

Getting the Most Value from Agile Feedback Systems:

Every Day, Every Sprint and
Every Release

Satish Thatte, PhD
SPC, CSM, CSPO, CSP

CEO, New Synergy Group

TriAgile 2016 Conference
Raleigh, NC
30 June 2016

About the Presenter

Satish Thatte



- **Agile/Lean Coach for 7 years**
 - New Synergy Group for 3 years (CEO and Founder)
 - VersionOne for 4 years
- **30+ years of technology and management experience**
 - Director at large, multinational companies: Texas Instruments, Bellcore, LG Electronics
 - VP of Engineering at start-up companies coaching and managing agile development teams
- Certified ScrumMaster (CSM), Certified Scrum Product Owner (CSPO), Certified Scrum Practitioner (CSP)
- SAFe Program Consultant (SPC)
- Over 70 client companies trained and coached
- M.S. and Ph.D. in Electrical Engineering from the University of Illinois at Urbana-Champaign
- 14 patents (13 US and one International)
- Location: Princeton, NJ



smthatte@gmail.com



@smthatte



<http://www.linkedin.com/in/satishthatte>

Blog: http://blogs.versionone.com/agile_management/author/sthatte/

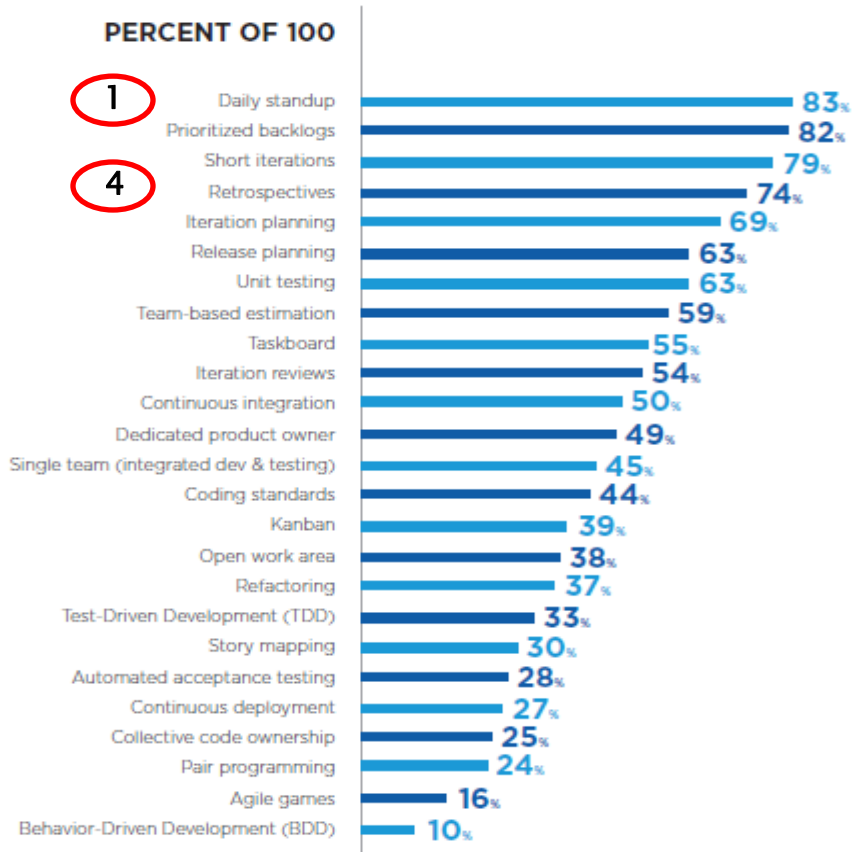
Popular Agile Methods



AGILE METHODS AND PRACTICES

Agile Techniques Employed

Continued...



Making Daily Scrums Really Effective and Efficient:

<http://bit.ly/ODMybj>

The Blog template downloaded over 1,000 times

Making Sprint Retrospectives really Effective:

<http://bit.ly/R0Kyfa>

The blog template downloaded over 300 times

Making Release Retrospective Strategic and Effective:

<http://bit.ly/GYGePS>

The blog template downloaded over 100 times

Strong “Full-Spectrum” Feedback is Essential for Agile to Work

- Agile methods are empirical: inspect and adapt
- Daily feedback system: benefit team members and teams
- Sprint feedback system: benefit teams and projects
- Release feedback system: benefit project, programs and organization
- Double feedback loops: Primary and Derivative feedbacks
- Many agile teams have dysfunctional feedback systems
 - No or delayed primary feedback
 - *Gray Light* Feedback: Blurred or noisy or mixed feedback signal
 - Feedback without relevant context
 - No derivate feedback: Little learning or continuous improvements
- The key to improving feedback systems
 - Strong *Full-Spectrum* feedback
 - Visual double feedback loops through boards, reports and metric

Feedback Framework: Evidence, Context, Consequence, Action

Feedback Loop in Action: An Example that Works Real Well

4. Action: Drivers slow down 10% - usually for several miles

1. Evidence: Car speed

Driver
Feedback

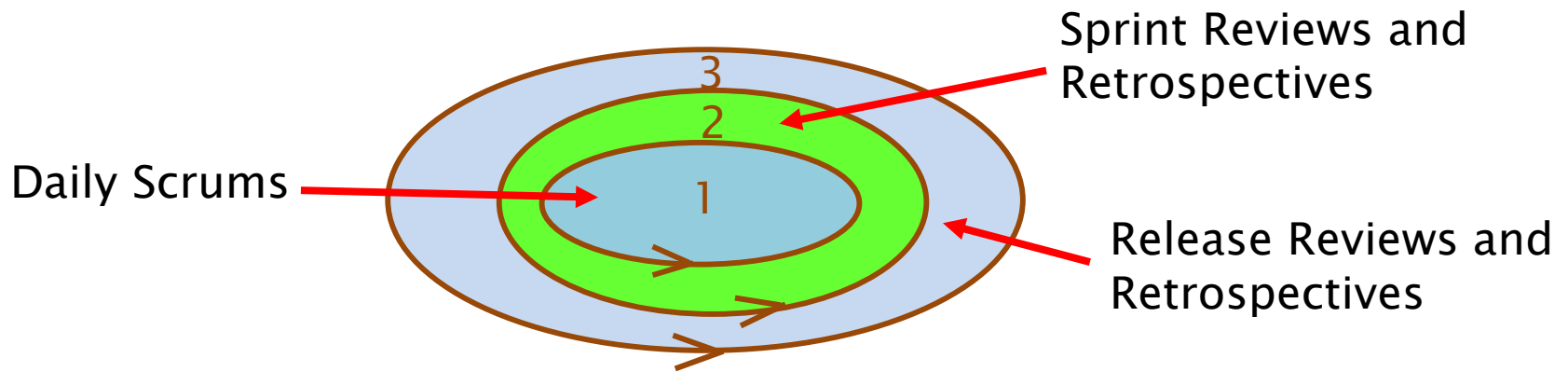
3. Consequence: Cops, traffic tickets, potential accidents

2. Context: Legal speed limit



Adapted from: "The feedback loop" by Thomas Goetz, *Wired* magazine, July 2011

Framework for Effective Agile Feedback Systems



- Does your agile system have Feedbacks or Open Loops?
- What's the quality of feedback?
 - Accurate or Ambiguous, Strong or Weak, Fast or Delayed
- How to improve the effectiveness of feedback?
 - Context, Consequence, Actions
- Cost and ease of use of feedback system

Daily Feedback System is experienced through Daily Scrums

Ineffective Daily Scrum

- What did you **do** since the last Daily Scrum?
 - Lots of development
- What will you **do** until the next Daily Scrum?
 - Will fix a bunch of defects
- What are the likely impediments?
 - Too many meetings, such as this Daily Scrum!
- Quality of feedback:
 - Ambiguous, Weak
- Very Weak in:
 - Evidence, Context, Consequence, Actions

Effective Daily Scrum

- What did you do since the last Daily Scrum **against your commitment?**
 - Completed implementation of US 37, but **could not complete** all unit testing.
- What's your **commitment** until the next Daily Scrum?
 - Complete unit testing of US37, and
 - Fix Defects 367, 431
- What do you need from your team members?
 - Need Joe (analyst) to explain US46
- What are the likely impediments?
 - Awaiting clarification on US59 from Dave (Product Owner)
- Quality of feedback: Accurate, Timely
 - Strong in: Evidence, Context, Consequence, Actions - **if in Visual Form**

Daily Scrum Taskboard: *Gray Light* Feedback Weak Feedback, Not Very Effective

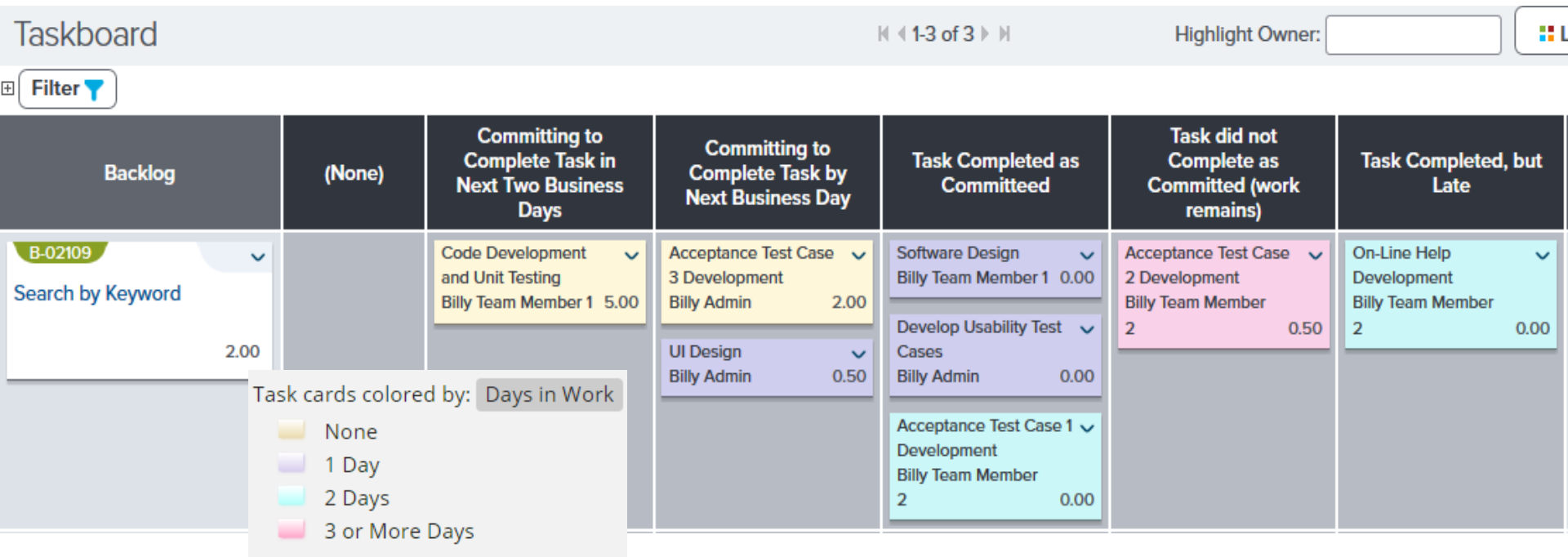
Taskboard 1-3 of :			
Filter			
Backlog	(None)	Task in Progress	Task Completed
<p>B-02083</p> <p>Search by Keyword</p> <p>2.00</p>	<p>Code Development and Unit Testing Robert Team Member 1 5.00</p> <p>Acceptance Test Case 3 Development Robert Admin 2.00</p>	<p>UI Design Robert Admin 0.50</p> <p>Acceptance Test Case 2 Development Robert Team Member 2 0.50</p>	<p>Software Design Robert Team Member 1 0.00</p> <p>Develop Usability Test Cases Robert Admin 0.00</p> <p>Acceptance Test Case 1 Development Robert Team Member 2 0.00</p> <p>On-line Help Development Robert Team Member 2 0.00</p>

Weak feedback between the **tasks** planned and presented **in the last Daily Scrum** and the **tasks actually completed since the last Daily Scrum**. **Feedback is not contextual**.

Tasks planned and presented **in today's Daily Scrum** may not drive **tasks that will be actually completed until the next Daily Scrum**

Weak in Evidence, Context, Consequence, and Actions on tasks for planning, completing and tracking.

Full-Spectrum Taskboard with Strong Feedback: Highly Effective and Efficient Daily Scrums



Strong feedback between the **tasks committed** to complete in the last Daily Scrum and the **tasks actually completed** since the last Daily Scrum. **Feedback is very contextual.**

Tasks committed in today's Daily Scrum will drive the **tasks that will be actually completed** until the next Daily Scrum. **Work on Tasks is driven by commitments made.**

Strong in Evidence, Context, Consequence, and Actions on tasks for planning, completing and tracking.

Using Colored Task Cards (to indicate the number of days a task is in progress: 1, 2, or 3 or more) gives insightful and actionable feedback.

Daily Scrum Testboard: *Gray Light* Feedback Weak Feedback, Not Very Effective

Testboard 1-3 of 3

Filter

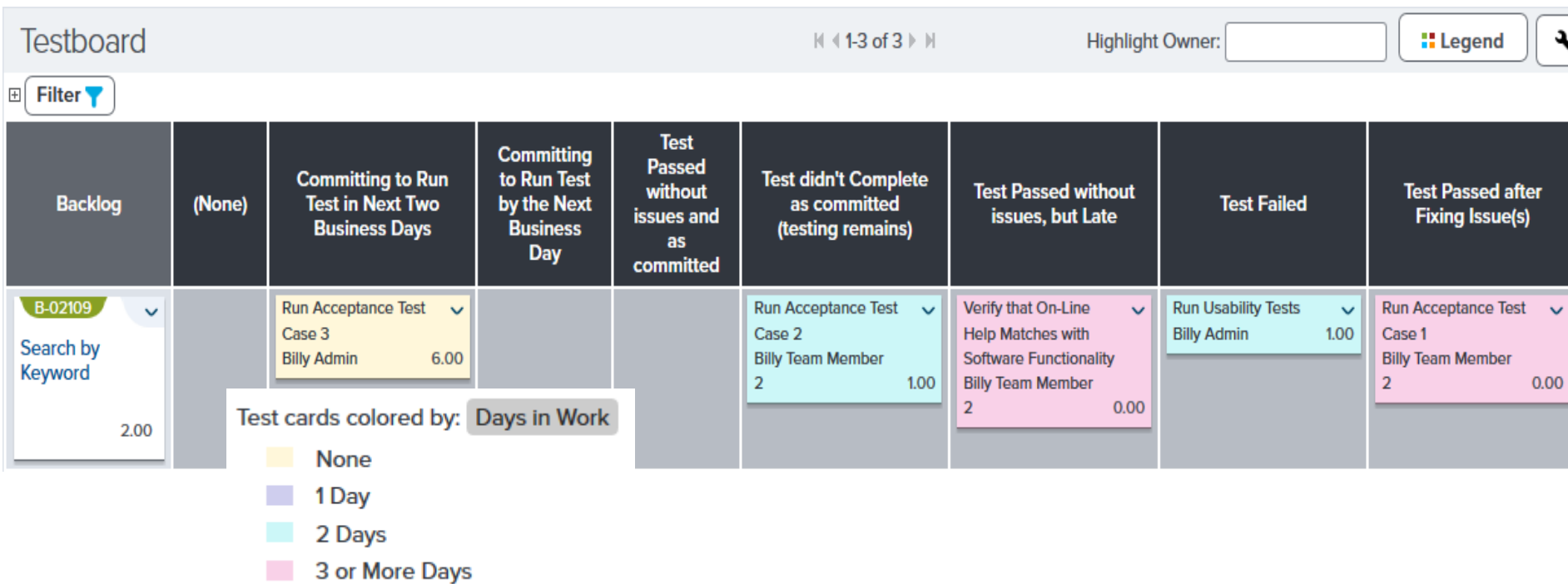
Backlog	(None)	Test in Progress	Test Failed	Test Passed
<p>B-02083</p> <p>Search by Keyword</p> <p>2.00</p>	<p>Run Acceptance Test 3</p> <p>Robert Admin 6.00</p>	<p>Run Acceptance Test 2</p> <p>Robert Team Member 2 1.00</p>	<p>Run Usability Test Cases</p> <p>Robert Admin 1.00</p>	<p>Run Acceptance Test 1</p> <p>Robert Team Member 2 0.00</p> <p>Verify that on-line help matches with software functionality</p> <p>Robert Team Member 2 0.00</p>

Weak feedback between the **tests** planned and presented **in the last Daily Scrum** and the **tests that were actually run and completed (passed or failed)** since the last Daily Scrum. Feedback is not contextual.

Tests planned and presented **in today's Daily Scrum** may not drive the **tests that will be actually run and completed until the next Daily Scrum**.

Weak in Evidence, Context, Consequence, and Actions on tests for planning, running and tracking.

Full-Spectrum Testboard with Strong Feedback: Highly Effective and Efficient Daily Scrums



Strong feedback between the **tests committed** to run in the last Daily Scrum and the tests actually run and completed (passed or failed) since the last Daily Scrum. **Feedback is very contextual.**

Tests committed to run in today's Daily Scrum will drive the tests that will be actually run until the next Daily Scrum (they may pass or fail). **Work on Tests is driven by commitments made.**

Strong in Evidence, Context, Consequence, and Actions on tests for planning, running, tracking.

Using Colored Test Cards to indicate the number of days a test is in progress: 1, 2, or 3 or more) gives insightful and actionable feedback.

Time-Flow Machine: Strong Full-Spectrum Story-Level Feedback

Example of a Struggling Agile Team, 4-Week (20 Workdays) Sprint

Back log	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1, 1		A	D	C	NT	C	TD	NT	TE	IMP	TE	NT	NT	TE	NT	DF	AT				
2, 3	S p r i n t P l a n n i n g	A	D	C	NT	C	IMP	C	NT	TD	NT	TE	NT	TE	NT	DF	NT	AT			
3, 2		A	A	IMP	D	NT	C	C	TD	NT	TD	NT	TE	IMP	TE	NT	DF	DF	AT		
4, 3			A	D	D	C	NT	C	C	C	TD	IMP	TD	NT	TE	NT	TE	DF	NT	AT	
5, 2					A	D	C	NT	C	C	TD	IMP	IMP	TE	NT	TE	NT	DF	NT		
6, 3						A	A	D	NT	C	NT	C	C	IMP	C	TD	TE	NT	TE		
7, 2					A	A	NT	D	C	C	IMP	IMP	TE	NT	TE	TE	TE	TE	DF		
8, 1									A	A	D	D	C	NT	C	C	NT	C	TD		

Micro Waterfalls

WIP = 8

Cycle time = 18

Little's Law :
 Ave. Cycle Time
 = Ave. WIP / Ave. Throughput
 = (68/18) / (4/18) = 68/4 = 17 days

Planned work = 17 SP
 Not Accepted = 8 SP
 Velocity = 9 SP

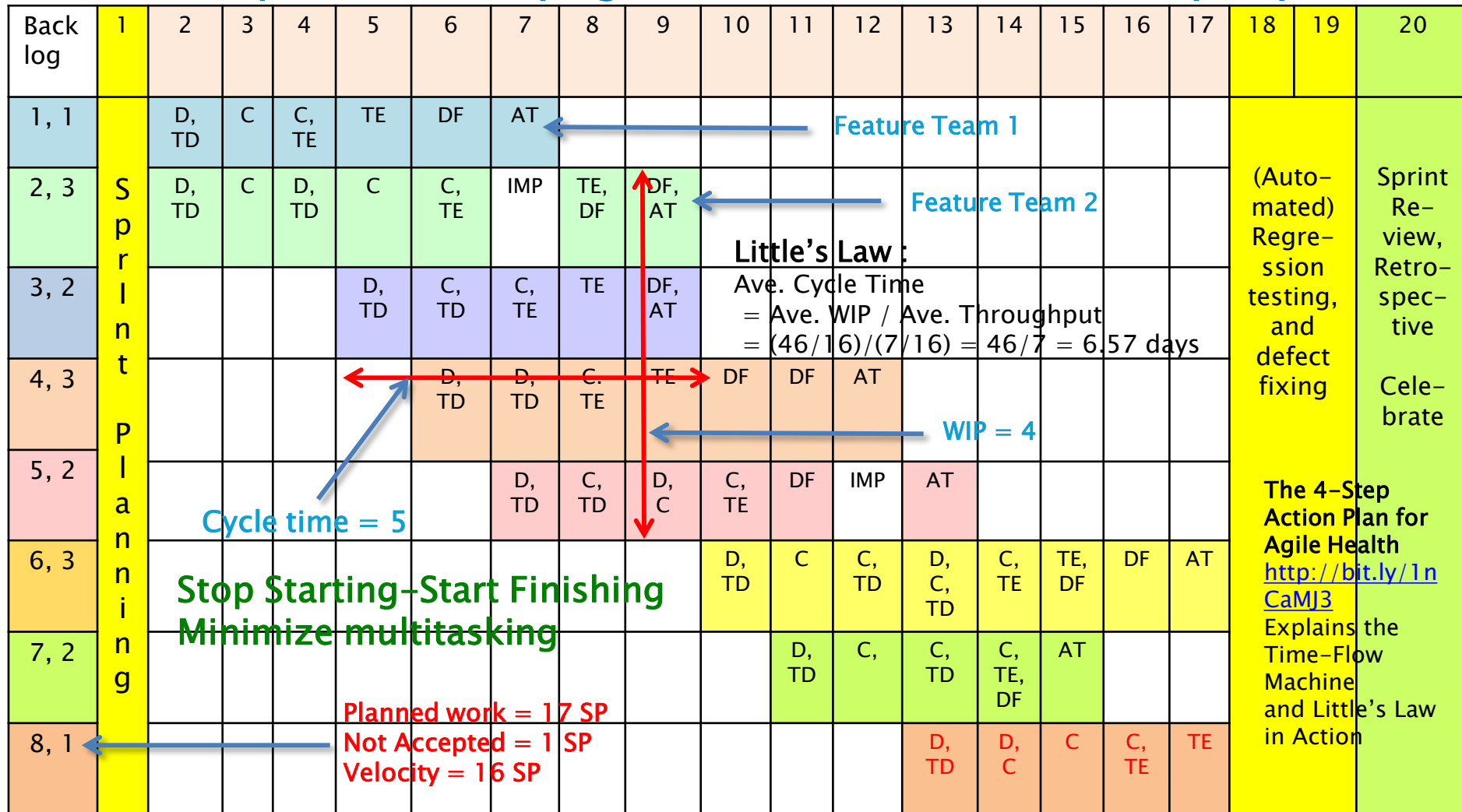
Sprint Review, Retrospective

No time for Regression testing

A: Analysis, D: Design, TD: Test Dev, C: Coding, TE: Test Exe, DF: Defect Fix, AT: Accept Test, IMP: Impediment, NT: No Team Member
 D, C, TD, TE, DF form Micro Waterfalls. Rampant Multitasking, High WIP, Low Throughput → High Cycle Time

Time-Flow Machine: Strong Full-Spectrum Story-Level Feedback

Example of a Healthy Agile Team, 4-Week (20 Workdays) Sprint



D: Design, TD: Test Development, C: Coding, TE: Test Execution, DF: Defect Fixing, AT: Accept Test, IMP: Impediment
D, TD, C, TE, DF concurrently by swarming feature teams; Low WIP, Low Multitasking, High Throughput → Low Cycle Time

Effective and Efficient Sprint Retrospectives

Product Name:

Release ID

Sprint ID

A) What worked well, & will be continued:

15 min

- Team consensus on up to top 3 factors

C) Present and Discuss Key Statistics and Impediments: 30 min

- Sprint goals planned vs. achieved in a sprint
- Planned vs. accepted story points (i.e., velocity) for the sprint
- Estimated effort vs. Actual effort spent in the Sprint
- Scope change, and any additional key statistics.
- Top 3 impediments or impediment patterns, and their causes

Review Derivative Feedback on Daily Feedback

- Review **Full-Spectrum** Taskboard and Testboard
- Review **Time-Flow** Machine

B) What was problematic & will be changed: 15 min

- Team consensus on up to top 3 factors

D) Develop **SMART Action plan to improve agile process: 45 min**

Did we meet our Sprint Goal? Any gaps?

- **S**pecific
- **M**easurable
- **A**chievable
- **R**ealistic
- **T**ime-bound

E) Capture the results of Sprint Retrospective in the agile tool, including SMART stories in the next sprint backlog: 15 min










Derivative Feedback on Daily Feedback: Review Full-Spectrum Taskboard during Sprint Retrospective

Taskboard





« 1-3 of 3 »

Highlight Owner:

Filter 

Backlog	(None)	Committing to Complete Task in Next Two Business Days	Committing to Complete Task by Next Business Day	Task Completed as Committed	Task did not Complete as Committed (work remains)	Task Completed, but Late
<p>B-02109 </p> <p>Search by Keyword</p> <p>2.00</p>				<p>UI Design  0.00 Billy Admin</p> <p>Software Design  0.00 Billy Team Member 1</p> <p>Develop Usability Test Cases  0.00 Billy Admin</p> <p>On-Line Help Development  0.00 Billy Team Member 2</p> <p>Acceptance Test Case 1 Development  0.00 Billy Team Member 2</p> <p>Acceptance Test Case 3 Development  0.00 Billy Admin</p>		<p>Code Development and Unit Testing  0.00 Billy Team Member 1</p> <p>Acceptance Test Case 2 Development  0.00 Billy Team Member 2</p>

Task cards colored by: Days in Work

-  None
-  1 Day
-  2 Days
-  3 or More Days

Goal: Maximize the number of SMART tasks completed as committed

Derivative Feedback on Daily Feedback: Review Full-Spectrum Testboard during Sprint Retrospective

Testboard ◀ 1-3 of 3 ▶ Highlight Owner: Legend

Filter

Backlog	(None)	Committing to Run Test in Next Two Business Days	Committing to Run Test by the Next Business Day	Test Passed without issues and as committed	Test didn't Complete as committed (testing remains)	Test Passed without issues, but Late	Test Failed	Test Passed after Fixing Issue(s)
<p>B-02109 </p> <p>Search by Keyword</p> <p>2.00</p>				<p>Run Acceptance Test Case 3 </p> <p>Billy Admin 0.00</p>		<p>Run Acceptance Test Case 2 </p> <p>Billy Team Member 2 0.00</p> <p>Verify that On-Line Help Matches with Software Functionality </p> <p>Billy Team Member 2 0.00</p>		<p>Run Acceptance Test Case 1 </p> <p>Billy Team Member 2 0.00</p> <p>Run Usability Tests </p> <p>Billy Admin 0.00</p>

Test cards colored by: Days in Work

- None
- 1 Day
- 2 Days
- 3 or More Days

Goal: Maximize the number of SMART tests completed as committed

Sprint SMART Actions:




Make SMART Stories for the Next Sprint Backlog

SMART Action Plan mapped on SMART Epics and SMART Stories	
<p>We had too many stories accepted in the last week of the sprint</p>	<ul style="list-style-type: none"> • SMART Story: Apply rigorous WIP controls to Kanban storyboard to reduce the cycle time. • SMART Story: Implement the Time-Flow Machine to observe flow daily, watch flow stoppage to take action, and use Time-Flow machine for derivative feedback during Sprint Retrospectives
<p>Build process was slow, which often caused delays waiting for build to complete</p>	<p>SMART Epic: Invest in Continuous Integration infrastructure for future sprints.</p> <ul style="list-style-type: none"> • SMART Story: Evaluate open-source options (such as Hudson) • SMART Story: Trial and experiment with Continuous Integration platform • SMART Story: Complete training of the team members for the selected Continuous Integration platform • SMART Story: Implement and deploy the Continuous Integration server

Release Feedback System: Experienced through Release Reviews and Release Retrospectives

- **Effective Release Reviews and Retrospectives** are essential for strategic improvements at the program or organizational level:
 - Time-to-market reduction, productivity improvements, quality improvements, release cost reductions, etc.
- **Review Primary feedback:** Primary feedback loop must indicate whether and how well the release-level goals are achieved
- **Review Derivative feedback** on a series of Sprint Retrospective feedbacks: Derivative feedback loop must identify any systemic issues coming from a sequence of sprint retrospectives to enable learning and process improvements by a program or the organization

Derivative Feedback on Sprint Feedback: Discuss during Release Retrospective

<input type="checkbox"/> Title	Sprint	Summary
<input type="checkbox"/>  Sprint 1 Retrospective	Sprint 1	<p>Worked well: Use of VersionOne as the agile tool provided us an integrated project database with good transparency</p> <p>Top Problem: Too many stories accepted in the last few days of the sprint</p> <p>SMART 1: Learn how to Apply rigorous WIP controls to Kanban storyboard.</p>
<input type="checkbox"/>  Sprint 2 Retrospective	Sprint 2	<p>Worked Well: Vertically split small features improved estimation and planning</p> <p>Top Problem: Daily Scrum meetings were not very effective as people usually weren't prepared for the meeting</p> <p>SMART 1: Implement the Kanban board with WIP control in VersionOne platform production environment</p> <p>SMART 2: Use the Recommended Taskboard for planning, tracking and reporting in Daily Scrums</p>
<input type="checkbox"/>  Sprint 3 Retrospective	Sprint 3	<p>Worked Well: Coordination among team members facilitated by Daily Scrum meetings</p> <p>Top Problem: Some stories were too big; we should have split them into smaller stories</p> <p>SMART 1: Evaluate open-source continuous Integration platform, such as Hudson</p> <p>SMART 2: Engage Satish Thatte (our Agile Coach) to coach us on writing well INVESTED stories</p>

Analyze Sprint Retrospective Summary over a sequence of many sprints:

- Are we actually improving as planned? Do we see expected benefits?
- Are we taking two steps forward and one step backward, or sideways
- Are there recurring issues? If so, what are their root causes?
- What actions are needed to fix the root causes?

Strategic Action Plan following Release Retrospectives

Strategic Metric	Measurement Approaches	Action Plan to Address Issues
A. Field data on product feature usage	Customer surveys for feature usage <ul style="list-style-type: none"> • Feature usage distribution and frequency • Features that are missing • Features that need to be improved or simplified 	<ul style="list-style-type: none"> • Discontinue rarely used features • Add important missing features or epics • Improve, simplify or streamline features with poor user experience
B. Concept to Customer value cycle time	<ul style="list-style-type: none"> • End-to-end cycle time • End-to-end process bottlenecks • (Value producing time / Total cycle time) 	Reduce end-to-end cycle time by simplifying and streamlining the process, reduce delays, and remove bottlenecks
C. Release cost	Fully loaded development and delivery costs: people, material, equipment, licenses, shared IT service charges, etc.	<ul style="list-style-type: none"> • Reduce development and delivery costs <ul style="list-style-type: none"> ○ Eliminate rarely used features (see Measure A) ○ Improve teamwork ○ Increase automation
D. Release productivity = (Release velocity / Release cost)	Normalize the velocity numbers across sprints and teams to account because story points across sprints and teams may represent different amounts of work.	<ul style="list-style-type: none"> • Improve team work, cross-functional team training • Reduce release cost (see Measure C)
E. Quality	Number of customer-reported new issues: Typically reported on a quarterly basis or for the entire release cycle.	<ul style="list-style-type: none"> • Improve quality with reviews for feature specification, design, code • Offer training and resources for test-driven development, refactoring, test automation, and technical debt reduction

Summary

	Daily Feedback System	Sprint Feedback System	Release Feedback System
Experienced with	Daily Scrums, Story Board, Burn-Down, Burn-up	Sprint Reviews and Retrospectives	Release Reviews and Retrospectives
Focused on	Individual team members & team	Agile teams and projects	Projects, Programs, Portfolios, and Organization
Primary Feedback	Feedback loop between daily commitment and its fulfillment: Enabled by Full-Spectrum Taskboard, Testboard, Time-Flow Machine	Feedback loop between Sprint goals and accomplishments; SMART action plan drives SMART stories in the next sprint, and SMART epics in the next release	Feedback loop between Release goals and accomplishments, captured with Strategic Metrics: Time-to-market reduction, productivity improvements, quality improvements, release cost reductions, etc.
Derivative Feedback	Feedback necessary to allow continuous improvements to deliver on daily commitments	Are SMART actions giving the expected results? Is the team regressing from or building upon prior improvements?	Is the program or organization regressing from or building upon prior improvements?
Tools	Daily Scrum with Revised Taskboard, Testboard, Sprint Time-Flow Machine	SMART Stories SMART Epics Sprint retro reports	Strategic metrics system, and implement action plan to improve strategically

Nurtures a learning organization with continuous improvements