Agilebars User Guide

Agilebars Sprint Scheduler is a visual sprint planning tool designed for Agile Scrum teams to manage product backlogs, plan sprints, track progress, and generate burndown charts. This guide focuses on **Agilebars-specific features**—the differences from Timebars and Costbars.

For features common across all products (Bar Creator, Spreadsheet Sync, Reports, Data Management, Cloud Publishing, etc.), see the Common Help Topics section at the end of this guide.

What Makes Agilebars Different?

Agilebars is specifically designed for **Agile Scrum sprint management** with these unique features:

1. **Dual-Mode Canvas**: Switch instantly between Timescale mode and Kanban mode
2. **2-Level Hierarchy**: Simplified structure (Projects and Tasks only—no portfolios or allocations)
3. **Kanban Board**: Visual workflow with 5 swim lanes (Backlog, Will Do, Doing, Finalizing, Done)
4. **Automatic Progress Tracking**: Move tasks between lanes to earn progress automatically
5. **Burndown Charts**: One-click generation based on lane movements and time-phased sizing
6. **Earned Value Scheduling**: Progress is calculated, not manually entered
7. **Sprint-Focused**: Optimized for 1-4 week sprint cycles

Best For:

- Scrum teams managing sprints
- Product owners tracking backlog reduction
- Agile coaches measuring team velocity
- Development teams visualizing work in progress

Getting Started with Agilebars

Prerequisites

Before diving into Agilebars, we recommend:

1. **Read Common Help**: Review the Common User Interface Guide for basics like:
   - How to create bars using Bar Creator
   - How to delete and duplicate bars
   - How to use filters and search
   - How to manage data with spreadsheets
2. **Visit the Sales Site**: Understand the value proposition:
   - Benefits Presentation
   - Features Presentation
   - Product Overview

3. **Get a License** (Optional):
    - Use Agilebars **free** for small projects (limited bars)
    - Purchase a subscription to remove limits
    - See video on licensing

Access Agilebars Sprint Planner Client Application

Visit: **ab.timebars.com**

The Agilebars Hierarchy

Unlike Timebars and Costbars (which support 5-level hierarchies), Agilebars uses a **simple 2-level hierarchy** optimized for Agile workflows:

Two Levels Only

Level 2 - Green Bars: Projects (Sprint Backlogs)

- Represents a sprint backlog or project
- Contains multiple work items (tasks)
- Color: Green
- Example: "Sprint 12 - Mobile App Features"

Level 4 - Blue Bars: Tasks (Work Items)

- Represents user stories, tasks, or backlog items
- Must be created under a Project (green bar)
- Color: Blue
- Example: "User Story: Login with OAuth"

No Other Levels

- **No L1 (Portfolios)**: Agilebars doesn't support portfolio-level management
- **No L3 (Sub-Projects)**: No need for work packages in sprint planning
- **No L5 (Allocations)**: Work sizing is assigned directly to tasks, not separate allocation bars

Hierarchy Rules:

- Green bars (Projects) cannot be nested under each other
- Blue bars (Tasks) **must** be dropped onto Green bars (Projects)
- Attempting to violate the hierarchy triggers a warning and prevents the action

---

Dual-Mode Canvas: Timescale and Kanban

The power of Agilebars lies in its ability to switch instantly between two modes:

Timescale Mode (Default)

What It Is:

- Traditional timeline view showing tasks on a calendar
- Horizontal bars represent work items across time
- Time-phases tasks from sprint start to sprint finish

When to Use:

- Planning sprint timelines
- Sizing work items with start/finish dates
- Visualizing sprint duration
- Setting up data before using Kanban mode

Key Point: Burndown charts are **time-phased**, so you must use Timescale mode to assign dates to tasks before generating accurate burndown charts.

Kanban Mode

What It Is:

- Visual board with swim lanes representing workflow stages
- Drag-and-drop tasks between lanes to update progress
- Real-time progress tracking based on lane position

When to Use:

- Daily sprint standups
- Tracking work in progress
- Updating task status quickly
- Visualizing team workload

How to Switch: Click the **"Switch Modes"** icon on the main menu (top center of screen)

- Switches from Timescale → Kanban
- Click again to switch back: Kanban → Timescale
- **No import/export required**—data syncs instantly

Kanban Board: Visual Workflow Management

What is a Kanban Board?

A Kanban board is a visual project management tool that helps teams track and manage work effectively using swim lanes and cards (bars).

Benefits:

- Clear visual representation of workflow
- Promotes team collaboration and transparency
- Manages work-in-progress limits
- Tracks and prioritizes tasks
- Identifies blockers quickly

- Drives continuous improvement

Digital vs. Physical: While traditional Kanban uses sticky notes on a wall, Agilebars provides a **digital Kanban board** that:

- Automatically calculates progress
- Generates burndown chart data
- Syncs with timescale view
- Stores all data in the browser

The 5 Agilebars Lanes

Tasks (blue bars) move through 5 swim lanes representing workflow stages:

1. Backlog

Purpose: Work items not yet committed to the sprint

Characteristics:

- 0% progress
- No actual start date set
- Represents the product backlog or sprint candidate items

Actions:

- Add new tasks here
- Prioritize and groom backlog items
- Move to "Will Do" when committing to sprint

---

2. Will Do

Purpose: Work committed to the sprint but not yet started

Characteristics:

- 0% progress (no progress earned by moving here)
- Actual start date: Not set
- Represents sprint commitment

Actions:

- Move from Backlog when sprint planning is complete
- Prioritize order for sprint execution
- Move to "Doing" when work begins

---

3. Doing

Purpose: Work actively in progress

Characteristics:

- **10% progress earned** when task enters this lane
- Actual start date: **Automatically set** when dropped here
- Represents active development/work

Progress Calculation:

- Moving to "Doing" earns **10% of the sizing value**
- 10% covers planning effort
- System automatically records actual start date

Actions:

- Move to "Finalizing" when development complete and ready for testing/review

---

4. Finalizing

Purpose: Work in final stages (testing, review, approval)

Characteristics:

- **75% total progress** (65% additional when entering)
- Represents testing, code review, proofreading, approval stages
- Team defines their own "finalizing" rules

Progress Calculation:

- Entering "Finalizing" earns additional **65% of sizing value**
- Total progress now: 10% (Doing) + 65% (Finalizing) = **75%**

Actions:

- Move to "Done" when all acceptance criteria met and item is complete

---

5. Done

Purpose: Work completed and accepted

Characteristics:

- **100% progress** (25% additional when entering)
- Actual finish date: **Automatically set** when dropped here
- Represents completed, accepted work

Progress Calculation:

- Entering "Done" earns final **25% of sizing value**
- Total progress: 10% + 65% + 25% = **100%**

Actions:

- Task is complete
- Contributes to sprint velocity
- Included in burndown chart as "work burned down"

Progress Calculation Rules

Understanding how progress accumulates is critical for accurate burndown charts:

Lane Transition	Progress Earned	Total Progress	Automatic Actions
Backlog → Will Do	0%	0%	None
Will Do → Doing	10%	10%	Actual Start Date set to today
Doing → Finalizing	65%	75%	None
Finalizing → Done	25%	100%	Actual Finish Date set to today

Important Notes:

- Progress is based on the **sizing value** of the task (hours or story points)
- The percentages (10%, 65%, 25%) represent portions of total work
- Moving backwards (e.g., Done → Doing) reverses progress calculations
- Progress is **earned automatically**—you never manually set percent complete

---

How to Use the Kanban Board

Step-by-Step Workflow:

1. **Set Report Date First**:
    - Choose **Tools > Canvas Settings**
    - Set **Report Date** to today's date
    - This ensures accurate date tracking when moving tasks
2. **Create Sprint Backlog (Green Bar)**:
    - Use Bar Creator to create a Project bar
    - Name it (e.g., "Sprint 12 - Q1 2025")
    - Set sprint start and finish dates in Timescale mode
3. **Add Work Items (Blue Bars)**:
    - Use Bar Creator to add Tasks under the Project
    - Drop blue Task Creator Bar onto green Project bar
    - Name each task (e.g., "User Story: Add Payment Gateway")
    - **Size each task** with hours or story points in Timescale mode
4. **Switch to Kanban Mode**:
    - Click **"Switch Modes"** icon on main menu

    - Tasks appear in Backlog lane by default

5. **Plan Sprint**:

    - Drag committed tasks from **Backlog** to **Will Do** lane
    - Prioritize order top to bottom

6. **Execute Sprint**:

    - As work starts, drag task from **Will Do** to **Doing**
    - Actual start date auto-records
    - Progress automatically updates to 10%

7. **Track Progress**:

    - When development complete, drag from **Doing** to **Finalizing**
    - Progress updates to 75%
    - Use for testing, review, approval phases

8. **Complete Work**:

    - When fully done, drag from **Finalizing** to **Done**
    - Progress updates to 100%
    - Actual finish date auto-records

9. **Update Report Date Daily**:

    - Important: Set Report Date to match the day you move bars
    - Ensures burndown chart calculates accurately

10. **Generate Burndown Chart**:

    - See Burndown Chart section below

Kanban Display Options

Toggle Between Bars and Text Boxes:

- Right-click on Kanban Canvas
- Select **"Toggle Bars vs Text Boxes"** from shortcut menu
- **Bars**: Visual bar representation (default)
- **Text Boxes**: Condensed text-only cards for more compact view

Large Screen Recommended: Choose a large, visible screen for optimal Kanban board visibility, especially during daily standups and team collaboration.

Prerequisites for Accurate Burndown Charts

Before generating burndown charts, complete these essential steps to ensure reliable forecasting and progress tracking.

Why Prerequisites Matter

If you take the time to estimate each work item with the skilled person responsible for completing it and ensure progress updates based on their input, you can begin to trust and rely on burndown charts as the primary reporting tool for problem-solving and predicting when the sprint will be completed.

Step 1: Define the Sprint Goal and Backlog

Start by understanding the sprint goal and the list of tasks or user stories that the development team commits to completing during the sprint. This forms the sprint backlog.

What to do:

- Clarify the sprint objective with product owner and stakeholders
- Identify all user stories, tasks, or work items for the sprint
- Ensure each item contributes to the sprint goal

Step 2: Estimate the Work

Estimate the effort required for each task and store the sizing in the **Work** field. This can be done using story points, task hours, or any other unit of measurement agreed upon by the team.

What to do:

- Work with the **skilled person responsible** for completing each task
- Assign realistic sizing based on their expertise and input
- Use consistent units (hours or story points) across all tasks
- Store estimates in the Work field for each blue Task bar

Why this matters: The Agilebars scheduling engine is modeled after the **Earned Value** concept, where progress is earned and task percent complete is calculated automatically based on sizing.

Step 3: Identify the Sprint Duration

Determine the duration of the sprint. Sprints typically last 1 to 4 weeks, and the length is defined during sprint planning.

What to do:

- Set sprint start and finish dates on the Green Project bar in Timescale mode
- Ensure all blue Task bars fall within the sprint timeframe
- Distribute tasks across the sprint timeline appropriately

Step 4: Complete a Progress Update

Collaborate with team members, gather the status, set the Report Date, and move bars to the correct stage based on input from the person doing the work.

What to do:

- Set **Report Date** to today (Tools > Canvas Settings > Report Date)
- Ask team members: "What lane should this task be in?"

- Move tasks between Kanban lanes based on actual progress
- Update daily during standup meetings

Progress Rules Reminder:

- Move from **Backlog** to **Will Do**: Earns no progress
- Move from **Will Do** to **Doing**: Earns 10% for planning effort
- Move from **Doing** to **Finalizing**: Earns 65% for work accomplished
- Move from **Finalizing** to **Done**: Earns 25% (total 100%) for final work accomplished

This process enforces updating progress, ensures accuracy, improves productivity, and achieves better results.

Set Report Date (Mandatory)

Every day that you will be moving a bar into lanes that drive progress (Doing, Finalizing, and Done), you **must** set the Report Date. Set it to match the date that the progress was given to you by your team.

How:

- Choose **Tools > Set Canvas > Report Date**
- Set to today's date
- Now start moving bars between Doing, Finalizing, and Done lanes

Why: The Report Date determines when actual start and actual finish dates are recorded, ensuring accurate historical tracking.

Burndown Charts

The burndown chart is Agilebars' primary reporting tool for visualizing sprint progress and forecasting completion.

What is a Burndown Chart?

A burndown chart shows the **amount of work remaining** over time during a sprint:

Two Lines:

1. **Ideal Trend Line (Planned)**: Diagonal line from total work at sprint start to zero at sprint end—represents perfect progress
2. **Actual Burndown Line (Forecast/Current)**: Shows actual remaining work based on task completion—updated as you move tasks between lanes

Axes:

- **X-Axis (Horizontal)**: Time (days in the sprint)
- **Y-Axis (Vertical)**: Remaining work (hours, story points, or sizing units)

Interpretation:

- **Below Ideal Line**: Team is ahead of schedule
- **On Ideal Line**: Team is on track

- **Above Ideal Line**: Team is behind schedule

How Burndown Charts Work in Agilebars

Agilebars uses **Earned Value Management** principles:

Key Concept: Progress is **earned**, not manually set. As tasks move through Kanban lanes, progress accumulates automatically, and work remaining decreases.

Data Sources:

1. **Task Sizing**: You assign hours or story points to each task in Timescale mode
2. **Lane Movements**: Moving tasks between lanes earns progress (10%, 75%, 100%)
3. **Time Phasing**: Dates assigned in Timescale mode distribute work across sprint timeline
4. **Calculation Engine**: Agilebars calculates remaining work for each day of the sprint

Why Time-Phasing Matters: Burndown charts are **time-phased**—they show work remaining by day. You must use **Timescale mode** to assign start/finish dates to tasks before the chart can accurately calculate daily remaining work.

How to Generate a Burndown Chart

Step-by-Step:

1. **Prepare Data in Timescale Mode**:

   - Switch to Timescale mode if in Kanban
   - Ensure all tasks have **start and finish dates** within the sprint
   - Verify all tasks have **sizing values** (hours or story points)
   - Tasks should span from sprint start to sprint finish

2. **Open Burndown Chart Page**:

   - Click on the **Project bar (green bar)** in either mode
   - In the popup, click the second row of text (e.g., "Pj:40")
   - This launches the Cost Schedule popup
   - Click **"Burndown Chart"** link
   - Burndown Chart page opens

3. **Calculate the Chart**:

   - Click the **"Refresh"** button on the Burndown Chart page
   - Wait for calculation (watch for popups showing progress)
   - When complete, you'll see green text: **"Current/Forecast Created!"**

4. **Run the Chart**:

   - Click **"Run"** button to render the chart
   - Chart displays with Ideal Line and Current/Forecast Line

5. **Create Baseline (First Time)**:
    - Under the **"Planned"** heading, click **[Create]** button
    - This saves the current forecast as the "Planned" baseline
    - Now you have two lines:
        - **Planned Line**: Your original sprint plan (won't change)
        - **Forecast Line**: Updates as you move tasks in Kanban mode

6. **Update Progress in Kanban**:
    - Switch back to Kanban mode
    - Move tasks between lanes as work progresses
    - Each movement updates progress and remaining work

7. **Refresh Chart**:
    - Return to Burndown Chart page
    - Click **"Refresh"** button again
    - Chart recalculates with updated progress
    - **Forecast Line** shows new actual progress
    - **Planned Line** remains unchanged (original baseline)

8. **Compare Lines**:
    - Compare Forecast Line to Planned Line
    - Assess if team is ahead, on track, or behind
    - Use insights for daily standups and sprint retrospectives

---

How to Read and Interpret the Burndown Chart

Understanding how to read the burndown chart is essential for making informed decisions during the sprint.

Chart Axes

Time Axis (X-Axis - Horizontal):

- Represents time in days or increments (e.g., Day 1, Day 2, etc.)
- Corresponds to the duration of the sprint
- Each point represents one day in the sprint timeline
- Starts at Day 0 (sprint start) and ends at final day (sprint end)

Work Remaining Axis (Y-Axis - Vertical):

- Represents the amount of remaining work in the sprint backlog
- Measured in your chosen units (story points, task hours, etc.)
- Starts high (total work at sprint start)
- Should decrease to zero by sprint end

Chart Lines

Ideal Trend Line (Planned Baseline):

- Diagonal line from the starting point to zero remaining work
- Represents the total work initially planned for the sprint (sum of all estimated tasks)
- Assumes the team completes an **equal amount of work each day**
- Shows perfect linear progress
- Remains **unchanged** throughout the sprint (represents original plan)

Actual Burndown Line (Current/Forecast):

- Updates as the sprint progresses
- Plots the actual amount of remaining work at the end of each day
- Connects the data points to form the actual burndown line
- **Recalculates** each time you refresh the chart
- Reflects real-world progress based on tasks moved to Done lane

Interpreting Progress

Compare the Actual line to the Ideal line:

Scenario	What It Means	Recommended Actions
Actual line below Ideal line	Team is **ahead of schedule**	Consider taking on additional scope if capacity allows; recognize team achievement
Actual line on Ideal line	Team is **on track**	Continue current pace; maintain focus on sprint goal
Actual line above Ideal line	Team is **falling behind**	Identify blockers; consider descoping; escalate if needed; adjust expectations
Actual line is flat (horizontal)	**No progress** being made	Critical: Investigate immediately; identify blockers; reassess sprint viability
Actual line drops sharply	**Burst of completions**	Positive: Tasks moved to Done; verify quality wasn't sacrificed for speed

Monitor Progress During the Sprint

Daily Monitoring:

1. **Update the chart daily** by refreshing after moving tasks in Kanban mode
2. **Compare actual to ideal** at each daily standup
3. **Identify trends early**—don't wait until the last few days
4. **Look for patterns**:
   - Consistent gap above ideal line = structural problem (under-capacity, poor estimates, blockers)
   - Spiky line = uneven progress (batch completions, dependencies)
   - Flat periods = work stalled (blockers, waiting on external dependencies)

Adjustments and Analysis

If Actual Line Consistently Falls Below Ideal:

- **Meaning:** Team is ahead of schedule
- **Analysis:**
  - Were estimates too conservative?
  - Did team increase capacity (overtime, additional resources)?
  - Were tasks simpler than expected?
- **Actions:**
  - Recognize and celebrate team performance
  - Consider pulling in additional work from backlog
  - Use this data to refine estimates for future sprints

If Actual Line Consistently Goes Above Ideal:

- **Meaning:** Team is falling behind
- **Analysis:**
  - Are estimates too optimistic?
  - Are there unexpected blockers or dependencies?
  - Is the team under-staffed or dealing with distractions?
  - Was scope added mid-sprint (scope creep)?
- **Actions:**
  - Identify and remove blockers immediately
  - Escalate issues to product owner or scrum master
  - Consider descoping—remove lower-priority work from sprint
  - Adjust sprint goal to reflect reality
  - Discuss root causes in retrospective

If Actual Line is Flat (No Progress):

- **Meaning:** Critical issue—no work is being completed
- **Analysis:**
  - Are there blocking dependencies?
  - Is the team waiting on external resources?
  - Are team members diverted to other priorities?
  - Is there a technical blocker preventing progress?
- **Actions:**
  - **Urgent:** Hold special meeting to identify blockers
  - Escalate to leadership if external blocker
  - Reassess sprint viability—may need to abort and replan
  - Document lessons learned for future sprints

Using Burndown Data for Decision-Making

Sprint Planning (Next Sprint):

- Review past burndown charts to understand team velocity
- Use average completion rate to plan realistic sprint capacity
- Adjust estimates based on historical accuracy

Daily Standups:

- Show burndown chart on screen during standup
- Use as conversation starter: "We're tracking above the ideal line—what blockers do we have?"
- Focus team on closing the gap or maintaining pace

Sprint Reviews:

- Present burndown chart to stakeholders
- Explain deviations from plan with evidence
- Build trust through transparency and empirical data

Sprint Retrospectives:

- Analyze why actual deviated from ideal
- Identify process improvements
- Celebrate when actual tracks ideal (predictability achievement)

Key Insights from Burndown Patterns

Pattern 1: Smooth Decline

- Actual line closely follows ideal line
- **Insight:** Excellent sprint execution, accurate estimates, minimal blockers
- **Action:** Maintain current practices; document what's working

Pattern 2: Step Function (Staircase Pattern)

- Actual line has sharp drops followed by flat periods
- **Insight:** Batch completions (e.g., all tasks done at sprint end)
- **Action:** Encourage continuous delivery; break tasks into smaller increments

Pattern 3: Late Surge

- Actual line flat for most of sprint, then drops sharply at end
- **Insight:** Procrastination, dependencies clearing late, or poor task breakdown
- **Action:** Improve task sizing; address dependencies earlier; encourage steady progress

Pattern 4: Early Plateau

- Actual line drops early, then flattens before sprint end
- **Insight:** Easy tasks done first, hard tasks left for later; or sprint over-committed
- **Action:** Better task sequencing; pair easy and hard tasks throughout sprint

Next Steps: Continuous Improvement

After Reading the Chart:

1. **Take action** based on what the chart reveals
2. **Communicate findings** with the team and stakeholders
3. **Adjust plans** if necessary (descope, add resources, remove blockers)
4. **Document insights** for retrospective discussion

5. **Refine estimates** for future sprints based on empirical data

Remember: The burndown chart is a visual representation of the team's progress, and it should be **updated regularly** to reflect the current state of the sprint. It is a valuable tool for transparency, communication, and identifying potential issues during the sprint.

Advanced Burndown Features

Additional Baselines:

- Create multiple baselines to compare different scenarios
- Under **"Additional Baselines"** heading, click buttons to create more
- Use radio buttons under **"Compare Current/Forecast to..."** to switch between baselines

Use Cases:

- Baseline 1: Original sprint plan
- Baseline 2: Mid-sprint re-plan after scope change
- Baseline 3: End-of-sprint final for retrospective analysis

Best Practices for Burndown Charts

1. **Size Tasks Accurately**:
   - Involve the person doing the work in sizing
   - Use consistent units (hours or story points)
   - Re-estimate if new information emerges
2. **Update Daily**:
   - Move tasks to correct lanes during daily standups
   - Set Report Date to today's date before moving tasks
   - Refresh burndown chart daily to track progress
3. **Time-Phase in Timescale**:
   - Assign realistic start/finish dates in Timescale mode
   - Distribute work evenly across sprint days
   - Avoid bunching all work at sprint start or end
4. **Use Empirical Data**:
   - Base decisions on actual progress (what has happened)
   - Trust the burndown chart if sizing and updates are accurate
   - Use chart to forecast sprint completion realistically
5. **Create Baseline Before Starting**:
   - Run chart and create Planned baseline **before** sprint work begins
   - This preserves original plan for comparison

- Update forecast as sprint progresses

What the Scrum Guide Says

From the 2020 Agile Scrum Guide:

> "Various practices exist to forecast progress, like burn-downs, burn-ups, or cumulative flows. While proven useful, these do not replace the importance of empiricism. In complex environments, what will happen is unknown. Only what has already happened may be used for forward-looking decision making."

What This Means:

- Burndown charts are **proven useful** but not mandatory
- **Empiricism** (evidence-based decisions) is paramount
- Use actual data (what has happened) for forecasting
- Don't rely solely on predictions—adapt based on reality

What Timebars Ltd. Says

Empiricism means basing decisions on real-world observations and data, not just forecasts.

However: If you take the time to:

1. **Size each work item** with the skilled person doing the work
2. **Update progress** accurately based on actual completion
3. **Use continuous assessment** and adjustment

...then you can **trust and rely on burndown charts** as the primary tool for predicting sprint finish.

Key: Accurate sizing + accurate progress tracking = reliable burndown forecasting.

Kanban Board KPIs

While the burndown chart visualizes progress, several **Key Performance Indicators (KPIs)** can be derived from Kanban and burndown data:

1. Velocity

Definition: The amount of work a team completes in a sprint

How Agilebars Helps:

- Burndown chart shows how much work gets done over time
- Sum of completed tasks (in Done lane) = sprint velocity
- Track velocity across sprints to predict future capacity

Use:

- Sprint planning: "We averaged 40 story points per sprint, so plan 40 for next sprint"

- Capacity forecasting for release planning

---

2. Sprint Progress

Definition: Percentage of sprint work completed at any point in time

How Agilebars Helps:

- Burndown chart shows work remaining vs. total work
- Progress = (Total Work - Remaining Work) / Total Work
- Visual comparison: Forecast line vs. Planned line

Use:

- Daily standups: "We're 60% done with 5 days left in the sprint"
- Early warning if team is falling behind

---

3. Resource Utilization

Definition: How effectively the team's capacity is being used

How Agilebars Helps:

- Steep burndown decline = team may be overburdened
- Flat burndown line = team may have capacity issues or blockers
- Lane distribution shows work-in-progress across stages

Use:

- Identify if team is over/under-allocated
- Balance workload across team members
- Detect bottlenecks (e.g., too much work stuck in Finalizing lane)

---

4. Predictability

Definition: Consistency in delivering planned work within sprints

How Agilebars Helps:

- Compare Forecast line to Planned line across sprints
- Large deviations = low predictability
- Forecast line tracking Planned line = high predictability

Use:

- Build stakeholder trust with consistent delivery
- Identify causes of variability (scope creep, poor estimates, external dependencies)
- Improve estimation accuracy over time

Agilebars Scheduling Engine

The Agilebars scheduling engine is unique among the Timebars product line, designed specifically for **Agile Earned Value Management**.

How It Works

Core Principle: Progress is **calculated**, not manually entered.

Workflow:

1. **Size Tasks**: Assign hours or story points to each task
2. **Move Between Lanes**: Drag tasks through Kanban workflow
3. **Earn Progress Automatically**: System calculates progress based on lane (10%, 75%, 100%)
4. **Calculate Remaining Work**: Total Work - Earned Progress = Work Remaining
5. **Generate Burndown**: Chart shows remaining work over time

No Manual Entry:

- You **never** manually set "percent complete" on tasks
- You **don't** track actual hours worked
- You **don't** bug developers for time estimates daily

Simple Questions: During daily standups, ask:

- Did the task start? → Move to Doing
- Is development done? → Move to Finalizing
- Is it completely done? → Move to Done

Agilebars does the rest of the calculations based on internal rules.

Benefits of Earned Value Approach

1. **Reduced Overhead**: No time tracking, no manual progress entry
2. **Objective Progress**: Based on workflow stage, not subjective estimates
3. **Consistent Reporting**: All tasks follow same progress rules
4. **Focus on Workflow**: Teams focus on moving work forward, not admin
5. **Reliable Forecasting**: Burndown charts reflect actual workflow progress

---

Tips for Agile Scrum Teams

Sprint Planning

1. **Groom Backlog in Timescale**:

    - Create all tasks in Timescale mode
    - Assign sizes (story points or hours)
    - Estimate start/finish dates for the sprint

2. **Switch to Kanban for Commitment**:

- Move to Kanban mode
- Drag committed tasks from Backlog → Will Do
- Visualize sprint capacity

3. **Create Baseline**:
   - Generate burndown chart
   - Click [Create] under Planned heading
   - Save original sprint plan

Daily Standups

1. **Use Kanban Board**:
   - Display on large screen during standup
   - Visually see all work in progress
   - Quickly identify blockers
2. **Update Report Date**:
   - Set Report Date to today (Tools > Canvas Settings > Report Date)
3. **Move Tasks**:
   - Update task positions based on team feedback
   - Drag completed work to Done lane
4. **Check Burndown** (Optional):
   - Refresh chart to see daily progress
   - Share with team if significantly off track

Sprint Retrospectives

1. **Review Burndown Chart**:
   - Compare Planned vs. Forecast lines
   - Discuss why forecast deviated from plan
   - Identify improvement areas
2. **Calculate Velocity**:
   - Sum story points completed (tasks in Done lane)
   - Compare to previous sprint velocities
   - Adjust next sprint capacity
3. **Analyze KPIs**:
   - Velocity: Are we consistent?
   - Predictability: Did we deliver what we committed?
   - Resource Utilization: Were we overburdened or underutilized?

Sprint Reviews

1. **Show Burndown Chart**:
    - Demonstrate sprint progress to stakeholders
    - Explain deviations from plan
    - Build trust with empirical data
2. **Export Data** (Optional):
    - Use Spreadsheet Sync to export sprint data
    - Analyze in Excel for reports
    - Archive sprint data for historical tracking

Common Help Topics

For features shared across Agilebars, Timebars, and Costbars, refer to these comprehensive guides:

Core Functionality

- Common User Interface Guide - Bar Creator, Canvas, Shortcuts, Tools
- Common Data Structure User Guide - Data hierarchy, IndexedDB, backups

Data Management

- Common Spreadsheet Sync User Guide - Bulk data import/export
- Common Risks Issues Change Requests User Guide - RIC management

Reporting

- Common Local Reports and Graphs Guide - In-app reports
- Common Cloud Reports and Dashboard Guide - Cloud dashboard

Cloud Features

- Common Cloud Publishing Guide - Publishing to Timebars Cloud

Quick Reference

Kanban Lane Progress Rules

Lane	Progress	Actual Start	Actual Finish
Backlog	0%	No	No
Will Do	0%	No	No
Doing	10%	☑ Set	No
Finalizing	75%	-	No

Lane	Progress	Actual Start	Actual Finish
Done	100%	-	☑ Set

Hierarchy

- **L2 (Green)**: Projects / Sprint Backlogs
- **L4 (Blue)**: Tasks / Work Items / User Stories

Burndown Workflow

1. Size tasks in Timescale mode
2. Switch to Kanban mode
3. Generate chart (Refresh → Run)
4. Create baseline ([Create] under Planned)
5. Move tasks between lanes as work progresses
6. Refresh chart to see updated forecast

Support

- **Email**: jcox@tbcox.com
- **Knowledge Base**: www.timebars.com/knowledgebase
- **Product Site**: www.timebars.com/agilebars
- **Phone**: (613) 255-5374

Agilebars empowers Agile teams to plan sprints, track progress visually, and deliver predictably—all with minimal overhead and maximum transparency.