COMP 3550

10.1 — HOW DO WE MEASURE SUCCESS?

Week 10: Measuring Team and Project Successes

WHAT DOES “SUCCESS” MEAN IN SOFTWARE PROJECTS?

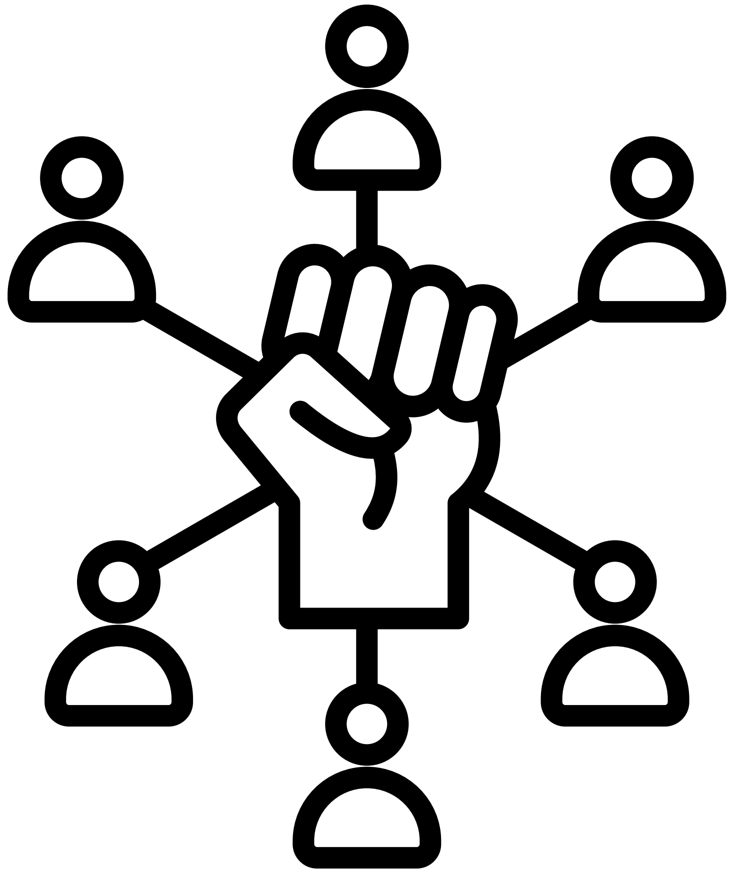
- Functional product
- Happy team
- On-time delivery
- Under Budget
- Learning something

Himbeault 2025 ©

TUCKMAN’S STAGES OF TEAM DEVELOPMENT

Forming

- Team members get to know each other
- Polite, cautious, figuring out roles



TUCKMAN'S STAGES OF TEAM DEVELOPMENT

Forming

- Team members get to know each other
- Polite, cautious, figuring out roles

Storming

- Conflicts emerge (ideas, working styles, priorities)
- Power struggles, frustration, testing boundaries

TUCKMAN'S STAGES OF TEAM DEVELOPMENT

Forming

- Team members get to know each other
- Polite, cautious, figuring out roles

Storming

- Conflicts emerge (ideas, working styles, priorities)
- Power struggles, frustration, testing boundaries

Norming

- Roles, processes, and norms settle
- More trust and cooperation
- Conflict becomes constructive

TUCKMAN’S STAGES OF TEAM DEVELOPMENT

Forming

- Team members get to know each other
- Polite, cautious, figuring out roles

Storming

- Conflicts emerge (ideas, working styles, priorities)
- Power struggles, frustration, testing boundaries

Norming

- Roles, processes, and norms settle
- More trust and cooperation
- Conflict becomes constructive

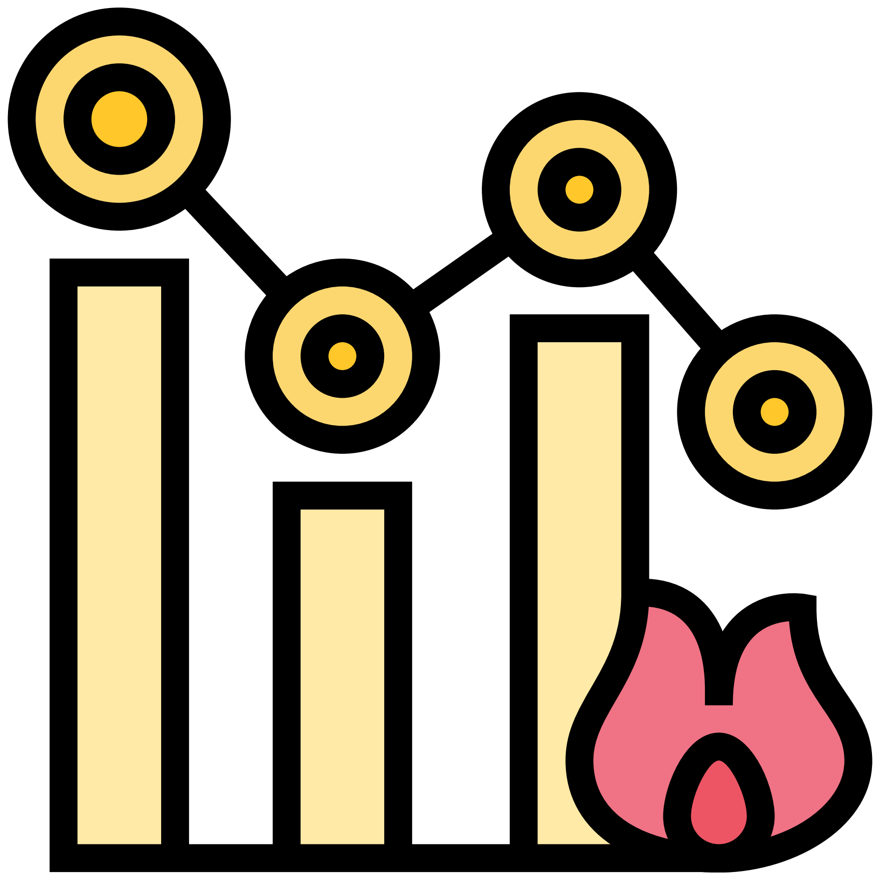
Performing

- High trust and autonomy
- Team focuses on goals, adapts quickly, and delivers results

BURNDOWN CHARTS

What They Show

- Work Remaining vs. Time
- Visual way to track progress against the plan
- Helps spot scope creep, slowdowns, or early completion



BURNDOWN CHARTS

What They Show

- Work Remaining vs. Time
- Visual way to track progress against the plan
- Helps spot scope creep, slowdowns, or early completion

How to Read One

- X-axis: Time (e.g., days in a sprint)
- Y-axis: Remaining work (story points, tasks, hours)
- Ideal Line: Smooth downward slope from start to finish
- Actual Line: Real progress — often jagged, sometimes above/below ideal

BURNDOWN CHARTS

What They Show

- Work Remaining vs. Time
- Visual way to track progress against the plan
- Helps spot scope creep, slowdowns, or early completion

How to Read One

- X-axis: Time (e.g., days in a sprint)
- Y-axis: Remaining work (story points, tasks, hours)
- Ideal Line: Smooth downward slope from start to finish
- Actual Line: Real progress — often jagged, sometimes above/below ideal

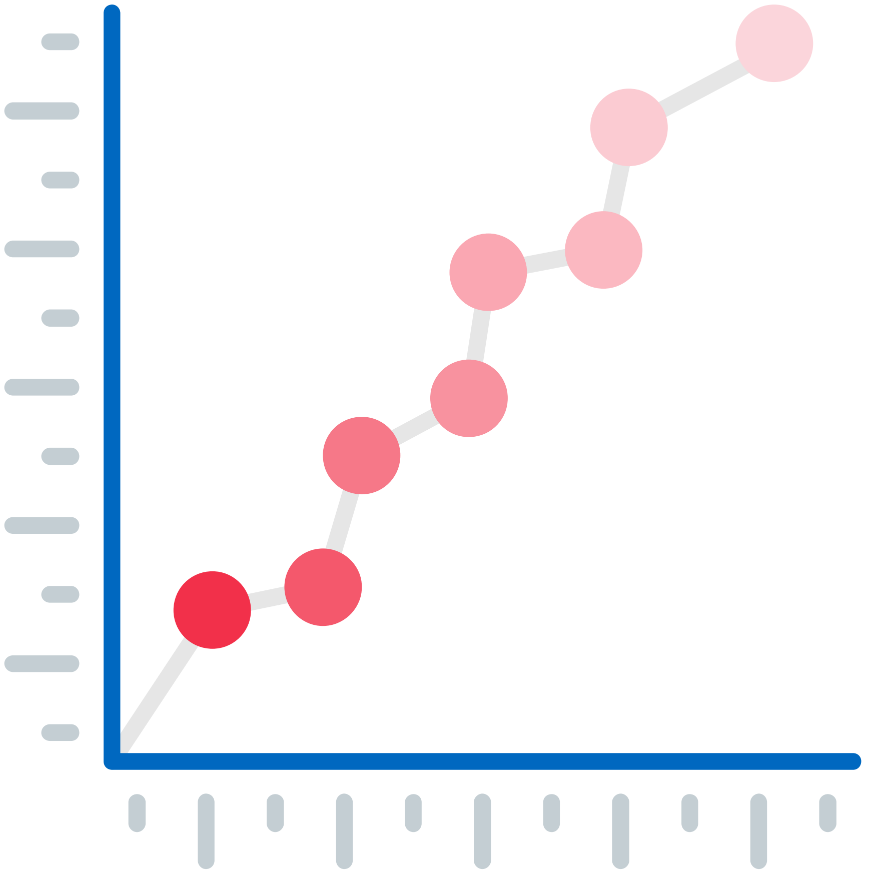
Interpreting Actual vs. Ideal

- Above Ideal: Behind schedule, blocked tasks, underestimated complexity
- Below Ideal: Ahead of schedule, tasks smaller than expected, or scope reduced
- Flat Line: No progress — potential blocker or dependencies not resolved

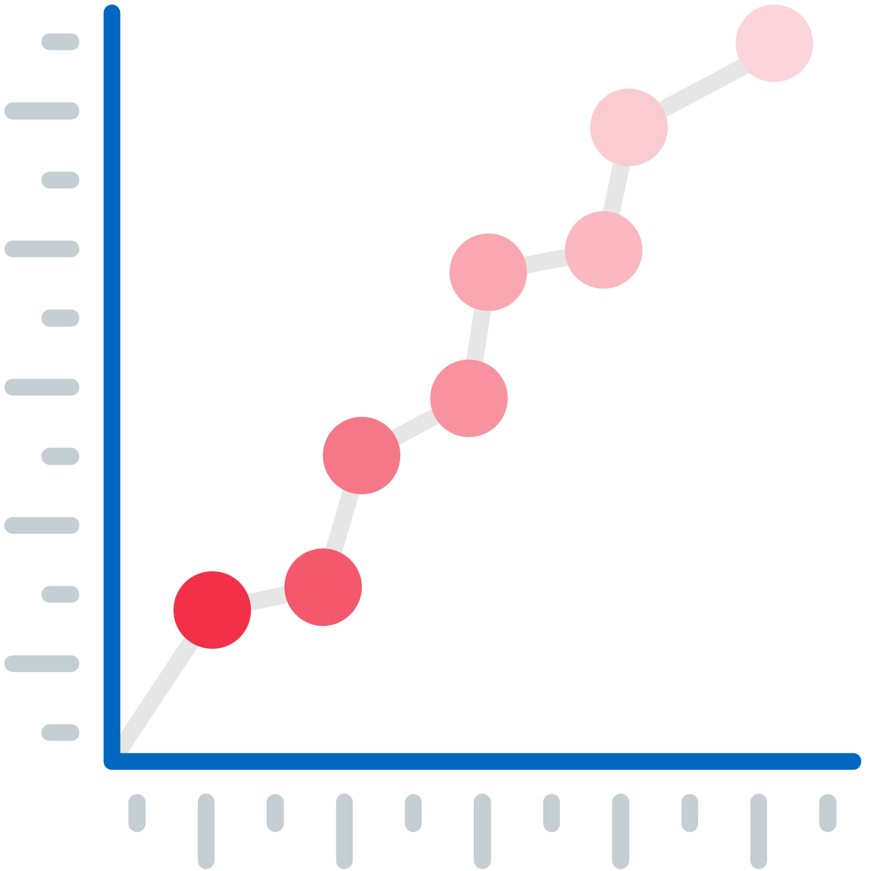
VELOCITY CHARTS

What They Show

- Team output per iteration (e.g., story points, tasks completed)
- Useful for forecasting how much the team can take on in future sprints
- Best viewed as trends over time, not a single number



VELOCITY CHARTS



What They Show

- Team output per iteration (e.g., story points, tasks completed)
- Useful for forecasting how much the team can take on in future sprints
- Best viewed as trends over time, not a single number

Reading the Chart

- X-axis: Iterations (Sprints)
- Y-axis: Completed work (points, tasks)
- Steady velocity → stable process & estimation
- Increasing velocity → improving productivity or changing scope
- Dropping velocity → potential blockers, capacity changes, or bigger stories

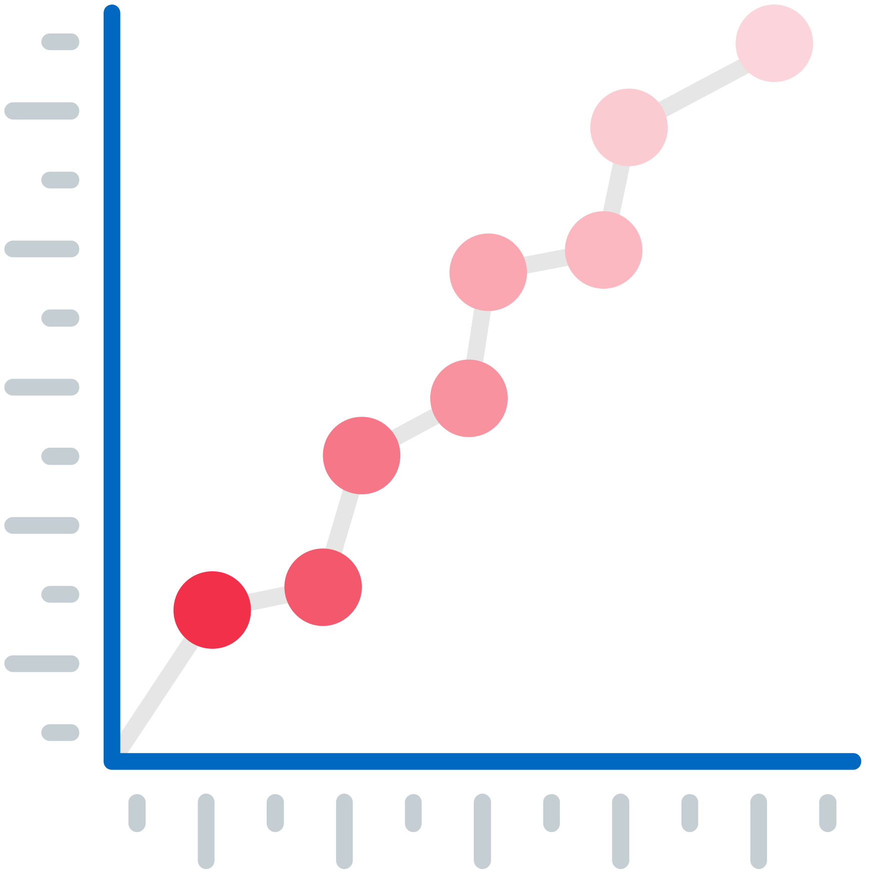
VELOCITY CHARTS

What They Show

- Team output per iteration (e.g., story points, tasks completed)
- Useful for forecasting how much the team can take on in future sprints
- Best viewed as trends over time, not a single number

Reading the Chart

- X-axis: Iterations (Sprints)
- Y-axis: Completed work (points, tasks)
- Steady velocity → stable process & estimation
- Increasing velocity → improving productivity or changing scope
- Dropping velocity → potential blockers, capacity changes, or bigger stories

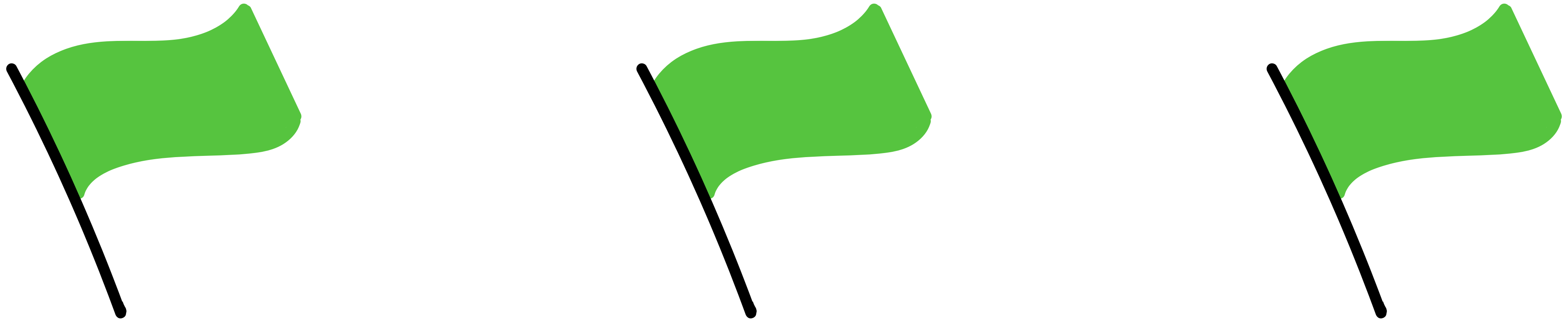


Common Pitfalls

1. **Points Inflation**
   - Over time, same work gets assigned more points
   - Makes velocity look higher without real productivity gain
2. **Uneven Effort**
   - Some sprints overloaded, others light
   - Leads to unreliable forecasts and team stress

SIGNS YOU'RE DOING WELL (AND NOT)

- **Green Flags — Things Going Right**
  - **Regular Commits —** steady, incremental progress
  - **Code Reviews —** constructive feedback, shared understanding
  - **Visible Progress —** features working in staging/demo environments
  - **Team Learning —** sharing knowledge, improving processes
  - **Healthy Communication —** blockers raised early, decisions documented



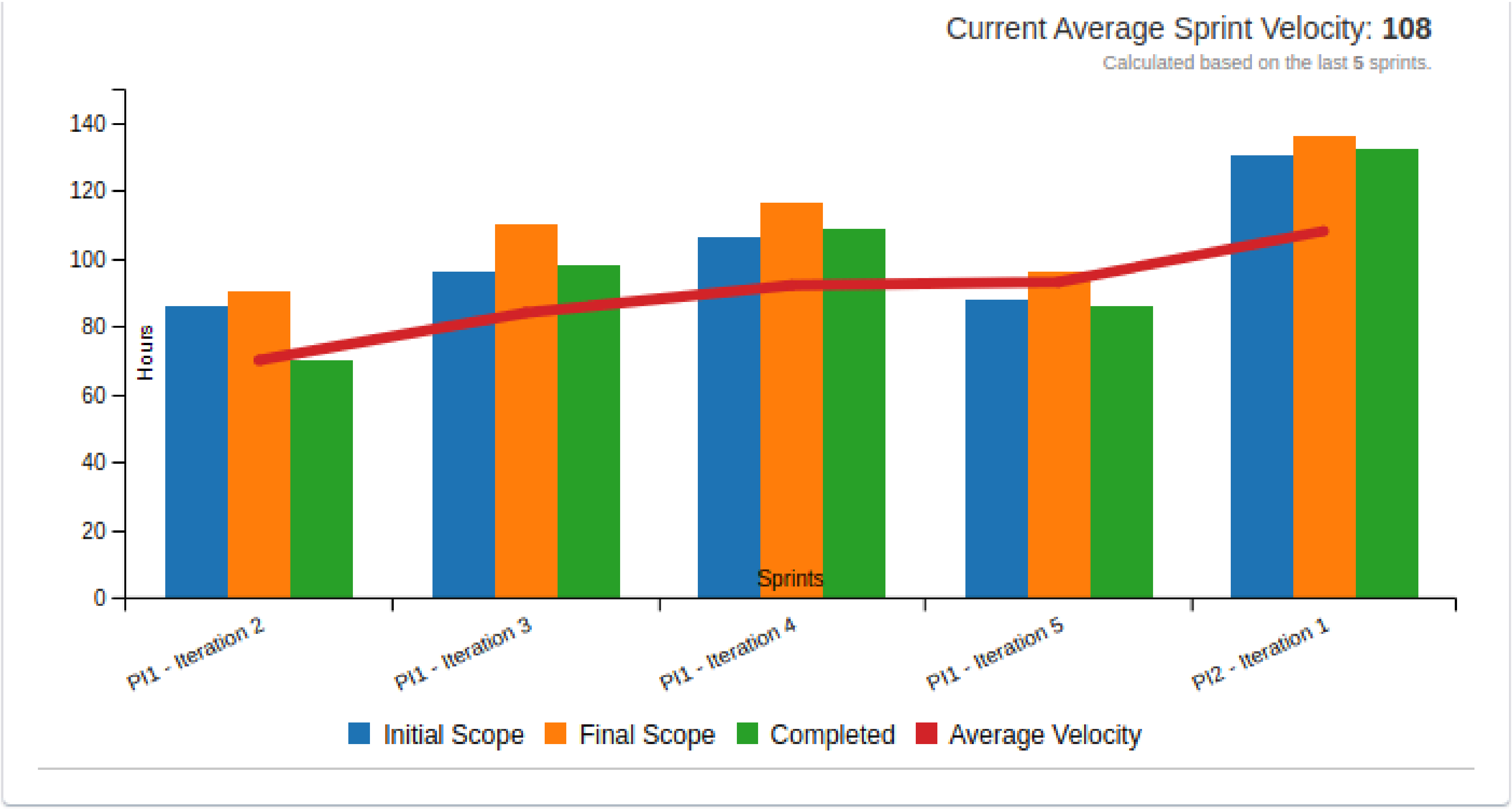
SIGNS YOU’RE DOING WELL (AND NOT)

- **Red Flags — Warning Signs**
  - **Missed Sprints —** repeated failure to hit planned goals
  - **Unclear Ownership —** no one knows who’s responsible for a task or area
  - **No Tests / No CI —** changes risky, bugs slip through unnoticed
  - **Long Periods of Silence —** no visible commits or updates
  - **Last-Minute Crunches —** repeated heroics to “save” delivery

PROJECT PAUSE & REFLECT

Calculate your own project's burndown and team velocity.
What surprises you? What did you expect?

Next, take a look at the following sample team velocity chart, what trends do you see? What can you say about this team?



https://community.atlassian.com/forums/Jira-questions/How-to-get-velocity-chart-if-Team-is-using-hours/qaq-p/2466912