Documentation Consultants'

SDLCforms

# Agile User Guide

SDLCforms.com

## Revision History

| Date | Version | Author | Change |
|---|---|---|---|
| 09/01/16 | 1.0 | Rebecca C. | Initial release |
| 09/23/16 | 1.1 | Ken F. | Incorporated Planning Poker Companion form. |
| 06/23/19 | 1.1 | Ken F. | Rearranged pages for better readability. |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

## Table of Contents

# 1 Introduction

This document is an Agile User Guide for reviewing the purpose of each form or template in the Documentation Consultants' SDLCforms Agile documentation.

This Agile User Guide includes the following:

1. A summary table briefly explaining the forms that are provided in the SDLCforms Agile packages.
2. A detailed explanation of the forms and templates.

## Audience

This document is intended for program managers, developers and quality assurance personnel who will utilize Documentation Consultants' SDLCforms Agile documentation on a daily basis within projects.

## 2 Overview of the Forms and Templates

| Form Name | Type | Description |
|---|---|---|
| **Project Charter** |  | The Project Charter is developed early in the project lifecycle and clearly sets the stage by aligning the team and stakeholders by setting goals and expectations. In most cases, the Project Charter, in conjunction with the Vision Statement, is the vehicle provided to the project sponsors as a part of the approval process. |
| **Vision Statement** |  | The Vision Statement identifies a future state for the product when the product reaches completion. The vision focuses on the conditions and defines the stakeholder's view of the product/service to be developed, specified in non-technical terms defining the stakeholder's key needs and features. |
| **Risk Management Register** |  | The Risk Management Register provides a vehicle for recording potential risks on a project, by recording, as a minimum, a brief explanation of the risk, the proposed mitigating strategy, the probability of the risk, size of loss and risk exposure (in days), actions needed to resolve the risk, and the current status. |
| **Project Data Sheet** |  | The objective of a Project Data Sheet (PDS) is in terms of scope, schedule and resources, how a project will deliver on a Vision Statement, and is a single summary of key business and quality objectives, product capabilities, and project management information. |
| **Theme/Epic Structure (Excel)** |  | The Theme / Epic Structure provides the ability to graphically identify a hierarchy of an optional high-level theme and /or epic and depicts the user stories within that hierarchy. |
| **Theme/Epic Structure (Visio)** |  | This is the optional Visio version of the Theme / Epic Structure (see Theme / Epic Structure (Excel)) to provide ease of working with "drag and drop" objects to update the document. |
| **Planning Poker Companion** |  | The Planning Poker Companion is a method for estimating story points for user stories accomplished by entering values into the document provided by the Agile Team with four optional methods for capturing and displaying results: Fibonacci Series, Numeric Series, Hours Approximation and Days Approximation. |
| **Story Points Allocation** |  | A user story is a particular business need assigned to the software development team. Using estimations of story points rather than time allows development teams to be less precise where a story point is a metric used to estimate the difficulty of implementing a given story.<br>Story Points Allocation provides a vehicle for recording story points that have been allocated for each user story. |

| | | |
|---|---|---|
| **Release Plan** |  | A Release Plan communicates, to the level of accuracy that is reasonable, when the release will be available, what features will be in the release, and what user stories are included in the release.<br>The Release Plan is a collaborative effort involving these roles: the Scrum master facilitates the meeting. The Product owner represents a general view of the product backlog. This effort provides insights into technical feasibility and dependencies. |
| **Sprint Planning & Meeting Schedule** |  | The Sprint Planning & Meeting Schedule provides a vehicle for recording a status of sprints and associated tasks for a given period.<br>Sprint planning is a collaborative effort involving a Scrum Master, who facilitates the meeting, a Product Owner, who clarifies the details of the product backlog items and their respective acceptance criteria, and the Entire Agile Team, who define the work and effort necessary to meet their sprint commitment. |
| **Sprint Backlog Status (Excel)** |  | Generally, the Sprint Backlog Status as an output of the Sprint Planning & Meeting Schedule to define the work that remains on the project. The sprint backlog contains many things not identified on the Product Backlog including specific tasks that have been decomposed from the accepted user stories.<br>Two versions of the Sprint Backlog Status are provided in the package - this document in Excel and one in Visio, as some organizations prefer to work in Excel, while the Visio version facilitates ease of "drag and drop" to move/copy objects within the document. |
| **Sprint Backlog Status (Visio)** |  | Generally, the Sprint Backlog Status is an output of the Sprint Planning & Meeting Schedule to define the work that remains on the project. The sprint backlog contains many things not identified on the Product Backlog including specific tasks that have been decomposed from the accepted user stories.<br>As mentioned above, two versions are provided in the package – either the version above in Excel or this Visio version, as this Visio version facilitates ease of "drag and drop" to move/copy objects within the document. |
| **14-Day Sprint Backlog-Capacity** |  | The 14-day Sprint Backlog contains the output of a sprint planning meeting. It contains a list of tasks that the Scrum team needs to complete during the sprint in order to convert a selected set of product backlog items into deliverable functionality within a period of 14 days.<br>The 14-day Sprint Backlog contains identification of the various sprints, and capacity for an elapsed period of 14 days. |
| **30-Day Sprint Backlog-Capacity** |  | The 30-day Sprint Backlog contains the output of a sprint planning meeting. It contains a list of tasks that the Scrum team needs to complete during the sprint in order to convert a selected set of product backlog items into deliverable functionality within a period of 30 days. |

| | | |
|---|---|---|
| | | The 30-day Sprint Backlog contains identification of the various sprints, and capacity for an elapsed period of 30 days. |
| **Product Backlog Grooming** |  | Product Backlog Grooming is a meeting that is held near the end of one sprint to ensure the backlog is ready for the next sprint.<br>During a product backlog refinement meeting, the team and product owner discuss the top items on the product backlog and record decisions in the Product Backlog Grooming artifact. |
| **Story Completion Checklist** |  | The Story Completion Checklist contains the story name, activity, work products and status. As Agile stories are finished, the product owner will work with their Agile team and the team's scrum master to make sure that finished stories are truly complete.<br>A definition of "Done" is a clear and concise list of requirements that software must adhere to for the team to call it complete. |
| **Iteration Retrospective** |  | An Agile retrospective is a meeting that's held at the end of an iteration. The process of retrospecting is at the heart and soul of Scrum. During the retrospective review, the team reflects on what transpired during the iteration and identifies actions for improvement going forward. |
| **Sprint Burn-Up Chart** |  | A Sprint Burn-Up Chart is a graph that shows the progress of work toward a *goal* line associated with a value on the vertical axis.<br>In the simplest form of burn up chart there are two lines on the chart:<br>1. A total work line (the project scope line)<br>2. A work completed line<br>As work is completed over *time* (the horizontal axis), the progress line moves up (burns up) to be nearer to the desired completion line. We can show projected outcomes on burn-up charts by calculating a trend line to see when work might be completed. |
| **Sprint Burndown Chart** |  | The Sprint Burndown Chart displays the remaining effort for a given period of time. Work remaining is the Y axis and working days is the X axis. When the x and y axis coincide, the work is complete.<br>When tracking product development using the Burndown Chart, teams can use a Sprint Burndown Chart and a Release Burndown Chart. |
| **Sprint Velocity Chart** |  | The Velocity Chart shows the amount of value delivered in each sprint, enabling you to predict the amount of work the team can get done in future sprints. It is useful during your sprint planning meetings, to help you decide how much work you can feasibly commit to. |

# 4 Detailed Explanation of Forms and Templates

## 4.1 Introduction

Documentation Consultants forms and templates (written in Microsoft Word and Excel) support organizational processes, specifically business and information systems that use or wish to use a Software Development Life Cycle (SDLC) methodology.

## 4.2 Description of Forms and Templates

The following tables provide an overview of each phase and a description of the forms and templates in the phase and documentation within the Agile phases. Use and updating/completion of the various forms is dependent upon project complexity, management's work philosophy, auditing requirements, and government regulations.

## 4.3 Envision Phase

### 4.3.1 Envision Phase Overview

This phase is to determine the product vision and project scope, the project community, and how the team will work together. The term 'envision' is a clear departure from traditional phase names such as initiate and plan, which while subtle, is also significant. This is because when envisioning you inadvertently accept a level of mishap and are therefore ready to make any necessary adjustments, in contrast to a set plan which has more rigorous connotations. The envision phase covers the 'Who? What? And How?'

Without a vision, all other efforts of creating a successful project will not bear fruit. The truth is, creating a compelling vision is very challenging. It takes work and leadership. Due to the myriad of paths that can be pursued it can often be daunting to create a cohesive vision. Compounding the problem is there is in fact no fixed rule for creating a great visionary statement.

Hence, it is important there is an effective leader to articulate a vision. The purpose of the envisioning phase is to clearly identify what needs to be done and how it is to be accomplished.

During the Envision Phase, the vision constantly changes as new information is gathered. Moving on from this stage; however, the vision should be reviewed periodically to ensure a clear understanding by all those involved in the project.

From a more technical perspective, the Envision Phase can be broken down into four practices:

- Vision
- Scope
- Community
- Approach.

When creating a vision, a product 'vision box' or project architecture could be used to streamline a vision. For example, a product vision box forces the project team to condense information into limited space in a box. A project data sheet forces the team to condense key scope plans and limitations on a single page. Limiting the team to a single page or box for example, increases focus and collaboration.

### 4.3.2 Envision Phase Forms & Templates

## Project Charter

| | |
|---|---|
| **Purpose:** | The Project Charter is developed early in the project lifecycle and clearly sets the stage by aligning the team and stakeholders by setting goals and expectations. In most cases, the Project Charter, in con junction with the Vision Statement, is the vehicle provided to the project sponsors as a part of the approval process. |
| **Description:** | From a project management perspective, the document:<br>• Establishes the vision state for the project<br>• Defines key goals and requirements<br>• Captures and sets customer expectations<br>• Defines project participants and their roles<br>• Defines limits and constraints<br>• Establishes all resource needs and overall cost targets<br>• Creates a high level view to the WBS and schedule<br>• Initiates negotiation and tradeoff<br>• Ultimately defines success<br><br>It is generally accepted that the desired intent is to limit the document to one page; however, as necessity arises, it may be necessary to provide the data on two pages.<br><br>The Project Charter consists of the following data elements:<br>• Product Vision<br>• Mission Statement<br>• Success Criteria<br>• Target Objectives<br>• Key Deliverables<br>• Team Members<br>• Timeline<br>• Assumptions and Constraints<br>• Risks and Issues. |

## Form Image


**Project Charter**
**Project Name**
Version

*Note: In any table, select and delete any blue line text; then click Home→Styles and select "Table Text" to restore the cells to the default value.*

| Product Vision | Date | Rev. | Focus |
|---|---|---|---|
| *Enter an abbreviated version of the product vision from the Vision Statement* | *99/99/99* | *A* | *Enter focus of new project / product* |

| Mission Statement | Success Criteria |
|---|---|
| *Enter a statement that describes the overall objectives of the project.* | *Enter how you will measure success in achieving project vision.* |
| | |
| | |
| | |
| | |
| | |

| Target Objectives | | Key Deliverables | |
|---|---|---|---|
| *Finalize organizational structure* | *99/99/99* | *Approved organization chart* | *Agile training complete* |
| *Obtain budget approval* | *99/99/99* | *Approved budget* | *Documentation deliverables defined* |
| *Hire contract personnel* | *99/99/99* | *Integration of team members* | |
| *Final review of team members* | *99/99/99* | | |
| *Train personnel in Agile* | *99/99/99* | | |

| Team Members | | Timeline | |
|---|---|---|---|
| **Name** | **Responsibility** | **Actions / Milestones** | **Target Date** |
| *Bill Gates* | *Project Manager* | *Enter any actions or milestones, such as:* | *99/99/99* |
| *Jessie Stewart* | *Product Manager* | • *Initial meeting to develop high level requirements and team organization chart* | *99/99/99* |
| *Joe Jones* | *Marketing* | • *Organization chart and budget approval meeting* | *99/99/99* |
| *Betty Barsam* | *Sales* | • *Create user stories.* | *99/99/99* |
| *Nancy Nansen* | *Development* | | |
| | | | |
| | | | |

| Assumptions and Constraints | Risks and Issues |
|---|---|
| *Enter and assumptions or constraints that will impact the project* | *Enter any risks or issues.* |

 Document: A2000 Page 1 of 1

## Vision Statement

| | |
|---|---|
| **Purpose:** | The Vision Statement identifies a future state for the product when the product reaches completion. The vision focuses on the conditions and defines the stakeholder's view of the product/service to be developed, specified in non-technical terms defining the stakeholder's key needs and features. |
| **Description:** | Often, the creation of the Vision Statement is lead by the product owner in close collaboration with the team<br><br>The vision must be clear and easy to understand by all team members with a common purpose, but must be written with carefully selected verbiage as to not be misunderstood by team members.<br><br>The Vision Statement consists of the following data elements:<br>• Title<br>• What target group does the vision address?<br>• How does the product create value for the users?<br>• What makes the product special and desirable?<br>• Business Goals – How will the product benefit the company?<br>• Who are the competitors and what are the strengths and weaknesses of this product versus the competitor's product?<br>• Where will the money come from to fund the project?<br>• What are the major cost factors to develop, market and maintain the project? |

## Form Image:


**Vision Statement**
**Project Name**
Version

*Note: In any table, select and delete any blue line text; then click Home→Styles and select "Table Text" to restore the cells to the default value.*

| Agile Vision Statement | | | |
|---|---|---|---|
| Title: | *Describe project / product purpose.* | | |
| **Target Group** | **Needs** | **Product** | **Business Goals** |
| *Which market segment does the project/ product address?*<br>*Who are the users and customers?* | *Define how does the project/product create value for the users?*<br>*What problem(s) does the vision solve?*<br>*What benefits does it provide?* | *Describe the product.*<br>*What makes the product special and desirable?*<br>*It is realistic and feasible to develop the project / product?* | *How will the project / product benefit the company?*<br>*What are the specific business goals?*<br>*What are the two or three most important goals?* |
| **Competitors** | **Revenue Sources** | **Cost Factors** | **Channels** |
| *Who are the project's / product's main competitors?*<br>*What are the strengths and weaknesses of this product versus the competitor's product?* | *How can the project / product be monetized to generate revenue?*<br>*What steps must occur to open up the revenue sources?* | *What are the main cost factors to develop, market, and maintain the project / product?*<br>*Which resources and activities incur the highest costs?* | *What channels will you use to market and sell the product to the target group of customers?*<br>*Identify the status of these channels. Do they exist or must they be created?* |

## Risk Management Register

| | |
|---|---|
| **Purpose:** | The Risk Management Register provides a vehicle for recording potential risks on a project, by recording, as a minimum, a brief explanation of the risk, the proposed mitigating strategy, the probability of the risk, size of loss (in days), and risk exposure (in days), actions needed to resolve the risk, and the current status. |
| **Description:** | It is imperative that the Risk Management Register is made available to the entire team to ensure that all team members are constantly on alert about these potential risks. At every sprint meeting, the Register must be reviewed and updated with new information obtained over the life of the sprint.<br><br>The following data elements are provided in the Register: |

| | |
|---|---|
| Risk | This brief explanation of the risk should be clearly understood by all team members. |
| Mitigation Strategy | How does the team propose to mitigate the risk? |
| Date Identified | The originating date. |
| Owner | What department/person owns the risk? Who is responsible to manage, control and take action in response to the risk? |
| Probability of Risk | Generally, this is stated as a percentage from 1% to 100%. |
| Size of Loss | The size of the loss is measured in days. |
| Risk Exposure | The risk exposure is measured in days. |
| Action to Manage Risk | What actions are underway to minimize or eliminate the risk? |
| Current Status | Suggested statuses = “Open,” “Closed,” or “Under Review.” |

## Form Image:


**Risk Management Register**
**Project Name**
**Version**

*Note: In any table, select and delete any blue line text; then click Home→Styles and select "Table Text" to restore the cells to the default value.*

| Risk | Mitigation Strategy | Date Identifed | Owner | Probability of Risk | Size of Loss (Days) | Risk Exposure (Days) | Action to Manage Risk | Current Status |
|---|---|---|---|---|---|---|---|---|
| *Utilizing new integration technology to connect to partnering systems.* | • *Monitor frequently.*<br>• *Search for potential new partners*<br>• *Consider internal staff taking over function* | *07/23/16* | *IT* | *30%* | *15* | *12* | *Review alternative to new integration technology* | *Open* |
| *Dependent systems are incompatible during integration* | • *Consider other dependent systems*<br>• *Consider middleware* | *08/14/16* | *IT* | *80%* | *12* | *9.5* | | *Closed on 09/27/16* |
| *There won't be time in the QA process to validate all possible combinations of cell phones* | • *Develop plan to test only stories with high point values*<br>• *Consider offering only most common phones* | *08/22/16* | *Marketing* | *87%* | *10* | *3* | *Consider using only 6 most popular phones.* | *Under Review* |
| *Strong possibility that system performance will be severely degraded due to huge volume of transactions* | • *Evaluate value of adding more servers.*<br>• *Re-evaluate if transaction volume is factual* | *09/12/16* | *Operations* | *90%* | *20* | *12* | *Consider adding high-speed servers to enhance performance.* | *Closed* |
| *Insufficient time will be allotted to provide end user documentation in 12 languages* | • *Consider reducing number of languages* | *09/24/16* | *Technical Support* | *30%* | *2* | *2* | *Consider reduction from 12 to 6 langages.* | *Open* |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

 Document: A4000 Page 1 of 1

## Project Data Sheet

**Purpose:**

The purpose of a Project Data Sheet (PDS) is in terms of scope, schedule and resources, how a project will deliver on a Vision Statement, and is a single summary of key business and quality objectives, product capabilities, and project management information. It is a simple document with a powerful impact whose condensed format appeals to the entire project community and constantly reminds them of the strategic aspects of the project.

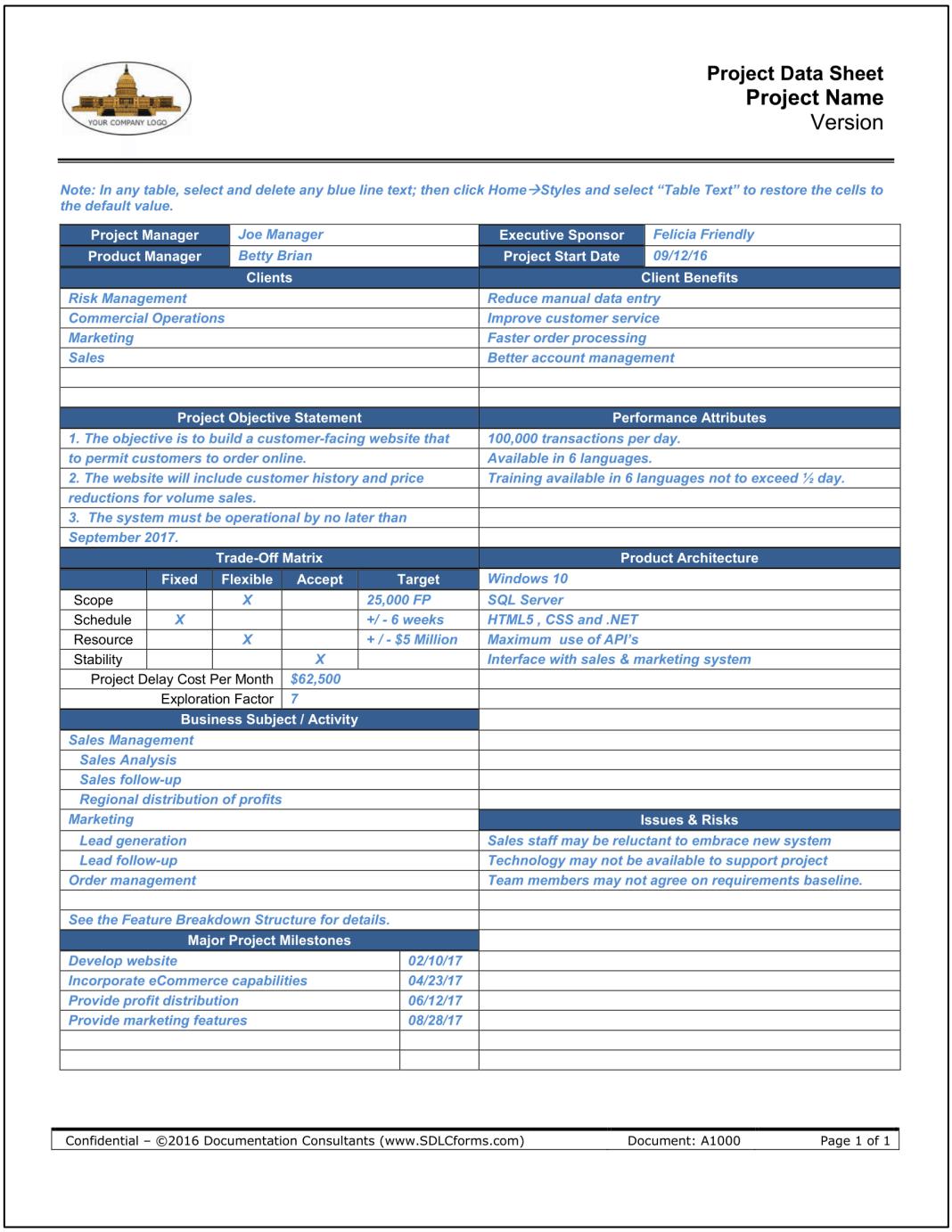
**Description:**

The Project Data Sheet (PDS) is an integral part in the evolution of a project plan. While the Vision Statement is an expansive view of what the product could become, the Vision Statement limits product development to the current scope, schedule and cost constraints.

The following data elements are provided in the Product Data Sheet:

| | |
|---|---|
| Project Manager | Identify the Project Manager. |
| Product Manager | This is the product owner. |
| Executive Sponsor | Who sponsors the project and has the responsibility for the project plan and constraints. |
| Project Start Date | When did the project start? |
| Clients | List key clients and customers. |
| Client Benefits | Identify the major b benefits the client has identified as the goals of the project. |
| Project Objective Statement | Provide a short and concise that identifies scope, schedule and cost information. |
| Performance Attributes | Identify performance and quality aspects of the product. |
| Trade-Off Matrix | For Scope, Schedule, Resources and Stability, identify whether the trade-off value of each is Fixed, Flexible or Accept. In addition, identify the target for each element. |
| Product Architecture | Provide a high level overview of the proposed product architecture (e.g., Windows 10, SQL Server. HTML, CSS, .NET, etc.) |
| Business Subject / Activity | Identify the key business groups and their objectives. |
| Issues and Risks | Identify factors that could impact the project. |
| Major Project Milestones | Identify critical deadlines for the project from inception to completion. |

## Form Image:



YOUR COMPANY LOGO

**Project Data Sheet**
**Project Name**
Version

*Note: In any table, select and delete any blue line text; then click Home→Styles and select "Table Text" to restore the cells to the default value.*

| | | | |
|---|---|---|---|
| **Project Manager** | *Joe Manager* | **Executive Sponsor** | *Felicia Friendly* |
| **Product Manager** | *Betty Brian* | **Project Start Date** | *09/12/16* |

| **Clients** | **Client Benefits** |
|---|---|
| *Risk Management* | *Reduce manual data entry* |
| *Commercial Operations* | *Improve customer service* |
| *Marketing* | *Faster order processing* |
| *Sales* | *Better account management* |
| | |
| | |

| **Project Objective Statement** | **Performance Attributes** |
|---|---|
| *1. The objective is to build a customer-facing website that* | *100,000 transactions per day.* |
| *to permit customers to order online.* | *Available in 6 languages.* |
| *2. The website will include customer history and price* | *Training available in 6 languages not to exceed ½ day.* |
| *reductions for volume sales.* | |
| *3. The system must be operational by no later than* | |
| *September 2017.* | |

| **Trade-Off Matrix** | | | | | **Product Architecture** |
|---|---|---|---|---|---|
| | **Fixed** | **Flexible** | **Accept** | **Target** | *Windows 10* |
| Scope | | *X* | | *25,000 FP* | *SQL Server* |
| Schedule | *X* | | | *+/ - 6 weeks* | *HTML5 , CSS and .NET* |
| Resource | | *X* | | *+ / - $5 Million* | *Maximum use of API's* |
| Stability | | | *X* | | *Interface with sales & marketing system* |
| Project Delay Cost Per Month | | | *$62,500* | | |
| Exploration Factor | | | *7* | | |

| **Business Subject / Activity** | |
|---|---|
| *Sales Management* | |
| *Sales Analysis* | |
| *Sales follow-up* | |
| *Regional distribution of profits* | |
| *Marketing* | **Issues & Risks** |
| *Lead generation* | *Sales staff may be reluctant to embrace new system* |
| *Lead follow-up* | *Technology may not be available to support project* |
| *Order management* | *Team members may not agree on requirements baseline.* |
| | |
| *See the Feature Breakdown Structure for details.* | |

| **Major Project Milestones** | |
|---|---|
| *Develop website* | *02/10/17* |
| *Incorporate eCommerce capabilities* | *04/23/17* |
| *Provide profit distribution* | *06/12/17* |
| *Provide marketing features* | *08/28/17* |
| | |
| | |

 Document: A1000 Page 1 of 1

## 4.4 Speculate Phase

### 4.4.1 Speculate Phase Overview

Unlike planning, speculating establishes a target and direction, but at the same time, it indicates that we expect much to change over the lifetime of a project. Unlike speculation, plans are usually conjectures about the future where people often expect the result to come directly from the plan. Deviations from the plan are therefore viewed as negative unlike with speculation where results are generally viewed as positive. Speculation is only one piece of information that will be examined to determine our course of action when iterating.

The result after speculating is a blueprint that outlines information about the products specification, platform architecture, resources, risk analysis, defect levels, business constraints and target schedules.

There are two crucial components of an iterative planning and development approach- **short iterative time boxes and features**. Short iterations help to accelerate the progress of a project by encouraging the project team to think with limited time to accomplish all aspects of product development.

For example, in iterative development, quality assurance activities are completed upon iterating. Thus, quality assurance staff members have to figure out how to be more effective and efficient than the previous iteration.

With regards to features, product development features are driven by first creating a product feature and then an extensive list of the features. Also, since more products are starting to include embedded software, hardware and software features are now strong contenders for feature-driven projects. The first concern for feature-based planning should be to make the process visible and understandable to the consumer team.

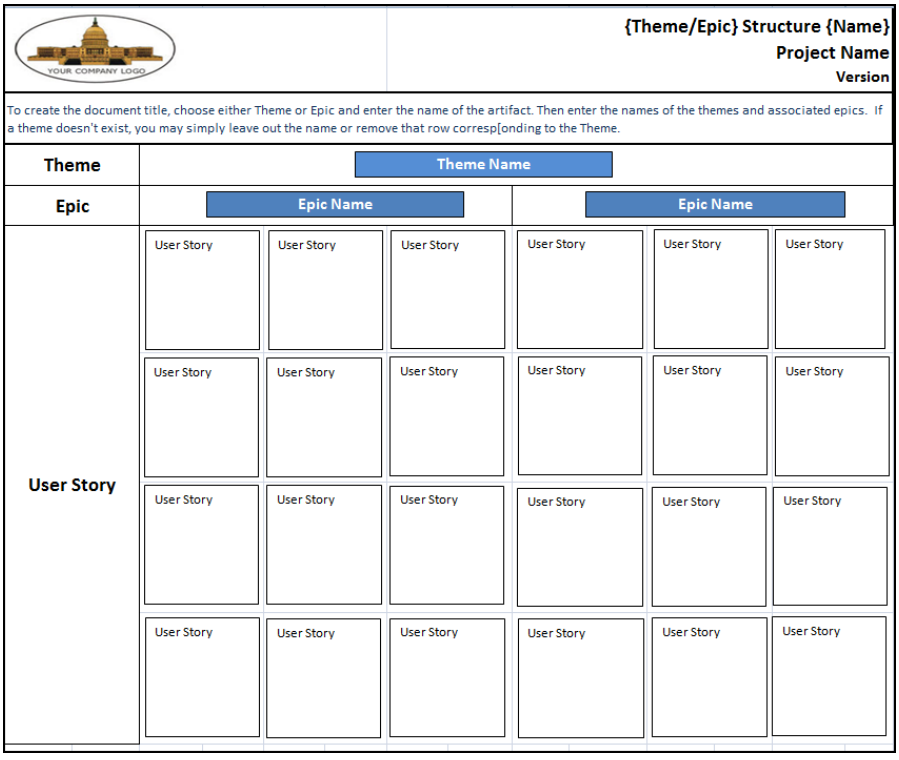
Often, the project team has allocated their time to making product specifications target the technical and engineering teams at the expense of consumer understanding. The project team should therefore note that features act as an interface between both engineers and consumers, and should work according to the understanding of both.

### 4.4.2. Speculate Phase Forms & Templates

## Theme / Epic Structure (Excel)

| | |
|---|---|
| **Purpose:** | The Theme / Epic Structure provides the ability to graphically identify a hierarchy of an optional high-level theme and /or epic and depicts the user stories within that hierarchy. |
| **Description:** | Definitions: A Theme is a collection of user stories. A Scrum epic is a large user story.<br><br>Collections of User Stories such as grouping of development work that logically goes together within a Sprint or Release (e.g. Collections of User Stories that affect the GUI design, database design, or implement core business logic or process).<br><br>Two versions of the Theme / Epic Structure are provided in the package - this document in Excel and one in Visio, as some organizations prefer to work in Excel, while the Visio version facilitates ease of "drag and drop" to move/copy objects within the document.<br><br>Note the title (in braces) of the artifact. Revise the title to specify whether theme or epic and the name of the theme or epic this illustration identifies.<br><br>If you are identifying only an Epic, then delete the row entitled “Theme, and delete the cells associated with the second occurrence of “Epic Name.”” |

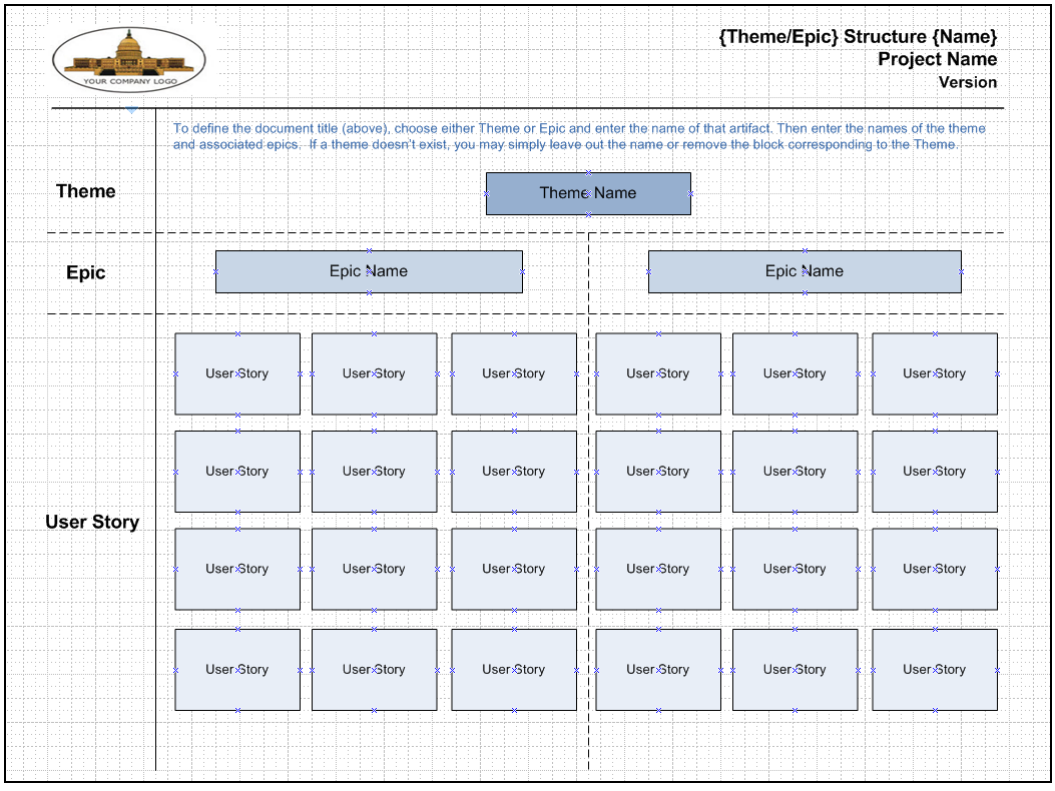
**Form Image**



## Theme / Epic Structure (Visio)

| | |
|---|---|
| **Purpose:** | This is the optional Visio version of the Theme / Epic Structure (see Theme / Epic Structure (Excel)) to provide ease of working with "drag and drop" objects to update the document. |
| **Description:** | Definitions: A Theme is a collection of user stories. A Scrum epic is a large user story.<br><br>Collections of User Stories such as grouping of development work that logically goes together within a Sprint or Release (e.g. Collections of User Stories that affect the GUI design, database design, or implement core business logic or process).<br><br>As opposed to the Excel version, you may drag-and-drop the various objects to restructure the diagram to define a specific theme or epic.<br><br>Note the title (in braces) of the artifact. Revise the title to specify whether theme or epic and the name of the theme or epic this illustration identifies.<br><br>If you are identifying only an Epic, then delete the objects entitled “Theme, and delete the objects associated with the second occurrence of “Epic Name.” |

**Form Image**



## Planning Poker Companion

| | |
|---|---|
| **Purpose:** | The Planning Poker Companion is a method for estimating story points for user stories accomplished by entering values provided by the Agile Team with four optional methods for capturing and displaying results: Fibonacci Series, Numeric Series, Hours Approximation and Days Approximation.<br>Averages are then calculated for: 1) Last Round Only, 2) Average All Rounds, and 3) Average Without First and Last Rounds. |
| **Description:** | Planning Poker (also called Scrum Poker) is a method for estimating story points for user stories accomplished by providing the Agile Team with cards of varying values by which they can select the card each individual believes approximates the value the user story contributes to the project.<br>The Planning Poker Companion provides a vehicle to record the card entries which contains all four types of tables to record those choices, which you may use optionally, or simply delete those charts you will never use.<br>**Table Design**<br>Since the four types of tables are designed as tables, you may add rows without the necessity of modifying any of the formulas. This will be accomplished automatically by Excel.<br>**Dropdown List Boxes**<br>Depending on which table you choose to use (Fibonacci, Hours Approximation, etc.), the Rounds columns in each table have been populated with the values that correspond to that selection. For example, in the Fibonacci Table, the cells have been populated with 0, 1, 2,3,5,8,13,22,55 and 89.<br>**Formulas:**<br>Formulas are embedded in the tables to calculate the following columns: Last Round Only, Average All Rounds, and Average W/O First & Last (Eliminate first and last entry).<br>**Last Round Only**<br>The Last Round Only column will display the last round that was entered. The formula checks each column to determine which Rounds column was the last column containing a numeric value.<br>=IF(ISBLANK(B6),0,IF(ISBLANK(C6),SUM(B6),IF(ISBLANK(D6),SUM(C6),IF(ISBLANK(E6),SUM(D6),IF(ISBLANK(F6),SUM(E6),IF(ISBLANK(G6),SUM(F6),IF(ISNUMBER(G6),SUM(G6)))))))<br>**Average All Rounds**<br>The Average All Rounds column will display an average of all rounds that have been entered. If there is no data in any cells, the value is 0.00.<br>=IF(SUM(B6:G6)>0, SUM(B6:G6) / COUNTA(B6:G6),0) |

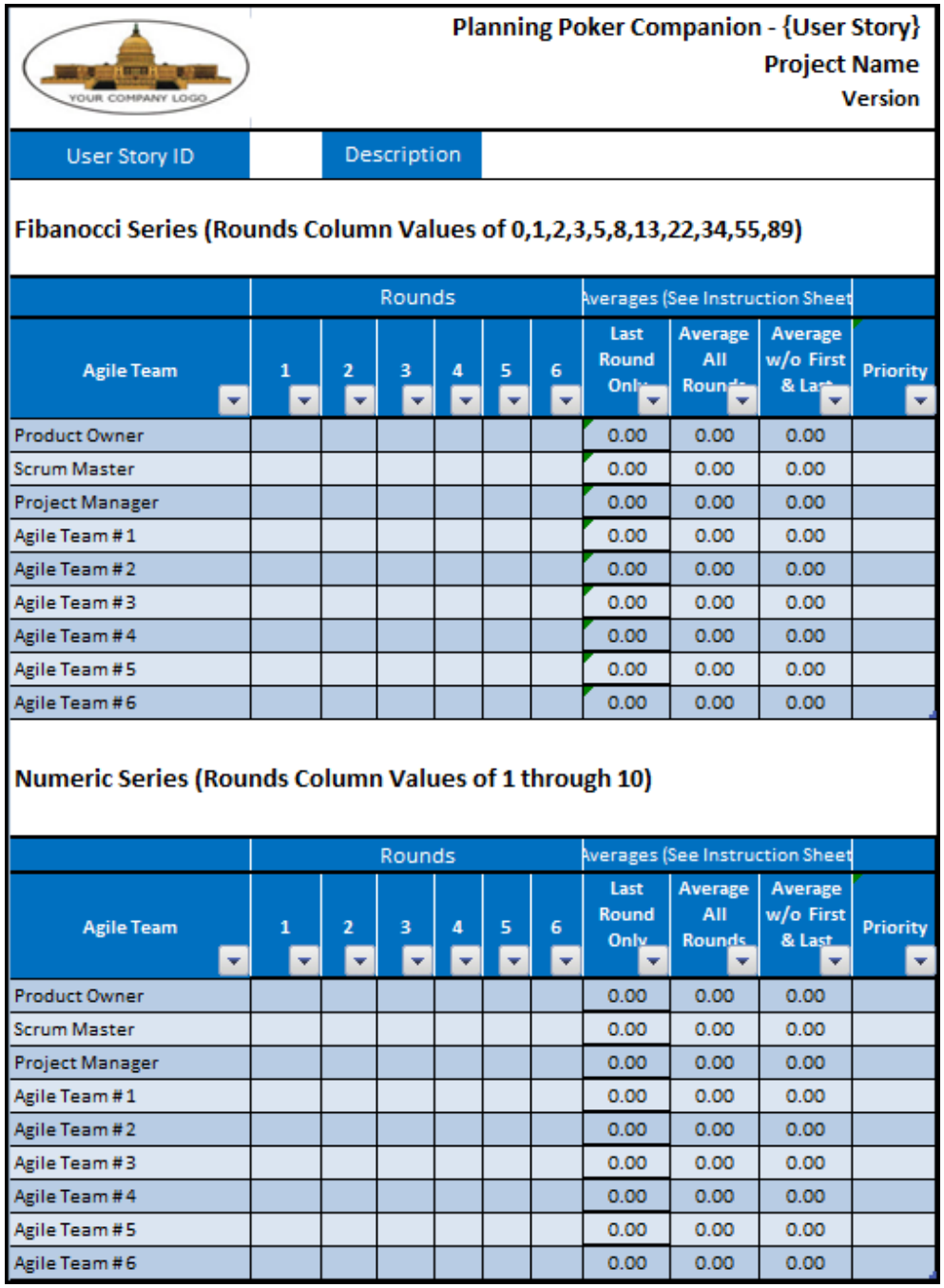
**Average Without First & Last**

The Average W/O First & Last column will provide an average of the values entered in all Rounds except for the first and last entries.

=IF(COUNTBLANK(B6:G6)=5,SUM(B6:G6),IF(COUNTBLANK(B6:G6)=4,SUM(B6:G6)/2,IF(COUNTBLANK(B6:G6)=3,SUM(C6:D6)/2,IF(COUNTBLANK(B6:G6)=2,SUM(C6:D6)/2,IF(COUNTBLANK(B6:G6)=1,SUM(C6:E6)/3,IF(COUNTBLANK(B6:G6)=0,SUM(C6:F6)/4,0))))))

If there are only two entries, it calculates an average. If there are three entries, it uses the last 2 entries. For example, if values have been entered in Rounds 1 through 5, the values shown in the Average W/O First & Last will be the average of Rounds columns 2 through 4.

## Form Image



Planning Poker Companion - {User Story}
Project Name
Version

| User Story ID | | Description | |
|---|---|---|---|

**Fibanocci Series (Rounds Column Values of 0,1,2,3,5,8,13,22,34,55,89)**

| | Rounds | | | | | | Averages (See Instruction Sheet | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Agile Team | 1 | 2 | 3 | 4 | 5 | 6 | Last Round Only | Average All Rounds | Average w/o First & Last | Priority |
| Product Owner | | | | | | | 0.00 | 0.00 | 0.00 | |
| Scrum Master | | | | | | | 0.00 | 0.00 | 0.00 | |
| Project Manager | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #1 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #2 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #3 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #4 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #5 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #6 | | | | | | | 0.00 | 0.00 | 0.00 | |

**Numeric Series (Rounds Column Values of 1 through 10)**

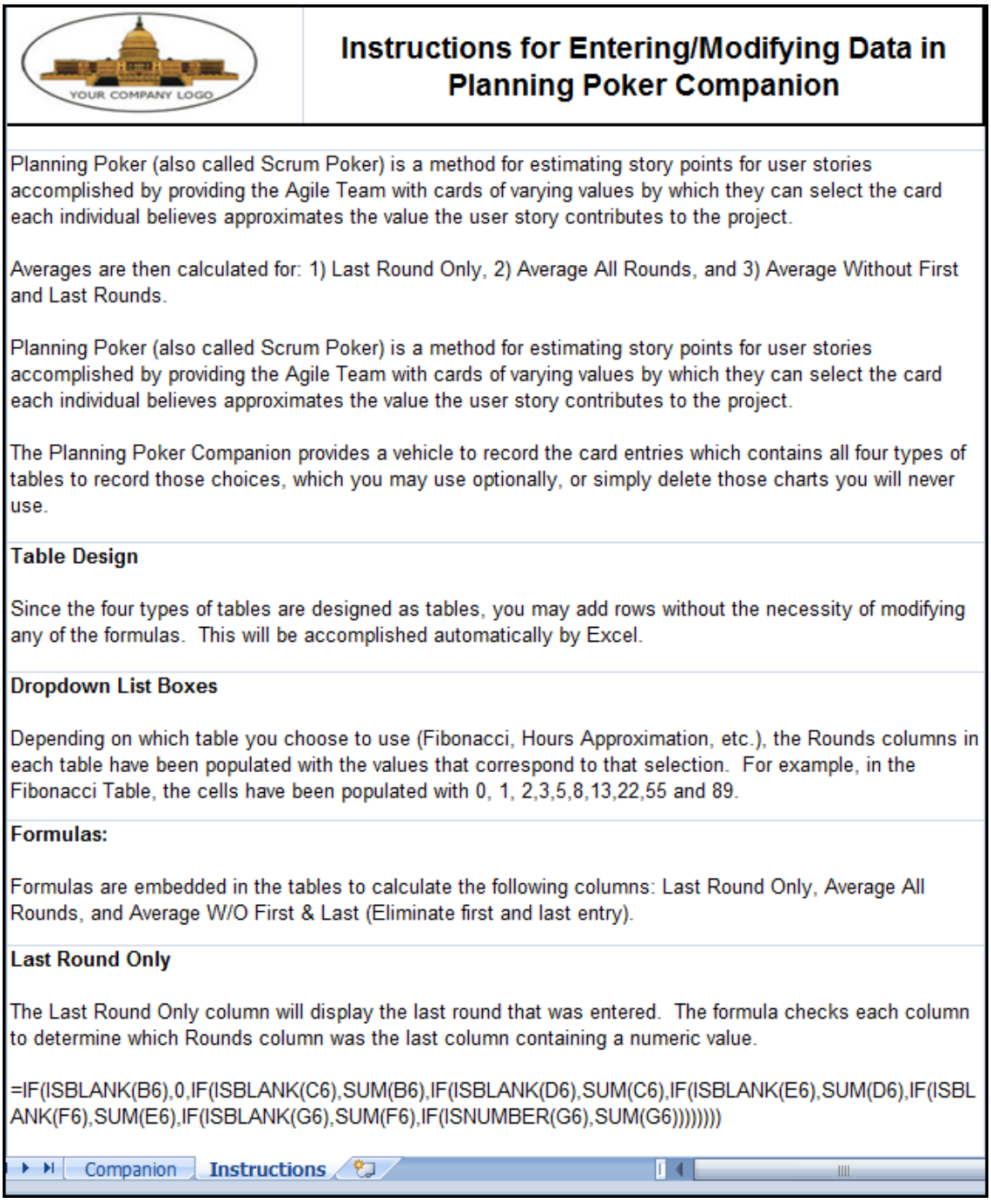
| | Rounds | | | | | | Averages (See Instruction Sheet | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Agile Team | 1 | 2 | 3 | 4 | 5 | 6 | Last Round Only | Average All Rounds | Average w/o First & Last | Priority |
| Product Owner | | | | | | | 0.00 | 0.00 | 0.00 | |
| Scrum Master | | | | | | | 0.00 | 0.00 | 0.00 | |
| Project Manager | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #1 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #2 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #3 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #4 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #5 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #6 | | | | | | | 0.00 | 0.00 | 0.00 | |

**Hours Approximation (Rounds Column Values of 0, .5, 1, 2.5, 10, 20, 40, 80)**

| | Rounds | | | | | | Averages (See Instruction Sheet) | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Agile Team | 1 | 2 | 3 | 4 | 5 | 6 | Last Round Only | Average All Rounds | Average w/o First & Last | Priority |
| Product Owner | | | | | | | 0.00 | 0.00 | 0.00 | |
| Scrum Master | | | | | | | 0.00 | 0.00 | 0.00 | |
| Project Manager | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #1 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #2 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #3 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #4 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #5 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #6 | | | | | | | 0.00 | 0.00 | 0.00 | |

**Days Approximation (Rounds Column Values of 0, .5, 1, 1.5, 2, 3, 4, 5, 6, 7, 8)**

| | Rounds | | | | | | Averages (See Instruction Sheet) | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Agile Team | 1 | 2 | 3 | 4 | 5 | 6 | Last Round Only | Average All Rounds | Average w/o First & Last | Priority |
| Product Owner | | | | | | | 0.00 | 0.00 | 0.00 | |
| Scrum Master | | | | | | | 0.00 | 0.00 | 0.00 | |
| Project Manager | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #1 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #2 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #3 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #4 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #5 | | | | | | | 0.00 | 0.00 | 0.00 | |
| Agile Team #6 | | | | | | | 0.00 | 0.00 | 0.00 | |


YOUR COMPANY LOGO
Instructions for Entering/Modifying Data in Planning Poker Companion
Planning Poker (also called Scrum Poker) is a method for estimating story points for user stories accomplished by providing the Agile Team with cards of varying values by which they can select the card each individual believes approximates the value the user story contributes to the project.
Averages are then calculated for: 1) Last Round Only, 2) Average All Rounds, and 3) Average Without First and Last Rounds.
Planning Poker (also called Scrum Poker) is a method for estimating story points for user stories accomplished by providing the Agile Team with cards of varying values by which they can select the card each individual believes approximates the value the user story contributes to the project.
The Planning Poker Companion provides a vehicle to record the card entries which contains all four types of tables to record those choices, which you may use optionally, or simply delete those charts you will never use.
Table Design
Since the four types of tables are designed as tables, you may add rows without the necessity of modifying any of the formulas. This will be accomplished automatically by Excel.
Dropdown List Boxes
Depending on which table you choose to use (Fibonacci, Hours Approximation, etc.), the Rounds columns in each table have been populated with the values that correspond to that selection. For example, in the Fibonacci Table, the cells have been populated with 0, 1, 2,3,5,8,13,22,55 and 89.
Formulas:
Formulas are embedded in the tables to calculate the following columns: Last Round Only, Average All Rounds, and Average W/O First & Last (Eliminate first and last entry).
Last Round Only
The Last Round Only column will display the last round that was entered. The formula checks each column to determine which Rounds column was the last column containing a numeric value.
=IF(ISBLANK(B6),0,IF(ISBLANK(C6),SUM(B6),IF(ISBLANK(D6),SUM(C6),IF(ISBLANK(E6),SUM(D6),IF(ISBLANK(F6),SUM(E6),IF(ISBLANK(G6),SUM(F6),IF(ISNUMBER(G6),SUM(G6))))))))
Companion
Instructions

## Story Points Allocation

| | |
|---|---|
| **Purpose:** | A user story is a particular business need assigned to the software development team. Using estimations of story points rather than time allows development teams to be less precise where a story point is a metric used to estimate the difficulty of implementing a given story.<br>Story Points Allocation provides a vehicle for recording story points that have been allocated for each user story, |
| **Description:** | It may be difficult, for example, to estimate how long a particular feature will take to develop but relatively easy to understand if it is more complex than others, in which case it should be assigned more story points.<br>Elements considered in assigning a story point include the complexity of the story, the number of unknown factors and the potential effort required to implement it. Story points are usually expressed according to a numerical range, such as an adaptation of a Fibonacci sequence, or in accordance with a mathematical series, or hours or days approximation (See Planning Poker Companion for examples). |

## Form Image


**Story Points Allocation**
**Project Name**
Version

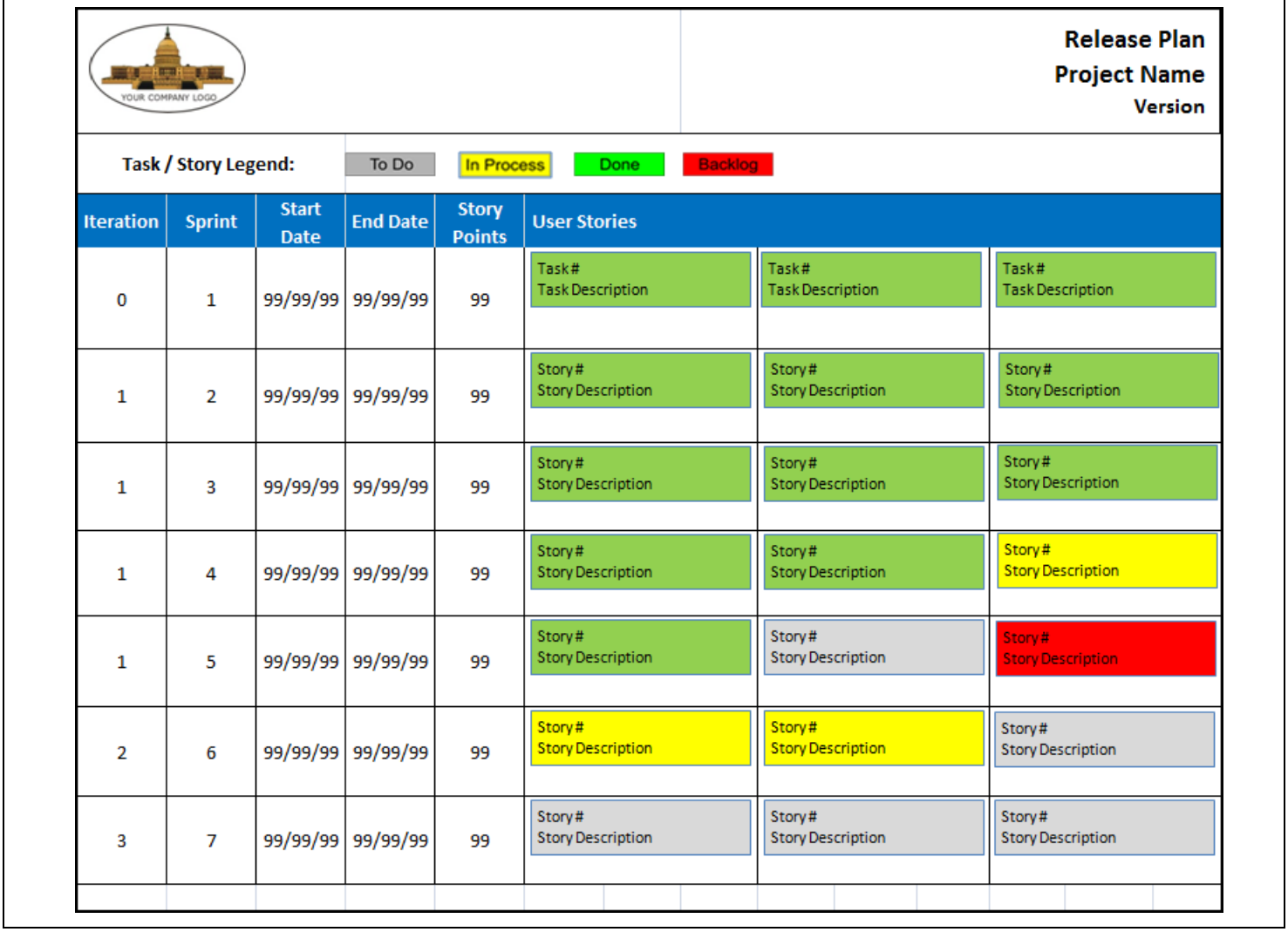
*Note: In any table, select and delete any blue line text; then click Home→Styles and select "Table Text" to restore the cells to the default value.*

| Sprint | Epic or Theme (if applicable) | User Story | Points |
|---|---|---|---|
| *1* | *Develop online customer feedback* | *Customers can enter their name, email address and phone number to identify who they are.* | *6* |
| | | *Customers will be able to rate the products on a scale of 1 to 10.* | *3* |
| | | *Customers will be able to enter comments on their ratings.* | *1* |
| | | *Average ratings per product will be provided for Marketing.* | *12* |
| | | *Marketing will have capability to respond to customer feedback* | *3* |
| *2* | *Provide reports on customer feedback* | *Reports will be provided to Sales & Marketing averaging all product ratings.* | *12* |
| | | *Reports will be formatted in both printed form and as Excel spreadsheets.* | *3* |
| | | *Reports will be provided by as weekly summaries and available on-demand.* | *6* |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

## Release Plan

| | |
|---|---|
| **Purpose:** | A Release Plan communicates, to the level of accuracy that is reasonable, when the release will be available, what features will be in the release, and what user stories are included in the release.<br>The Release Plan is a collaborative effort involving these roles: the Scrum master facilitates the meeting. The Product owner represents a general view of the product backlog.<br>This effort provides insights into technical feasibility and dependencies. |
| **Description:** | The Release Plan consists of the following date elements:<br>• Iteration<br>• Sprints included<br>• Start and end dates<br>• Collective story points for each sprint<br>• User stories included in each iteration/sprint color coded to signify:<br>  ○ To Do items<br>  ○ In Process items<br>  ○ Done items<br>  ○ Backlog items. |

## Form Image

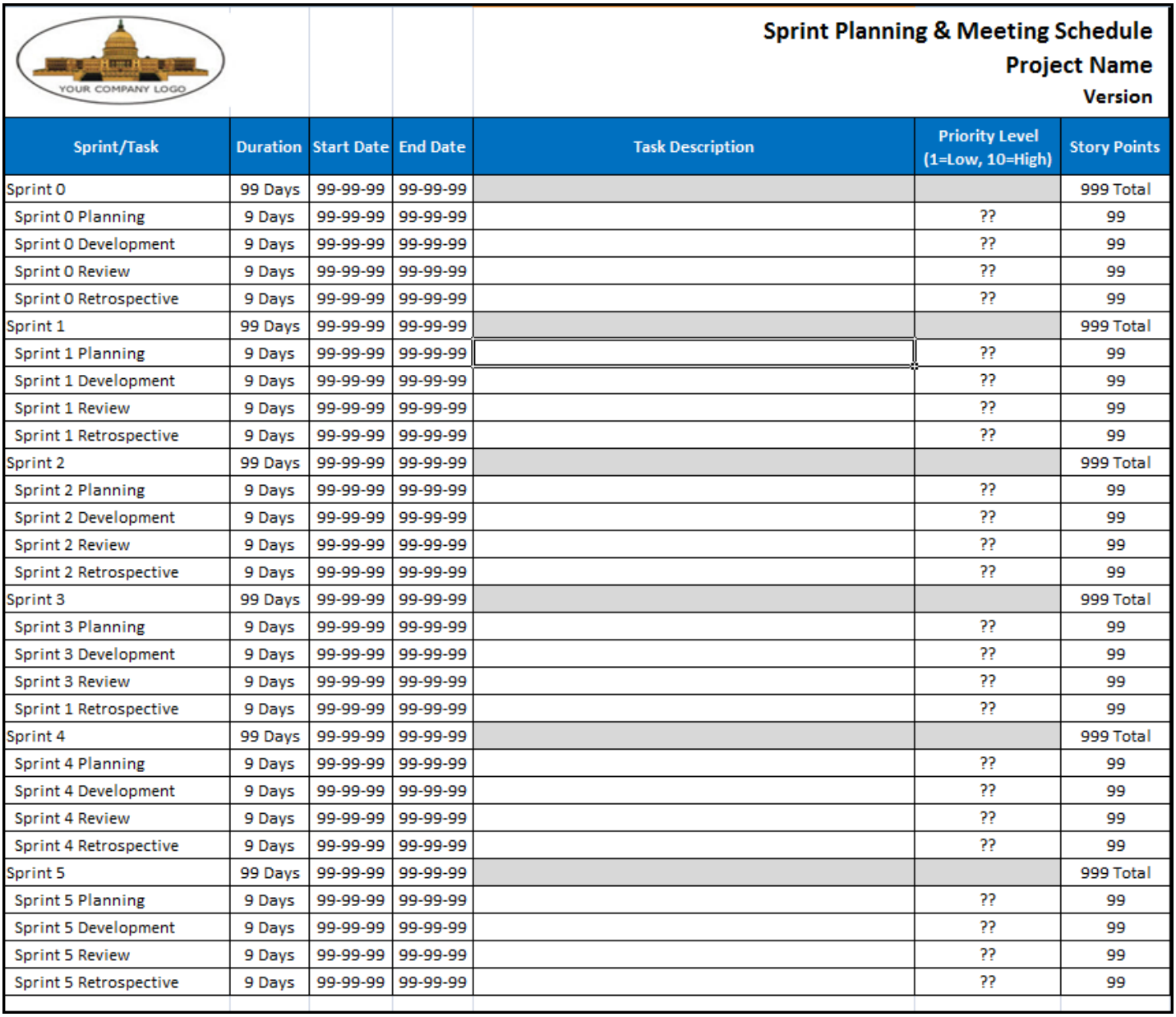



| Iteration | Sprint | Start Date | End Date | Story Points | User Stories | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 99/99/99 | 99/99/99 | 99 | Task # Task Description | Task # Task Description | Task # Task Description |
| 1 | 2 | 99/99/99 | 99/99/99 | 99 | Story # Story Description | Story # Story Description | Story # Story Description |
| 1 | 3 | 99/99/99 | 99/99/99 | 99 | Story # Story Description | Story # Story Description | Story # Story Description |
| 1 | 4 | 99/99/99 | 99/99/99 | 99 | Story # Story Description | Story # Story Description | Story # Story Description |
| 1 | 5 | 99/99/99 | 99/99/99 | 99 | Story # Story Description | Story # Story Description | Story # Story Description |
| 2 | 6 | 99/99/99 | 99/99/99 | 99 | Story # Story Description | Story # Story Description | Story # Story Description |
| 3 | 7 | 99/99/99 | 99/99/99 | 99 | Story # Story Description | Story # Story Description | Story # Story Description |

## Sprint Planning & Meeting Schedule

| | |
|---|---|
| **Purpose:** | The Sprint Planning & Meeting Schedule provides a vehicle for recording a status of sprints and associated tasks for a given period.<br>Sprint planning is a collaborative meeting attended by Scrum Master, who facilitates the meeting, a Product Owner, who clarifies the details of the product backlog items and their respective acceptance criteria, and the Entire Agile Team, who define the work and effort necessary to meet their sprint commitment. |
| **Description:** | During the sprint planning meeting, the product owner describes the highest priority features to the team. The team asks enough questions that they can turn a high-level user story of the product backlog into the more detailed tasks of the sprint backlog.<br>The Sprint Planning & Meeting Schedule consists of the following data elements:<br>• Identification of the Sprint and associated tasks.<br>• The duration (in days)<br>• The start and end dates<br>• The task description<br>• The priority level of each task<br>• The story points associated with that task<br>• The total number of story points for that sprint. |

## Form Image



YOUR COMPANY LOGO

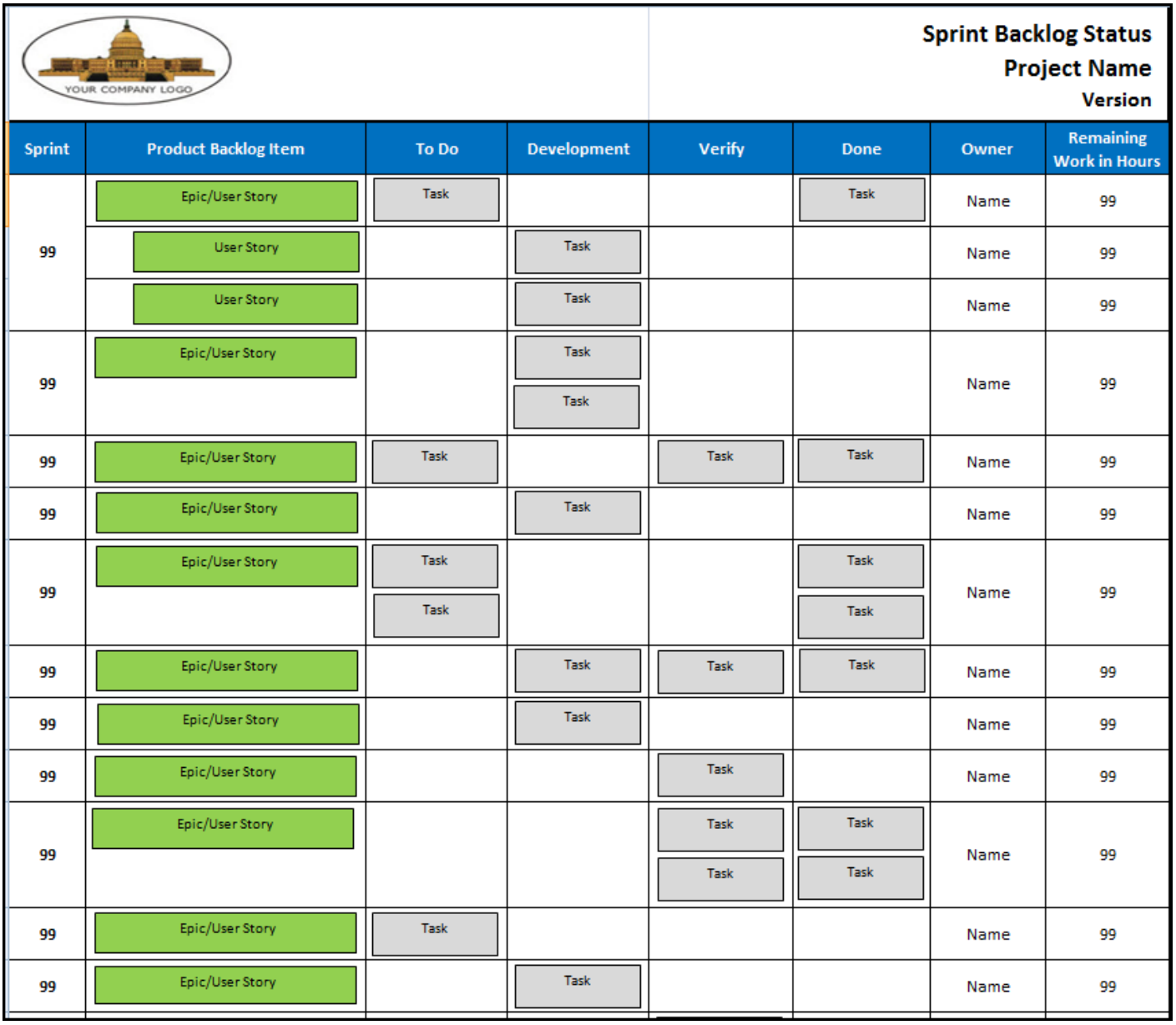
**Sprint Planning & Meeting Schedule**
**Project Name**
**Version**

| Sprint/Task | Duration | Start Date | End Date | Task Description | Priority Level (1=Low, 10=High) | Story Points |
|---|---|---|---|---|---|---|
| Sprint 0 | 99 Days | 99-99-99 | 99-99-99 | | | 999 Total |
| Sprint 0 Planning | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 0 Development | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 0 Review | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 0 Retrospective | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 1 | 99 Days | 99-99-99 | 99-99-99 | | | 999 Total |
| Sprint 1 Planning | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 1 Development | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 1 Review | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 1 Retrospective | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 2 | 99 Days | 99-99-99 | 99-99-99 | | | 999 Total |
| Sprint 2 Planning | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 2 Development | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 2 Review | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 2 Retrospective | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 3 | 99 Days | 99-99-99 | 99-99-99 | | | 999 Total |
| Sprint 3 Planning | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 3 Development | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 3 Review | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 1 Retrospective | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 4 | 99 Days | 99-99-99 | 99-99-99 | | | 999 Total |
| Sprint 4 Planning | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 4 Development | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 4 Review | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 4 Retrospective | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 5 | 99 Days | 99-99-99 | 99-99-99 | | | 999 Total |
| Sprint 5 Planning | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 5 Development | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 5 Review | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |
| Sprint 5 Retrospective | 9 Days | 99-99-99 | 99-99-99 | | ?? | 99 |

## Sprint Backlog Status (Excel)

| | |
|---|---|
| **Purpose:** | Generally, the Sprint Backlog Status as an output of the Sprint Planning & Meeting Schedule to define the work that remains on the project. The sprint backlog contains many things not identified on the Product Backlog including specific tasks that have been decomposed from the accepted user stories.<br><br>Two versions of the Sprint Backlog Status are provided in the package - this document in Excel and an alternative in Visio, as some organizations prefer to work in Excel, while the Visio version facilitates ease of "drag and drop" to move/copy objects within the document. |
| **Description:** | This document can be used to identify only those tasks that are necessary within a single sprint or, if your preference, a complete status of all sprints.<br><br>The following data elements are provided in the Product Data Sheet:<br>• Sprint identification<br>• Product Backlog item (Theme or Epic)<br>• Tasks (To Do, Development, Verify, Done)<br>• Owner<br>• Remaining work in Hours. |

## Form Image

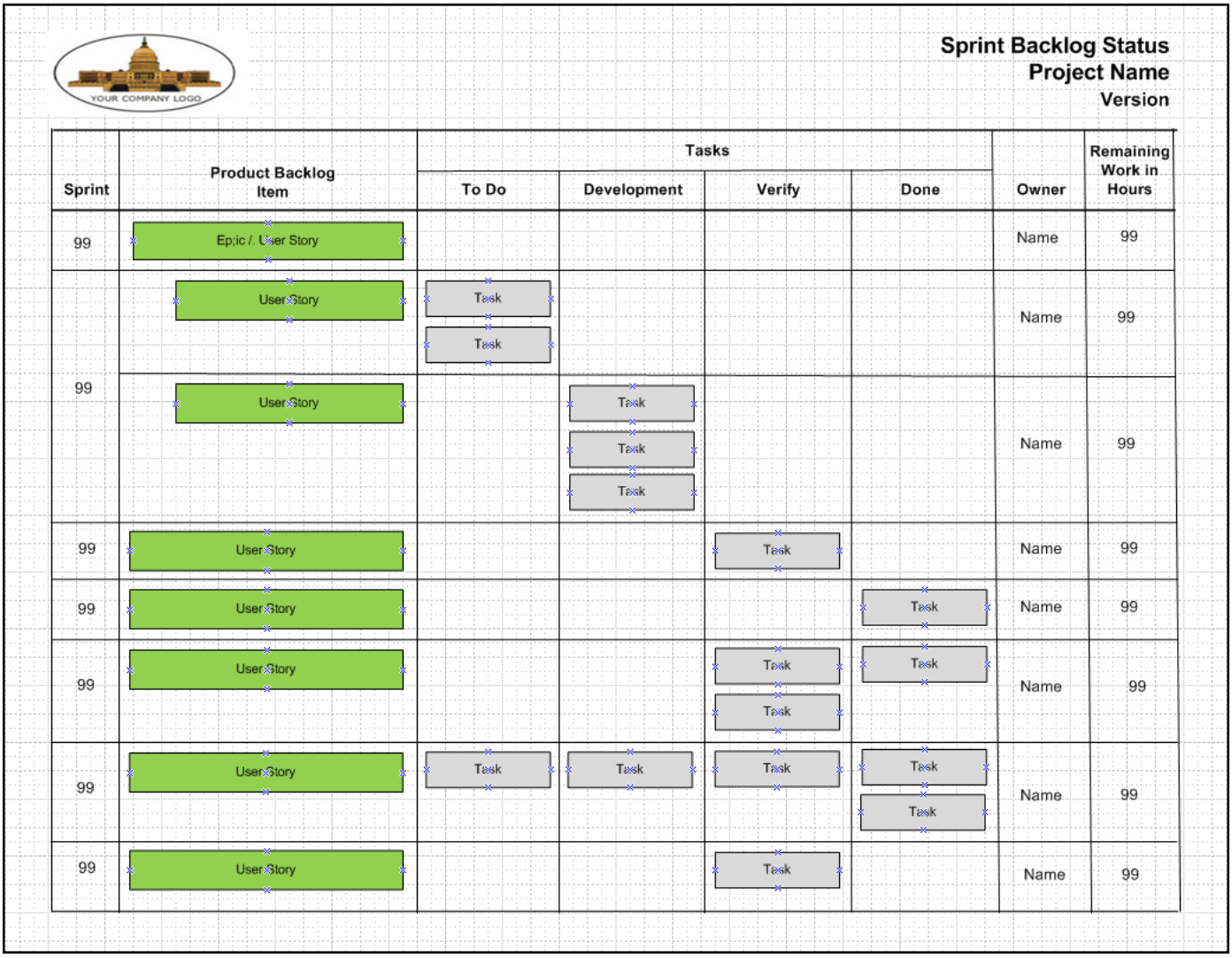



| Sprint | Product Backlog Item | To Do | Development | Verify | Done | Owner | Remaining Work in Hours |
|---|---|---|---|---|---|---|---|
| 99 | Epic/User Story | Task | | | Task | Name | 99 |
| | User Story | | Task | | | Name | 99 |
| | User Story | | Task | | | Name | 99 |
| 99 | Epic/User Story | | Task<br>Task | | | Name | 99 |
| 99 | Epic/User Story | Task | | Task | Task | Name | 99 |
| 99 | Epic/User Story | | Task | | | Name | 99 |
| 99 | Epic/User Story | Task<br>Task | | | Task<br>Task | Name | 99 |
| 99 | Epic/User Story | | Task | Task | Task | Name | 99 |
| 99 | Epic/User Story | | Task | | | Name | 99 |
| 99 | Epic/User Story | | | Task | | Name | 99 |
| 99 | Epic/User Story | | | Task<br>Task | Task<br>Task | Name | 99 |
| 99 | Epic/User Story | Task | | | | Name | 99 |
| 99 | Epic/User Story | | Task | | | Name | 99 |

## Sprint Backlog Status (Visio)

| | |
|---|---|
| **Purpose:** | The Sprint Backlog Status as an output of the Sprint Planning & Meeting Schedule to define the work that remains on the project. The sprint backlog contains many things not identified on the Product Backlog including specific tasks that have been decomposed from the accepted user stories.<br><br>Two versions of the Sprint Backlog Status are provided in the package – either the previous version in Excel or this version in Visio, as this Visio version facilitates ease of "drag and drop" to move/copy objects within the document. |
| **Description:** | This document can be used to identify only those tasks that are necessary within a single sprint or, if your preference, a complete status of all sprints.<br><br>The following data elements are provided in the Sprint Backlog Status:<br>• Sprint identification<br>• Product Backlog item (Theme or Epic)<br>• Tasks (To Do, Development, Verify, Done)<br>• Owner<br>• Remaining work in Hours. |

## Form Image


YOUR COMPANY LOGO
Sprint Backlog Status
Project Name
Version
Sprint
Product Backlog Item
Tasks
To Do
Development
Verify
Done
Owner
Remaining Work in Hours
99
Epic /. User Story
Name
99
User Story
Task
Task
Name
99
99
User Story
Task
Task
Task
Name
99
99
User Story
Task
Name
99
99
User Story
Task
Name
99
99
User Story
Task
Task
Task
Name
99
99
User Story
Task
Task
Task
Task
Task
Name
99
99
User Story
Task
Name
99

## 14-Day Sprint Backlog – Capacity

**Purpose:**

The 14-day Sprint Backlog contains the output of a sprint planning meeting. It contains a list of tasks that the Scrum team needs to complete during the sprint in order to convert a selected set of product backlog items into deliverable functionality within a period of 14 days.

The 14-day Sprint Backlog contains identification of the various sprints, and capacity for an elapsed period of 14 days.

**Description:**

The spreadsheet consists of three sheets: 1) a 14-day sprint backlog table, 2) a capacity sheet, and 3) an instruction sheet providing instructions for updating the tables.

**Formulas:** Formulas are embedded in the chart to calculate:

- Remaining Time for Work Item Each Day = Max (all previous Spent entries).
- Day Totals – total of Spent and Remain columns.

**Sprint Backlog Sheet**

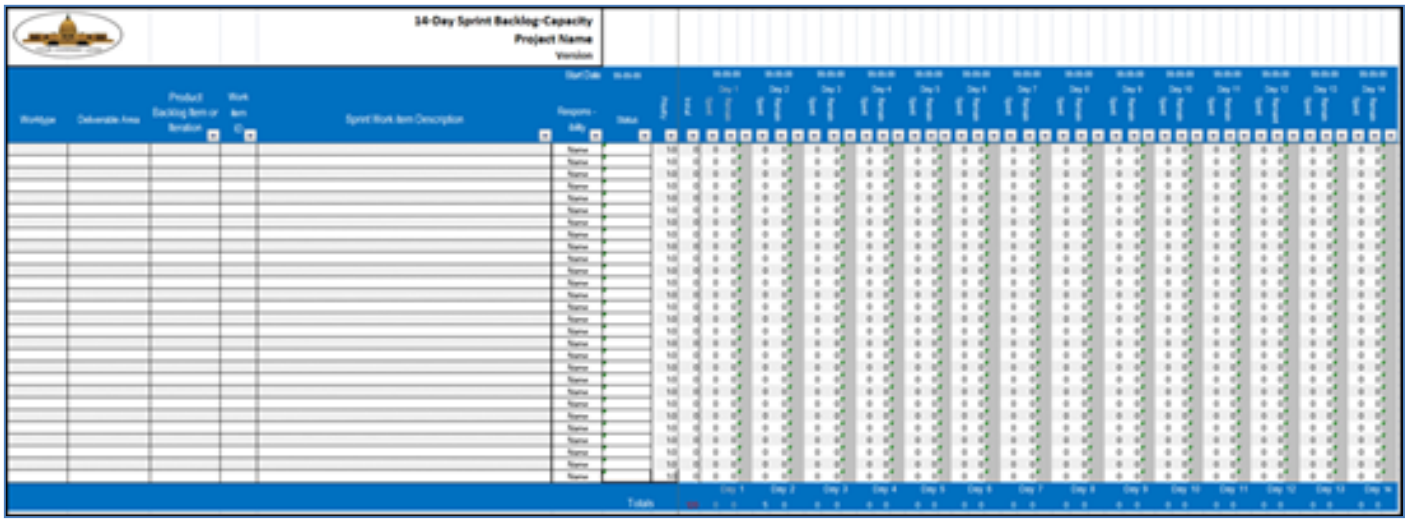
| | |
|---|---|
| **Worktype** | Suggested entries include "Feature," "Tax," "Precondition," an "Spike." |
| **Deliverable Area** | Enter the Deliverable Area. |
| **Product Backlog Item or Iteration** | Enter this description. |
| **Work Item ID** | Enter the appropriate ID. |
| **Sprint Work Item Description** | Enter a description of the work to be accomplished. |
| **Responsibility** | Enter who will be responsible for the work. |
| **Status** | Suggested entries include "Complete," :In Progress," "Pending "Postponed," or "cancelled." |
| **Priority** | Determine a scale for the Priority. |
| **Initial (Estimate)** | Enter the initial estimate for the work item. |
| **Spent** | As each day passes, enter the time spent during the day worki on that item. |
| **Remain** | This amount is automatically calculated by the embedded form and is equal to the Initial value minus the time Spent each day |
| **Day Totals** (Bottom of Spreadsheet) | At the bottom of the spreadsheet, totals for each day will displa |

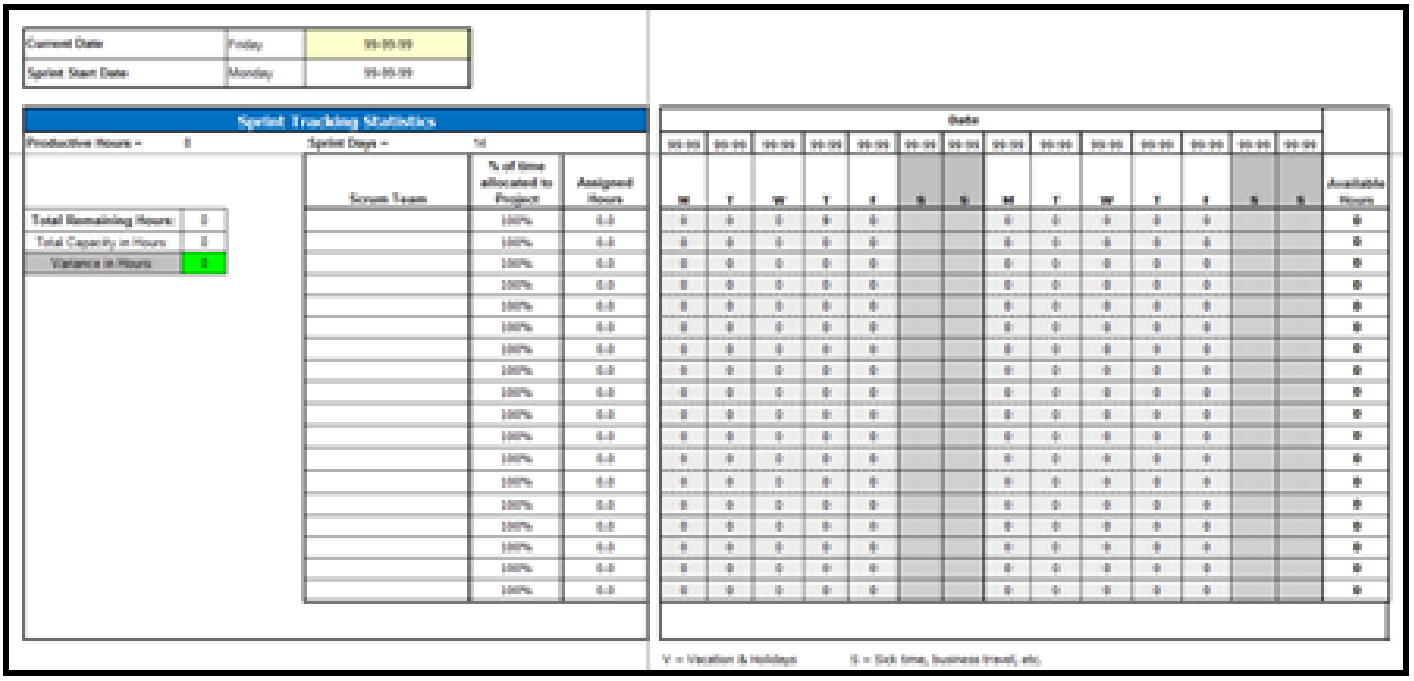
**Capacity Sheet**

The initial tasks you need to complete to prepare this sheet for a project meeting are as follows:

| | |
|---|---|
| **Current Date** | Enter the day of the week. |
| **Sprint Start Date** | Enter the date in a 99-99-99 format. |
| **Scrum Team** | Enter the names of the individuals on the Scrum Team. |
| **% of Time Allocated to Project** | Enter the % of time each individual can devote to the team |
| **Assigned Hours** | Enter the total assigned hours for each individual. |
| **Date** | Enter the date for each of the days. |
| **Hours** | Enter the amount of hours worked each day. |

## Form Image


14-Day Sprint Backlog-Capacity
Project Name
Version


Current Date
Friday
Sprint Start Date
Monday
Sprint Tracking Statistics
Productive Hours =
Sprint Days =
Total Remaining Hours
Total Capacity in Hours
Variance in Hours
Scrum Team
% of time allocated to Project
Assigned Hours
Date
Availability Hours
V = Vacation & Holidays
S = Sick time, business travel, etc.

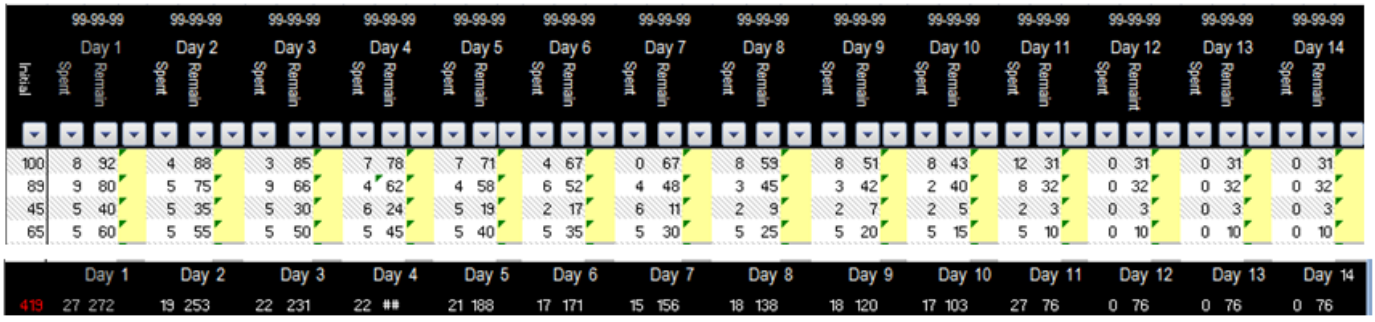
## Instructions for Entering Data in Sprint Backlog & Capacity Sheets

### 1. Sprint Backlog Sheet

The initial tasks you need to complete to prepare this sheet for a project meeting are as follows:

| | |
|---|---|
| **Worktype** | Enter a Worktype. Suggested entries include "Feature," "Tax," "Precondition," and "Spike." |
| **Deliverable Area** | Enter the Deliverable Area. |
| **Product Backlog Item or Iteration** | Enter this description. |
| **Work Item ID** | Enter the appropriate ID. |
| **Sprint Work Item Description** | Enter a description of the work to be accomplished. |
| **Responsibility** | Enter who will be responsible for the work. |
| **Status** | Suggested entries include "Complete," :In Progress," "Pending," "Postponed," or "cancelled." |
| **Priority** | Determine a scale for the Priority. |
| **Initial (Estimate)** | Enter the initial estimate for the work item. |
| **Spent** | As each day passes, enter the time spent during the day working on that item. |
| **Remain** | This amount is automatically calculated by the embedded formulas and is equal to the Initial value minus the time Spent each day. |
| **Day Totals** (Bottom of Spreadsheet) | At the bottom of the spreadsheet, totals for each day will display. |

An example of typical entries for each day are shown below reflecting the Initial Value and how the Remain values reduce each day as Spent are removed from the total Initial value. At the bottom of the spreadsheet are the totals.



| Initial | 99-99-99 Day 1 Spent | Remain | 99-99-99 Day 2 Spent | Remain | 99-99-99 Day 3 Spent | Remain | 99-99-99 Day 4 Spent | Remain | 99-99-99 Day 5 Spent | Remain | 99-99-99 Day 6 Spent | Remain | 99-99-99 Day 7 Spent | Remain | 99-99-99 Day 8 Spent | Remain | 99-99-99 Day 9 Spent | Remain | 99-99-99 Day 10 Spent | Remain | 99-99-99 Day 11 Spent | Remain | 99-99-99 Day 12 Spent | Remain | 99-99-99 Day 13 Spent | Remain | 99-99-99 Day 14 Spent | Remain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 100 | 8 | 92 | 4 | 88 | 3 | 85 | 7 | 78 | 7 | 71 | 4 | 67 | 0 | 67 | 8 | 59 | 8 | 51 | 8 | 43 | 12 | 31 | 0 | 31 | 0 | 31 | 0 | 31 |
| 89 | 9 | 80 | 5 | 75 | 9 | 66 | 4 | 62 | 4 | 58 | 6 | 52 | 4 | 48 | 3 | 45 | 3 | 42 | 2 | 40 | 8 | 32 | 0 | 32 | 0 | 32 | 0 | 32 |
| 45 | 5 | 40 | 5 | 35 | 5 | 30 | 6 | 24 | 5 | 19 | 2 | 17 | 6 | 11 | 2 | 9 | 2 | 7 | 2 | 5 | 2 | 3 | 0 | 3 | 0 | 3 | 0 | 3 |
| 65 | 5 | 60 | 5 | 55 | 5 | 50 | 5 | 45 | 5 | 40 | 5 | 35 | 5 | 30 | 5 | 25 | 5 | 20 | 5 | 15 | 5 | 10 | 0 | 10 | 0 | 10 | 0 | 10 |

| | Day 1 | | Day 2 | | Day 3 | | Day 4 | | Day 5 | | Day 6 | | Day 7 | | Day 8 | | Day 9 | | Day 10 | | Day 11 | | Day 12 | | Day 13 | | Day 14 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 419 | 27 | 272 | 19 | 253 | 22 | 231 | 22 | ## | 21 | 188 | 17 | 171 | 15 | 156 | 18 | 138 | 18 | 120 | 17 | 103 | 27 | 76 | 0 | 76 | 0 | 76 | 0 | 76 |

## 30-Day Sprint Backlog – Capacity

**Purpose:**

The 30-day Sprint Backlog contains the output of a sprint planning meeting. It contains a list of tasks that the Scrum team needs to complete during the sprint in order to convert a selected set of product backlog items into deliverable functionality within a period of 30 days.

The 30-day Sprint Backlog contains identification of the various sprints, and capacity for an elapsed period of 30 days.

**Description:**

The spreadsheet consists of three sheets: 1) a 30-day sprint backlog table, 2) a capacity sheet, and 3) an instruction sheet providing instructions for updating the tables.

**Formulas:** Formulas are embedded in the chart to calculate:

Remaining Time for Work Item Each Day = Max (all previous Spent entries).

Day Totals – total of Spent and Remain columns.

**Sprint Backlog Sheet**

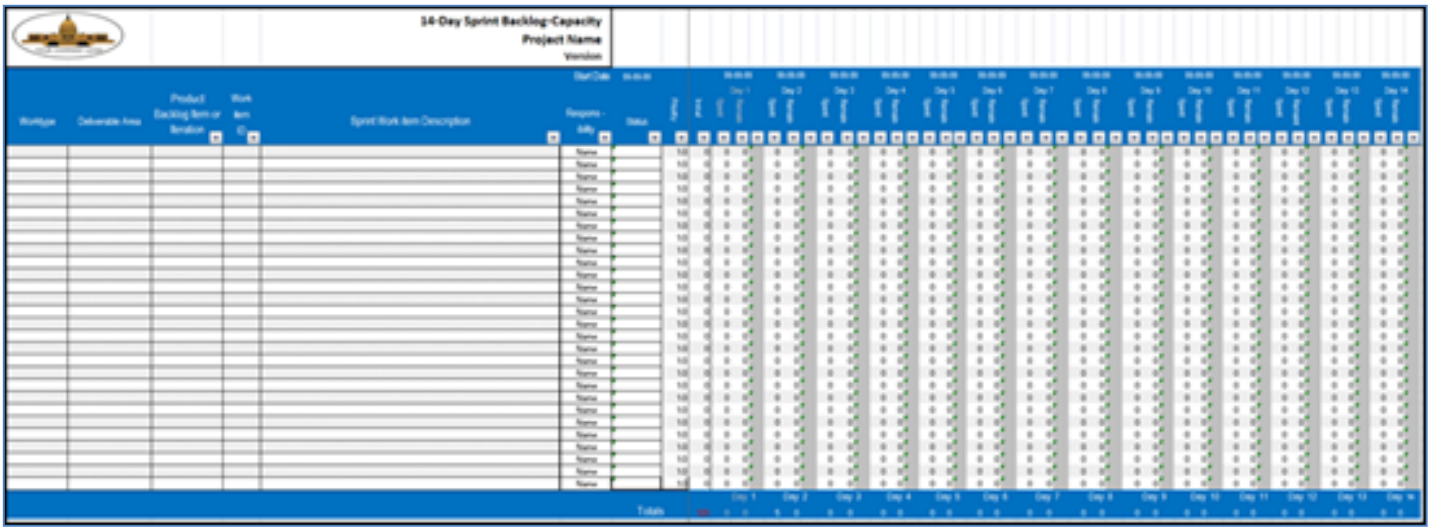
| | |
|---|---|
| **Worktype** | Suggested entries include "Feature," "Tax," "Precondition," and "Spike." |
| **Deliverable Area** | Enter the Deliverable Area. |
| **Product Backlog Item or Iteration** | Enter this description. |
| **Work Item ID** | Enter the appropriate ID. |
| **Sprint Work Item Description** | Enter a description of the work to be accomplished. |
| **Responsibility** | Enter who will be responsible for the work. |
| **Status** | Suggested entries include "Complete," :In Progress," "Pending," "Postponed," or "Cancelled." |
| **Priority** | Determine a scale for the Priority. |
| **Initial (Estimate)** | Enter the initial estimate for the work item. |
| **Spent** | As each day passes, enter the time spent during the day working on that item. |
| **Remain** | This amount is automatically calculated by the embedded formula and is equal to the Initial value minus the time Spent each day. |
| **Day Totals** (Bottom of Spreadsheet) | At the bottom of the spreadsheet, totals for each day will display. |
| | |

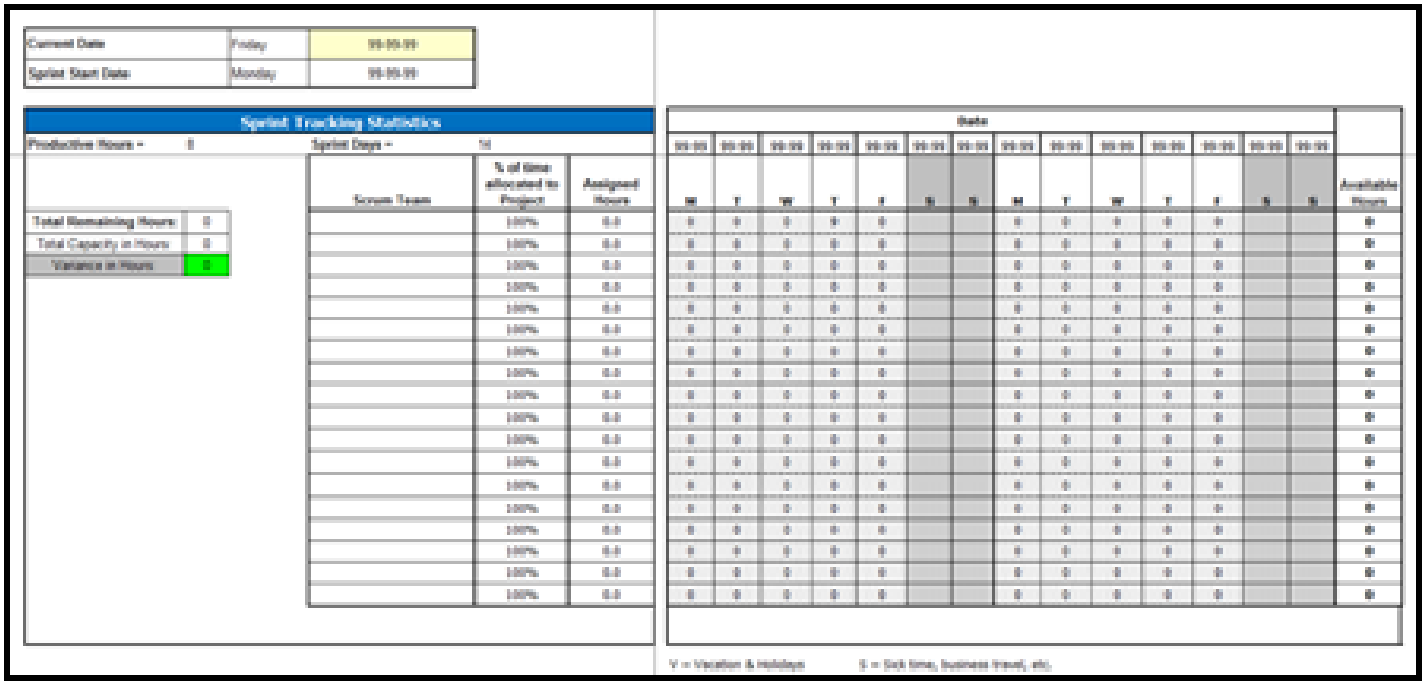
**Capacity Sheet**

The initial tasks you need to complete to prepare this sheet for a project meeting are as follows:

| | |
|---|---|
| **Current Date** | Enter the day of the week. |
| **Sprint Start Date** | Enter the date in a 99-99-99 format. |
| **Scrum Team** | Enter the names of the individuals on the Scrum Team. |
| **% of Time Allocated to Project** | Enter the % of time each individual can devote to the team. |
| **Assigned Hours** | Enter the total assigned hours for each individual. |
| **Date** | Enter the date for each of the days. |
| **Hours** | Enter the amount of hours worked each day. |

## Form Image


14-Day Sprint Backlog:Capacity
Project Name
Version


Current Date
Friday
Sprint Start Date:
Monday
Sprint Tracking Statistics
Productive Hours =
Sprint Days =
Scrum Team
% of time allocated to Project
Assigned Hours
Total Remaining Hours:
Total Capacity in Hours
Variance in Hours
Date
Available Hours
M
T
W
T
F
S
S
V = Vacation & Holidays
S = Sick time, business travel, etc.


## Instructions for Entering Data in Sprint Backlog & Capacity Sheets

### 1. Sprint Backlog Sheet

The initial tasks you need to complete to prepare this sheet for a project meeting are as follows:

| | |
|---|---|
| **Worktype** | Enter a Worktype. Suggested entries include "Feature," "Tax," "Precondition," and "Spike." |
| **Deliverable Area** | Enter the Deliverable Area. |
| **Product Backlog Item or Iteration** | Enter this description. |
| **Work Item ID** | Enter the appropriate ID. |
| **Sprint Work Item Description** | Enter a description of the work to be accomplished. |
| **Responsibility** | Enter who will be responsible for the work. |
| **Status** | Suggested entries include "Complete," :In Progress," "Pending," "Postponed," or "cancelled." |
| **Priority** | Determine a scale for the Priority. |
| **Initial (Estimate)** | Enter the initial estimate for the work item. |
| **Spent** | As each day passes, enter the time spent during the day working on that item. |
| **Remain** | This amount is automatically calculated by the embedded formulas and is equal to the Initial value minus the time Spent each day. |
| **Day Totals** (Bottom of Spreadsheet) | At the bottom of the spreadsheet, totals for each day will display. |

An example of typical entries for each day are shown below reflecting the Initial Value and how the Remain values reduce each day as Spent are removed from the total Initial value. At the bottom of the spreadsheet are the totals.

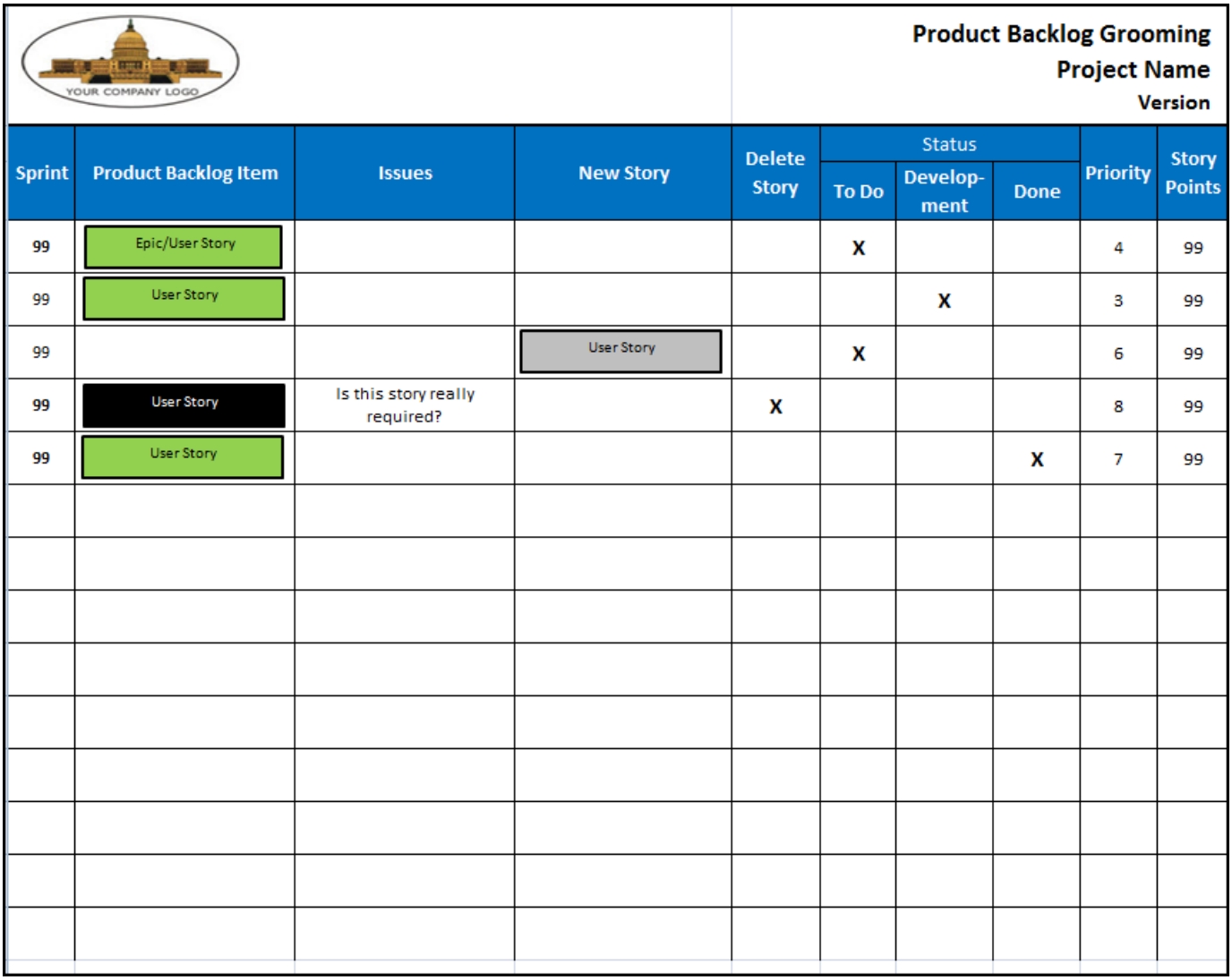
| | 99-99-99 Day 1 | | 99-99-99 Day 2 | | 99-99-99 Day 3 | | 99-99-99 Day 4 | | 99-99-99 Day 5 | | 99-99-99 Day 6 | | 99-99-99 Day 7 | | 99-99-99 Day 8 | | 99-99-99 Day 9 | | 99-99-99 Day 10 | | 99-99-99 Day 11 | | 99-99-99 Day 12 | | 99-99-99 Day 13 | | 99-99-99 Day 14 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Initial | Spent | Remain | Spent | Remain | Spent | Remain | Spent | Remain | Spent | Remain | Spent | Remain | Spent | Remain | Spent | Remain | Spent | Remain | Spent | Remain | Spent | Remain | Spent | Remain | Spent | Remain | Spent | Remain |
| 100 | 8 | 92 | 4 | 88 | 3 | 85 | 7 | 78 | 7 | 71 | 4 | 67 | 0 | 67 | 8 | 59 | 8 | 51 | 8 | 43 | 12 | 31 | 0 | 31 | 0 | 31 | 0 | 31 |
| 89 | 9 | 80 | 5 | 75 | 9 | 66 | 4 | 62 | 4 | 58 | 6 | 52 | 4 | 48 | 3 | 45 | 3 | 42 | 2 | 40 | 8 | 32 | 0 | 32 | 0 | 32 | 0 | 32 |
| 45 | 5 | 40 | 5 | 35 | 5 | 30 | 6 | 24 | 5 | 19 | 2 | 17 | 6 | 11 | 2 | 9 | 2 | 7 | 2 | 5 | 2 | 3 | 0 | 3 | 0 | 3 | 0 | 3 |
| 65 | 5 | 60 | 5 | 55 | 5 | 50 | 5 | 45 | 5 | 40 | 5 | 35 | 5 | 30 | 5 | 25 | 5 | 20 | 5 | 15 | 5 | 10 | 0 | 10 | 0 | 10 | 0 | 10 |

| | Day 1 | | Day 2 | | Day 3 | | Day 4 | | Day 5 | | Day 6 | | Day 7 | | Day 8 | | Day 9 | | Day 10 | | Day 11 | | Day 12 | | Day 13 | | Day 14 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 419 | 27 | 272 | 19 | 253 | 22 | 231 | 22 | ## | 21 | 188 | 17 | 171 | 15 | 156 | 18 | 138 | 18 | 120 | 17 | 103 | 27 | 76 | 0 | 76 | 0 | 76 | 0 | 76 |

## Product Backlog Grooming

| | |
|---|---|
| **Purpose:** | Product Backlog Grooming is a meeting that is held near the end of one sprint to ensure the backlog is ready for the next sprint.<br><br>The intent of a "grooming" meeting is to ensure that the backlog remains populated with items that are relevant, detailed and estimated to a degree appropriate with their priority, and in keeping with current understanding of the project or product and its objectives. |
| **Description:** | During a product backlog refinement meeting, the team and product owner discuss the top items on the product backlog and record decisions in the Product Backlog Grooming artifact.<br><br>Backlog refinement is really a checkpoint rather than an effort to fully resolve issues.<br><br>From this meeting, the following decisions can be made:<br><br>• Removing user stories that no longer appear relevant<br>• Creating new user stories in response to newly discovered needs<br>• Re-assessing the relative priority of stories<br>• Assigning estimates to stories which have yet to receive one<br>• Correcting estimates in light of newly discovered information<br>• Splitting user stories which are high priority but too coarse grained to fit in an upcoming iteration.<br><br>The Product Backlog Grooming artifact contains the following data elements:<br><br>• Sprint number<br>• Product Backlog Item description<br>• What are the issues confronting this item?<br>• Identify desired new stories<br>• Delete stories that are no longer required<br>• Provide a status of the user story (To Do, Development, Done)<br>• Identify the Priority<br>• Reallocate Story Points. |

## Form Image

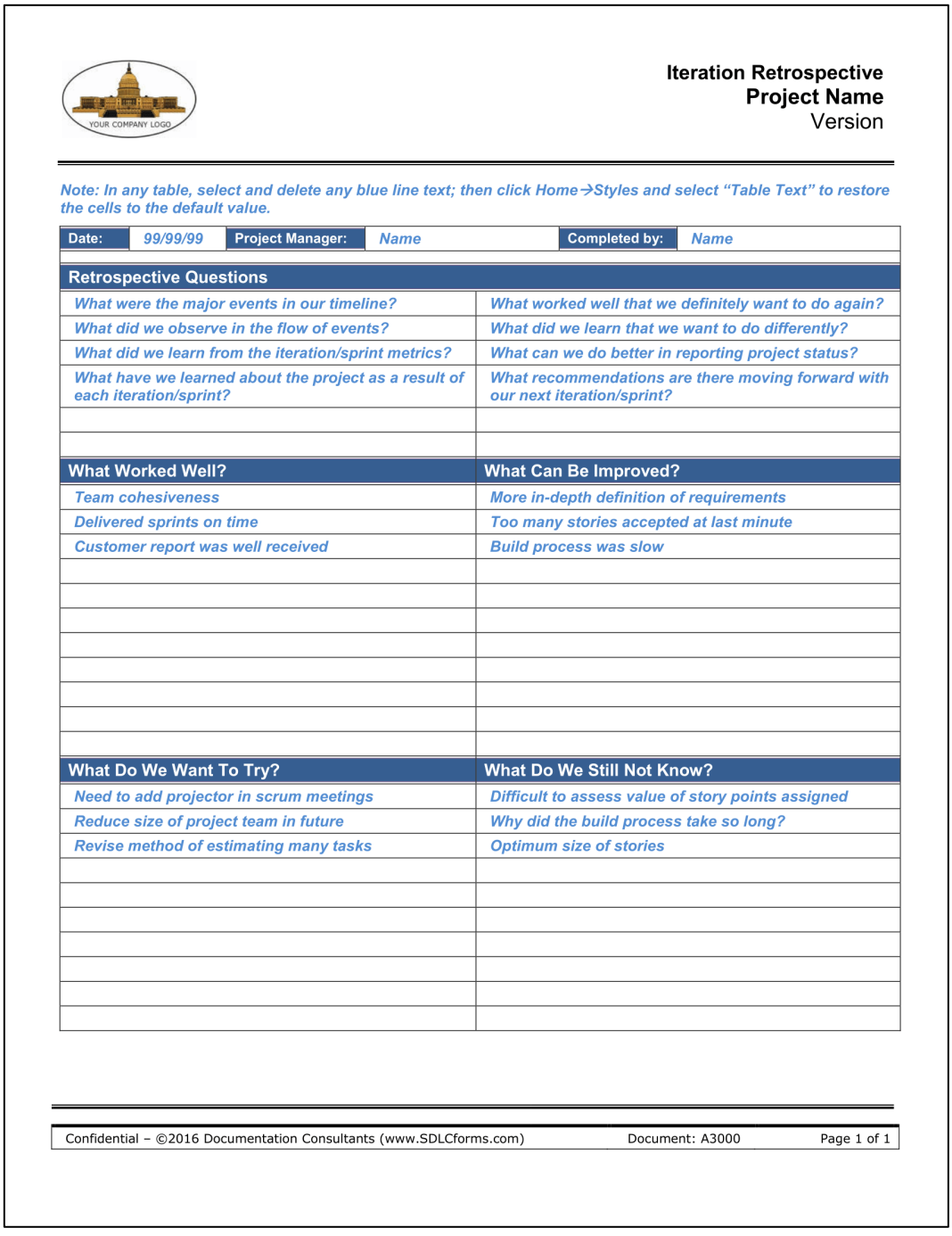



| Sprint | Product Backlog Item | Issues | New Story | Delete Story | Status | | | Priority | Story Points |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | To Do | Develop-ment | Done | | |
| 99 | Epic/User Story | | | | X | | | 4 | 99 |
| 99 | User Story | | | | | X | | 3 | 99 |
| 99 | | | User Story | | X | | | 6 | 99 |
| 99 | User Story | Is this story really required? | | X | | | | 8 | 99 |
| 99 | User Story | | | | | | X | 7 | 99 |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |

## Iteration Retrospective

| | |
|---|---|
| **Purpose:** | An Agile retrospective is a meeting that's held at the end of an iteration. The process of retrospecting is at the heart of Scrum. During the retrospective review, the team reflects on what transpired during the iteration and identifies actions for improvement going forward. |
| **Description:** | The Agile retrospective can be thought of as a "Lessons Learned" meeting. The team reflects on how everything went and then decides what changes they want to make in the next iteration. The retrospective is accomplished by team collaboration, and team members should decide together how the meetings will be run and how decisions will be made about improvements.<br><br>During the retrospective, team members are asked the following questions:<br>• What worked well and we definitely want to do again?<br>• What did we learn from the iteration/sprint metrics?<br>• What did we learn that we want to do differently?<br>• Did we learn what the optimum story points are for an iteration?<br>• What did we learn in the flow of events?<br>• What can we do better in reporting project status?<br>• What did we NOT investigate or talk about that likely could stand improvement?<br><br>The Iteration Retrospective contains the following data elements:<br>• Date of the retrospective<br>• Project manager's name.<br>• Who completed the retrospective?<br>• What questions will be asked in the retrospective meeting?<br>• What worked well?<br>• What can be improved?<br>• What do we want to try?<br>• What do we still not know? |

## Form Image



YOUR COMPANY LOGO

**Iteration Retrospective**
**Project Name**
Version

*Note: In any table, select and delete any blue line text; then click Home→Styles and select "Table Text" to restore the cells to the default value.*

| Date: | 99/99/99 | Project Manager: | Name | Completed by: | Name |
|---|---|---|---|---|---|

| Retrospective Questions | |
|---|---|
| What were the major events in our timeline? | What worked well that we definitely want to do again? |
| What did we observe in the flow of events? | What did we learn that we want to do differently? |
| What did we learn from the iteration/sprint metrics? | What can we do better in reporting project status? |
| What have we learned about the project as a result of each iteration/sprint? | What recommendations are there moving forward with our next iteration/sprint? |
| | |
| | |

| What Worked Well? | What Can Be Improved? |
|---|---|
| Team cohesiveness | More in-depth definition of requirements |
| Delivered sprints on time | Too many stories accepted at last minute |
| Customer report was well received | Build process was slow |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

| What Do We Want To Try? | What Do We Still Not Know? |
|---|---|
| Need to add projector in scrum meetings | Difficult to assess value of story points assigned |
| Reduce size of project team in future | Why did the build process take so long? |
| Revise method of estimating many tasks | Optimum size of stories |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

 Document: A3000 Page 1 of 1

## 4.5 Explore Phase

### 4.5.1 Explore Phase Overview

We now draw our attention to the exploratory phase, which is founded on action. What is known as a Complex Adaptive System (CAS) is a collection of agents who explore to achieve a goal by interacting with each other according to a set of rules. A CAS experiments with various paths, selects and executes viable ones, compares the results against its goals and adapts as necessary. In a more project-specific sense, the project-manager's goal is to:

- Help the team articulate and understand the goal and constraints
- Help the team interact efficiently
- Facilitate an effective decision-making process
- Be prepared for the inevitable eventuality of the project going off track.

The project manager as the team builder contributes in six ways to project success:

- Focus the team on delivering results
- Molding a group of individuals into a team
- Developing their capabilities
- Providing team with required resources
- Coaching the customers
- Orchestrating the team's rhythm.

Thus, the purpose of the project leader is to create highly efficient exploratory teams. Exploration and experimentation are the foundations of new product development that involve the risk of making mistakes, failing, and then learning from this failure. Project managers should respond by making risk-taking a lot less risky. This can be achieved by adhering to the three explore phase practices illustrated in the diagram below.

Project community on the other hand is geared towards project towards project managers. The objective of coaching and team development is to unleash the capability of the team by helping team members continuously improve their domain knowledge, self-discipline and team skills.

Other practices include daily team integrating meetings to coordinate team members' activities on a daily basis. Participatory decision-making to provide the project community with specific practices to make various analytical decisions and finally, daily interaction with the customer team including product manager to ensure the product stays on track to meet customer requirements.

Thus, all practices work in synergy creating an effective strategy to explore via various iterative processes.

### 4.5.2 Explore Phase Forms & Templates

## Story Completion Checklist

| | |
|---|---|
| **Purpose:** | The Story Completion Checklist contains the story name, activity, work products and status. As Agile stories are finished, the product owner will work with their Agile team and the team's scrum master to make sure that finished stories are truly complete.<br><br>A definition of "Done" is a clear and concise list of requirements that software must adhere to for the team to call it complete. |
| **Description:** | In order to status a user story as "Done," it must meet the following criteria:<br><br>• Is the story fully tested?<br>• Are necessary automated tests created?<br>• Are all tests checked into the appropriate test case management system?<br>• Have the customer requirements been fully met?<br>• Have customers/stakeholders viewed and accepted the tasks?<br>• Are changes fully documented in internal specs, etc.?<br>• Are changes fully reflected in customer-facing documentation (i.e. user manuals, etc.).?<br>• Are all scrum team members happy with the functionality?<br>• Are there any remaining caveats or gotchas in use?<br>• Have negative test been demonstrated as well as positive?<br>• Is all necessary error checking in place?<br>• Have interfacing components/features been tested?<br>• Are all external dependencies met?<br>• Are any necessary legal agreements in place to use the work?<br>• Have workflows been reviewed by usability teams?<br>• Have text strings been reviewed by legal/UI/usability?<br>• Are all necessary text string translations in place?<br>• Are all localization requirements met?<br>• Have any newly discovered work items been recorded in the backlog?<br>• Are any bugs that are not going to be fixed documented in release notes/problem tracking items?<br>• Has the story been marked as complete in agile project management systems?<br>• Have all builds/branches been integrated into the mainline of code?<br>• Is the team sure that no other work is necessary?<br><br>The Story Completion Checklist provides a vehicle to record whether the story has been completed and Inspected and the number of defects found. |

## Form Image


**Story Completion Checklist**
**Project Name**
Version

*Note: In any table, select and delete any blue line text; then click Home→Styles and select "Table Text" to restore the cells to the default value.*

| | | | | | |
|---|---|---|---|---|---|
| Story Name | *Provide customer feedback* | Developers | *Betty Smith, Brad Bards* | | |
| Feature Name | *Develop online form for customer* | Inspected By: | *Joe Jones* | Date: | *99/99/99* |

**Story Completion Checklist**

Legend: C = Completed, I = Inspected, # = Number of defects

| Activity | C | I | # | Activity | C | I | # |
|---|---|---|---|---|---|---|---|
| *Story scope and estimate reconsidered* | ☐ | ☐ | *3* | *Manual tests completed* | ☐ | ☐ | |
| *Development environment created* | ☐ | ☐ | *1* | *Automated tests completed* | ☐ | ☐ | |
| *Requirements drafted* | ☐ | ☐ | *0* | *Test design completed* | ☐ | ☐ | |
| *User interface created* | ☐ | ☐ | *0* | *Code complete* | ☐ | ☐ | |
| *Technical design drafted* | ☐ | ☐ | *0* | *Manual tests executed and passed* | ☐ | ☐ | |
| *User documentation drafted* | ☐ | ☐ | *2* | *Technical design documentation completed* | ☐ | ☐ | |
| *Test design drafted* | ☐ | ☐ | *0* | *User interface documentation completed* | ☐ | ☐ | |
| | ☐ | ☐ | | *User documentation completed* | ☐ | ☐ | |
| *Requirements completed* | ☐ | ☐ | *2* | *Story integrated (& integration tests passed)* | ☐ | ☐ | |
| *Existing manual tests defined* | ☐ | ☐ | *0* | | ☐ | ☐ | |
| *Tests drafted* | ☐ | ☐ | *1* | | ☐ | ☐ | |
| *Code drafted* | ☐ | ☐ | *0* | | ☐ | ☐ | |
| | ☐ | ☐ | | | ☐ | ☐ | |
| | ☐ | ☐ | | | ☐ | ☐ | |
| | ☐ | ☐ | | | ☐ | ☐ | |
| | ☐ | ☐ | | | ☐ | ☐ | |
| | ☐ | ☐ | | | ☐ | ☐ | |
| | ☐ | ☐ | | | ☐ | ☐ | |
| | ☐ | ☐ | | | ☐ | ☐ | |
| | ☐ | ☐ | | | ☐ | ☐ | |
| | ☐ | ☐ | | | ☐ | ☐ | |
| | ☐ | ☐ | | | ☐ | ☐ | |
| | ☐ | ☐ | | | ☐ | ☐ | |
| | ☐ | ☐ | | | ☐ | ☐ | |
| | ☐ | ☐ | | | ☐ | ☐ | |
| | ☐ | ☐ | | | ☐ | ☐ | |
| | ☐ | ☐ | | | ☐ | ☐ | |
| | ☐ | ☐ | | | ☐ | ☐ | |

## 4.6 Adapt Phase

### 4.6.1 Adapt Phase Overview

Control and correction are common terms applied to this lifecycle phase. Plans are made, results are monitored, and corrections are made—implying that the plans were right and the actual results, if different from the plan, are wrong. “Adapt” implies modification or change rather than success or failure. In projects guided by the philosophy that responding to change is more important than following a plan, attributing failure to variation from the plan isn’t productive.

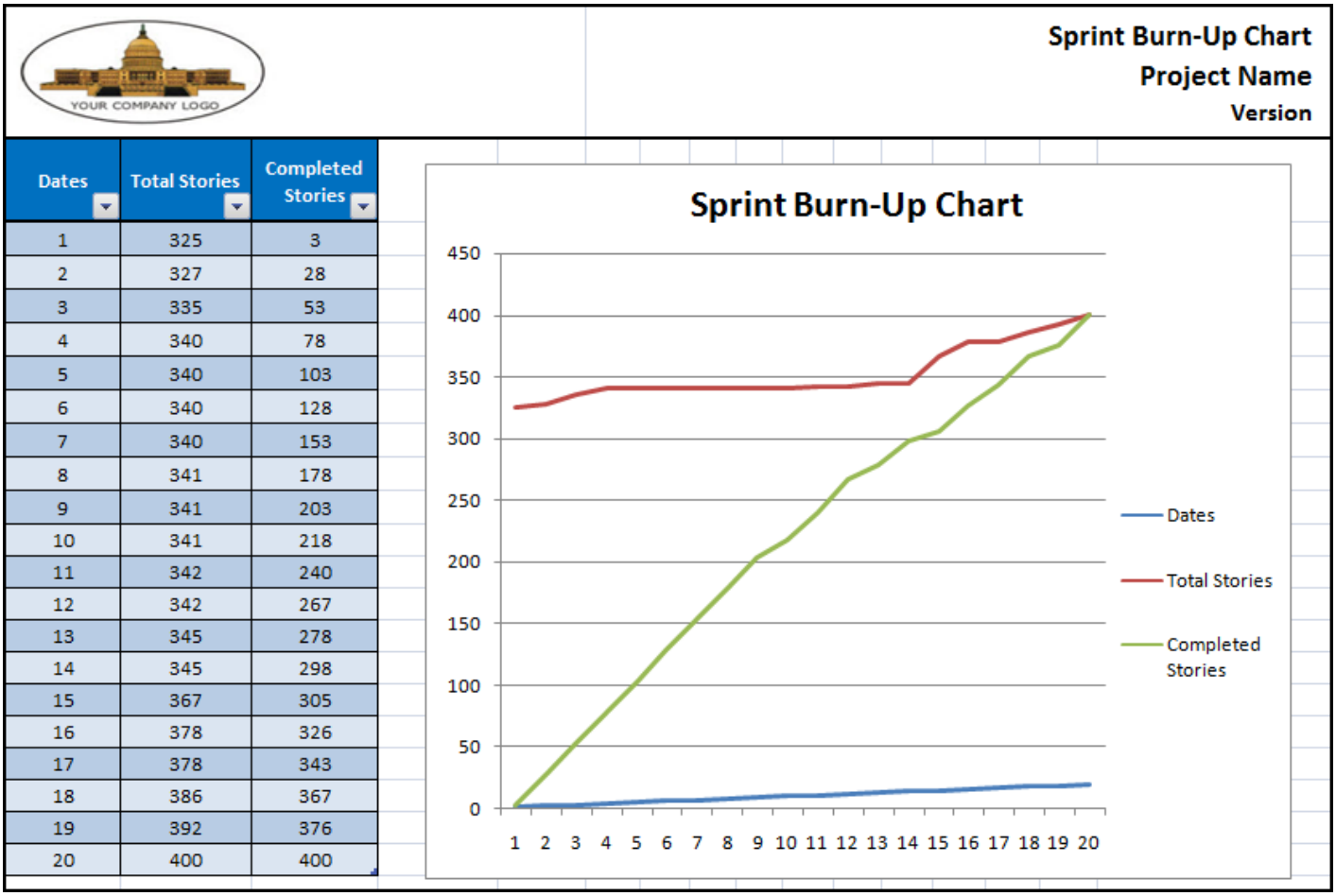
A purely ad hoc process fails to learn from its mistakes, whereas the incorporation and retention of lessons learned are key pieces of APM. After the Envision phase, the loop will generally be Speculate-Explore-Adapt, with each iteration successively refining the product. However, periodically revisiting the Envision phase may be necessary as the team gathers new information.

### 4.6.2 Adapt Phase Forms & Templates

## Sprint Burn-Up Chart

| | |
|---|---|
| **Purpose:** | A Sprint Burn-Up Chart is a graph that shows the progress of work toward a *goal* line associated with a value on the vertical axis.<br>In the simplest form of burn up chart there are two lines on the chart:<br>3. A total work line (the project scope line)<br>4. A work completed line<br>As work is completed over *time* (the horizontal axis), the progress line moves up (burns up) to be nearer to the desired completion line. We can show projected outcomes on burn-up charts by calculating a trend line to see when work might be completed.<br>These charts can allow you to instantly identify certain types of problems, such as scope creep or a deviation from the planned project path. These problems can then be discussed and corrective action can be taken at an early stage, rather than when it is too late, |
| **Description:** | The vertical axis is amount of work, and is measured in units customized to your own project. Some common units are number of tasks, estimated hours or story points (in agile project management methodologies). The horizontal axis is time, usually measured in days.<br>The Sprint Burn-Up Chart contains entries which permit the generation of the two lines:<br>• Dates<br>• Total Stories to be completed<br>• Number of stories completed.<br>**Data Grid**: The data grid is constructed as a table, which means whenever you update the table by adding or deleting rows or columns or simply changing the values in the table, the graph will be automatically updated. |

**Form Image**




| Dates | Total Stories | Completed Stories |
|---|---|---|
| 1 | 325 | 3 |
| 2 | 327 | 28 |
| 3 | 335 | 53 |
| 4 | 340 | 78 |
| 5 | 340 | 103 |
| 6 | 340 | 128 |
| 7 | 340 | 153 |
| 8 | 341 | 178 |
| 9 | 341 | 203 |
| 10 | 341 | 218 |
| 11 | 342 | 240 |
| 12 | 342 | 267 |
| 13 | 345 | 278 |
| 14 | 345 | 298 |
| 15 | 367 | 305 |
| 16 | 378 | 326 |
| 17 | 378 | 343 |
| 18 | 386 | 367 |
| 19 | 392 | 376 |
| 20 | 400 | 400 |

## Sprint Burndown Chart

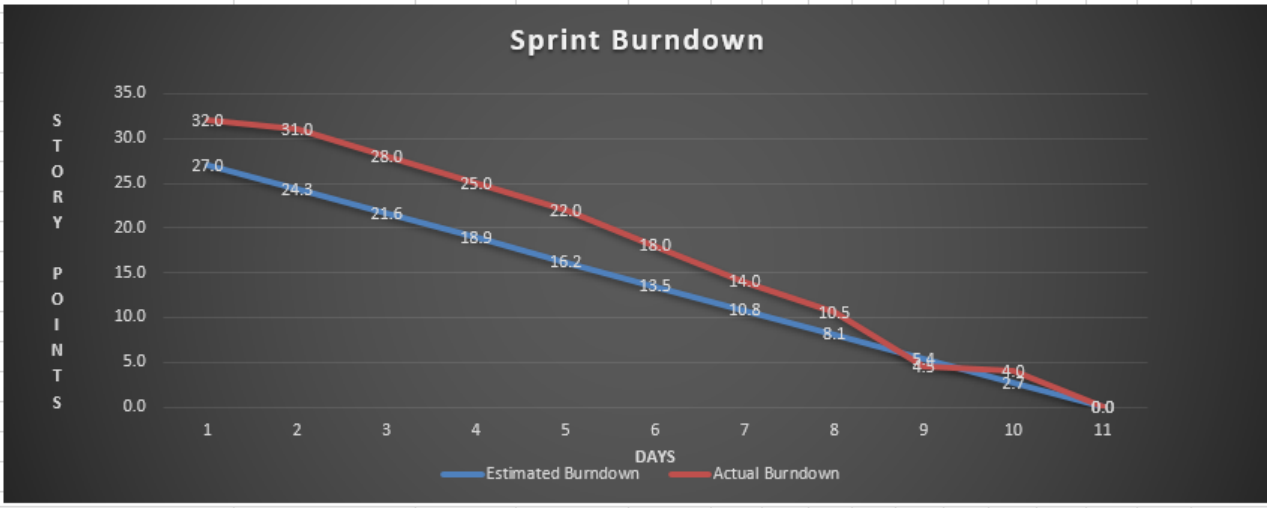
| | |
|---|---|
| **Purpose:** | The Sprint Burndown Chart displays the remaining effort for a given period of time. Work remaining is the Y axis and working days is the X axis. When the x and y axis coincide, the work is complete.<br><br>When tracking product development using the Burndown Chart, teams can use a Sprint Burndown Chart and a Release Burndown Chart. |
| **Description:** | The Sprint Burndown Chart consists of the following data elements:<br>• Product Backlog Item (PBI)<br>• Task<br>• Name<br>• Status (To Do, In-Progress, Closed)<br>• Original Estimate of time<br>• An array of 10 days by which work effort per day can reduce the time remaining to zero.<br>• Sprint Review.<br><br>**Data Grid**: The data grid is constructed as a table, which means whenever you update the table by adding or deleting rows or columns or simply changing values in the table, the graph is automatically updated.<br><br>**Formulas:** Formulas are embedded in the chart to calculate:<br>• Time remaining each day after each previous work effort. |

## Form Image

YOUR COMPANY LOGO

**Sprint Burndown Chart**
**Project Name**
**Version**

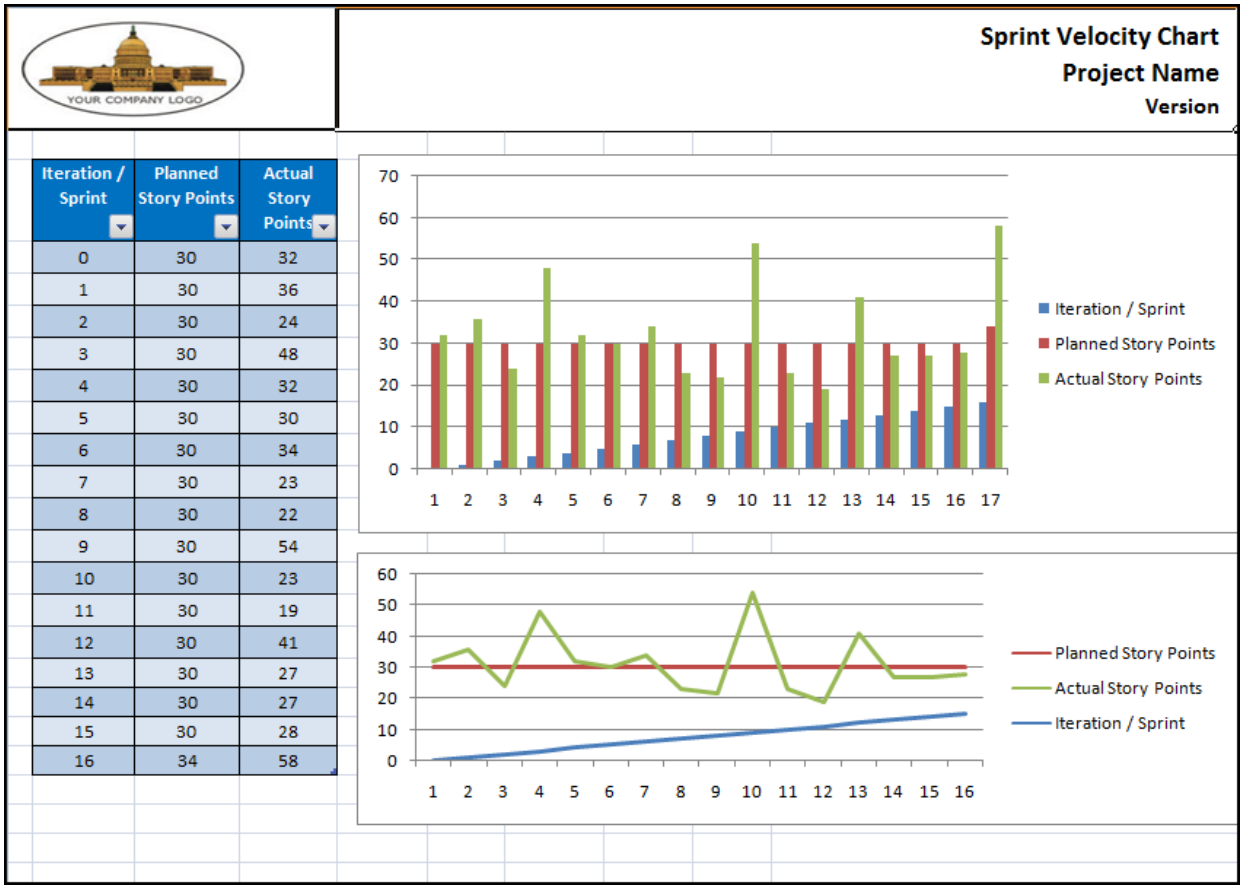
| Product Backlog Item (PBI) | Task | Name | Status | Original Esimate | Day 1 | Day 2 | Day 3 | Day 4 | Day 5 | Day 6 | Day 7 | Day 8 | Day 9 | Day 10 | Sprint Review |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PBI Item 1 | Task 1 | Name1 | Closed | 7.0 | 0.0 | 0.0 | 3.0 | 2.0 | 2.0 | 0.0 | 0.0 | Day | 0.0 | 0.0 | 0.0 |
| PBI Item 2 | Task 2 | Name2 | Closed | 3.0 | 0.0 | 3.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| PBI Item 3 | Task 3 | Name3 | In Progress | 4.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 2.0 | 0.0 | 0.0 | 0.0 |
| PBI Item 4 | Task 4 | Name4 | Closed | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| PBI Item 5 | Task 5 | Name5 | Closed | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.5 | 0.0 | 0.0 |
| PBI Item 6 | Task 6 | Name6 | Closed | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 |
| PBI Item 7 | Task 7 | Name7 | In Progress | 8.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 4.0 | 0.0 | 2.0 | 0.0 | 2.0 | 2.0 |
| PBI Item 8 | Task 8 | Name8 | Closed | 2.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 0.0 | 0.0 | 2.0 | 0.0 | 0.0 | 0.0 |
| PBI Item 9 | Task 9 | Name9 | Closed | 1.0 | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 0.0 |
| | | **Estimated Burndown** | | 27.0 | 24.3 | 21.6 | 18.9 | 16.2 | 13.5 | 10.8 | 8.1 | 5.4 | 2.7 | 0.0 | |
| | | **Daily Complete** | | 32.0 | 1.0 | 3.0 | 3.0 | 3.0 | 4.0 | 4.0 | 3.5 | 6.0 | 0.5 | 4.0 | 2.0 |
| | | **Actual Burndown** | | 32.0 | 31.0 | 28.0 | 25.0 | 22.0 | 18.0 | 14.0 | 10.5 | 4.5 | 4.0 | 0.0 | |



## Sprint Velocity Chart

| | |
|---|---|
| **Purpose:** | The Velocity Chart shows the amount of value delivered in each sprint, enabling you to predict the amount of work the team can get done in future sprints. It is useful during your sprint planning meetings, to help you decide how much work you can feasibly commit to. |
| **Description:** | 1. Velocity measures how much functionality a team delivers in a sprint.<br>2. Velocity measures a team's ability to turns ideas into new functionality in a sprint.<br>**Data Grid**: The Velocity Chart data grid is constructed as a table, which means whenever you update the table by adding or deleting rows or columns or simply changing values in the table, the graph is automatically updated.<br>**Formulas:** Formulas are embedded in the chart to calculate the charts:<br>Both the upper and lower charts: =SERIES(Velocity!$C$3,,Velocity!$C$4:$C$20,2) |

**Form Image**



## 4.7 Close Phase

### 4.6.1 Close Phase Overview

Projects are partially defined by the presence of both a beginning and an end. Many organizations fail to identify a project's end point, often causing perception problems among customers. Projects should end—with a celebration.

The key objective of the Close phase, and the "mini" close at the end of each iteration, is learning and incorporating that learning into the work of the next iteration or passing it on to the next project team.