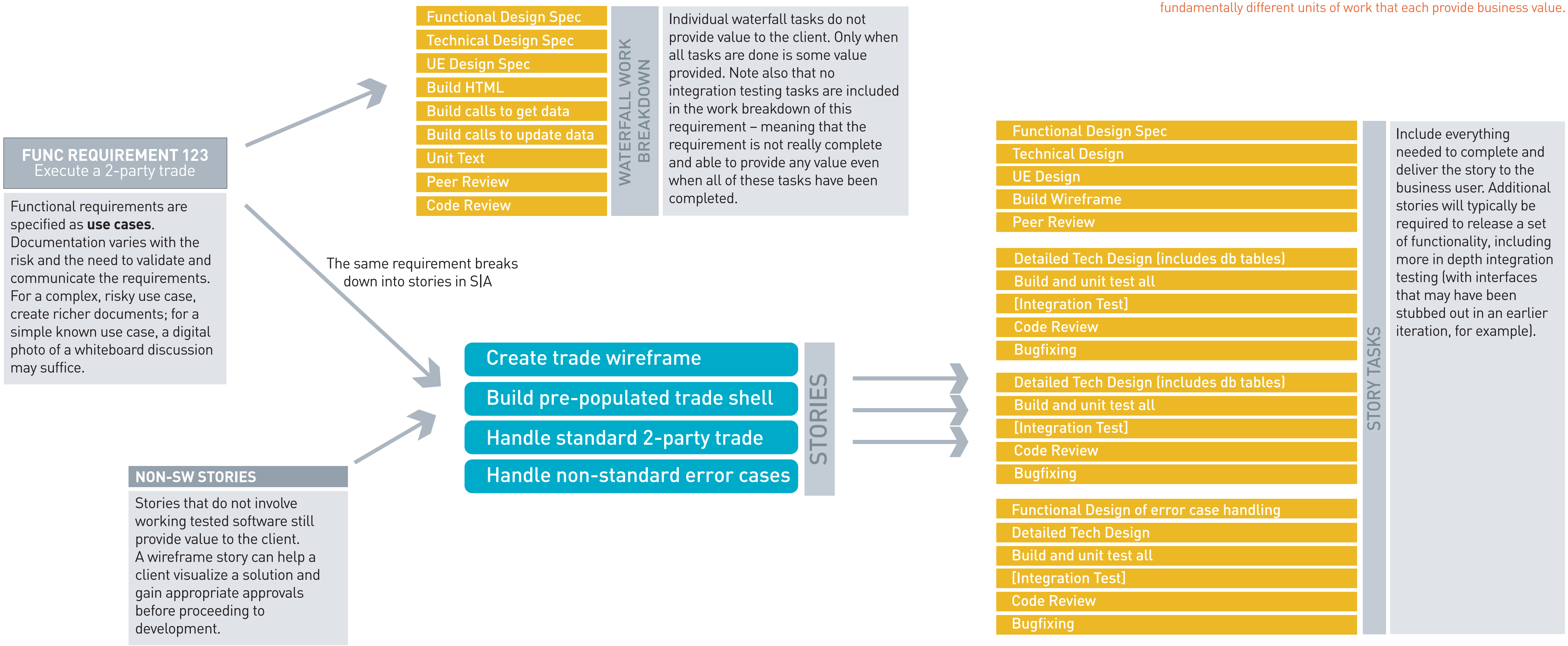
Sapient

S|A 3 **Requirements to Stories**

To build a system, you will translate requirements into units of work. Waterfall models decompose requirements into tasks, but S|A decomposes requirements into **stories**, fundamentally different units of work that each provide business value.



FUNC REQUIREMENT 123
Execute a 2-party trade

Functional requirements are specified as **use cases**. Documentation varies with the risk and the need to validate and communicate the requirements. For a complex, risky use case, create richer documents; for a simple known use case, a digital photo of a whiteboard discussion may suffice.

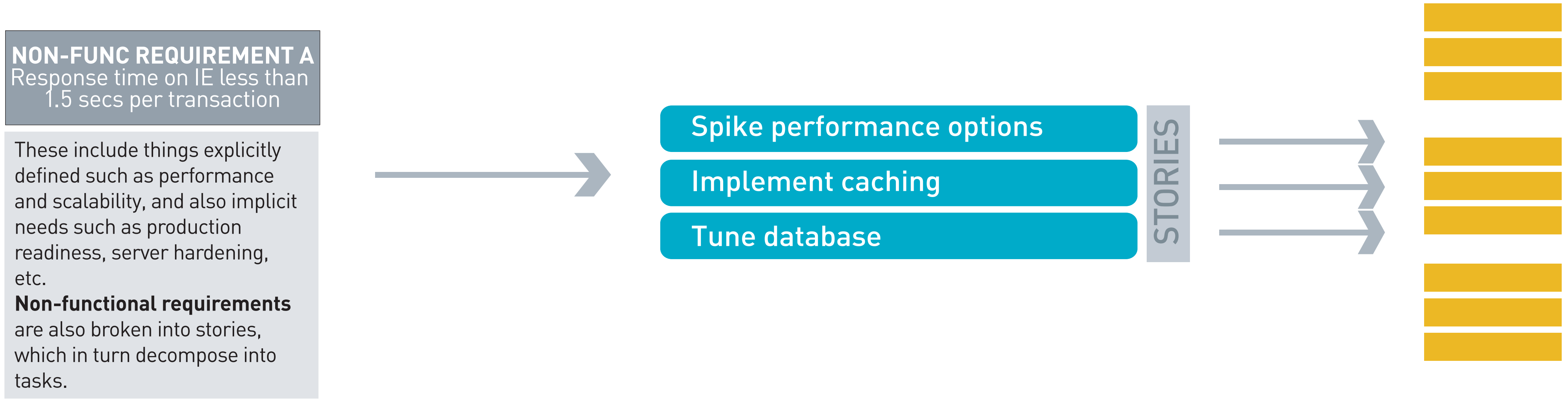
Individual waterfall tasks do not provide value to the client. Only when all tasks are done is some value provided. Note also that no integration testing tasks are included in the work breakdown of this requirement – meaning that the requirement is not really complete and able to provide any value even when all of these tasks have been completed.

The same requirement breaks down into stories in S|A

Include everything needed to complete and deliver the story to the business user. Additional stories will typically be required to release a set of functionality, including more in depth integration testing (with interfaces that may have been stubbed out in an earlier iteration, for example).

NON-SW STORIES

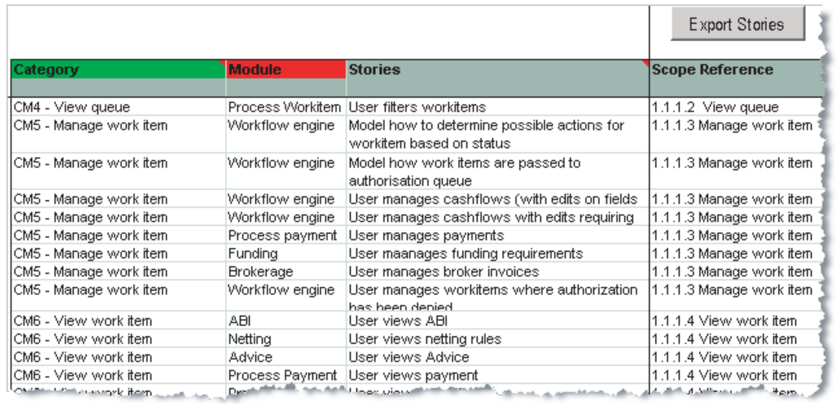
Stories that do not involve working tested software still provide value to the client. A wireframe story can help a client visualize a solution and gain appropriate approvals before proceeding to development.



NON-FUNC REQUIREMENT A
Response time on IE less than 1.5 secs per transaction

These include things explicitly defined such as performance and scalability, and also implicit needs such as production readiness, server hardening, etc.

Non-functional requirements are also broken into stories, which in turn decompose into tasks.



Category	Module	Stories	Scope Reference
CM4 - View queue	Process Workitem	User filters workitems	1.1.1.2 View queue
CM5 - Manage work item	Workflow engine	Model how to determine possible actions for workitem based on status	1.1.1.3 Manage work item
CM5 - Manage work item	Workflow engine	Model how work items are passed to authorisation queue	1.1.1.3 Manage work item
CM5 - Manage work item	Workflow engine	User manages cashflows (with edits on fields	1.1.1.3 Manage work item
CM5 - Manage work item	Workflow engine	User manages cashflows with edits requiring	1.1.1.3 Manage work item
CM5 - Manage work item	Process payment	User manages payments	1.1.1.3 Manage work item
CM5 - Manage work item	Funding	User maanages funding requirements	1.1.1.3 Manage work item
CM5 - Manage work item	Brokerage	User manages broker invoices	1.1.1.3 Manage work item
CM5 - Manage work item	Workflow engine	User manages workitems where authorization has been denied	1.1.1.3 Manage work item
CM6 - View work item	ABI	User views ABI	1.1.1.4 View work item
CM6 - View work item	Netting	User views netting rules	1.1.1.4 View work item
CM6 - View work item	Advice	User views Advice	1.1.1.4 View work item
CM6 - View work item	Process Payment	User views payment	1.1.1.4 View work item

Stories are captured in Estimation Tool, with linkage to relevant requirements in the Scope Matrix.

Sapient

S|A 3 **Story Definition**

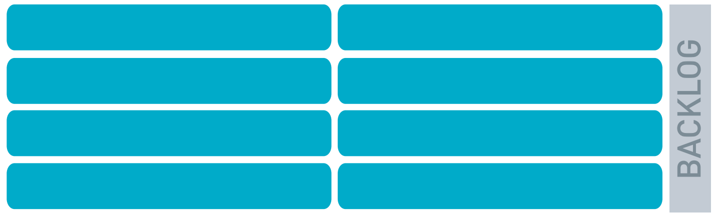
The story is the fundamental unit of work in an agile project. To decompose a requirement into stories, use an iterative process, breaking requirements into work units, splitting or merging stories until each meets four characteristics: valuable, understandable, testable, and completable.

STORY ATTRIBUTES

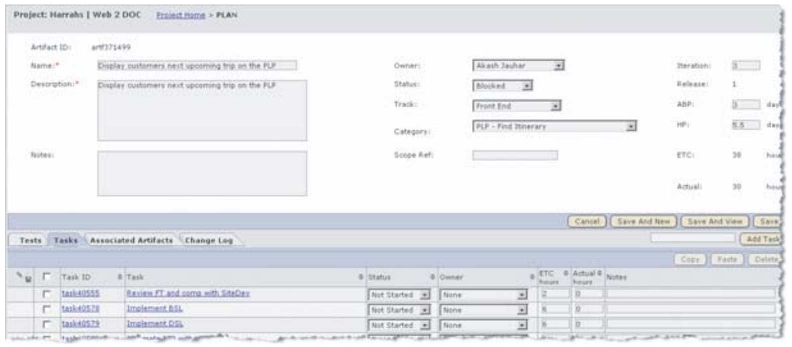
Name: 1 line	*Create simple trade wireframe*
Description: 2-4 lines	*Create simple trade wireframe using the established style guide, reviewed by key marketing contacts*
Tests: 1+ tests to validate completion of the story	***Test 1:** Wireframe conforms to style guide* ***Test 2:** Wireframe has been signed off by marketing team*
Tasks: 0+ granular activities to be executed to accomplish to meet the story tests. Task size should be small – able to be completed in a day.	***Task 1:** Conduct brainstorm session with marketing* ***Task 2:** Mock up 2 options* ***Task 3:** Facilitate marketing decision* ***Task 4:** Finalize chosen wireframe* ***Task 5:** Internal peer review* ***Task 6:** Final marketing review* ***Task 7:** Final mock up updates*

Just in time detail: both **Tests** and **Tasks** are defined only at the start of the iteration where that story is slated for delivery. This minimizes planning and estimation waste in the event the story is mapped to an iteration, and also allows the team to benefit from knowledge gained in previous iterations.

BACKLOG



The collection of all stories associated with the project is known as the **backlog**.



Project: Harrahs | Web 2 DOC   Project Home > PLAN

Artifact ID: artf371499

Name:* Display customers next upcoming trip on the PLF

Description:* Display customers next upcoming trip on the PLF

Notes:

Owner: Akash Jauhar

Status: Blocked

Track: Front End

Category: PLP - Find Itinerary

Scope Ref:

Iteration: 3

Release: 1

ABP: 3 days

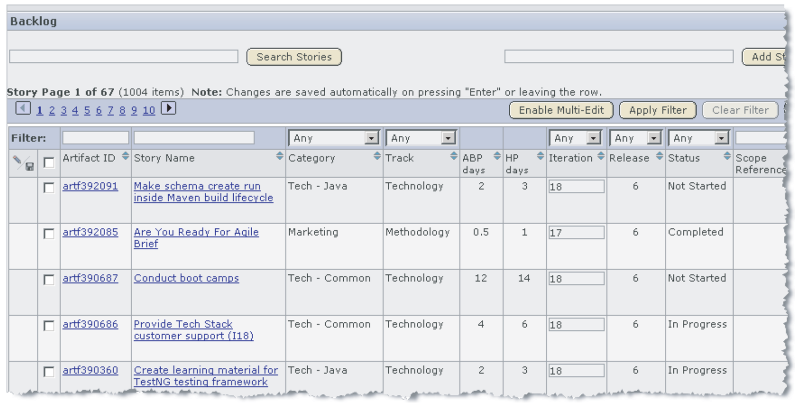
HP: 5.5 days

ETC: 38 hours

Actual: 30 hours

	Task ID	Task	Status	Owner	ETC hours	Actual hours	Notes
□	task40555	Review FT and comp with SiteDev	Not Started	None	2	0	
□	task40578	Implement BSL	Not Started	None	8	0	
□	task40579	Implement DSL	Not Started	None	8	0	

Story Detail page in ResultSpace^{RM}



Backlog

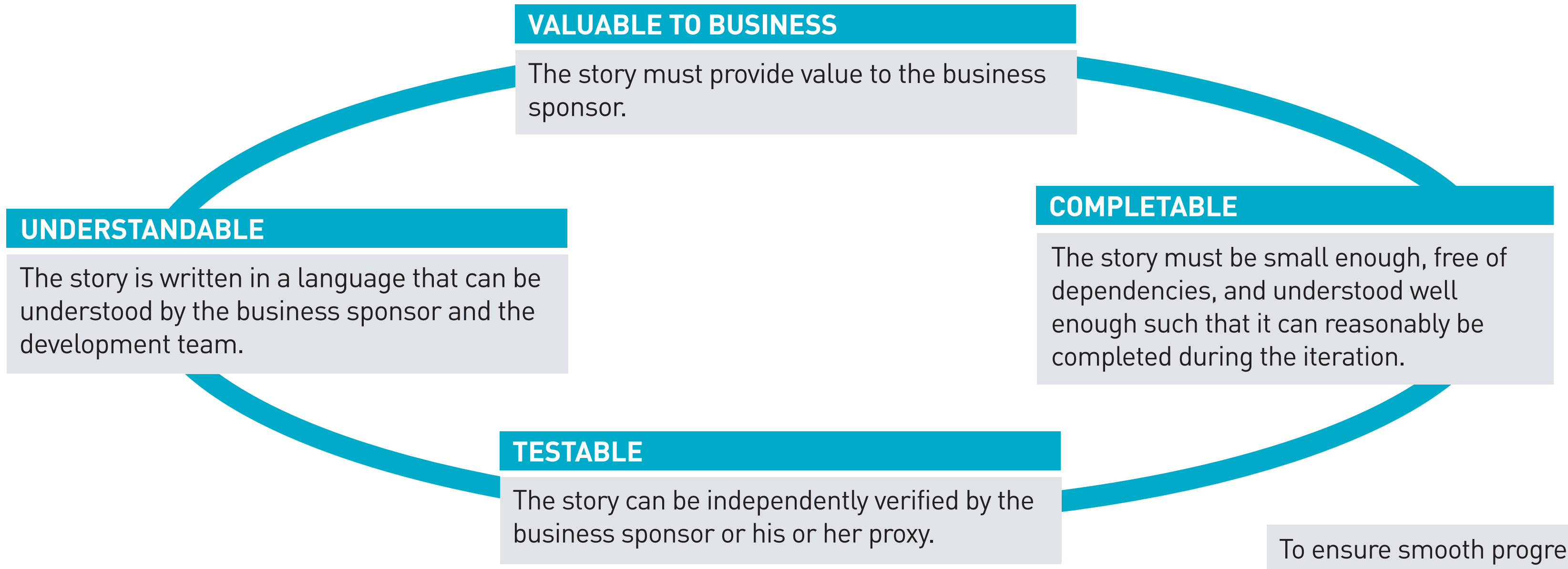
Story Page 1 of 67 (1004 items) **Note:** Changes are saved automatically on pressing "Enter" or leaving the row.

1 2 3 4 5 6 7 8 9 10

	Artifact ID	Story Name	Category	Track	ABP days	HP days	Iteration	Release	Status	Scope Reference
Filter:			Any	Any			Any	Any	Any	
□	artf392091	Make schema create run inside Maven build lifecycle	Tech - Java	Technology	2	3	18	6	Not Started	
□	artf392085	Are You Ready For Agile Brief	Marketing	Methodology	0.5	1	17	6	Completed	
□	artf390687	Conduct boot camps	Tech - Common	Technology	12	14	18	6	Not Started	
□	artf390686	Provide Tech Stack customer support (I18)	Tech - Common	Technology	4	6	18	6	In Progress	
□	artf390360	Create learning material for TestNG testing framework	Tech - Java	Technology	2	3	18	6	In Progress	

Backlog page in ResultSpace^{RM}

STORY CHARACTERISTICS



VALUABLE TO BUSINESS

The story must provide value to the business sponsor.

UNDERSTANDABLE

The story is written in a language that can be understood by the business sponsor and the development team.

COMPLETABLE

The story must be small enough, free of dependencies, and understood well enough such that it can reasonably be completed during the iteration.

TESTABLE

The story can be independently verified by the business sponsor or his or her proxy.

To ensure smooth progress, each story should ideally be able to be completed in less than half the duration of the iteration. To focus and engage the team and the business sponsor, and to allow productive prioritization discussions, keep the total number of stories per track per iteration to a reasonable number (less than 20).

Sapient

S|A 3 **Estimation**

After decomposing the scope of the project into stories, a subset of the development team estimates the work at a high level in order to do release and iteration planning.

WIDE-BAND MODIFIED DELPHI (WMD):

To efficiently estimate stories, a small group of team members with varying backgrounds assembles to use the WMD estimation technique. Each person independently assigns two values to each story: an Aggressive But Possible ("ABP") estimate (in person days), assumed to be a target that could be met under ideal conditions, perhaps 50% of the time; and a Highly Probable ("HP") value which would be the number giving 90-95% certainty. They then meet to discuss their estimates, and create a total ABP and HP for the backlog.

"EPIC"

Some stories break the rule of fitting within an iteration – at estimation time, so little is known about the story that it is estimated at a large range. Later in the project once more is known, split this story into smaller stories that can fit in an iteration.

ABIHLASH		BOBBY		MARY		FINAL	
ABP	HP	ABP	HP	ABP	HP	ABP	HP
2	3	2	3	2	5	2	4
1	2	10	20	5	7	5	20
1	2	1	2	1	3	1	2
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
20	20	25	35	20	20	25	35
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
						250	350

MAJOR DIFFERENCES

Discuss these and reach a consensus value based on clarified assumptions, etc.

MINOR DIFFERENCES

In the estimation team, you do not discuss these – simply take a numerical average.
In this manner, the entire set of stories can be efficiently estimated.

STORY POINTS

To facilitate discussions with the business sponsor regarding relative story size, each story is assigned a relative weighting by taking the initial ABP value of that story. Based on the definition of ABP, it should be remembered that the story point size is not equivalent to the effort it will take to complete the story.

$$\frac{250 + 350}{2} = 300 \text{ DAYS}$$

(total estimated effort)

EXPECTED EFFORT

By using a statistical average, the ABP and HP totals are used to calculate the expected effort for the project. This effort is used (together with leadership staff, risk buffers, and other add-ins) to define the staffing for the project.

TOTAL PERSON DAYS	
ABP + HP/2	300
Risk buffer	30
Add-ins	70
TOTAL	400

STAFFING

Total Person Days drives a staffing level. By combining this with time drivers, project dependencies, and other client milestones, the team leadership derives the total duration of the project in terms of Releases and Iterations.

The total story points for the project are distributed across the iterations. For an ongoing, steady-sized team, the distribution would be flat. Any other configuration would lead to a different distribution of points per iteration:

25	25	25	25	25	25	25	25	25	25

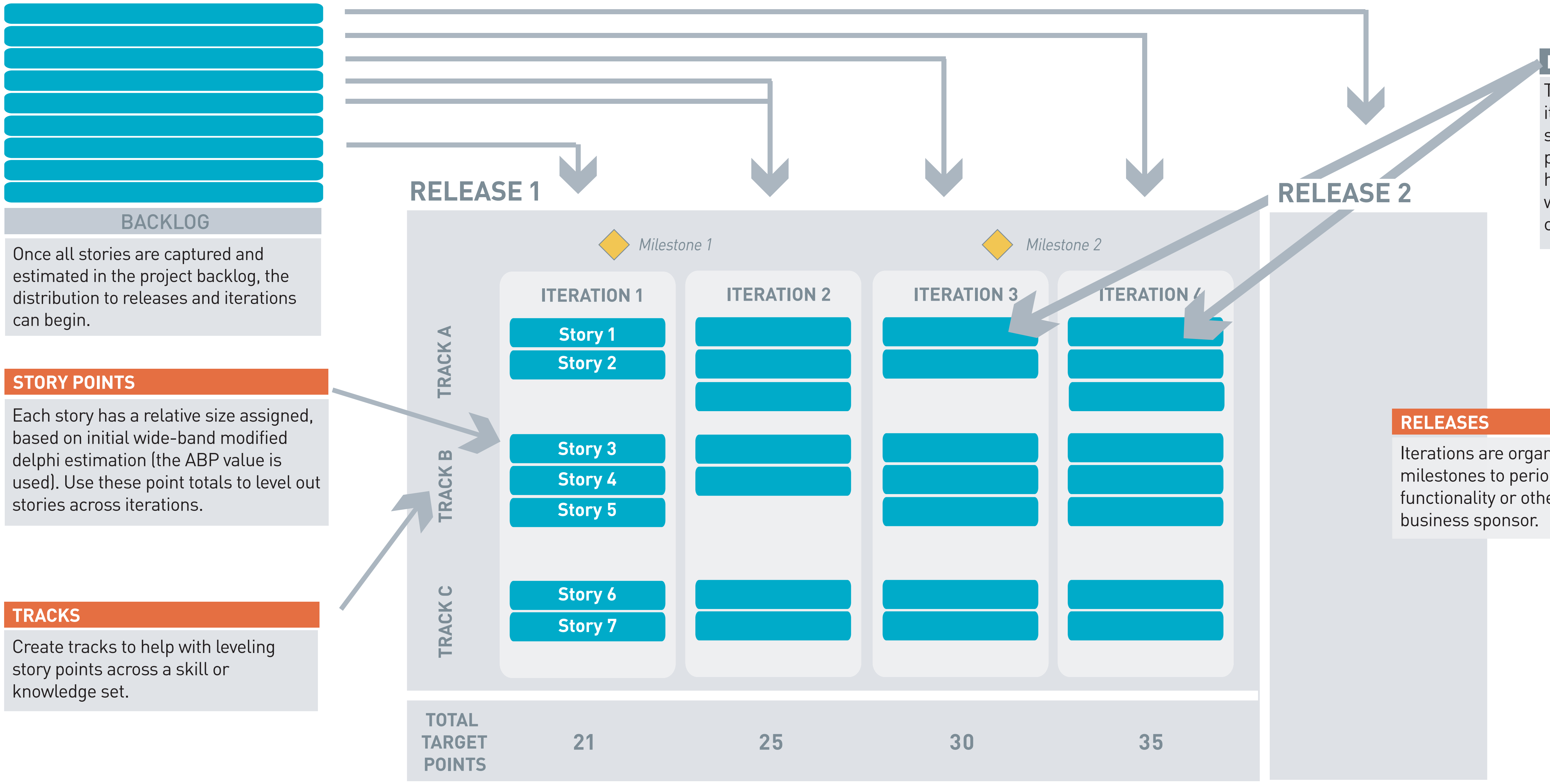
20 WEEKS TOTAL (10 TWO-WEEK ITERATIONS)
4 PERSON TEAM

Note that there is not a 1-to-1 mapping of person days available per iteration and the story points per iteration (because of risk and other add-ins).

Sapient

S|A 3 **Mid-Level Planning**

For a project with a fixed scope of work, create an initial plan by assigning stories to iterations within the first release, and to releases beyond that. Use dependencies, track leveling, and iteration leveling as techniques to distribute stories. For a co-located planning team, use stickies (one per story) and a whiteboard to facilitate the process.



BACKLOG

Once all stories are captured and estimated in the project backlog, the distribution to releases and iterations can begin.

STORY POINTS

Each story has a relative size assigned, based on initial wide-band modified delphi estimation (the ABP value is used). Use these point totals to level out stories across iterations.

TRACKS

Create tracks to help with leveling story points across a skill or knowledge set.

DEPENDENCIES

Try to place dependent stories in separate iterations to simplify planning. Strive to simplify and avoid dependencies where possible and track those that do exist in your head. If you must include dependent stories within an iteration, work to eliminate any cross-track dependencies.

RELEASES

Iterations are organized around meaningful milestones to periodically release functionality or other deliverables to the business sponsor.

ITERATIONS

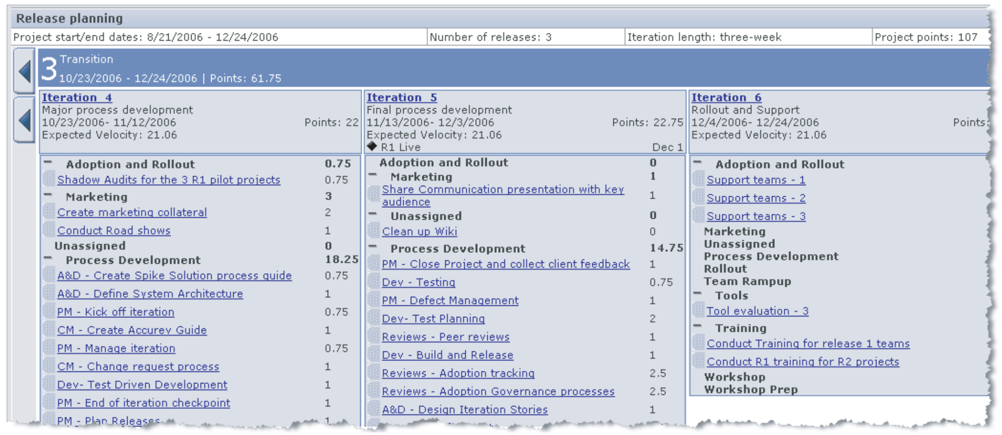
Work is executed in a series of consecutive 1-4 week time-boxes. Stories must be completed within this time-box. Incomplete stories may be moved to a subsequent iteration for completion.

TEAM VELOCITY RAMP UP

Before the team learns how to work together its productivity will be lower than average. To account for this, plan to complete fewer story points in earlier iterations (and to make up for this with more productive iterations towards the end of the project or release).

SPLIT STORIES

If a story includes a large amount of design that needs to be validated by someone outside the team, split that story in two and spread it across iterations. For example, don't define a complex use case and then implement it as part of the same story. Too many dependencies would exist on signoff with people during the iteration, leading to risk on not completing the stories in the iteration.



Release planning

Project start/end dates: 8/21/2006 - 12/24/2006 | Number of releases: 3 | Iteration length: three-week | Project points: 107

3 Transition
10/23/2006 - 12/24/2006 | Points: 61.75

Iteration 4
Major process development
10/23/2006- 11/12/2006 Points: 22
Expected Velocity: 21.06

Adoption and Rollout	**0.75**
Shadow Audits for the 3 R1 pilot projects	0.75
Marketing	**3**
Create marketing collateral	2
Conduct Road shows	1
Unassigned	**0**
Process Development	**18.25**
A&D - Create Spike Solution process guide	0.75
A&D - Define System Architecture	1
PM - Kick off iteration	0.75
CM - Create Accurev Guide	1
PM - Manage iteration	0.75
CM - Change request process	1
Dev- Test Driven Development	1
PM - End of iteration checkpoint	1
PM - Plan Releases	1

Iteration 5
Final process development
11/13/2006- 12/3/2006 Points: 22.75
Expected Velocity: 21.06
◆ R1 Live Dec 1

Adoption and Rollout	**0**
Marketing	**1**
Share Communication presentation with key audience	1
Unassigned	**0**
Clean up Wiki	0
Process Development	**14.75**
PM - Close Project and collect client feedback	1
Dev - Testing	0.75
PM - Defect Management	1
Dev- Test Planning	2
Reviews - Peer reviews	1
Dev - Build and Release	1
Reviews - Adoption tracking	2.5
Reviews - Adoption Governance processes	2.5
A&D - Design Iteration Stories	1

Iteration 6
Rollout and Support
12/4/2006- 12/24/2006 Points:
Expected Velocity: 21.06

Adoption and Rollout
Support teams - 1
Support teams - 2
Support teams - 3
Marketing
Unassigned
Process Development
Rollout
Team Rampup
Tools
Tool evaluation - 3
Training
Conduct Training for release 1 teams
Conduct R1 training for R2 projects
Workshop
Workshop Prep

Release planning feature in ResultSpace^{RM} used to assign stories to iterations.

Sapient

S|A 3 Kickoff and Manage an Iteration

Perform just in time detailed definition of the story, and then move through a state of Not Started, through to Completed. Minimize work in progress stories so that productivity will be as constant as possible. Only when an unforeseen issue exists that blocks a story within an iteration does the status deviate from this normal progression.

CREATE SIMPLE TRADE WIREFRAME

TESTS

- Wireframe conforms to styleguide
- Wireframe has been signed off by marketing

TASKS

	ETC
Lorem ipsum dolor sit amet	2
Adipiscing elit, sed diam nonummy	2
Mod tincidunt	1
Erat volutpatt wisi enim ad minim	1
Veniam, quis nostrud	3
Exerci tation ullamcorper suscipit lobortis	6
Nisl ut aliquip ex ea commodo consequat	8
Duis autem vel eum iriure dolor in	4
Hendrerit in vulputate	2

MINIMIZE WORK IN PROGRESS

Teams should work to complete in progress stories before starting new stories whenever possible. This reduces the risk of in progress stories getting delayed and the project team's velocity suffering as a result.

TASKING

At the start of the iteration the sub team responsible for a story works together to list the tests that will define completeness for the story. They will checkpoint these with the business user.

At the same meeting, create a set of tasks for the story (including tasks required to test the story), assigning an estimate in hours to each task. If the total estimated effort for the tasks is significantly different from the earlier estimate (the Aggressive But Possible and the Highly Probable), then the story owner escalates to the Project Manager to take appropriate action. For story estimates that exceed the rough estimate, it may be necessary to split the story, or to remove another story from the iteration's plan. Any case where the story list changes from that agreed upon with the business sponsor, the latter should be informed immediately.

TOTAL ETC

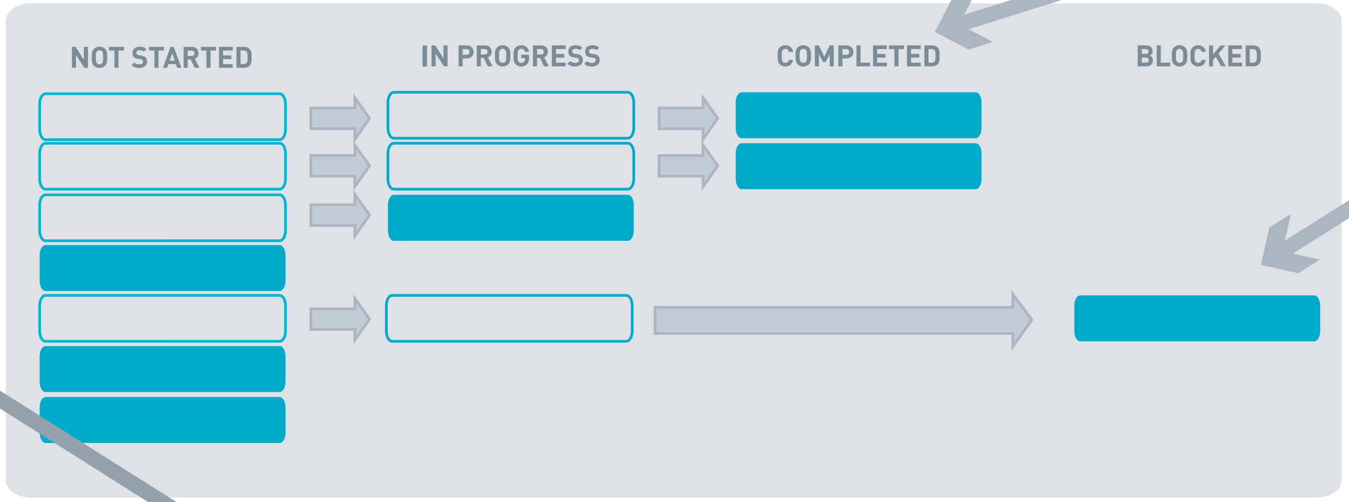
Is critical so the team knows if they are on track to complete the story during the iteration.

TESTS

Are critical so the team knows when they are done.

HALF-WAY THROUGH

At the mid-point in the iteration, this visual representation shows a reasonable distribution of the stories. A minimal number are In Progress, some have already been Completed.

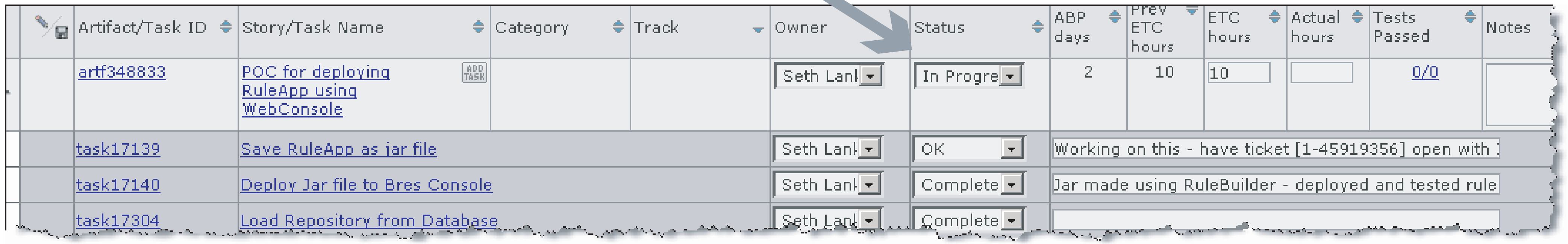


VELOCITY

The team should strive foremost to deliver value to the client, which is represented only by Completed stories (i.e., those that for which all tests pass). Velocity is the sum of all story points for each Completed story each iteration. Stories that end the iteration incomplete contribute zero to the team's velocity and are instead rolled over the the next iteration.

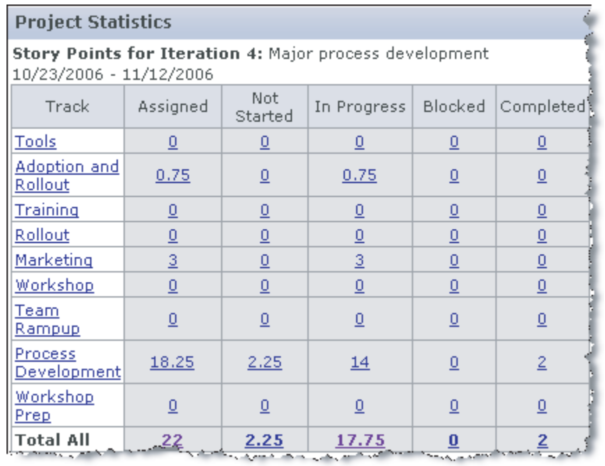
BLOCKED STORIES

An issue should be actively worked for any story that is marked as Blocked. A story may be blocked because of an unforeseen dependency, a reviewer out sick, etc. If a story is unlikely to become unblocked, the team should consult the business sponsor and consider pulling in another story from the next iteration.



	Artifact/Task ID	Story/Task Name	Category	Track	Owner	Status	ABP days	Prev ETC hours	ETC hours	Actual hours	Tests Passed	Notes
	artf348833	POC for deploying RuleApp using WebConsole			Seth Lanl	In Progre	2	10	10		0/0	
	task17139	Save RuleApp as jar file			Seth Lanl	OK	Working on this - have ticket [1-45919356] open with					
	task17140	Deploy Jar file to Bres Console			Seth Lanl	Complete	Jar made using RuleBuilder - deployed and tested rule					
	task17304	Load Repository from Database			Seth Lanl	Complete						

The Iteration Plan view in ResultSpace^RM



Project Statistics

Story Points for Iteration 4: Major process development
10/23/2006 - 11/12/2006

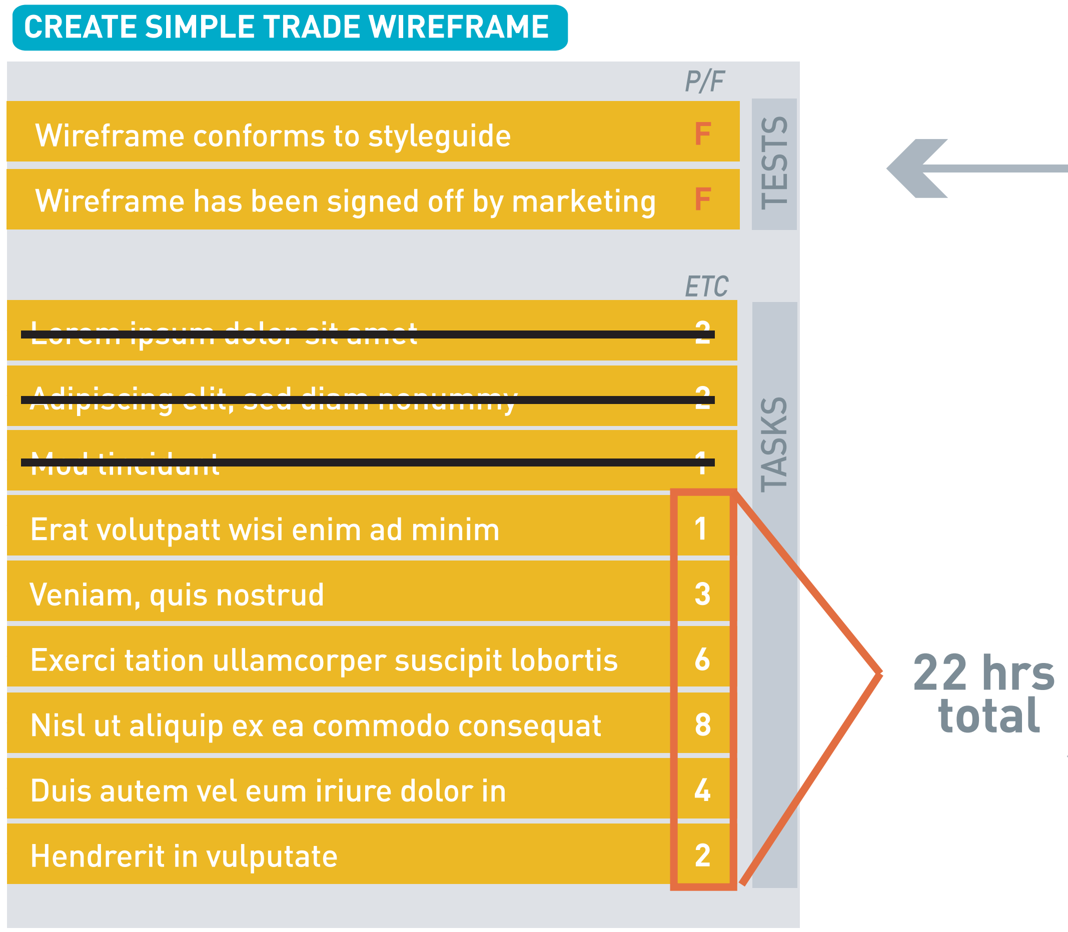
Track	Assigned	Not Started	In Progress	Blocked	Completed
Tools	0	0	0	0	0
Adoption and Rollout	0.75	0	0.75	0	0
Training	0	0	0	0	0
Rollout	0	0	0	0	0
Marketing	3	0	3	0	0
Workshop	0	0	0	0	0
Team Rampup	0	0	0	0	0
Process Development	18.25	2.25	14	0	2
Workshop Prep	0	0	0	0	0
Total All	22	2.25	17.75	0	2

Status of all stories per iteration in ResultSpace^RM

Sapient

S|A 3 **Tracking Progress – Story Level**

Once a story is started, work the story aggressively until it is complete. Each day, the subset of the team working the story updates the Estimate to Complete (ETC) for the story, giving everyone visibility into the likelihood of completing the iteration's stories.



CREATE SIMPLE TRADE WIREFRAME

	P/F
Wireframe conforms to styleguide	F
Wireframe has been signed off by marketing	F

TESTS

	ETC
~~Lorem ipsum dolor sit amet~~	~~2~~
~~Adipiscing elit, sed diam nonummy~~	~~2~~
~~Mod tincidunt~~	~~1~~
Erat volutpatt wisi enim ad minim	1
Veniam, quis nostrud	3
Exerci tation ullamcorper suscipit lobortis	6
Nisl ut aliquip ex ea commodo consequat	8
Duis autem vel eum iriure dolor in	4
Hendrerit in vulputate	2

TASKS

22 hrs total

CONCURRENT TESTING

During the iteration, testers are working side by side with the developers on a story. Initially they define manual tests, then begin executing them as soon as the developer gives the signal. Only the tester can set the test to Pass.

In the case of a non-development story, someone not working on the story should be assigned to peer-test the story. In the example shown, the tester would look for evidence of a check against the style guide, and some form of documented signoff (email or other).

Maintaining discipline around test definition, tracking and execution is one of the most challenging aspects of Sapient|Approach.

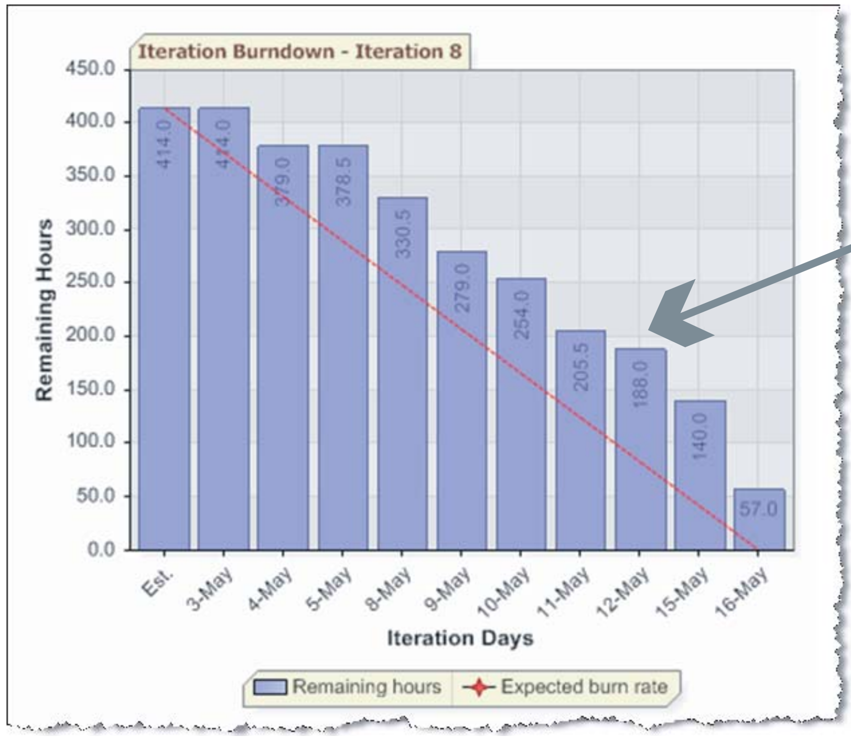
ESTIMATE TO COMPLETE (ETC)

Each day the story owner re-evaluates the story based on what's done and what remains to be completed. Make a new estimate based on the remaining tasks and what you have learned.

Story ETC = ∑ remaining task estimates (in hours)

BURNDOWN CHART

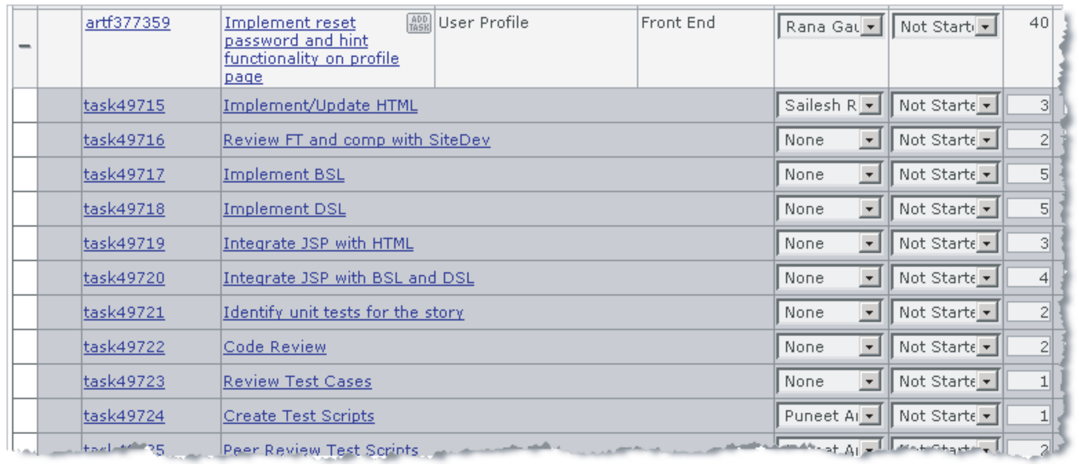
The total hours remaining to complete the story each day drive the **Burndown Chart** which compares the actual reported amount to complete for all stories, to a theoretical straight line rate of completion.



Effort "burn-down" used as primary indicator of progress within an iteration on a daily basis (from ResultSpace^{RM} Progress Charts)

SECOND WEEK COMPARISON

Looking at this mid-iteration, it would be clear that not all stories are likely to be completed. In the team retrospective, you could discuss what caused the blip and potentially take corrective action in the next iteration.



−	artf377359	Implement reset password and hint functionality on profile page	User Profile	Front End	Rana Gau	Not Start	40
	task49715	Implement/Update HTML			Sailesh R	Not Starte	3
	task49716	Review FT and comp with SiteDev			None	Not Starte	2
	task49717	Implement BSL			None	Not Starte	5
	task49718	Implement DSL			None	Not Starte	5
	task49719	Integrate JSP with HTML			None	Not Starte	3
	task49720	Integrate JSP with BSL and DSL			None	Not Starte	4
	task49721	Identify unit tests for the story			None	Not Starte	2
	task49722	Code Review			None	Not Starte	2
	task49723	Review Test Cases			None	Not Starte	1
	task49724	Create Test Scripts			Puneet A	Not Starte	1
	task49725	Peer Review Test Scripts			Puneet A	Not Starte	2

The ResultSpace^{RM} task list is a living list. Each day as progress is made, tasks are completed, added, changed, and estimates updated as needed.

Sapient

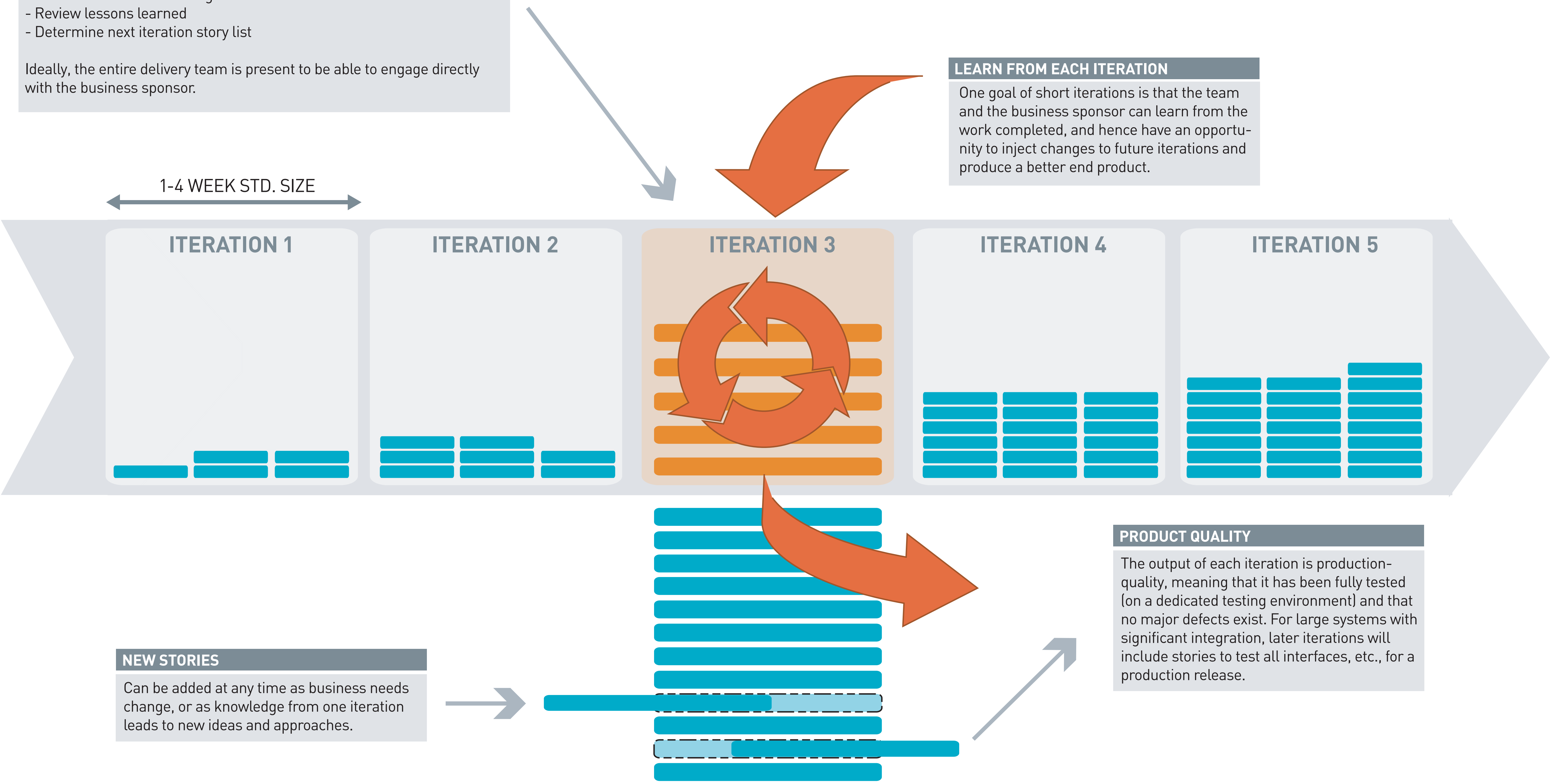
S|A 3 **Client Checkpoint & the Story Stack**

At the end of each iteration, the team checkpoints progress and gets input from the business sponsor. In this setting, the backlog of stories becomes an important concept in Sapient|Approach planning -- you and the business sponsor must treat it as a virtual stack of work, where the most valuable and riskiest items always bubble to the surface.

END ITERATION CHECKPOINT

On the last day, or very close to this date, the team holds a meeting with the business sponsor to:
- Review completed stories – done in a hands-on mode
- Resolve critical outstanding issues
- Review lessons learned
- Determine next iteration story list

Ideally, the entire delivery team is present to be able to engage directly with the business sponsor.



LEARN FROM EACH ITERATION

One goal of short iterations is that the team and the business sponsor can learn from the work completed, and hence have an opportunity to inject changes to future iterations and produce a better end product.

PRODUCT QUALITY

The output of each iteration is production-quality, meaning that it has been fully tested (on a dedicated testing environment) and that no major defects exist. For large systems with significant integration, later iterations will include stories to test all interfaces, etc., for a production release.

NEW STORIES

Can be added at any time as business needs change, or as knowledge from one iteration leads to new ideas and approaches.

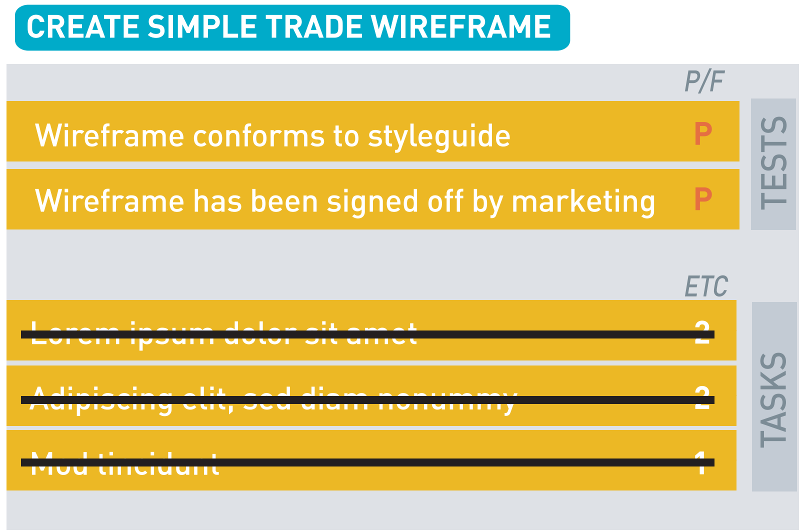
AN AGILE PLAN

Continually reshuffle and revisit the backlog of stories to push the highest value ones to the top of the stack.

Sapient

S|A 3 Incremental Delivery

In each iteration you deliver actual business value to the business sponsor. This is in contrast to a waterfall approach where the solution comes together only at the end of the project. The incremental approach gives confidence to all parties of project completion, and supports an evolutionary approach where knowledge gained during the project can be fed back into the project, resulting in a better solution.



CREATE SIMPLE TRADE WIREFRAME

	P/F	
Wireframe conforms to styleguide	P	TESTS
Wireframe has been signed off by marketing	P	
	ETC	
~~Lorem ipsum dolor sit amet~~	~~2~~	TASKS
~~Adipiscing elit, sed diam nonummy~~	~~2~~	
~~Mod tincidunt~~	~~1~~	

"DONE IS DONE"

To successfully build a foundation for incremental completion of stories, each story must be clearly "done" as follows:
- All tests must pass
- All tasks must be complete
- No outstanding P1 or P2 defects must exist

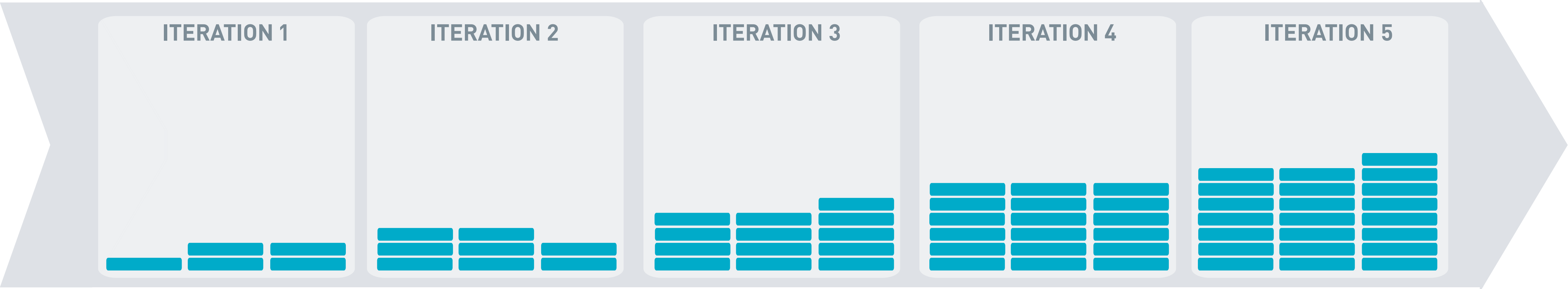
Once a story has been completed, It is acceptable to rework stories in later iterations under two circumstances:
1. **refactoring** – while working on a story, whenever your team identifies a way to improve code already completed, this is encouraged and supported with automated test scripts that continually validate all completed stories. You may capture a major refactoring as a separate story but basic refactoring effort should be assumed to be part of every delivery story.
2. business driver – a business sponsor or someone on your team may identify a need to re-work a previously completed story. In this case a new story is created and inserted into an iteration by potentially removing another story if need be.

INCREMENTAL ACCEPTANCE

By involving the business user on a regular basis to approve completed work, "final" acceptance of the overall scope of work is greatly facilitated. A working, trustful relationship is established that makes the final handoff much less contentious. Note, this incremental acceptance does not need to be a formal process (though in some cases this is advisable).

ENABLING PROCESSES

Core technology processes allow fully-functioning and tested software to be delivered in very short time windows – essential to incremental delivery. These include
Test-Driven Development,
Continuous Integration,
Full lifecycle QA,
and Automated Testing.



CHANGE IS WELCOME

Although it may seem contradictory to incremental delivery, it is expected and encouraged that your development team and the business sponsor will learn from the work completed in each iteration, and alter the planned set of stories for future iterations, either by introducing new stories (e.g., a reworking of a UI that is not satisfactory to all parties even following 3 stories), or by shifting the timing of certain stories from one iteration to another.

"Iteration" implies the potential to successively improve and revise work already completed. It does not mean that each story will be continually reworked until the business user is happy – once a story has been completed it is not reopened.

Sapient

S|A 3 **Tracking Progress – Project Level**

At the completion of each iteration, compare your actual progress – measured as completed story points – against the expected ideal progress. You and the business sponsor can immediately see and address any anomalies.

VELOCITY

The total of the completed story points for each iteration is defined as the velocity of that iteration – the rate at which the team can generate business value. The team's velocity one iteration is used to plan the stories that can be completed in the next iteration. Velocity is expected to vary from one iteration to the next, but be relatively consistent assuming a similar team size.

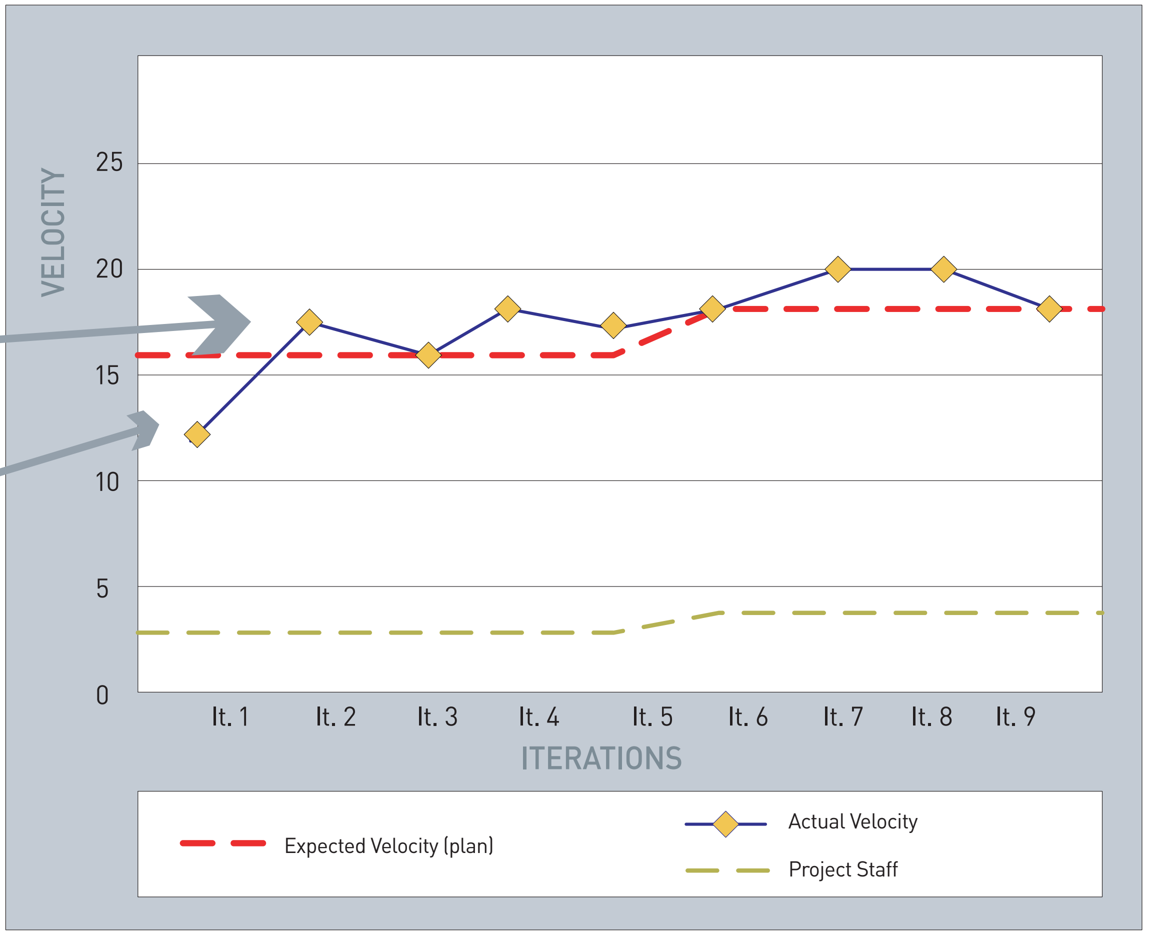
ITERATION 2 - COMPLETED

5	Story 1
1	Story 2
3	Story 3
1	Story 4
2	Story 5
3	Story 6
2	Story 7

TOTAL POINTS: 17

RAMP-UP TIME

As a new team begins working together, it will take one or two iterations to become as efficient as possible. Hence, expectations should be set that initial iterations' velocity will be lower than the norm.

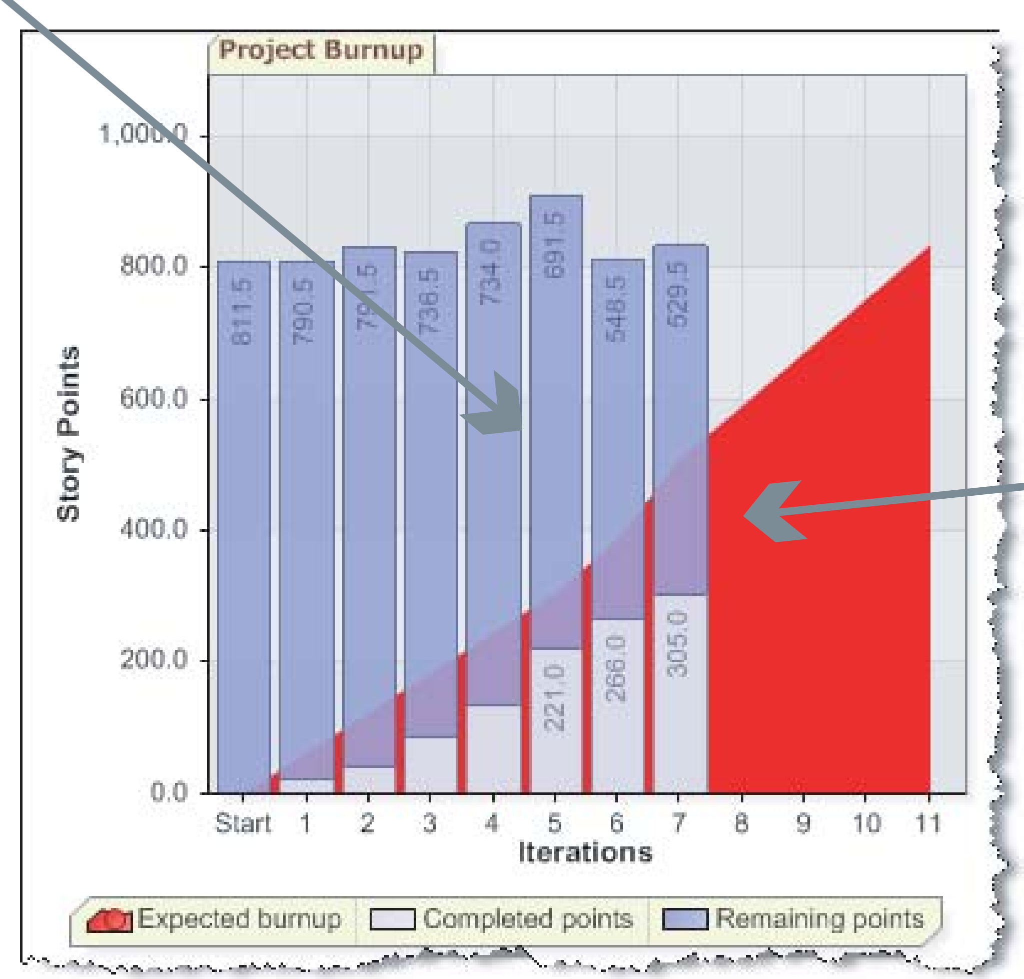


VARIATIONS IN TOTAL POINTS

Caused by scope changes during the course of the project.

BURNUP CHART

The cumulative completed story points are tracked against the total project story points in a view that compares expected and ideal progress to the team's actual progress.



The Project Burnup is one of the Progress Charts available in the Plan feature of ResultSpace^{RM}. It can be viewed by Release, or by the entire Project.

ACTUAL VS. EXPECTED

The expected is the minimum to be maintained to stay on track to complete the total identified scope for the project. Falling below the expected indicates a problem to be explored: most likely productivity issues or estimation issues. As noted earlier, initial iterations will most likely fall below the expected; later iterations may well exceed the expected. In this example, the project is unlikely to complete the scope without significant changes to the team, scope, or project duration.