



Agile Training Series

Pitfalls of Using Velocity for Planning

Jonathan Jenkins

October 2016

About Jonathan AKA Jarhead or Gunny

Agile

- Agile Architect / Coach
- Scaled Agile Framework Consultant (SPC)
- Certified Scrum Professional (CSP)
- Certified Scrum Master (CSM)
- Practitioner of
 - eXtreme Programming (XP)
 - Kanban
 - Marine Corps Maneuver Warfare

Professional

- Professor Villanova University
- 20+ years software/database development
- 15+ years project management
- Microsoft SharePoint Administrator / Developer
- Database Administrator (MS SQL)
- Active Clearance (TS/SCI)
- Former Political Consultant

Education

- B.S. Computer Science
- M.S. Information Technology (Upsilon Pi Epsilon)
- MBA (Ongoing)
- Certificate in Visual Basic

Service

- Gunnery Sergeant of Marines
- Iraq War Veteran
- Air Traffic Controller
- Marine Air Ground Task Force Planner

Working Agreements

- Phones on silent or off
- Respect the speaker
- Everyone has a voice but only one voice at a time
- No sidebar conversations
- Respect the time box
- Give fast feedback

Learning Objectives

- Define the intent of the Velocity measure
- Know the expected benefits of Velocity
- Understand why teams shouldn't focus on meeting their Velocity
- Identify disruptions to the estimating and planning processes
- Approach to starting out a new team

Enabling Objectives

- Common Language
- Common Definitions
- Generally Accepted Practices / Techniques

Velocity

The 40,000' View and General Perception

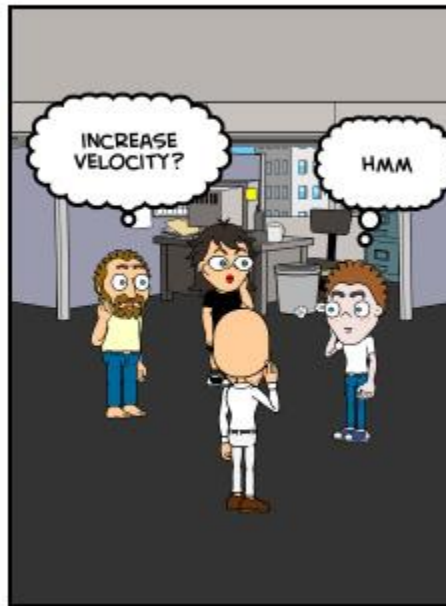
The 40,000' View

- Sum of estimates associated with work completed over a period of time
 - Used as a planning technique
 - For individual teams
 - Helps determine how much stuff they can be delivered
 - Over the next period of time
- Lagging indicator not a leading indicator
- 2000 – Mention in context of eXtreme Programming
- 2002 – Scrum community adopts (Go XP!)

Common Thoughts on Velocity

- Metric to determine what a team can deliver in a Sprint
- Team caps their commitment based on the team's velocity
- Velocity increases as team matures

OBSERVER EFFECT - AGILE



BY ONTIMENOW.COM



WWW.BITSTRIPS.COM

Common Language & Definitions

| Word / Term | Comment |
|------------------|--|
| Sprint | Timebox usually measured in weeks. Same as Iteration |
| Release | Timebox of more than 1 Sprint |
| Commit | Obligate yourself to do the best job possible. Not a guarantee |
| Guarantee | Promise to do something – no exceptions |
| Ideal Day | 8 hours a day |
| Man Day | Generally, 5 to 6 hours a day |

| Word / Term | Comment |
|-----------------------------|---|
| Definition of Ready | The “stuff” the team needs to know in order to have confidence in committing to work on “something” |
| Definition of Done | The quality and activities the team needs to do in order for work to be completed |
| Levels of Completion | 0% or 100% |
| Points | A technique to estimate other than traditional methods (hours, days, months, etc.). |
| Stories | A “Just-in-Time” business requirement |
| Work Item | Story, technical story, defect, spike, etc. |

Relative Estimations

- Estimates don't need to be accurate, just consistent
- Estimates are mean to be relative to other story estimates
 - Compare to other stories from the past
 - Evaluate consistently (reference stories)
- Velocity corrects for the inaccuracies of our estimates
- Instead of time utilize the ABC's



Ambiguity

Bigness

Complexity

Relative Estimations



Ambiguity

Bigness

Complexity

- Example using Fibonacci 0, 1, 2, 3, 5, 8, 13, 21, etc.
 - A work item of size 3 is 3x that of a size 1
 - A work item of size 3 is 50% more than that of a size 2
 - The scale is about size **NOT** effort
- Some of my theories
 - It is likely a team can complete 5 work items of size 2 before a single work item of size 8
 - Velocity is more predictable when teams consider ABC for each work item
 - Velocity is more predictable when team consistently decomposes work items

Velocity

- Sum of estimates associated with work items completed in a given timebox
- Used as a planning technique by teams to determine how much stuff they can do in their next timebox

| Work Item | Size | Status |
|-----------|------|----------|
| ABC | 5 | Done |
| XYZ | 3 | Done |
| PDF | 3 | Not Done |
| DSF | 1 | Done |

What is the team's velocity?

Velocity

- Lagging indicator not a leading indicator
- Measured in the same units as work item estimates
 - Story points
 - Man days
 - Ideal days
 - Hours
- If estimated backlog equals 70 then it would take ~6 Sprints

Expected Benefits

Expected Benefits

- Consistent approach to planning
- Able to track how a team is doing in a Sprint (Sprint burndown)
- Able to plan out the delivery for an entire project

| Backlog Size | Velocity | Sprints to Complete |
|--------------|----------|---------------------|
| 50 | 10 | 5 |

The Reality

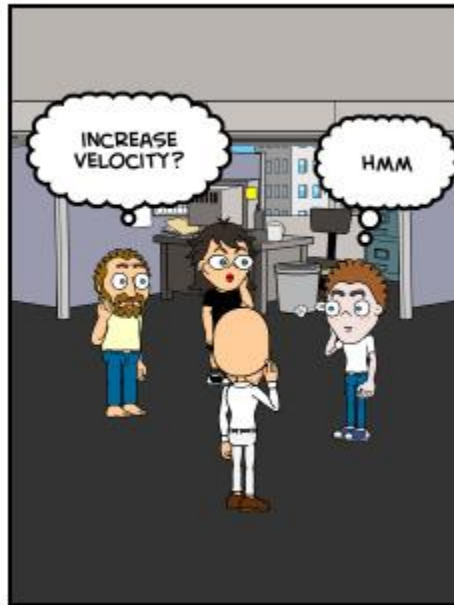
The Reality – External Influences

- Teams Change
 - People leave or join
 - Change roles with the team
 - Vacations, sick days, unexpected leave
- New technologies continually get introduced
- Work in new line of business without institutional knowledge

The Reality – Internal Influences

- Teams do not utilize a DoR and/or DoD
- Teams do not refine stories to have them “Sprint Ready”
- Stories not ready when Sprint starts
- Generally takes 4 to 6 Sprints for a team to start finding their rhythm

OBSERVER EFFECT – AGILE



BY ONTIMENOW.COM



WWW.BITSTRIPS.COM

The Reality – Internal Influences

- Sprint disruptions
 - Production support
 - Continuous “Drive byes” from non-team members
- Lack of focus
 - Evolving requirements in the Sprint
 - Shift of in-Sprint priorities
- Work not done (reasons)
 - It was not started
 - It failed testing
 - DoD not met
- Inconsistent approach to estimating
 - Single person provides the estimate (product owner, analysts, dev lead, etc.)
 - Non-delivery team members estimate
 - Management forces team to reduce / increase their estimates

Controversy

A Little Controversy

- Points are for long-term measure not necessarily Sprint planning
- If capacity remains then add more work into the Sprint for them
- Focus on maximizing throughput and capacity not Velocity
- Average velocity means nothing

Suggestions

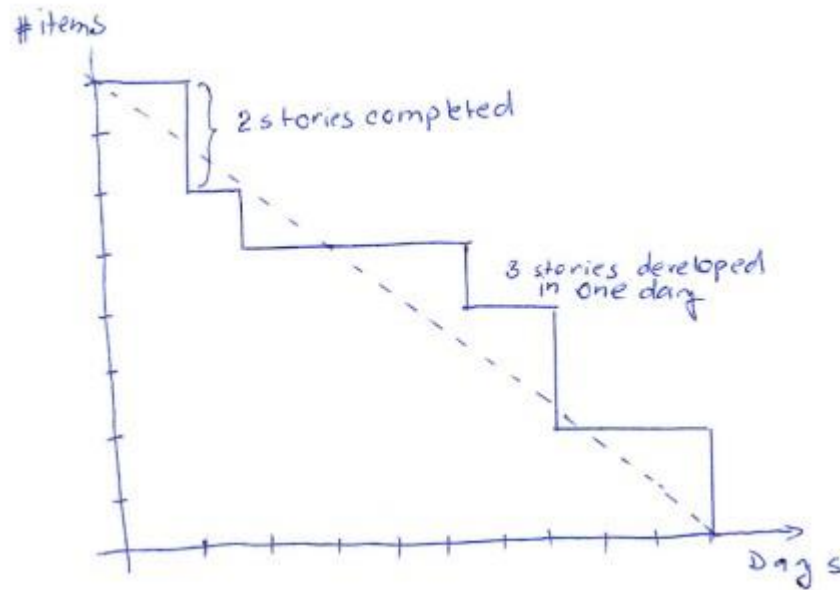
Suggestions

- Document the context of what was / was not delivered
- Don't use just Velocity
 - Utilize capacity planning as well
 - Take into account context of previous Sprint(s)
- Don't utilize average velocity
- Eliminate / reduce overtime
- Focus on root cause of not being able to meet all commitments
- Utilize retrospectives to discuss commitments, capacity, and Velocity
- Keep working on improving work items so they are small enough to deliver within a couple days
- Maintain a quality mind-set
- Utilize DoR before a work item is associated to a Sprint
- Utilize DoD when estimating a work item
- Ensure Sprint plans keep everyone employed each day
- Build story lists

Burndown Charts

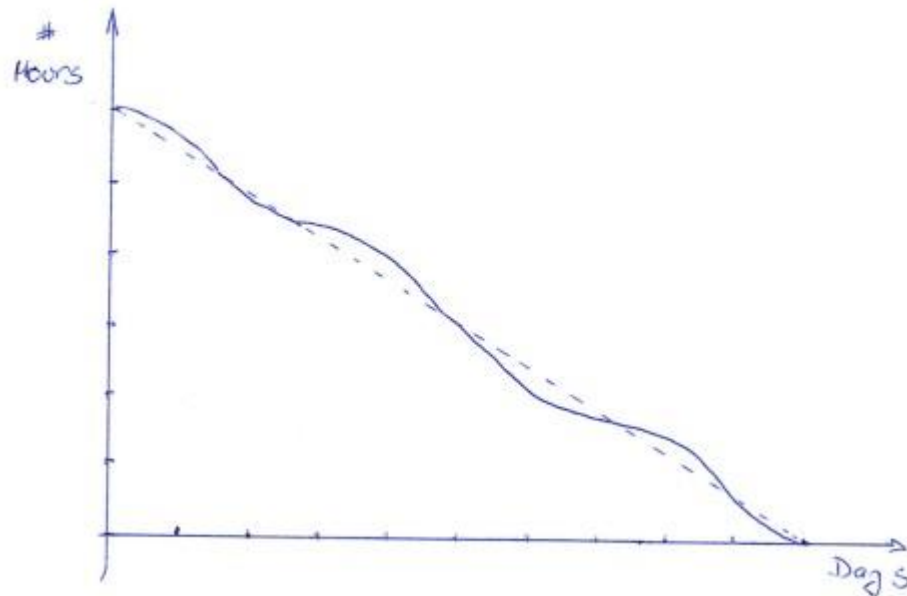
Burndown – Number of Stories

- Vertical (up and down) – number of stories
- Horizontal (across) – timebox
- Not a good approach as items has no relation to effort or size



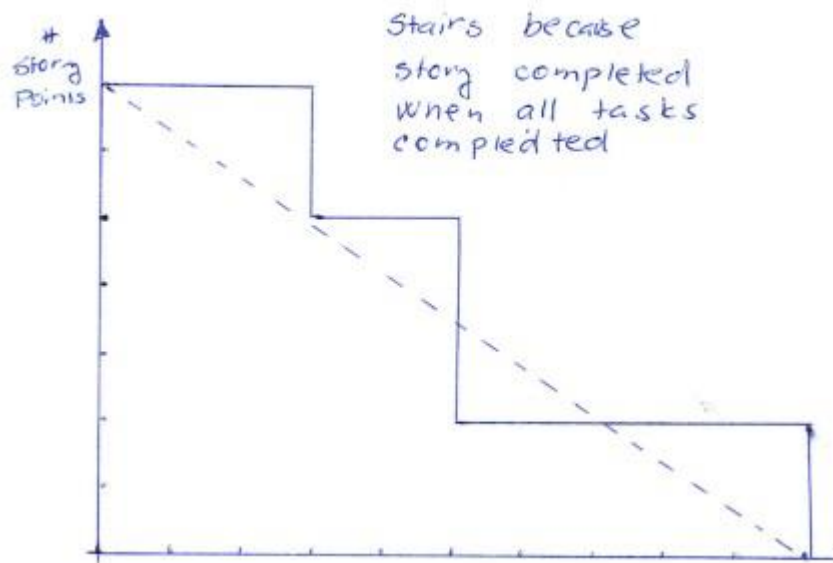
Burndown – Time

- Vertical (up and down) – hours
- Horizontal (across) – timebox
- Shows remain time necessary for completion
- Better than items as its more granular but can lead to micromanagement
- Good as can show spikes
- Good for new teams



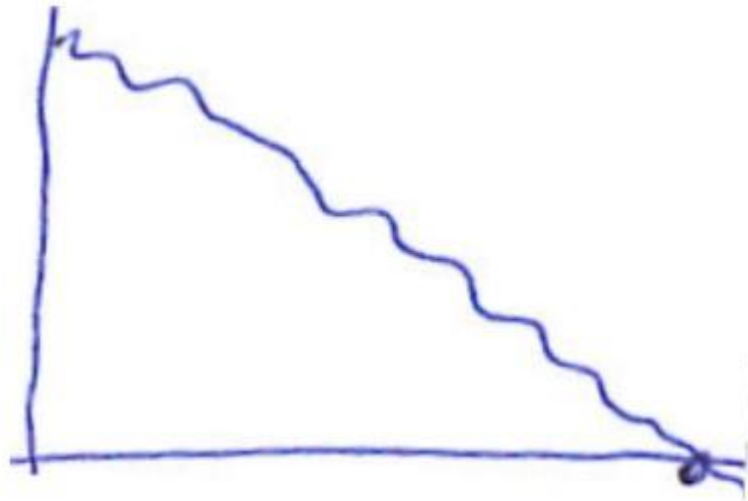
Burndown – Remaining Size

- Vertical (up and down) – story points (not velocity?)
- Horizontal (across) – timebox
- Shows remaining size of all stories to be completed
- Better than items as its more granular but can lead to micromanagement
- Good as can show when change made to Sprint backlog



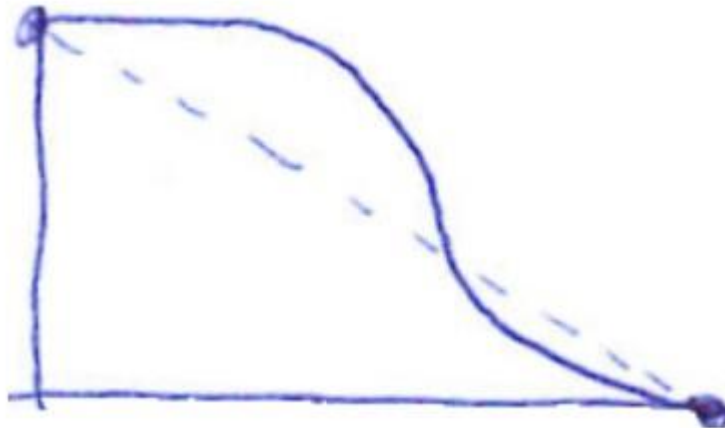
Burndown – Ideal Team

- Demonstrates the team is able to quickly adjust
- Inference is that the Sprint backlog did not change
- Inference that the Scrum Master is helping keep noise away from team
- Corrective Action: None



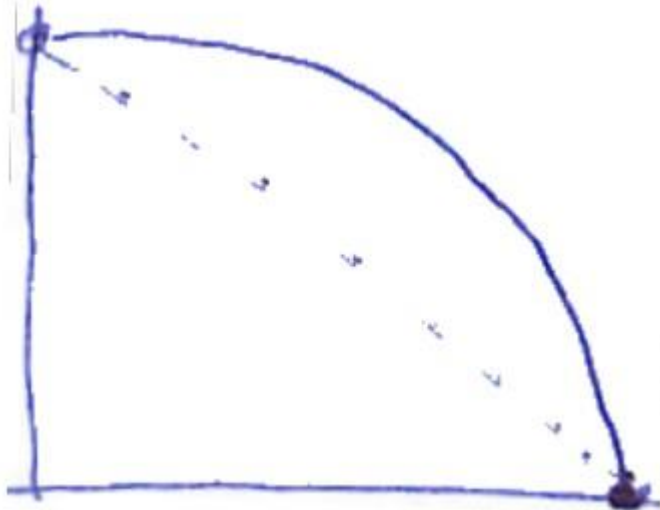
Burndown – Great Team

- Demonstrates the team is able to self-organize self-manage
- Team met the Sprint goals while also taking on more work than their velocity
- Corrective Action:
 - Utilize retrospective for them to understand why the late progress in 1st half
 - Inspect if capacity throughout the Sprint is sufficient



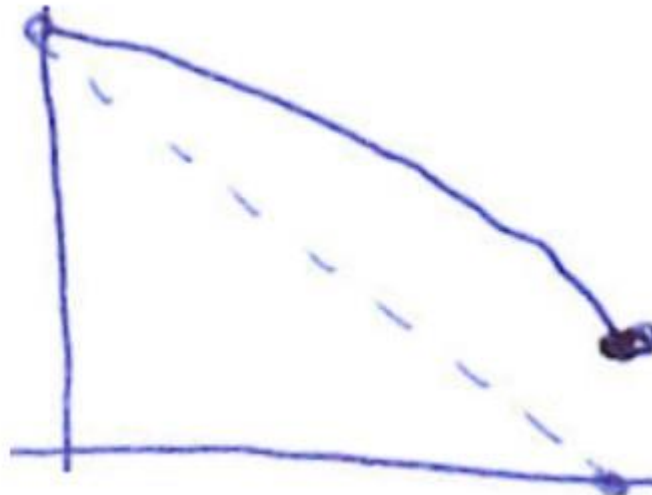
Burndown – Nice Team

- Typical of many teams
- Team was able to meet commitments on time
- Either adapted their scope or put in overtime
- Appears they may be self-reflecting
- Corrective Action:
 - Likely the teams needs to talk more when plan appears to be going awry
 - Ensure items remain prioritized so lowest value ones can be pushed out of Sprint



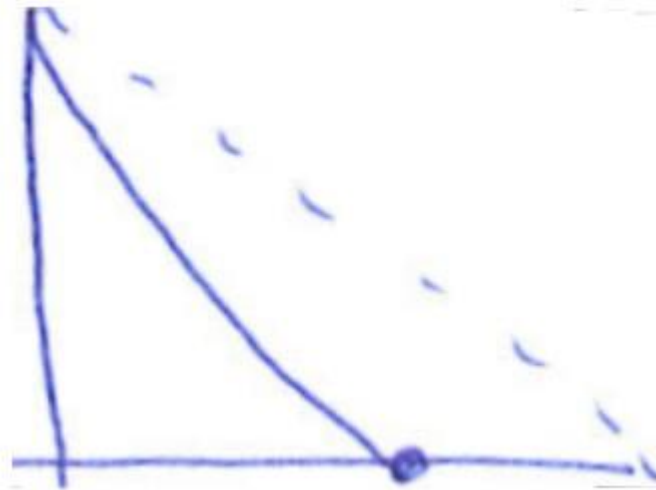
Burndown – Grrrrr, Its too Late

- Team did not meet their commitments
- Team was late the entire Sprint
- Team is unable to adapt
- Corrective Action:
 - Inspect their refinement and planning practices / activities
 - Likely the teams needs to talk more when plan appears to be going awry
 - Ensure items remain prioritized so lowest value ones can be pushed out of Sprint
 - Reduce scope



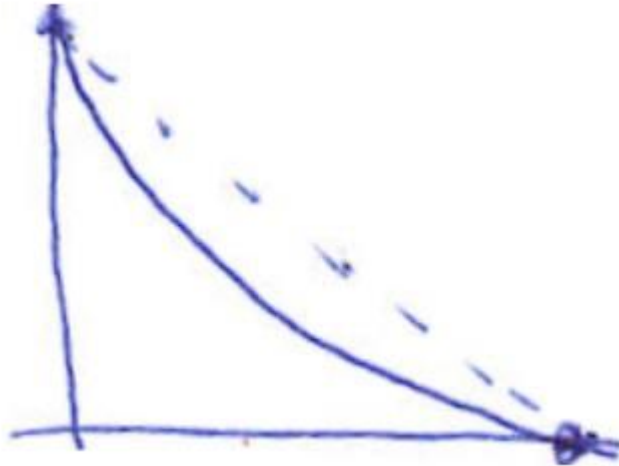
Burndown – Grrrrr, Its too Early

- Team finishes their work earlier than expected
- Team did not take in additional stories despite having the capacity to do so
- Stories may have been over estimated
- Velocity and capacity planning don't seem to be known
- Corrective Action:
 - Reboot the team if this repeats next Sprint
 - Engage the business and Scrum Master as Sprint kicks off



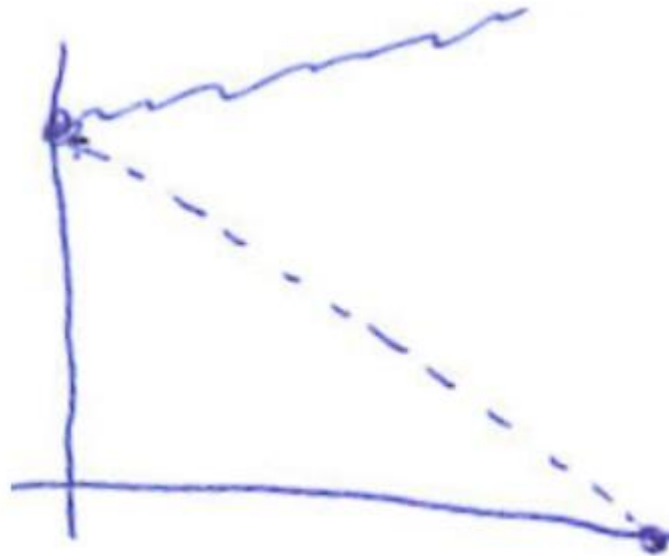
Burndown – Planning? Estimating Problems?

- Could be that the team committed to less than they could do
- Could be that the Product Owner didn't have enough stories
- Could be that the team over estimated the complexity of items
- Corrective Action:
 - Work with team for more productive refinement and estimating activities
 - Ensure Scrum Master understands they need to pull in more work items



Burndown – OMG

- Team got nothing done
- Team just kept adding more work
- Team did not mark progress potentially
- Team never identified they had a problem
- Corrective Action:
 - Provide team with coaching and mentoring for a few Sprints
 - Focus on the team solving its own problems



Quiz Time!

The Reality – External Influences

- Is velocity a leading indicator or a lagging indicator?
- What frameworks / methods could benefit with velocity?
- Is velocity about size or effort?
- What are the ABC's?
- What is capacity planning?

Questions? Comments?