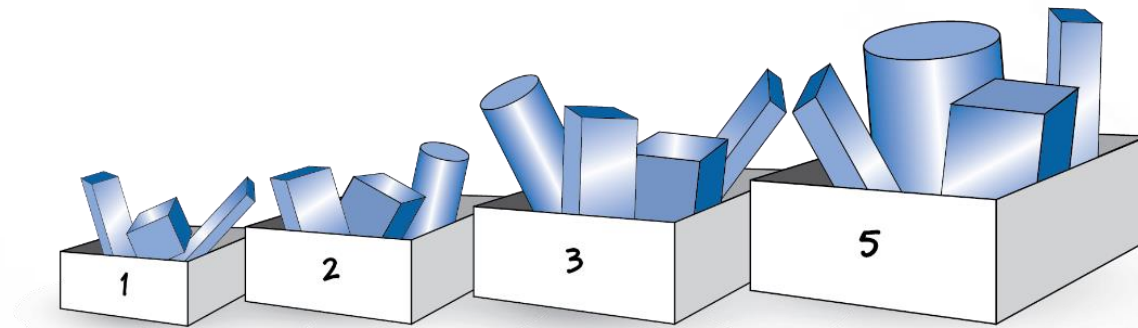
- (Fun!) Technique for sizing PBIs Based On:

- Consensus
- Expert Opinions
- Intense Discussion
- Relative Sizing
- Accurate Grouping/Binning
- Leverages Estimating History



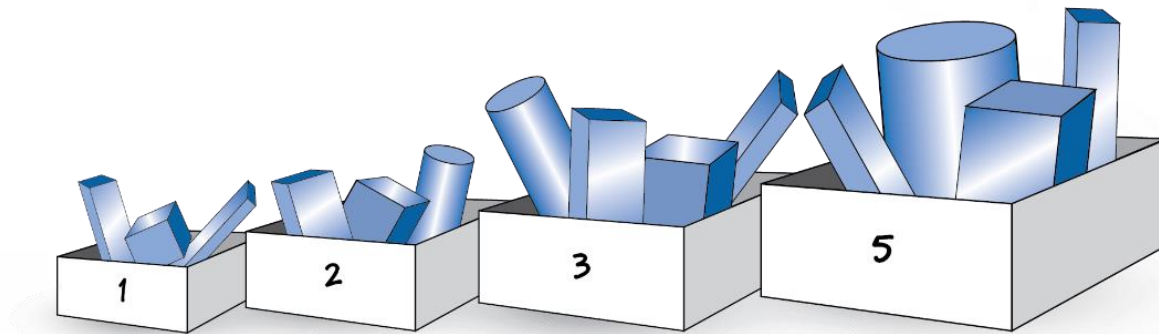
# Planning Poker

- **Goal:** Accuracy over Precision
- **Common Set of Cards** (could be others) uses a modified Fibonacci Sequence,
  - 1, 2, 3, 5, 8, 13, 20, 40, and 100
  - Alternative based on binary: 1,2,4,8,16,32...
- You are using relative comparisons of the work items to each other



# Planning Poker

- Group (bin) together **like-sized PBIs** and assign them same number on our scale
- Full Scrum team participates
  - Product Owner presents, describes, & clarifies
  - ScrumMaster coaches & keeps game moving
  - Dev. Team collaboratively generates estimates





# Planning Poker

- **Playing the game**

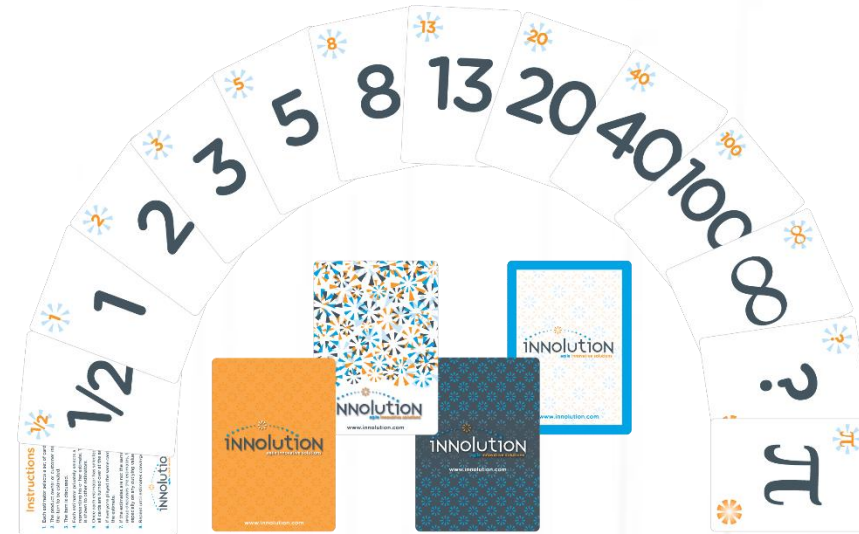
1. Each Dev. Team member has a deck of cards
2. PO selects PBI to estimate, discuss
3. Each estimator **privately** selects a card
4. Estimates are **simultaneously** exposed
5. Exposed estimates:
  - All have same card – consensus, becomes PBI estimate
  - Different cards – engage in discussion to expose assumptions and misunderstandings (ask high and low estimators to explain)
    - Return to Step 3 until **consensus is reached**
    - Consensus is usually reached within 2 to 3 rounds



**This deck includes other cards including  $\frac{1}{2}$ , infinity, question, and pi. Some decks also include a zero card (meaning too small or already done)**

# Planning Poker

- Benefits
  - Consensus estimates usually better than one person
  - Individual knowledge that is not shared by everyone on the team can be discussed
  - The PBI discussion is really helpful and valuable



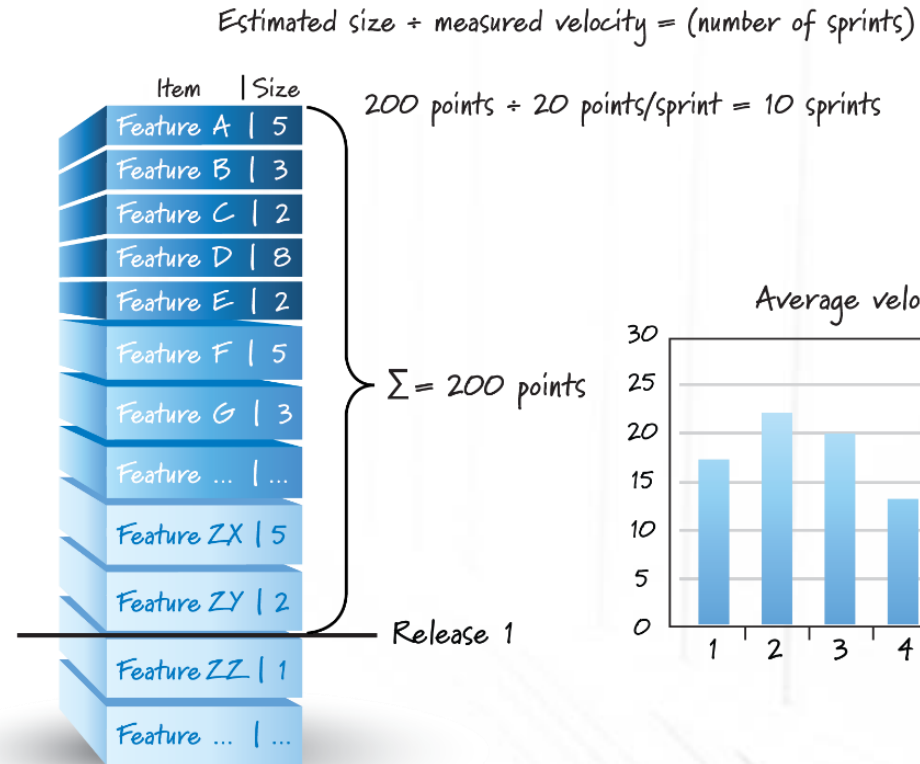
Copyright © 2012, Kenneth S. Rubin and Innovation, LLC. All Rights Reserved.

This deck includes other cards including  $\frac{1}{2}$ , infinity, question, and pi. Some decks also include a zero card (meaning too small or already done)

# What is Velocity?

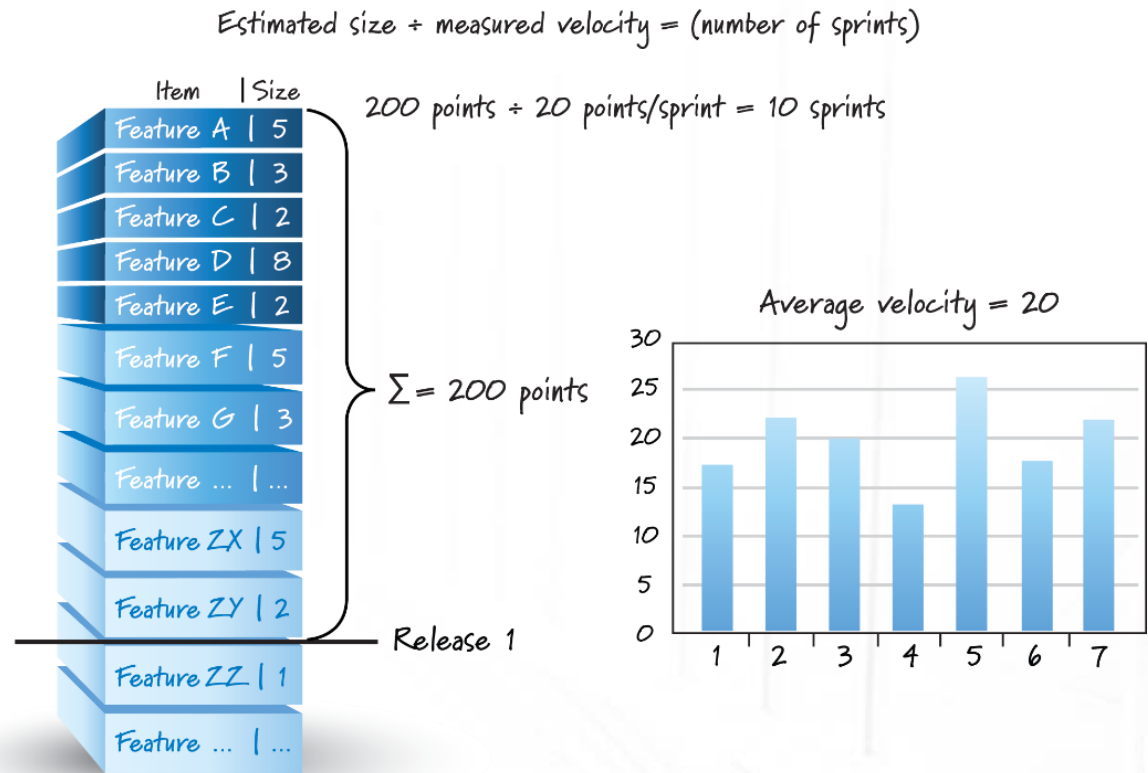
- **Velocity**

- Is the amount of work completed each sprint
- Is calculated by **adding size of completed PBIs** by end of sprint
  - PBIs are either done or not
- Measures **output** (size), not **outcome** (value)



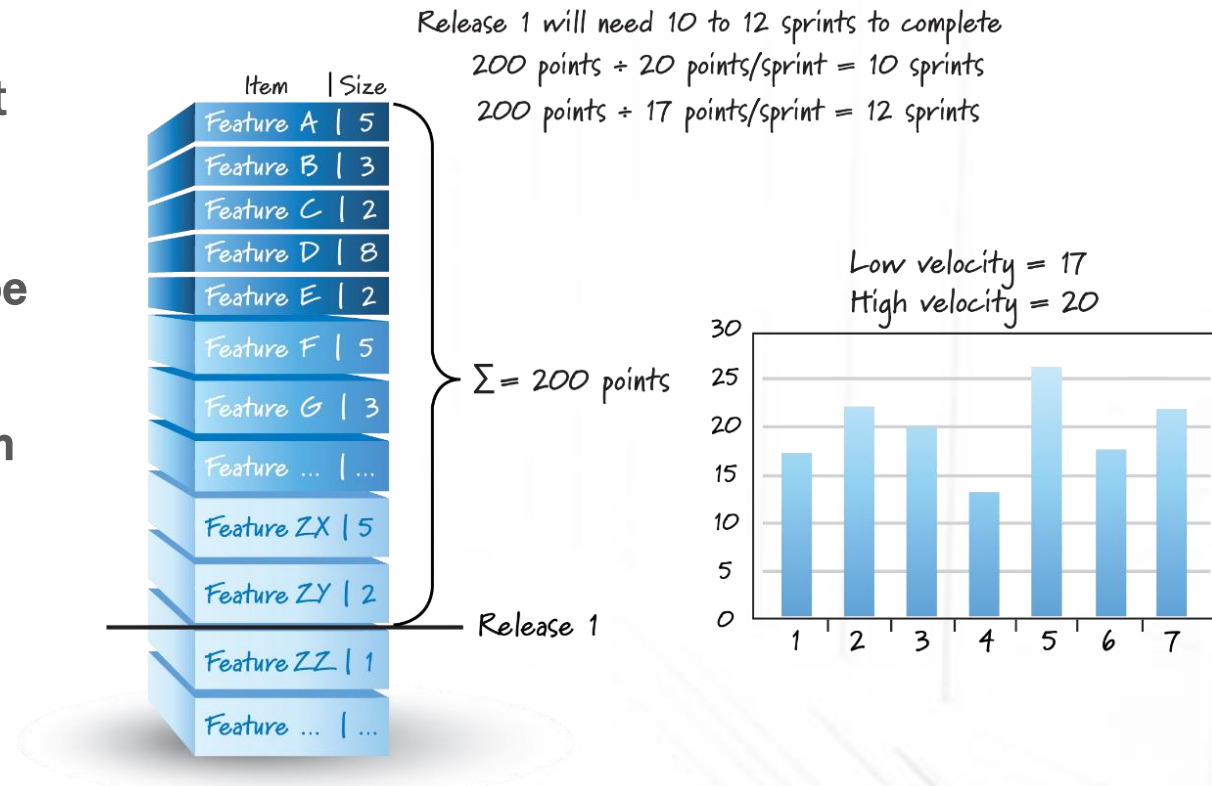
# What is Velocity?

- **Velocity**
  - Is essential for Scrum planning – both **release-level** and **sprint-level**
  - Is a diagnostic metric the team can use to evaluate and improve its use of Scrum
- **PBI size** does not necessarily = most valuable to PO



# Calculate a Velocity Range

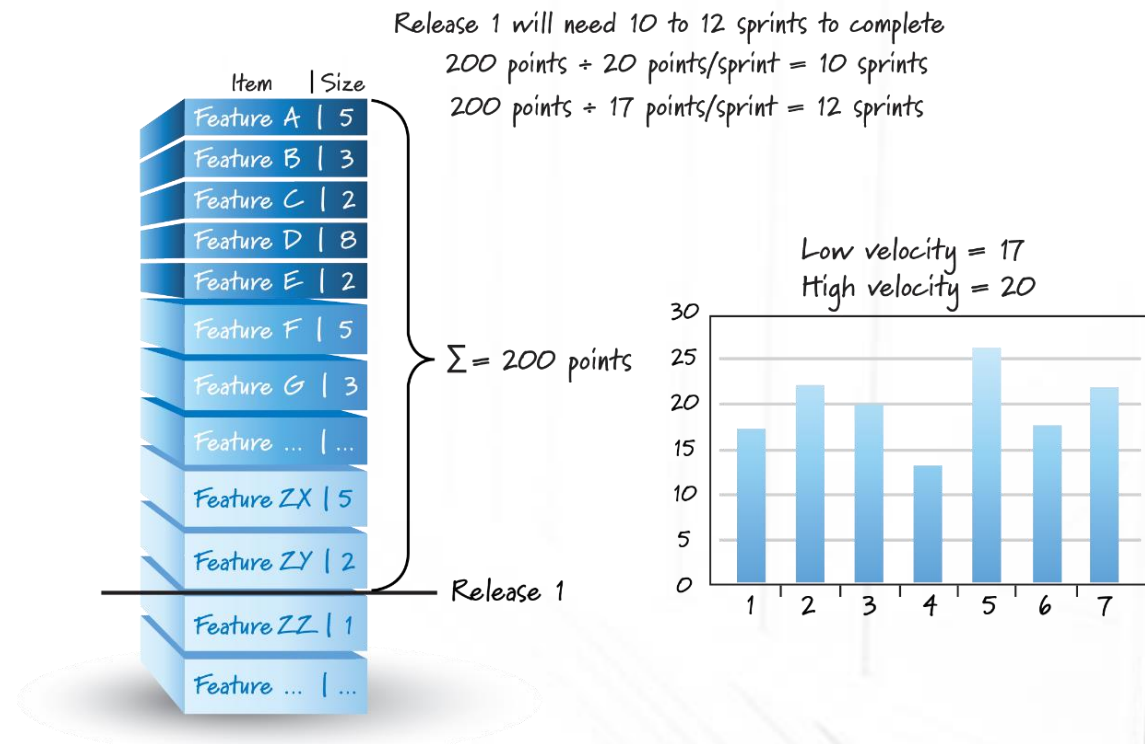
- Using a **Velocity Range** is a good way to be more accurate without being overly precise
  - Problem: Questions about **how much can be done, when will it be done, and what will it cost** are asked at the start of a project when the **least is probably known for preciseness**
  - The range indicates **10 to 12 sprints**





# Forecasting Velocity

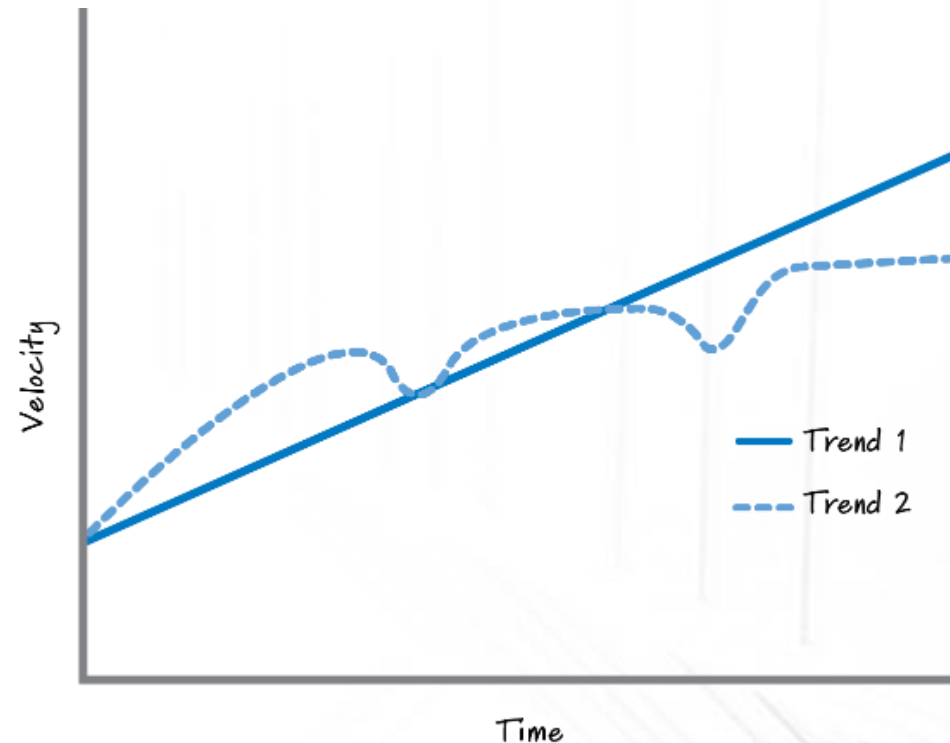
- Long-lived teams make it easy to determine velocity (range)
- New teams not so...
  - Have the team perform one sprint plan to determine PBIs it thinks it can complete; that becomes its initial velocity
  - Could have team perform two sprints and then create a high/low range for initial velocity
  - Once team actually performs a sprint a real velocity would be known and that should replace the forecast





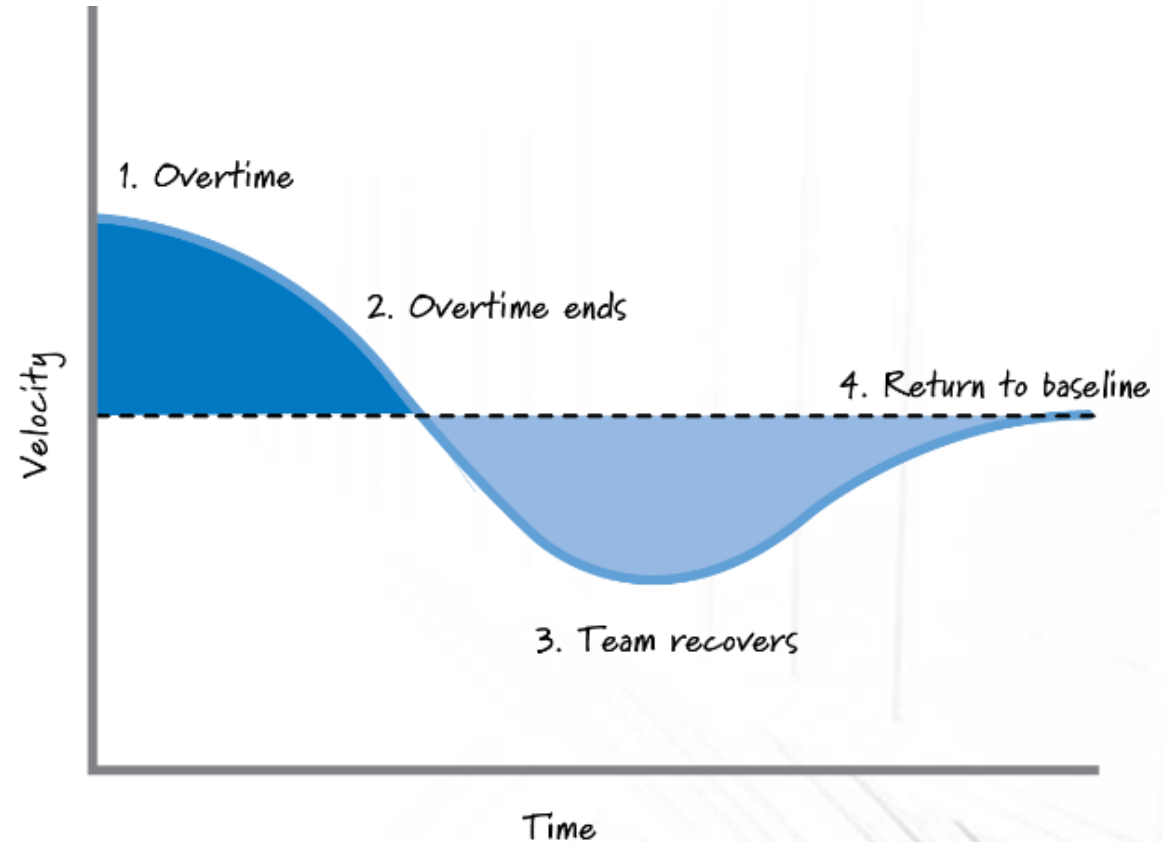
# Affecting Velocity

- A team's velocity cannot continue to increase **forever** (Trend 1 in top graph)
- Velocity fluctuates and at some point it could **plateau** (Trend 2 in top graph)
- New tools, increased training, new team members can have a **positive effect** on velocity



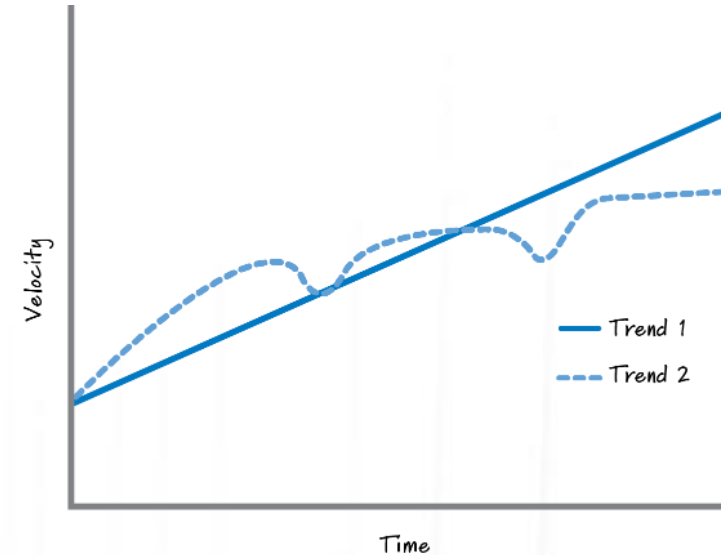
# Affecting Velocity

- New/additional “anything” usually causes a **dip in velocity** until the team adapts to the “new” normal
- **Overtime** (bottom graph) is usually thought of as a way to increase velocity which it might in the short run but over time it leads to lower velocity and product quality

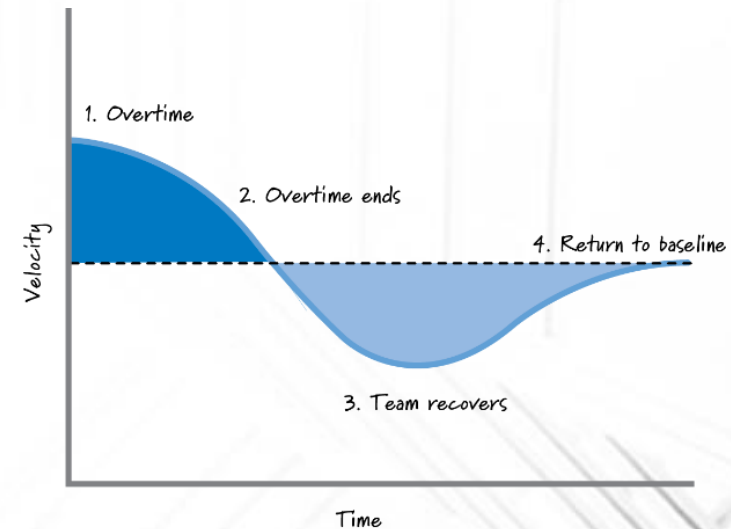


# Using Velocity

- Velocity is used as a key planning tool and for a team's diagnostic metric
- Velocity should **not** be used as a performance metric to attempt to judge team productivity
  - Using it this way can motivate wasteful and dangerous behavior
  - Teams typically do not use a **common baseline** across teams for determining PBI size
  - Causes “gaming” the system and **point inflation**



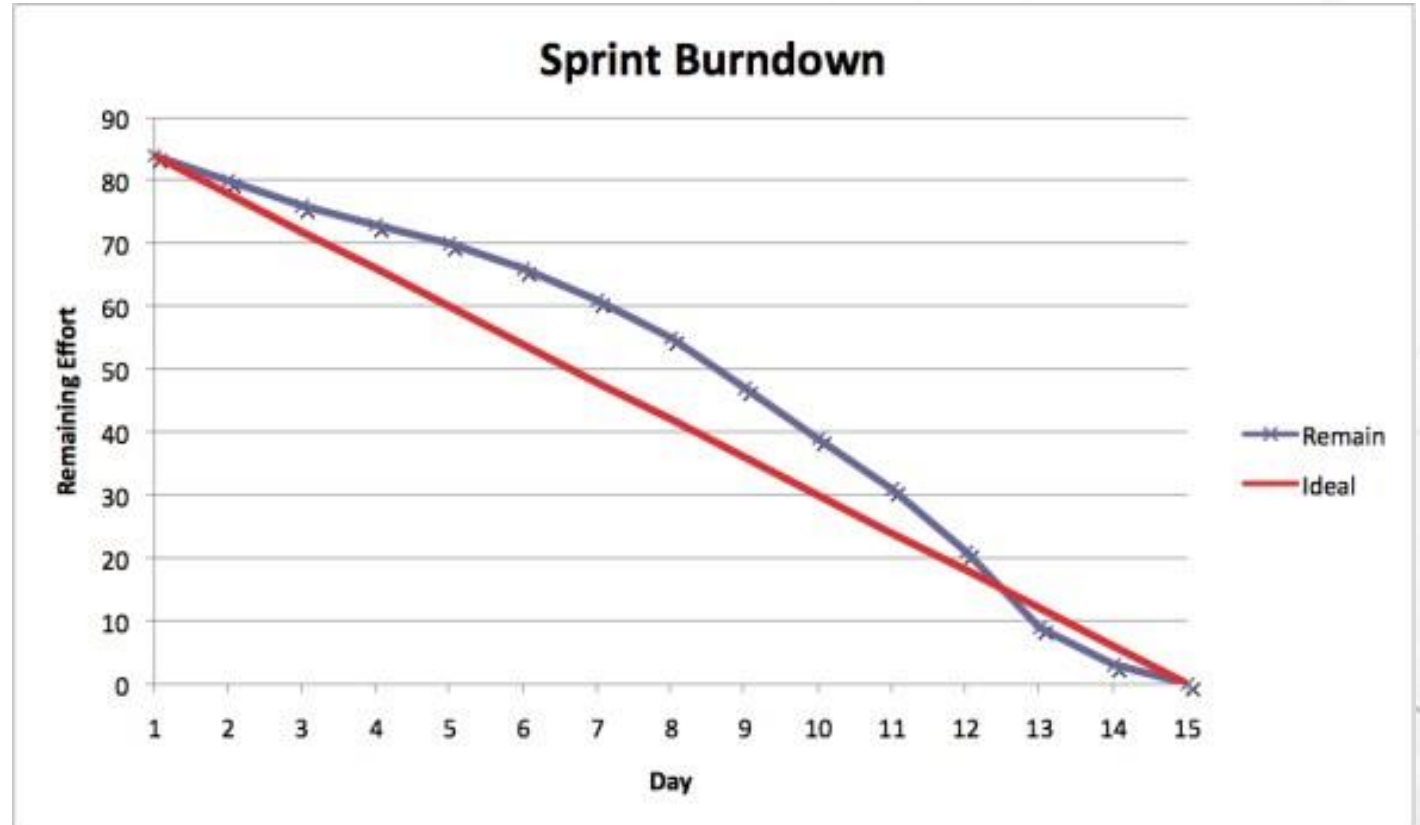
Copyright © 2012, Kenneth S. Rubin and Innolution, LLC. All Rights Reserved.



Copyright © 2012, Kenneth S. Rubin and Innolution, LLC. All Rights Reserved.

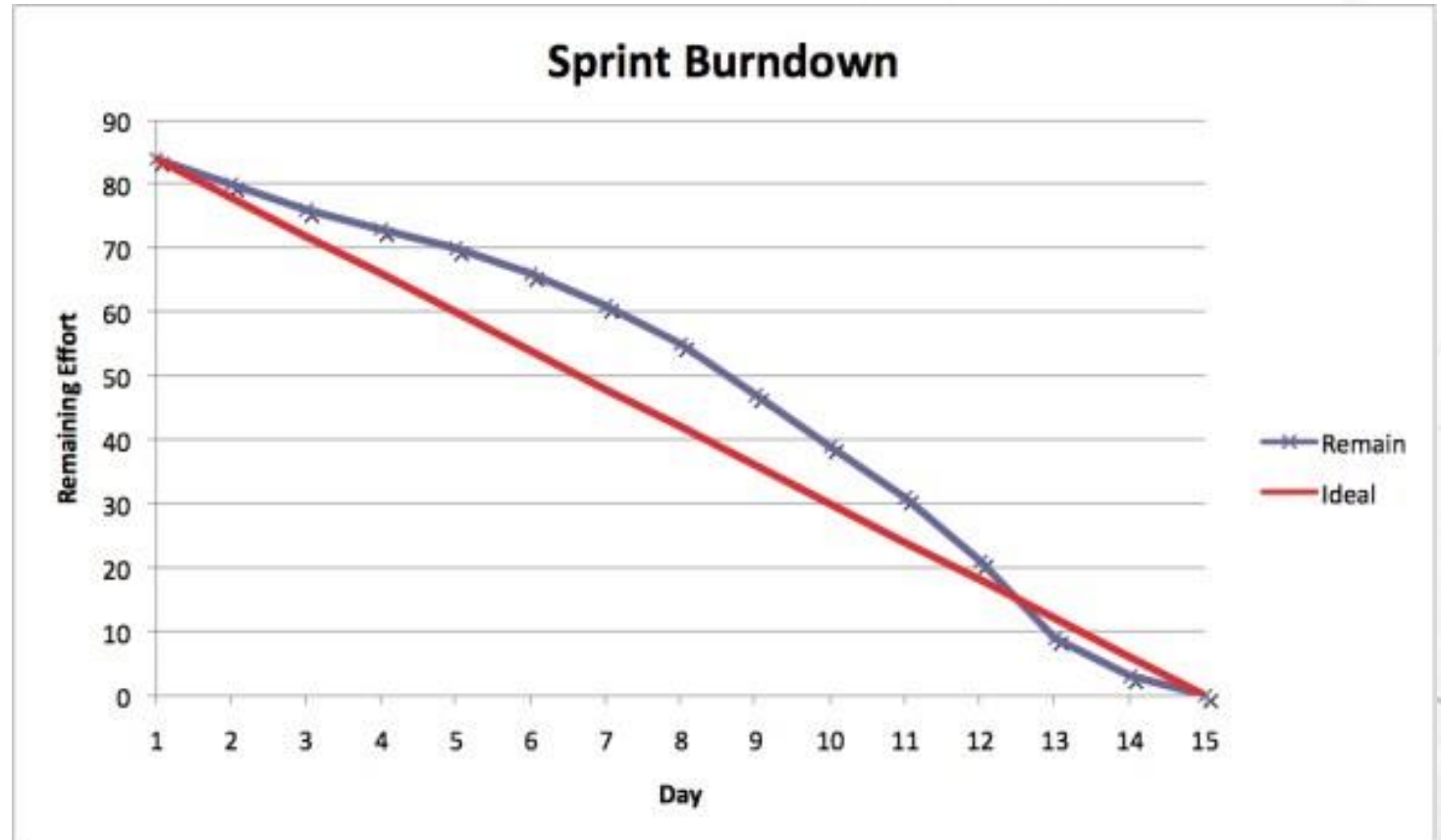
# Sprint Burndown

- Shows the days of the Sprint and the effort left to work on each day.
- Tells us how much work is left to be done by the group



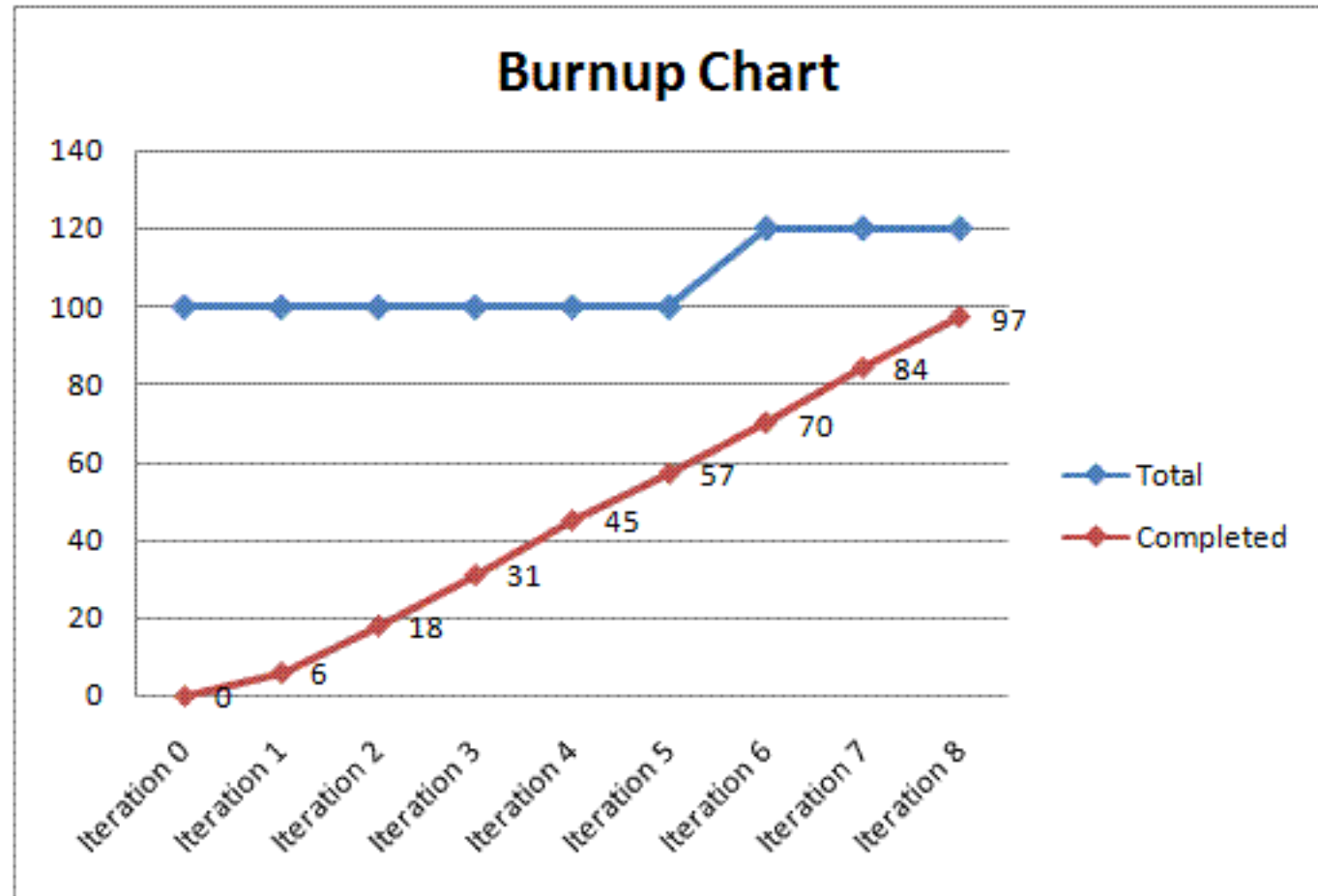
# Sprint Burndown

- Can help us see if:
  - The work is being completed
  - Are there any spikes? (work got added during the Sprint)
  - Did the team start off on the right track?
  - Did it complete all the story efforts by the last day?



# Sprint Burnup

- Similar to burndown but starts with the total work and the completed work the team has done.
- As the team completes the work, the line trends up to reach the total work line.
- The total work can also change up or down (although ideally it shouldn't).





# What other thing a Burndown chart tells us about the team?

- What other items do you suspect we can learn about how the team is doing?

