



# Beyond Guesswork: Empirical Strategies for Agile Release Planning

Optimizing Agile Outcomes with Data-Driven Planning

Agile Boston

10 April 2024

**Dan LeFebvre**

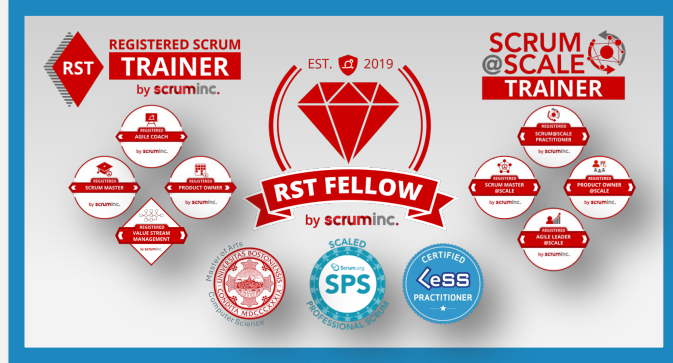
President & Principal Consultant

# Dan LeFebvre

President & Principal Consultant

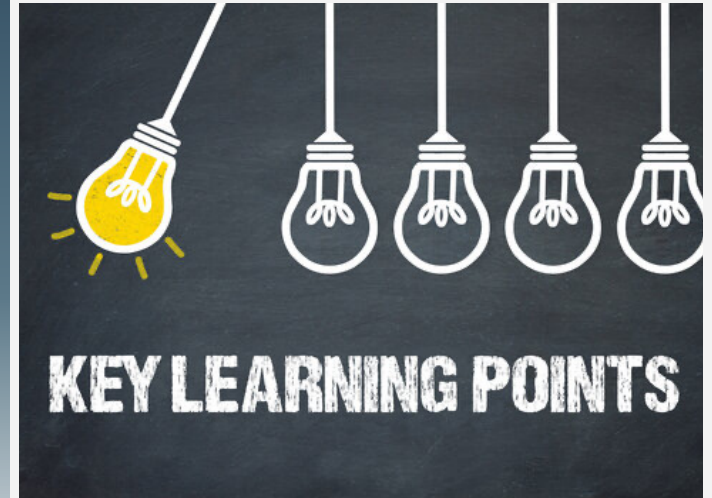


- Extensive experience in product development as a developer and leader
- Using agile practices since 2003
- Agile Coach since 2006
- Executive Coach since 2009



# Key Learnings

- Introduction to Empiricism in Agile
- Sizing and Estimation Techniques
- Release Burndown Charts
- Multi-Team Coordination
- Planning for Multiple Releases
- Q&A



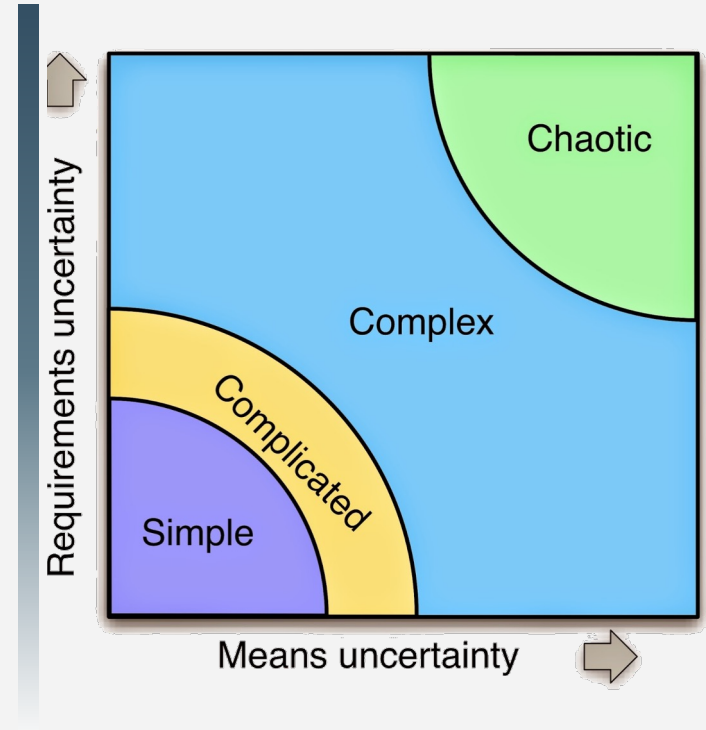
# Introduction to Empiricism

- Empiricism - the practice of making decisions based on observed and measured evidence.
- Progress is based on the reality of completed work, not on forecasts or predictions.
- Benefits of the Empirical Approach:
  - ◆ Enhances responsiveness to change.
  - ◆ Improves accuracy in planning.
  - ◆ Fosters a culture of continuous improvement and learning.



# The Need for Empiricism in Release Planning

- Challenges with Traditional Planning
  - ◆ Complexity makes predictive planning ineffective.
  - ◆ Variance is considered “bad” and, therefore, is avoided.
- Advantages of Empirical Planning
  - ◆ Empiricism bases decisions on observable outcomes.
  - ◆ Adjustments to progress and processes can be made in flight.
  - ◆ Encourages a learning mindset.



# Empirical Release Planning Steps

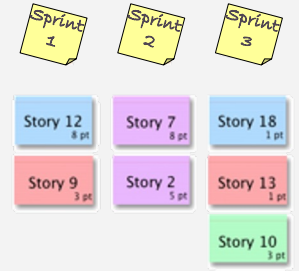
## ① Definition



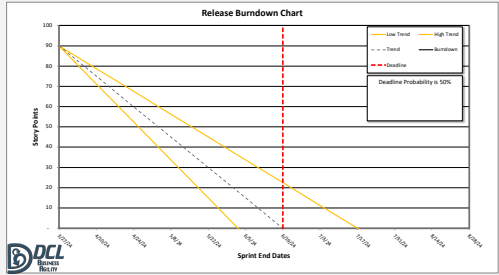
## ② Sizing



## ③ Ordering



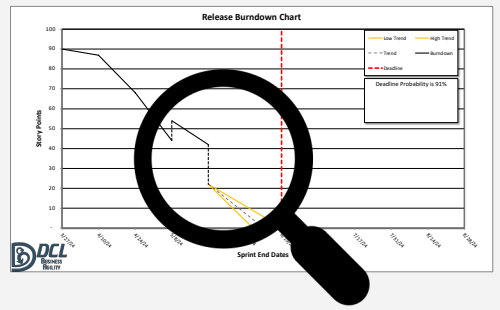
## ④ Projection



## ⑤ Execute



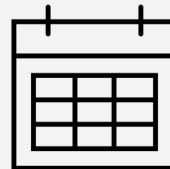
## ⑥ Inspect & Adapt



# Sizing and Estimation

“Estimate Size; Derive Duration”

- Mike Cohn



# Sizing vs. Estimation

- Sizing is about the “bigness” of the work.
  - ◆ Developers use relative sizing to compare the item's complexity, volume, and uncertainty.
  - ◆ This correlates well with the relative effort required.
  - ◆ Uses “non-time-based” units like points.
- Estimation is often about the duration of the work.
  - ◆ Developers use their expertise to guess the duration.
  - ◆ They often neglect the developer's skill (especially if not themselves) or the effects of multitasking and process inefficiencies.



# Methods and Tools for Effective Sizing

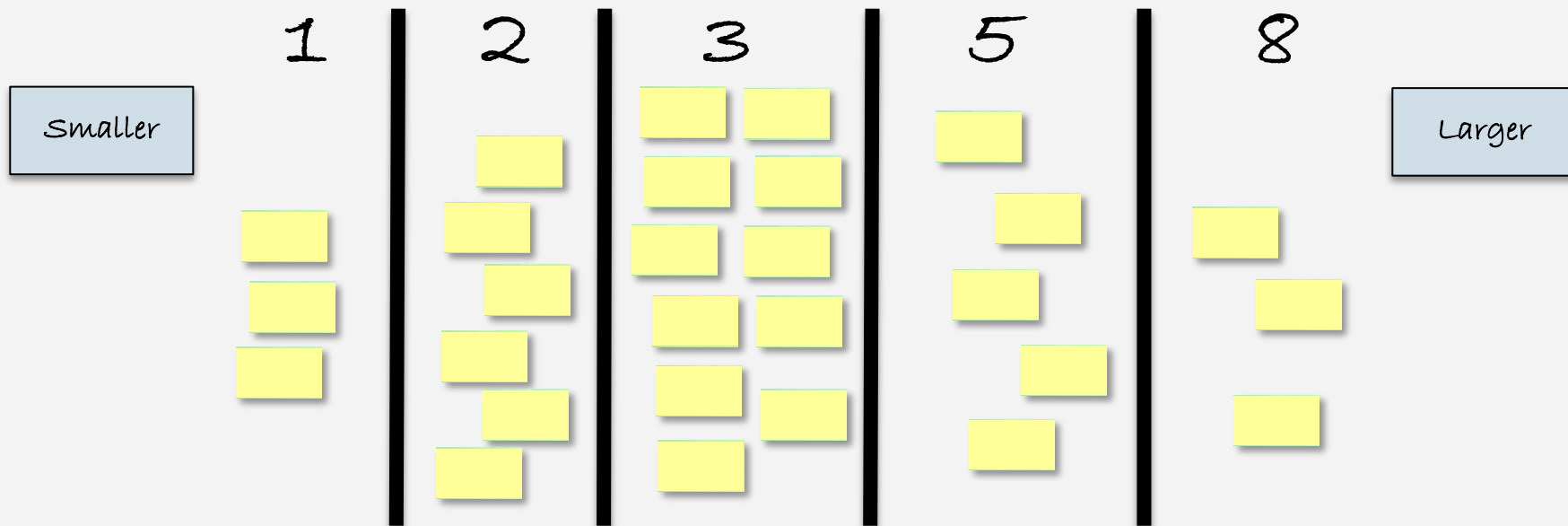
- The size of the entire release needs to be calculated for release planning.
- The two best methods are Affinity Sizing and Large Scale Sizing



Requirement	Build Email Client			
Features	4 Release 1	8 Release 2	88 Release 3	18 Release 4
Epic		17 8 3		
Stories		4 5 8 7	We can now get a rough estimate of the entire piece of work	We eventually reach a level of decomposition where the team is working with their normal <b>Story Point</b> estimates (things that fit into a single Sprint).

# Affinity Sizing

- Great for large number of items to be sized (>15)

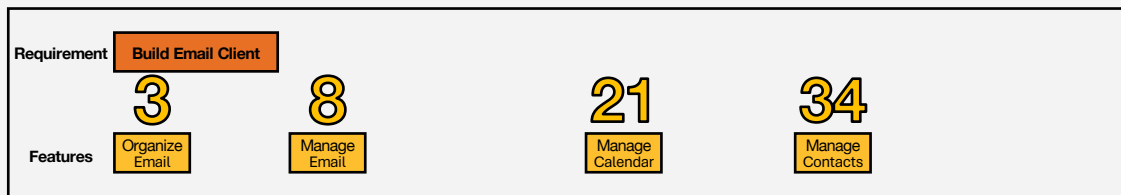


Source:

<http://www.gettingagile.com/2008/07/04/affinity-estimating-a-how-to/>

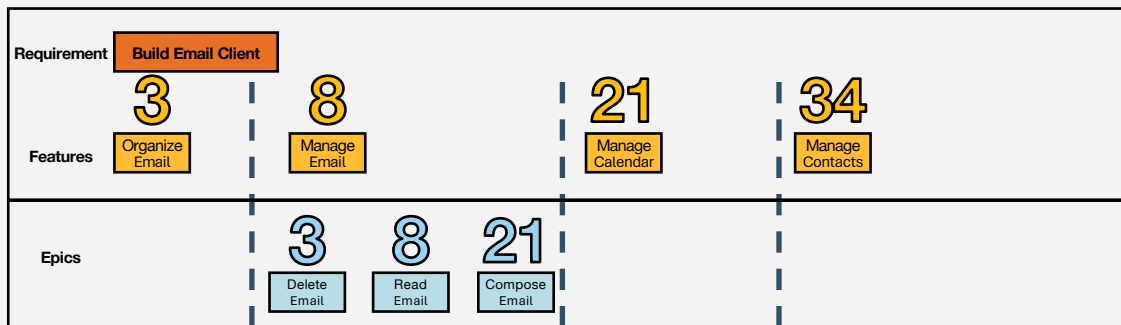
# Large Scale Sizing

- Use Affinity Sizing to determine the relative size of the top level.

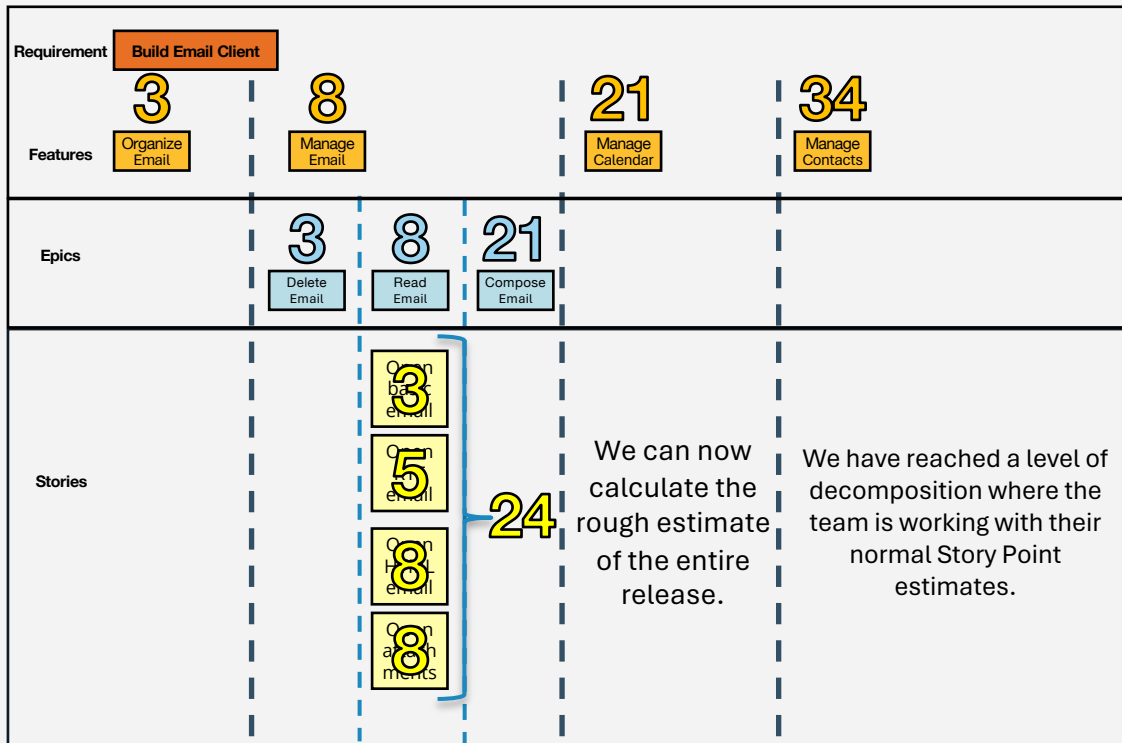


# Large Scale Sizing

- Define Epics for a medium-sized top-level item.
- Use Affinity Sizing to determine the relative sizes of the next level down.



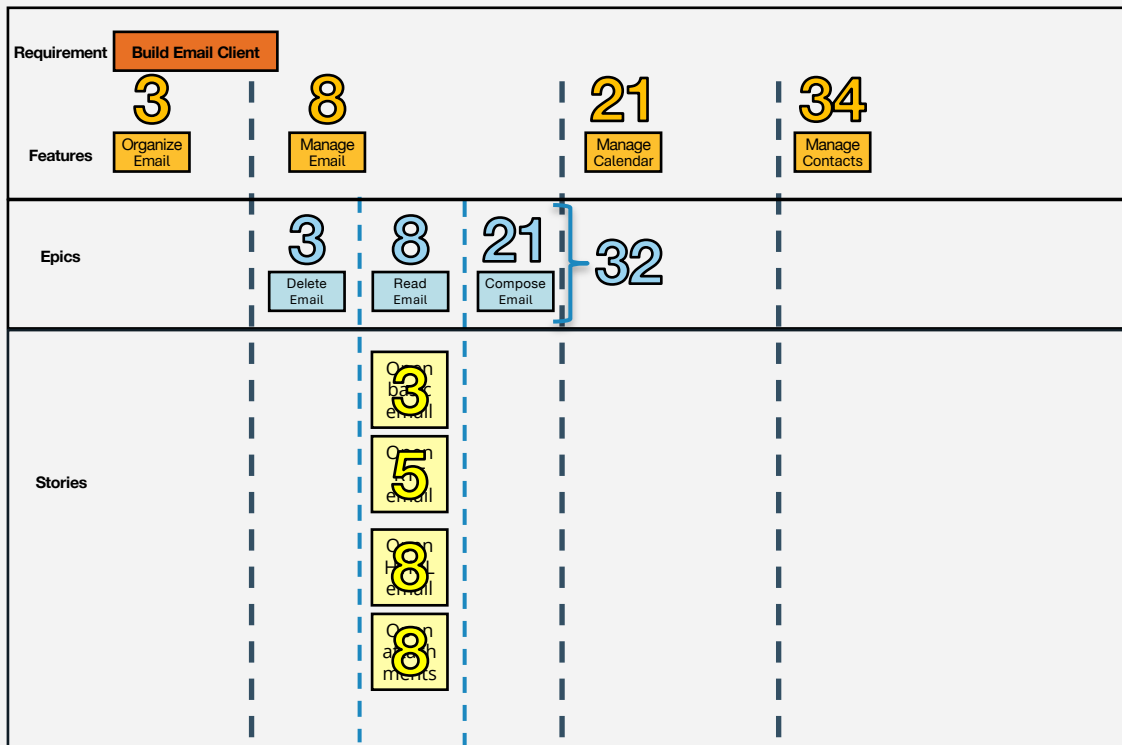
# Large Scale Sizing



1) Calculate SP/EP ratio

$$\frac{SP}{EP}$$

# Large Scale Sizing



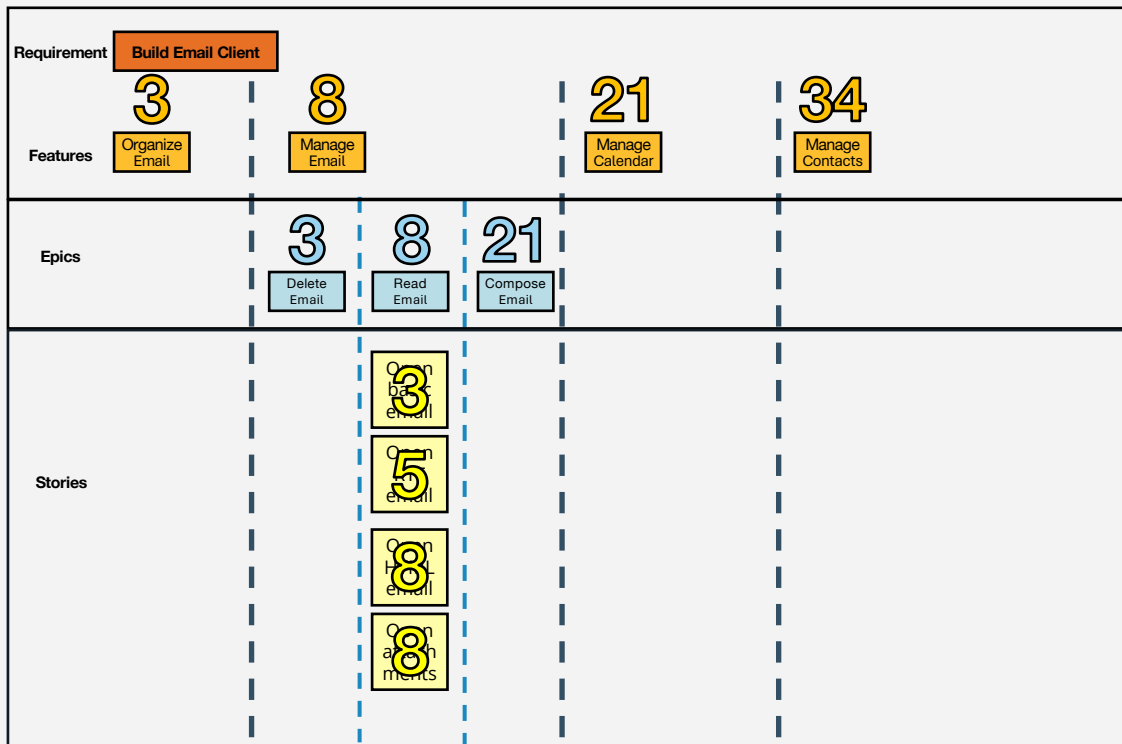
2) Calculate EP/FP ratio

$$\frac{EP}{FP}$$

1) Calculate SP/EP ratio

$$\frac{SP}{EP} = \frac{24}{8} = \frac{3}{1}$$

# Large Scale Sizing



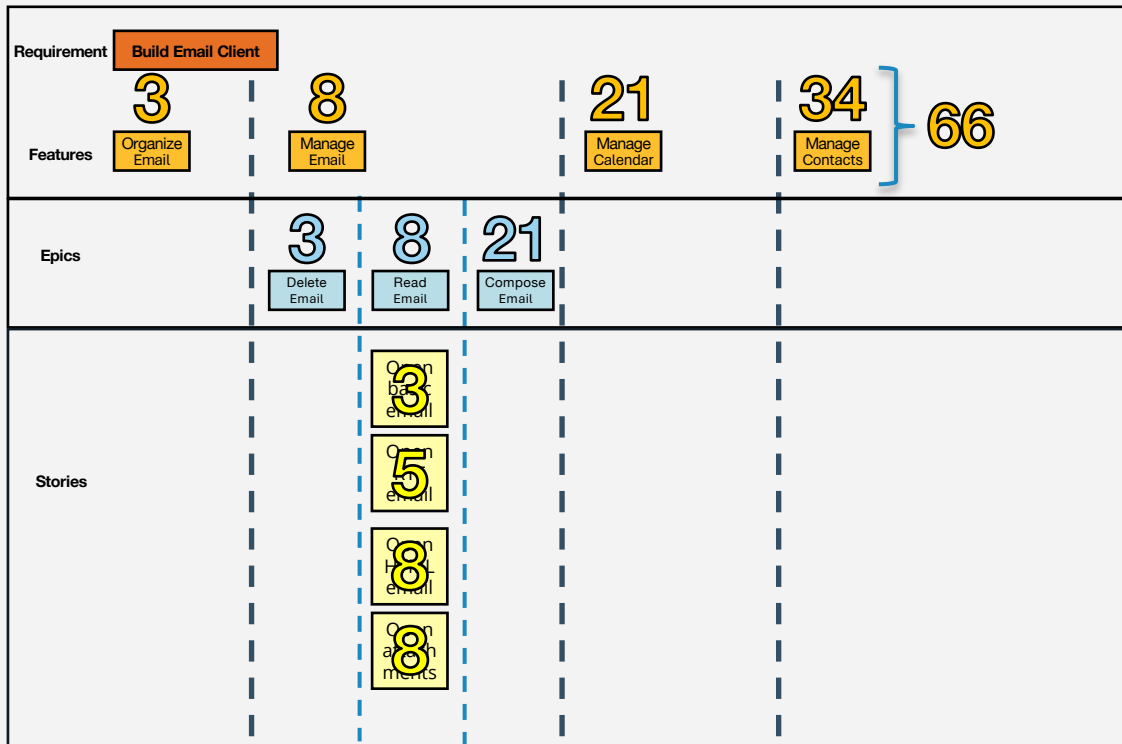
2) Calculate EP/FP ratio

$$\frac{EP}{FP} = \frac{32}{8} = \frac{4}{1}$$

1) Calculate SP/EP ratio

$$\frac{SP}{EP} = \frac{24}{8} = \frac{3}{1}$$

# Large Scale Sizing



3) Calculate Total SP

$$SP = \sum FP \times \frac{EP}{FP} \times \frac{SP}{EP}$$

2) Calculate EP/FP ratio

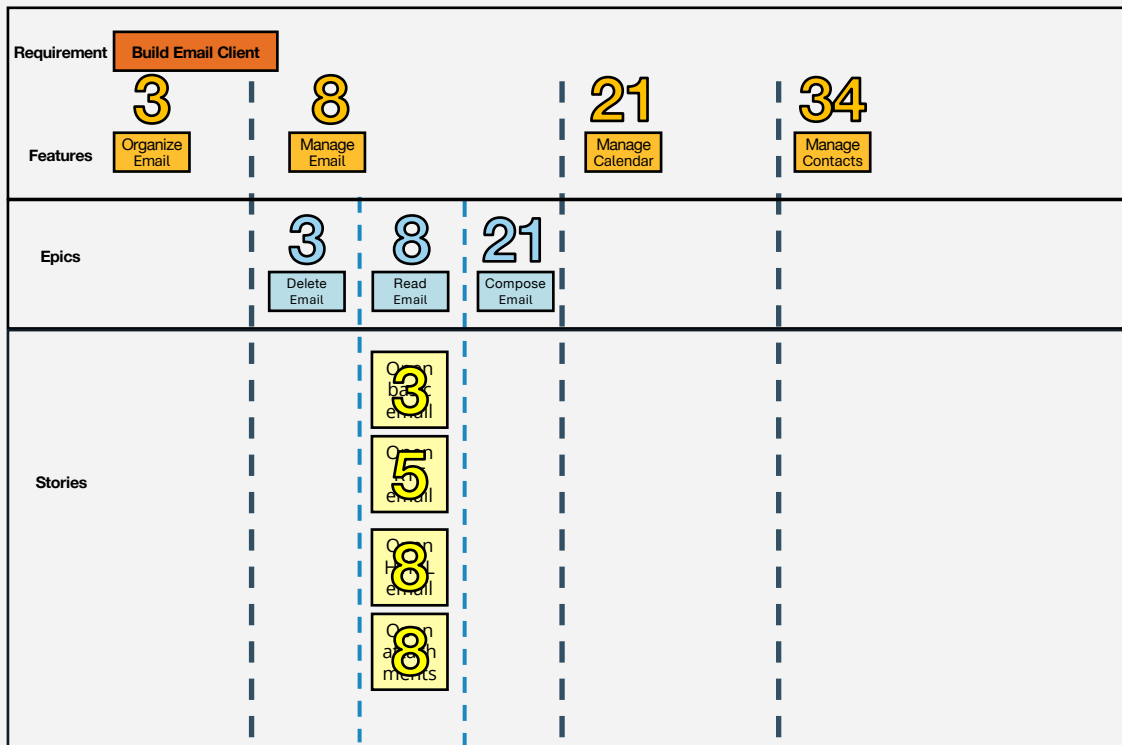
$$= \frac{32}{8} = \frac{4}{1}$$

1) Calculate SP/EP ratio

$$= \frac{24}{8} = \frac{3}{1}$$



# Large Scale Sizing



3) Calculate Total SP

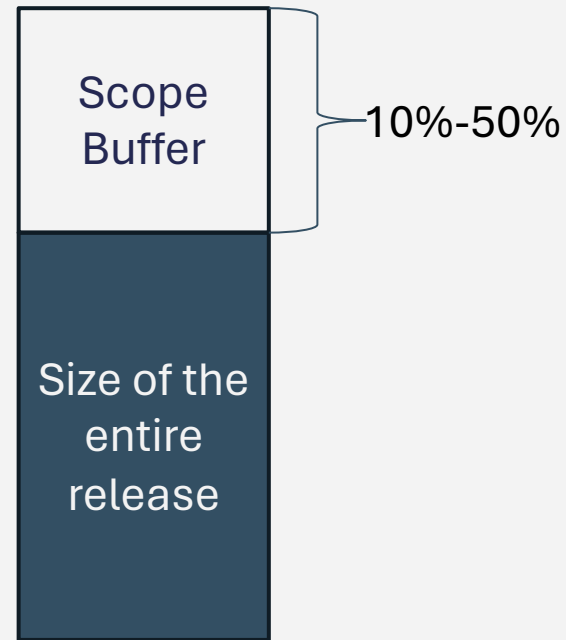
$$SP = \sum FP \times \frac{EP}{FP} \times \frac{SP}{EP}$$

$$SP = 66 \times \frac{4}{1} \times \frac{3}{1}$$

$$SP = 792$$

# Accounting for Scope Uncertainty

- In complex environments, the backlog will usually grow based on discovered needs, stakeholder feedback, and rework based on new knowledge.
- Add a scope buffer to protect against size growth.



# Release Burndown Charts

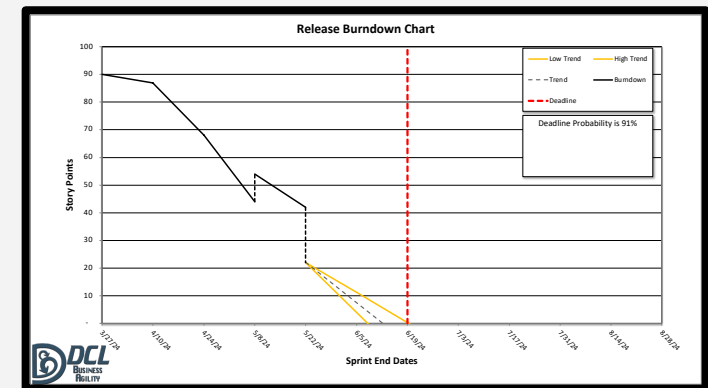
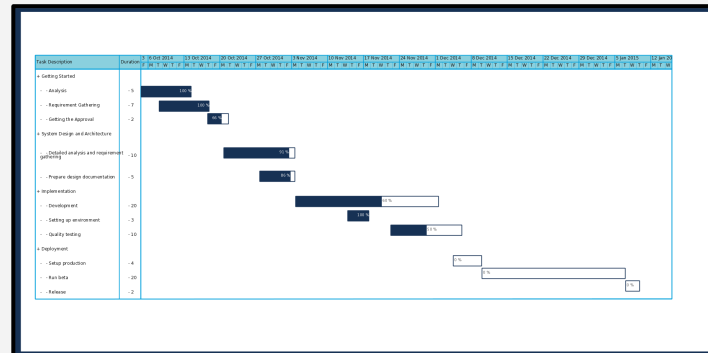
# Calculating Durations for the Project

## • With Estimation

- ◆ Create a Gantt chart including the durations of the tasks and dependencies to create the timeline.
- ◆ Constant updates of the chart are needed to account for variances.

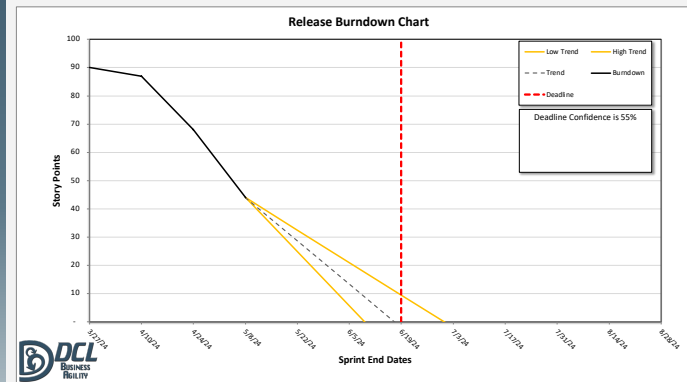
## • With Sizing

- ◆ Add up the sizes and divide by the amount of work the team can deliver in an iteration (velocity or throughput).
- ◆ Use the rolling average of the last 3 to 8 iterations to generate a trend to project completion.

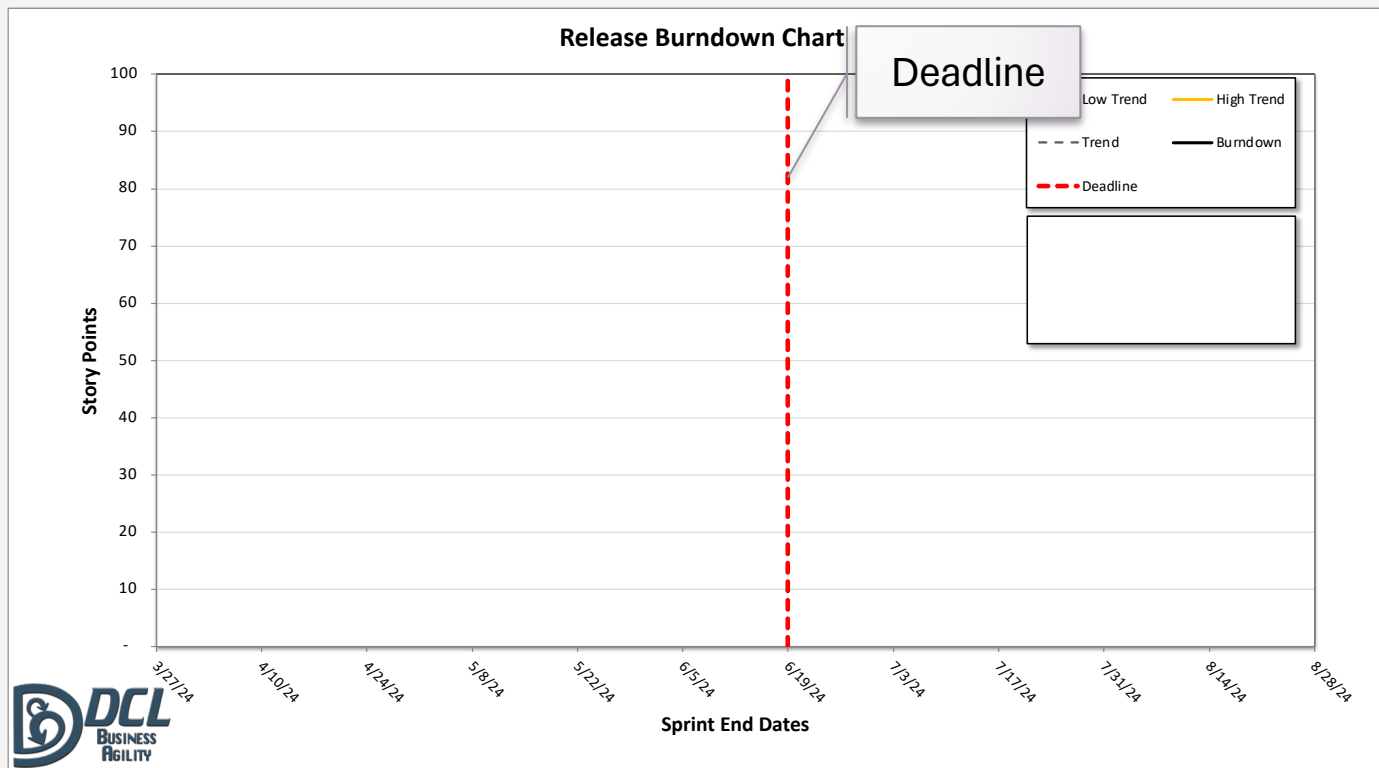


# What Are Release Burndown Charts?

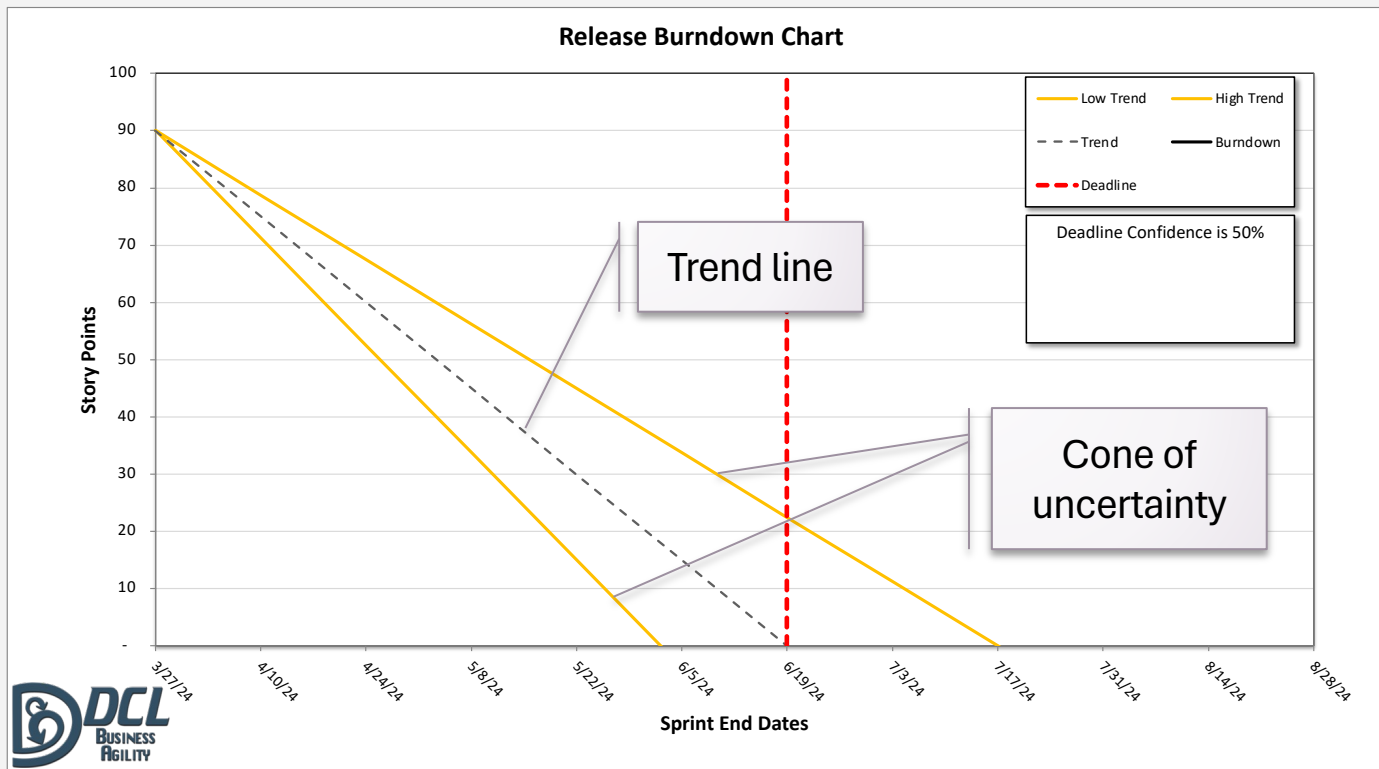
- A visual tool used to track the amount of work remaining in a release backlog over time.
- Provides a clear, at-a-glance view of the release progress against the planned schedule and scope.
- It helps identify deviations from the plan early, allowing for timely interventions.
- Facilitates communication with stakeholders by offering a transparent overview of the release plan.



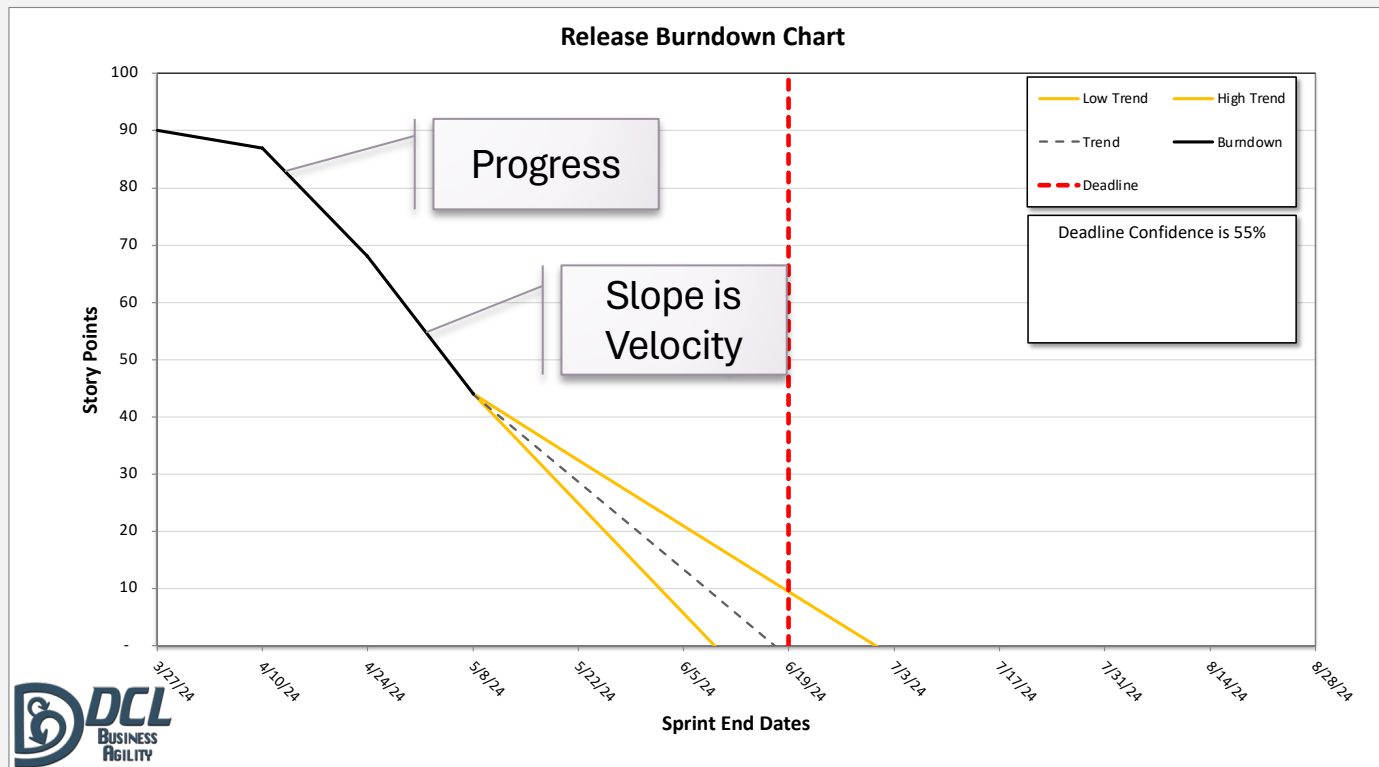
# Reading and Interpreting Burndown Charts



# Reading and Interpreting Burndown Charts

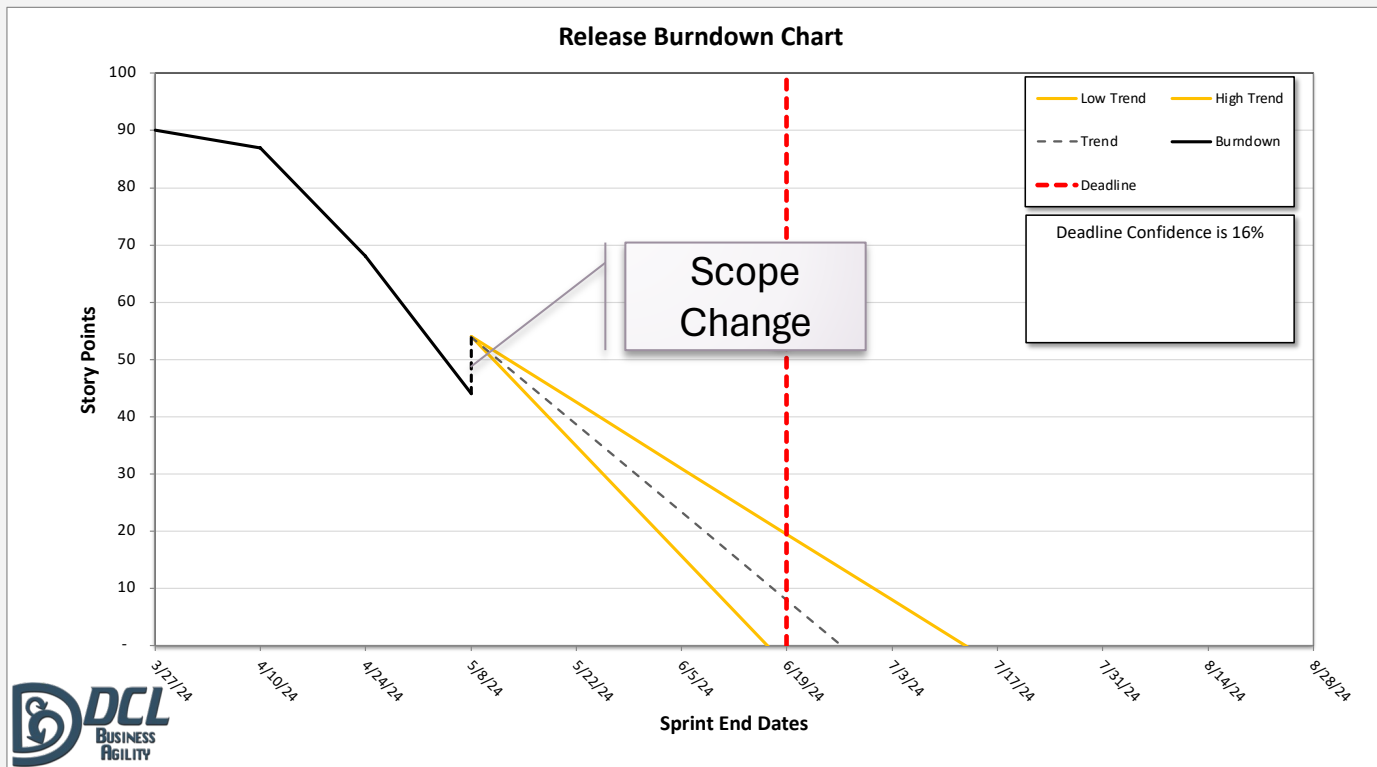


# Reading and Interpreting Burndown Charts

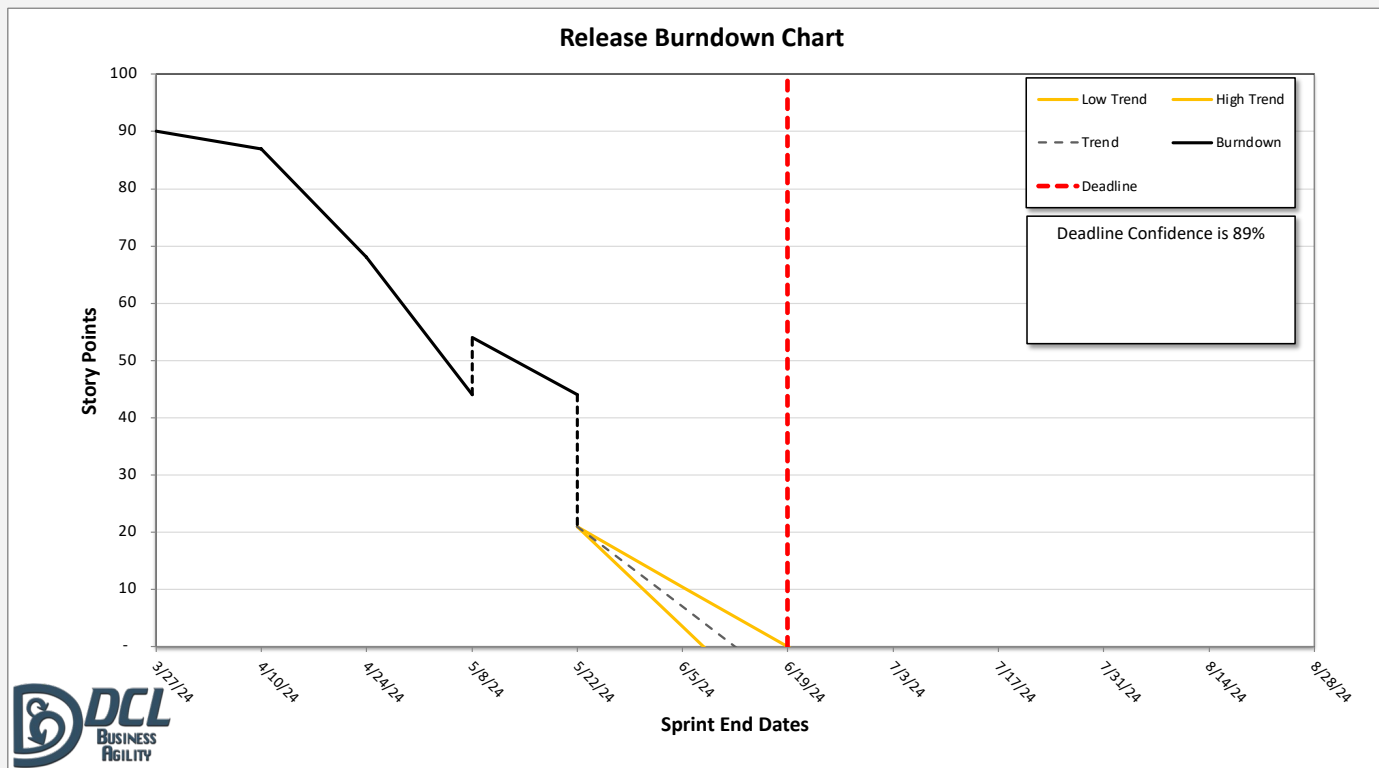




# Reading and Interpreting Burndown Charts



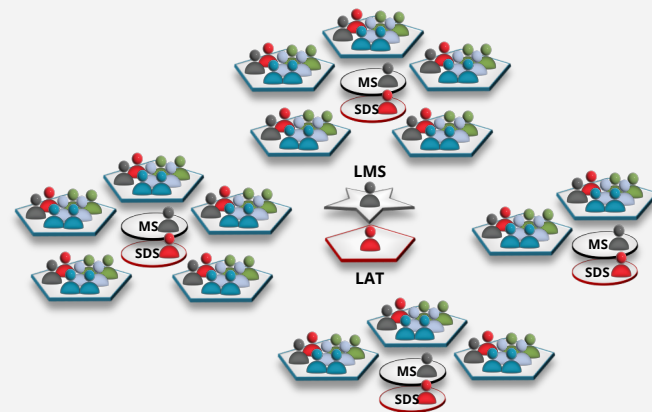
# Reading and Interpreting Burndown Charts



# Multi-Team Coordination

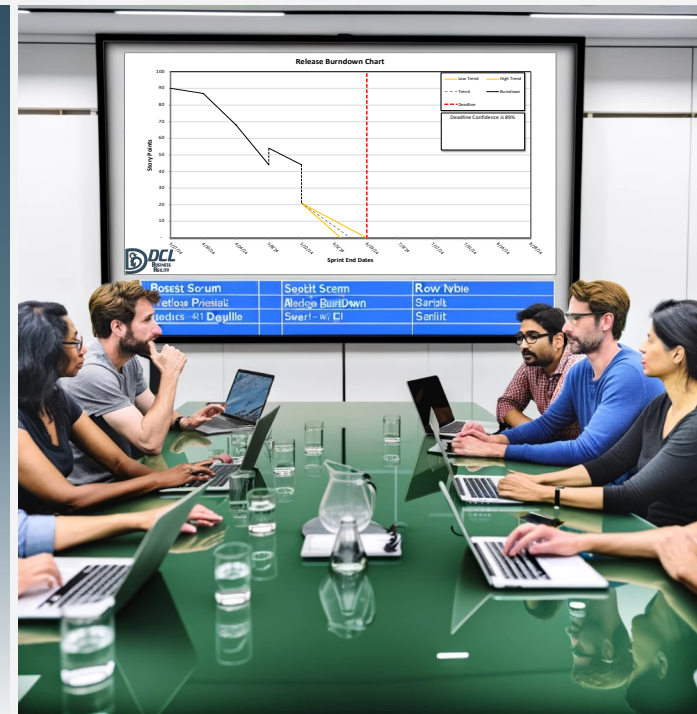
# Challenges of Multi-Team Planning & Tracking

- Integration and Alignment
- Communication Overhead
- Dependency Management
- Consistent Estimation and Sizing
- Understanding the current state



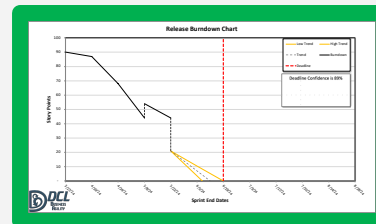
# Strategies for Effective Multi-Team Planning

- Form a Product Owner team with each team's PO.
- Meet every sprint to align on Sprint Goal & dependencies.
- Each PO will pull epics and re-prioritize their backlog based on the meeting outcomes.
- This meeting can be called a MetaScrum, Scaled Refinement, or Scaled Sprint Planning.

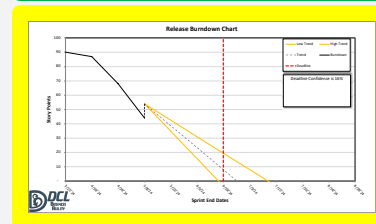


# Strategies for Effective Multi-Team Tracking

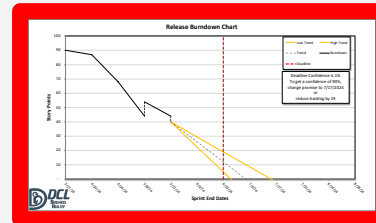
- First Technique – Multi-Team Large-Scale Sizing
  - ◆ Large Scale Size the features and epics using representative developers from each team.
  - ◆ Use Release Burndown Chart on Epic or Feature points.
- Second Technique - “Longest Pole” method
  - ◆ Each team creates its own Release Burndown Chart.
  - ◆ The lowest confidence or latest date is the release date.
    - The PO team can move work around, reduce scope, or recommit as a team.



Deadline Confidence is 89%



Deadline Confidence is 16%



Deadline Confidence is 1%  
To get a confidence of 90%,  
change promise to 7/17/2024  
or  
reduce backlog by 19

# Planning for Multiple Releases

When a team must work on more than 1 release at a time.

# Strategy for Multi-Release Planning

- Sometimes, teams must work on multiple products or releases simultaneously.
- Produce a Release Burndown Chart for each release.
- The team backlog includes items from each release.
- At the end of the sprint, apply the points from done items to the appropriate release burndowns.

## Sprint Backlog

✓8

✓5

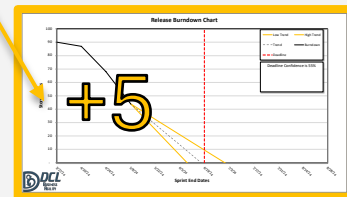
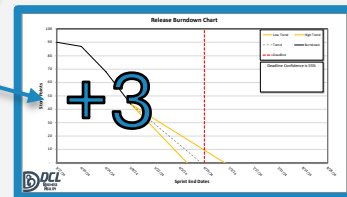
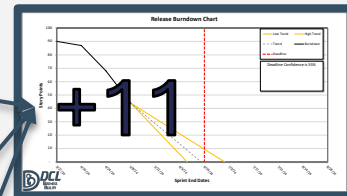
✓3

✓1

13

5

✓2





# Key Takeaways

- Decision-making should be grounded in empirical evidence.
- Embrace relative sizing to inform more accurate planning.
- Leverage both Affinity Sizing and Large-Scale Sizing.
- Use release burndown charts to visualize progress and adapt plans in response to actual performance.
- Foster collaboration through regular PO team meetings to align goals and dependencies.
- Actively seek and incorporate feedback from stakeholders and past releases to ensure products remain relevant and competitive.

# Thank You!

Slides



Schedule a free 30-minute consultation



Release Burndown Tool



dclba.com



Dan LeFebvre

President & Principal Consultant



[dan@dclba.com](mailto:dan@dclba.com)  
[info@dclba.com](mailto:info@dclba.com)



[linkedin.com/in/danlefebvre](https://www.linkedin.com/in/danlefebvre)  
[linkedin.com/company/DCLBA](https://www.linkedin.com/company/DCLBA)

# Questions

