Agile Metrics and Stakeholders

Jason Ebueng
K. Ngo
K. Nguyen
T. Yau

Overview

This presentation will cover Stakeholders in Agile and how they work with Agile Metrics.

The types of Stakeholders be established.

Agile Alliance’s Stakeholder Management Framework will be demonstrated.

Finally key metrics will be highlighted in regards to Stakeholders.

Stakeholders

Stakeholders are anybody invested in the success of the project.

These includes Engineers, Project Managers, Business Owners, and Customers.

Stakeholders are important in rapid product feedback while maintaining healthy communications between various parties.

Stakeholders Types

Internal Stakeholders

These members are closer to the project development. These include:

- Employees
- Managers
- Owners

External Stakeholders

These stakeholders are outside of the main project team, but still have a vested interest. These include:

- Competitors
- Suppliers
- Government Regulators
- Creditors
- Shareholders
- Customers

Stakeholder Management

Agile Alliance has a framework for managing Stakeholder relationships.

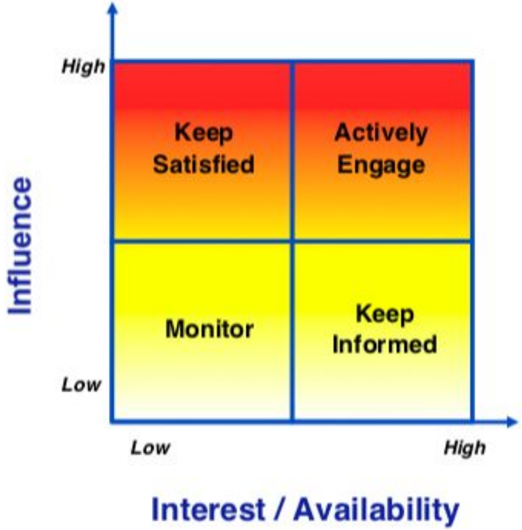


Figure from Stakeholder Management, Drew Jemilo

Stakeholder Management

Given the Stakeholder's availability and influence over a project, a relationship can be established with the right scope and information.

High availability and influential Stakeholders are actively engaged and involved with the project, such as Project Managers and Engineers.

Low influential and interested parties should be satisfied with a relationship where they can view reports and data.

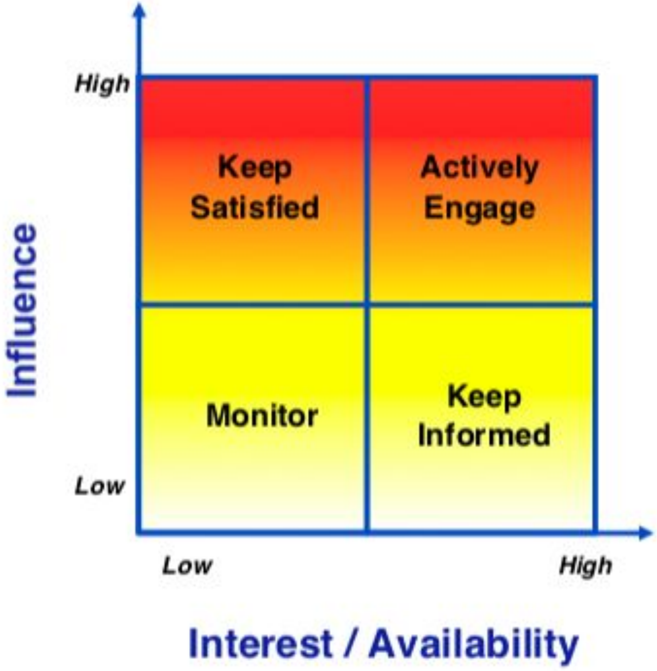


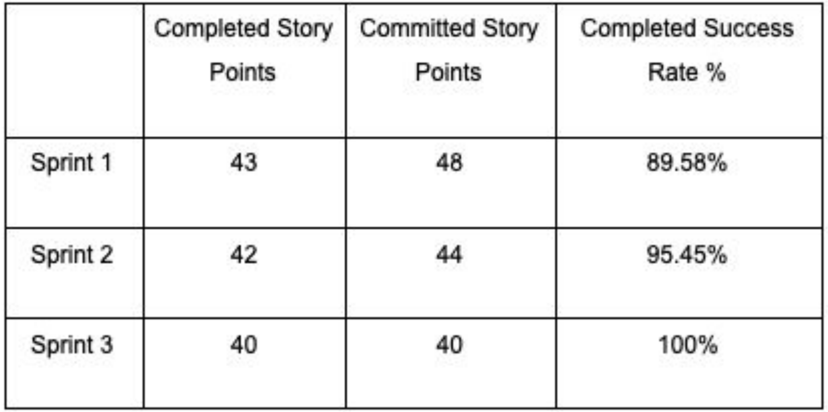
Figure from Stakeholder Management, Drew Jemilo

Key Metrics

Story Points are used for scoring and estimating tasks in the Product Backlog.

It’s important to score these points appropriately to give a sense of scope and resources for the task.

Stakeholders with at least moderate influence and availability will find this information useful.



	Completed Story Points	Committed Story Points	Completed Success Rate %
Sprint 1	43	48	89.58%
Sprint 2	42	44	95.45%
Sprint 3	40	40	100%

Table 3.1 - completed story rate for each sprint

Key Metrics

Velocity is the measure of Story Points in a sprint compared against time.

This is useful for reviewing Scope of a project in regards to the current resources, such as Team Size.

Highly influential Stakeholders benefit the most from this metric, such as Project Managers.

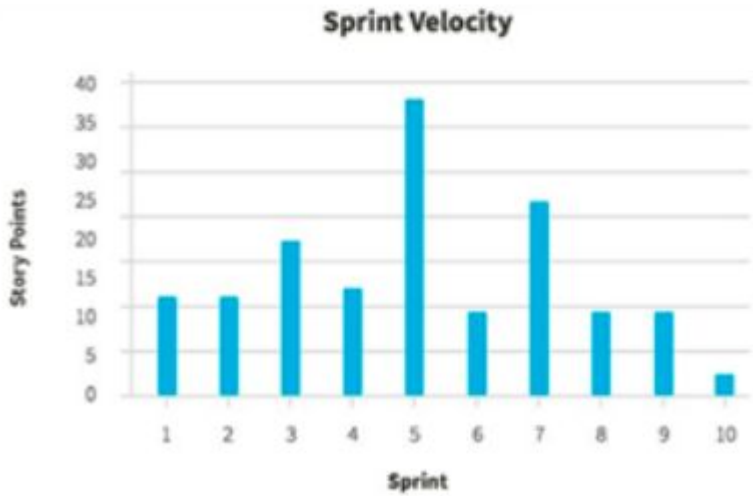


Figure 3.1 – Bar graph of Sprint Velocity

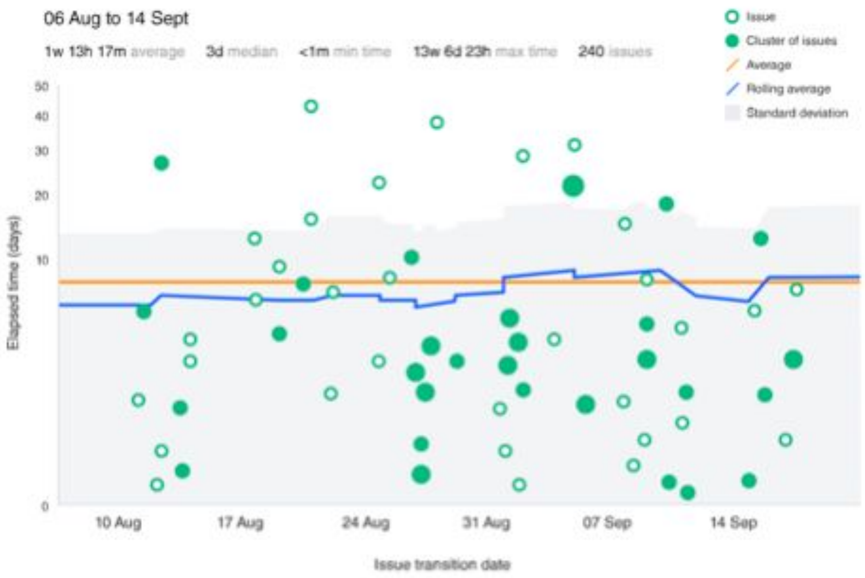
Key Metrics

Errors, Bugs, and Defects are important metrics for quality. They can be collected during the Sprint cycle and afterwards as well.

New user stories and tasks can be developed to put back into the tasklist or backlog, which is then need to be managed with limited resources.

Stakeholders of all influence and availability can find this metric useful, as it’s a general indicator of quality.

2.4.5: Control Chart



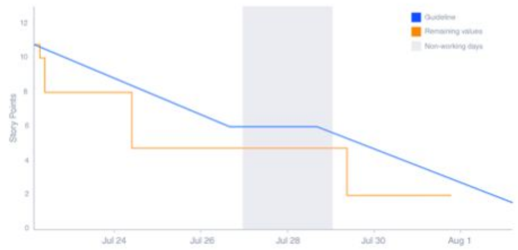
Key Metrics

For Stakeholders, Scope is available through a Project's Vision and Scope or requirements documentation.

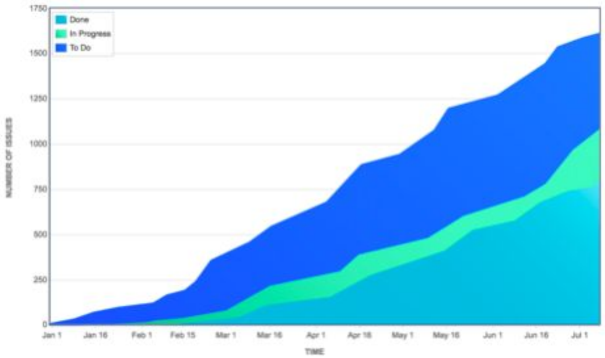
Scope analysis requires multiple key metrics such as Sprints, Velocity, and Defects.

Stakeholders who are at least moderately influential and influential will benefit most from Scope analysis.

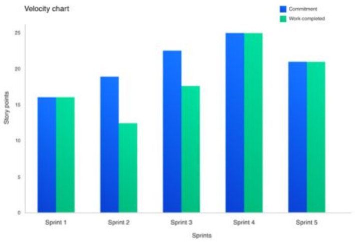
Burndown Chart



2.4.6: Cumulative Flow Diagram



2.4.4: Velocity Chart



Effectiveness Metrics

Effectiveness

- Business goals
  - Operating under schedule, cost, and time constraints
  - Increasing value
  - Increasing quality
  - Meeting customer demands & increasing customer satisfaction

Note on effectiveness metrics

- Effectiveness metrics may be counterproductive to process improvement if imposed by upper management or outsiders
  - Metrics should be used by Agile team to improve process capability
- Metrics should be used to answer questions and understand process capability
- Only metrics that align with business goals should be measured
  - Keep process simple and Agile
- Metrics should be used in combination with other metrics to provide the big picture

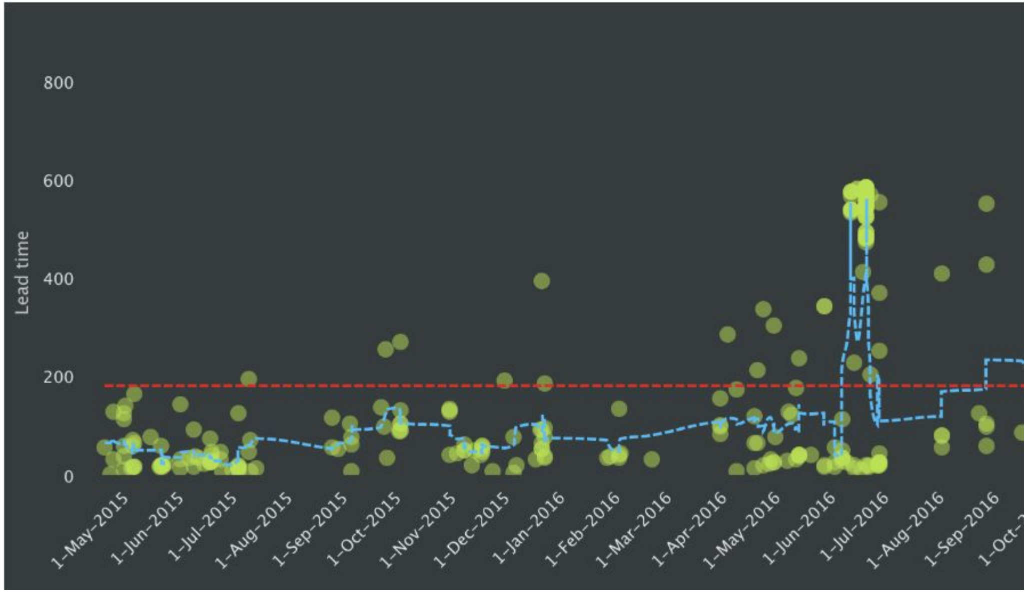
Effectiveness at Meeting Constraints

- While Agile focuses on increasing quality and value, it must operate under project constraints
    - Time, cost, scope
- Important measures for meeting constraints
    - Lead time
    - Queue metrics
    - WIP

Lead Time

- Amount of time it takes to complete the production of one unit from customer order to unit completion
- Aim to keep lead time as low as possible
  - Keep tasks from becoming stale
  - Holding cost
- Commonly visualized and controlled using control charts

Lead time

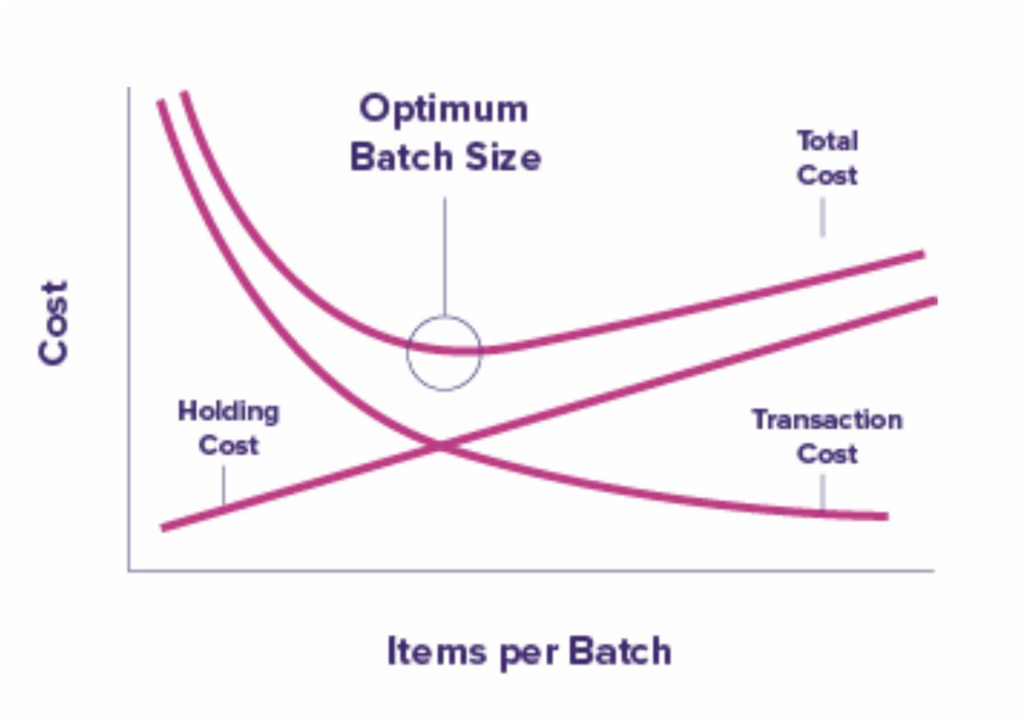


- Source: https://www.scrumdesk.com/start/manual-for-scrumdesk-start/project-lead-time-chart/

Queues and WIP

- Monitored in conjunction with lead time
- Critical to agile processes
- Holding cost associated with queues
- The larger the queue and the longer you hold, the higher the cost
- WIP
  - Critical to controlling the process
  - Indication of Agile team collaboration
    - Low WIP - high collaboration
    - High WIP– low collaboration

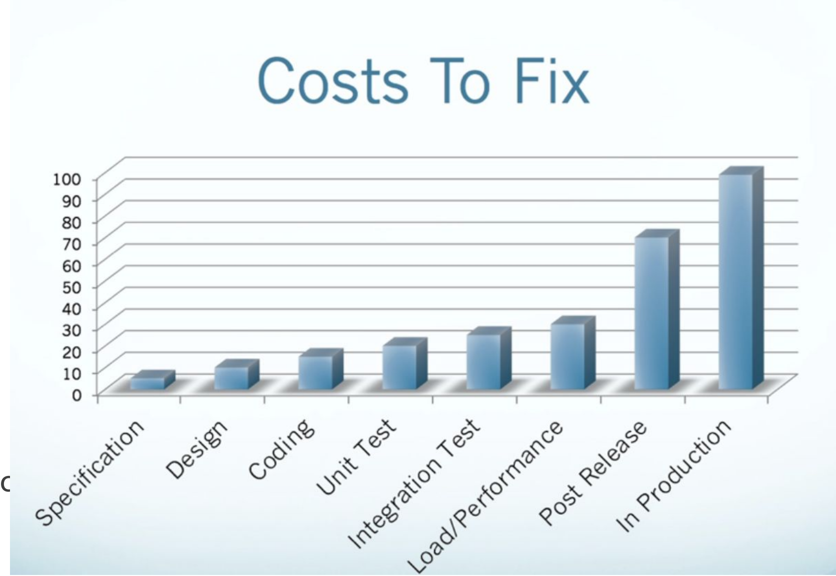
Queues Holding Cost



Source: https://www.conjur.org/blog/devops-dishwasher/

Effectiveness meeting quality goals

- Monitoring defects is critical
- Cost to fix defects increases exponentially throughout the software development life cycle



- Sourc

Code coverage

- Code coverage
  - White-box approach
  - Measures the number of lines of code that are covered by the test cases
  - higher the code coverage, the higher the degree the source code is executed, and the lower the chance a defect will go undetected

Test coverage

- Test coverage
  - Black-box approach
  - measures the degree to which requirements have been tested in the code.

- Measuring test coverage and code coverage in tandem help provide a bigger picture of system quality

Agile Team Metrics

Measure progress, health, environment

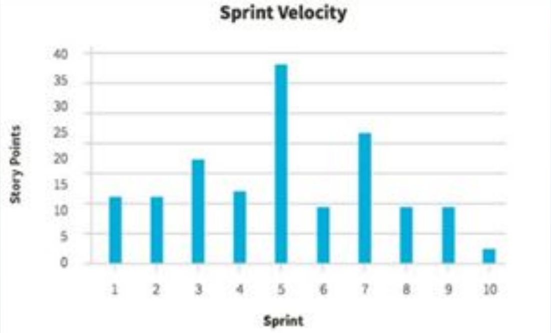
Agile Team – Daily Scrum

- Daily Scrums Metric
    - Stand up meetings everyday
    - monitor sprint progress
    - report problems
    - report tasks progress
    - Promote communication

Agile Team – Sprint Retrospectives

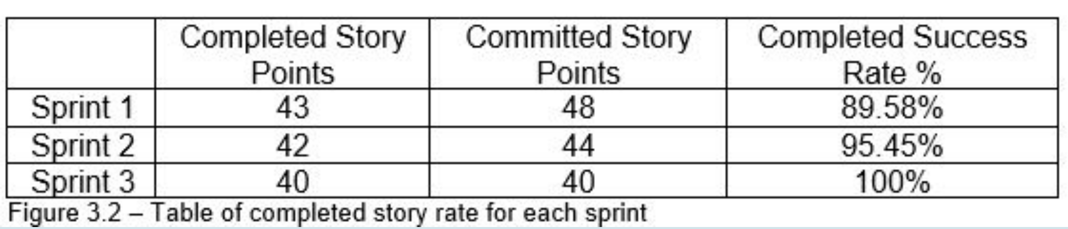
- Meetings to discuss how well the sprint went
- Share best practices for improvements
- Evaluate team's health
- Evaluate team's process and productivity
    - Team Velocity
    - Completed Success Rate
    - Estimation Accuracy

Agile Team – Sprint Retrospectives



- Team Velocity Metric
  - measures average number of stores completed in a sprint
  - Purpose: estimate how much can be accomplished to better plan next sprint
  - Prevent team from overcommitting

Agile Team – Sprint Retrospectives

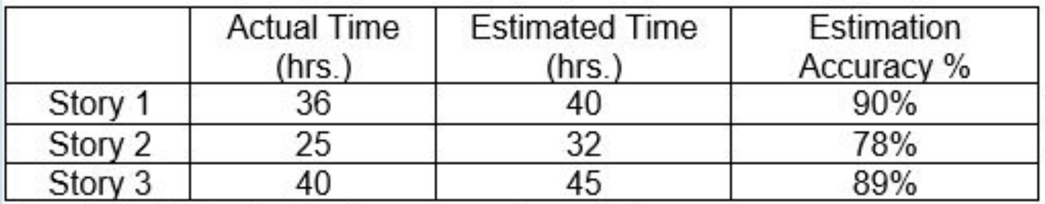
- Completed Success Rate Metric
  - Purpose: looks at percentage rate of completed stories over committed story
  - 

	Completed Story Points	Committed Story Points	Completed Success Rate %
Sprint 1	43	48	89.58%
Sprint 2	42	44	95.45%
Sprint 3	40	40	100%

Figure 3.2 – Table of completed story rate for each sprint

Agile Team – Sprint Retrospectives

- Estimation Accuracy Metrics
  - Calculates % of actual time spent to estimated time needed
  - Purpose: record accuracy of time to better estimate time needed for upcoming sprint.



	Actual Time (hrs.)	Estimated Time (hrs.)	Estimation Accuracy %
Story 1	36	40	90%
Story 2	25	32	78%
Story 3	40	45	89%

Figure 3.4 – Table of estimation accuracy for each story

Agile Team Satisfaction Metric

- Satisfaction Surveys Metric
  - Assess health of team by conducting surveys to see how much the team is satisfied with their work
- Happiness Ratings Metric
  - Assess happiness with members rating how they feel
- Team Morale Ratings Metric
  - Rate statements on a scale
    - I fee fit and strong in my team
    - I am proud of the work that I do for my team
    - I am enthusiastic about the work that I do for my team
    - I find the work that I do for my team of meaning and purpose

Agile Team Satisfaction Metric continues

- Peer Feedback Metric
  - Each member chooses one peer, observer their chosen peer in a given time frame, and share their observation
  - Improves self-management
- Team Feedback Metric
  - Stakeholders rate and give feedback to team base on their observations

Agile Team Member Turnover Metric

- Monitor the health environment of the agile team
- Replacement of team members can assess the team’s working environment
- Low turnover = healthy environment
- High turnover = poor environment

Agile Delivery

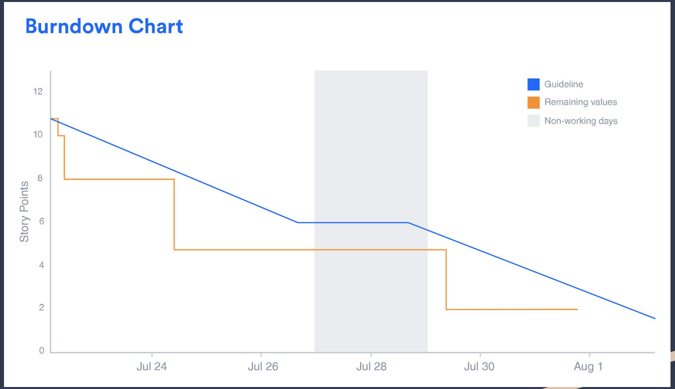
Agile Delivery: Introduction

- Metrics and visualizations are indispensable when aiming to maximize the efficiency of software delivery in any Agile software organization
- Metrics and visualizations, when produced correctly, can provide useful information for project leads or any other consumers to make informed decisions for and provide guidance to the software team

Agile Delivery: Introduction (continued)

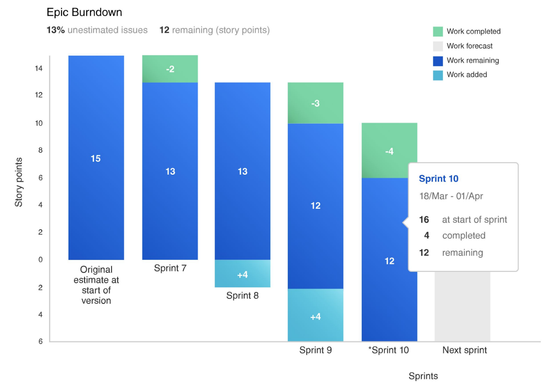
- Atlassian, a provider of software development and collaboration tools, has developed a suite of visualization tools that are popularly used in the software industry today to visualize a variety of agile metrics produced from software development processes
- Each visualization tool has its own purpose and conveys its own set of information that project leads, management, or even team members may find useful when developing project roadmaps, scheduling software releases, or making hiring decisions
- In our research on the topic of Agile Delivery, we will focus on the following 5 different charts: Sprint Burndown, Epic and Release Burndown, Velocity, Control Chart, and Cumulative Flow Diagram

Sprint Burndown Chart



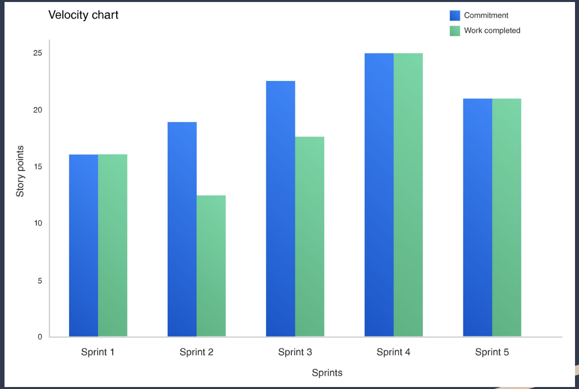
- The Sprint Burndown Chart is one of the most useful types of visualization tools for any Agile organization that plans their work into a number of iterations, called sprints
- To keep track of progress on these tickets during the sprint, a Sprint Burndown Chart can be utilized to visualize the completion of tickets over time within the Sprint
- The Sprint Burndown Chart plots time on the x-axis against Story Points on the y-axis, where Story Points are values assigned to each ticket that represents approximately how much work is involved for that ticket
- A software team might aim for 0 remaining story points (orange line hitting x-axis) to demonstrate completion of work for a given sprint.

Epic and Release Burndown Chart



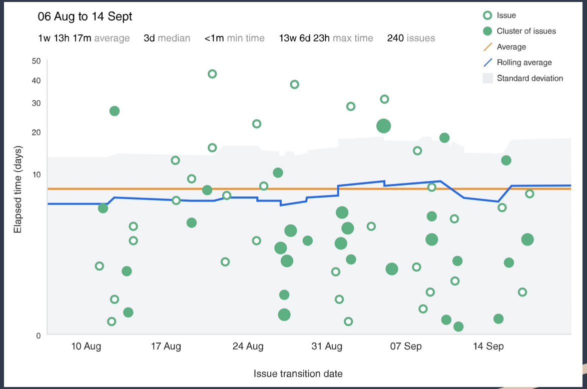
- The Epic and Release Burndown Charts are also important types of visualizations that are used to track development progress on a larger scale
- Development teams may work on several different Epics within the same sprint, so the Epic Burndown Chart can be used to capture progress on all the Epics being worked on
- Release Burndown Chart can be used to keep track of all the tickets from each Epic that are planned to be released, giving the team a visual of how much work remains to be completed before releasing the product
- Due to the possibility of scope creep within any given sprint, the software team can use the Epic and Release Burndown Charts to realize any anomalies in their workflow

Velocity Chart



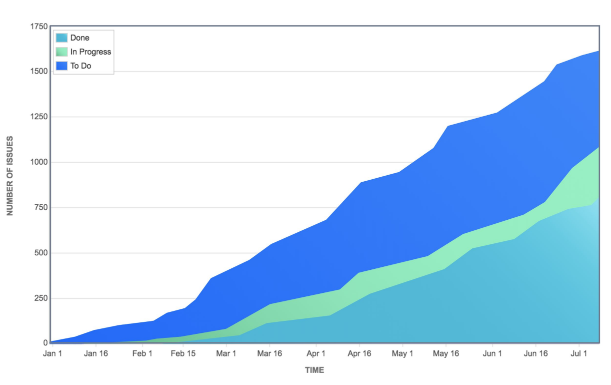
- The Velocity Chart may be used to keep track of the team’s velocity across sprints, or the total amount of work actually completed by the team, and compare that against the total amount of work they were committed to do
- Velocity is often measured using the sum of story points or the total number of hours spent working on tickets during the sprint
- The Velocity Chart is helpful because it may be used to estimate how quickly a team can complete all the tasks in the backlog, allowing project leads to more effectively estimate how many sprints it would take to complete a certain amount of work

Control Chart



- The Control Chart is often used to keep track of Cycle Time, the time it takes for the statuses of tickets to change
- For example, the Control Chart may tell the reader how much time was needed for a developer to move the ticket from “In Progress” to “Done”
- It is useful to visualize Cycle Times because a team with consistent cycle times might imply that they are dependable in getting work done
- A software team can use the Control Chart to improve their development process by aiming for short and consistent cycle times

Cumulative Flow Diagram



- The Cumulative Flow Diagram keeps track of the total number of issues in different statuses for a team
- It is useful to determine whether or not the flow of work is consistent across time, allowing a project lead to detect any bottlenecks in the workflow
- A Cumulative Flow Diagram with a smooth pattern over time suggests a consistent flow of work by the team

This marks the end of the presentation.
Thank you very much!