Managing and Leadership

Lecture 5

SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://about.softuni.bg

Have a Question?

Software University

sli.do

#Agile

Table of Contents

1. Scrum events
2. Key Metrics
3. Managing Risk
4. Agile Leadership
5. Q&A

Scrum Events

The **ceremonies** behind Scrum

Scrum Events – "Sprint Planning"

Software University

- **Sprint planning**
  - It all starts with planning
  - At the beginning of each iteration in the form of a meeting
  - The **input** for the Spring Planning is the Product Backlog
  - There are other inputs used in the later Planning sessions – can you think what those are?
  - The whole "Scrum Team" participate
  - It should not take more than 2-4 hours
  - It should cover ALL for the upcoming sprint
  - The **output** is the Sprint Backlog

"Sprint Planning" – Roles

- The roles are distributed during the Scrum Planning Phase
  - **Product Owner**
    - Reads the US-s to be added in the upcoming Sprint
    - Explains the priorities
    - Explains (Acceptance Criteria) the precise expectations
  - **Scrum Master**
    - Facilitator
    - Makes sure all voices are heard
    - Watches for barriers, pitfalls, and unforeseen blockers
    - Timeboxing
  - **The Dev Team**
    - Ensure each US is well understood
    - Ensure the plan for the Sprint is realistic
    - Actively participate during the organization of the Sprint Backlog

Sprint Planning – Best practices (1)

- During the Planning Phase think of the three basic questions
  - **What**
    - The Product Owner describes the objective of the sprint and what backlog items contribute to that goal. The Scrum Team decides what can be done in the coming sprint and what they will do during the sprint to make that happen. This is a two-way conversation.
  - **How**
    - The Development Team plans the work necessary to deliver the sprint goal. Ultimately, the resulting sprint plan is a negotiation between the development team and product owner based on value and effort.
  - **Who**
    - The Product Owner defines the goal based on the value that they seek. The Development Team needs to understand how they can or cannot deliver that goal. If either is missing from this event it makes planning the sprint almost impossible.

Sprint Planning – Best Practices (2)

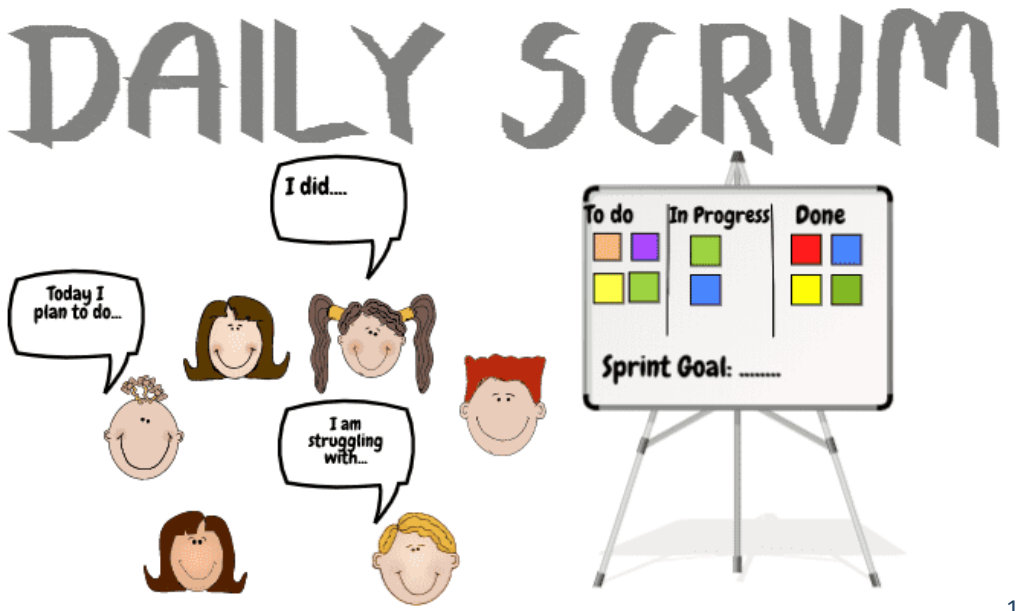
- Make sure the goal is clear, realistic, and achievable
- Consider the time required for fixing bugs, system issues, unpredicted problems…
- 80/20 rule
- Do not overcommit

Scrum Events

- **Sprint planning**
  - It all starts with planning
  - At the beginning of each iteration
  - 2-4 hours workshop
- **Daily Scrum**
  - Daily meet
  - 15 min/daily
  - Stay focused on the goal

Scrum Events – Daily Scrum (1)

- The most important event during the delivery phase
- …and yet, it should not take more than 15 min
- Answers the three main questions
  - Why did I do it yesterday?
  - What am I planning to do today?
  - Are there obstacles ahead of me?
- An event for the Dev Team!



Scrum Events – Daily Scrum (2)

- Who does participate?
  - **The event is for the Dev Team**
    - The SM –to observe/ensure the event follows the basic rules
    - The PO is invited
- Why 15 min only?
- The Daily Scrum… is it every day?
- Does everyone from the Dev Team need to participate?
- How about distributed teams and different time zones?

Scrum Events – Daily Scrum (2)

- **What is the true value** of the Daily Scrum?
  - A suitable time Developers are allowed to adjust their plan
  - A checkpoint to inspect the progress towards the Sprint Goal/s
  - It increases the probability that the Development Team will meet the Sprint Goal
  - Team members often meet immediately after the Daily Scrum for detailed discussions
  - *Isn't it a burden to hold such a meeting every day?*

Scrum Events – Daily Scrum (3)

- **What is the true value** of the Daily Scrum?
  - Daily Scrums improve communications, eliminate other meetings, identify impediments, promote quick decision-making, and improve the Development Team's level of knowledge across the team.
  - The structure of the meeting is set by the Developers and should be conducted to serve the purpose (not a particular guide).
  - Another common term is "Daily Stand Up"…

Scrum Events – Daily Scrum (4)

- What do you need to be careful about?
- Signs of unproductive event/s and the role of the SM?
- What if 15 min are not enough?
- Same place / Same time / Same people
- Create a safe environment where issues can be raised

Location

Size

Setup

Light

Air

Privacy

Scrum Events – "Sprint Review" (1)

- Time: At end of Sprint
- Duration: **4 hours** for 4-week Sprints, proportionally **shorter** for shorter Sprints
- Who: **Scrum Team, the customer, key stakeholders**
- Input: Sprint Backlog (User Stories), product Increment (Tasks)
- Output: revised Product Backlog

Scrum Events – "Sprint Review" (2)

- The time that the team shows what has been produced during the sprint
- It is also meant to inspect the Increment and adapt the Product Backlog if necessary
- During the meeting, the team collaborates on what steps should be taken next to optimize the value of the software
- It is an informal, but important meeting held at the end of each Sprint.
- Attended by the SCRUM team + stakeholders (clients)

The Sprint Review – Agenda

Software University

START	DUR.	ACTIVITY	DESCRIPTION	WHO
09:10	10 min	**Goal of the Sprint**	• Introduction into Sprint Goals • Information about an impact of the sprint on release plans • Review of the product roadmap	Product Owner
09:20	5 min	**Review of the top requirements**	• Review of up to three top requirements related to sprint goal	Product Owner
09:25	15 min	**Sprint Status**	• Share information about plan vs reality • Review of the sprint statistics • Review of the important technology changes or improvements, enablers • Statistics of bug fixes • Current program increment (release) statistics	Scrum Master
9:40	15 min	**Demonstration**	• Live demonstration of up to three completed requirements	Team
9:55	10 min	**Feedback**	• Collecting the feedback from stakeholders	Scrum Master
10:05	5 min	**Closing**	• Information about the next sprint review • Publishing sprint review presentation	Scrum Master

The Sprint Review – Roles and Responsibilities

- **Product Owner**
  - An introduction to sprint goals
  - An introduction of planned top requirements that the team has committed to deliver
  - Sprint status overview
  - Information about defects and improvements
- **Scrum Master**
  - Moderator of the meeting
  - Evidence of feedback
- **Development Team**
  - Informs about the sprint status
  - Live demonstration of functionality

The Sprint Review: Hints & Tips

Software University

- Preparation is key
- Send reports / information / agenda in advance
- Keeping a routine is not a bad thing
- Keep this "timeboxed"
- Make everyone feel comfortable
- Speak the language everyone understands. Avoid becoming too technical
- Explain why something has(not) been done during the sprint
- Talk about "how this functionality would solve a problem", not necessarily about the functionality itself
- Stick to the Agenda
- Dry run!

Scrum Events

- **Sprint planning**
  - It all starts with planning
  - At the beginning of each iteration
  - 4-8 hours workshop
- **Daily Scrum**
  - Daily meet
  - 15 min/daily
  - Stay focused on the goal
- **Sprint review**
  - At the end of each Sprint
  - Demonstrating what has been achieved
- **Sprint retrospective**
  - How to improve the next Sprint

Sprint Retrospective (1)

- When: Immediately following the Sprint Review
- Duration: up to **3 hours** for 4-week Sprints, proportionally **shorter** for shorter Sprints
- Who: **Scrum Team**
- Goal: Inspect and adapt with regard to **people**, **relationships**, **processes**, and **tools**
- Do not confuse the Sprint Review and the Sprint Retrospective

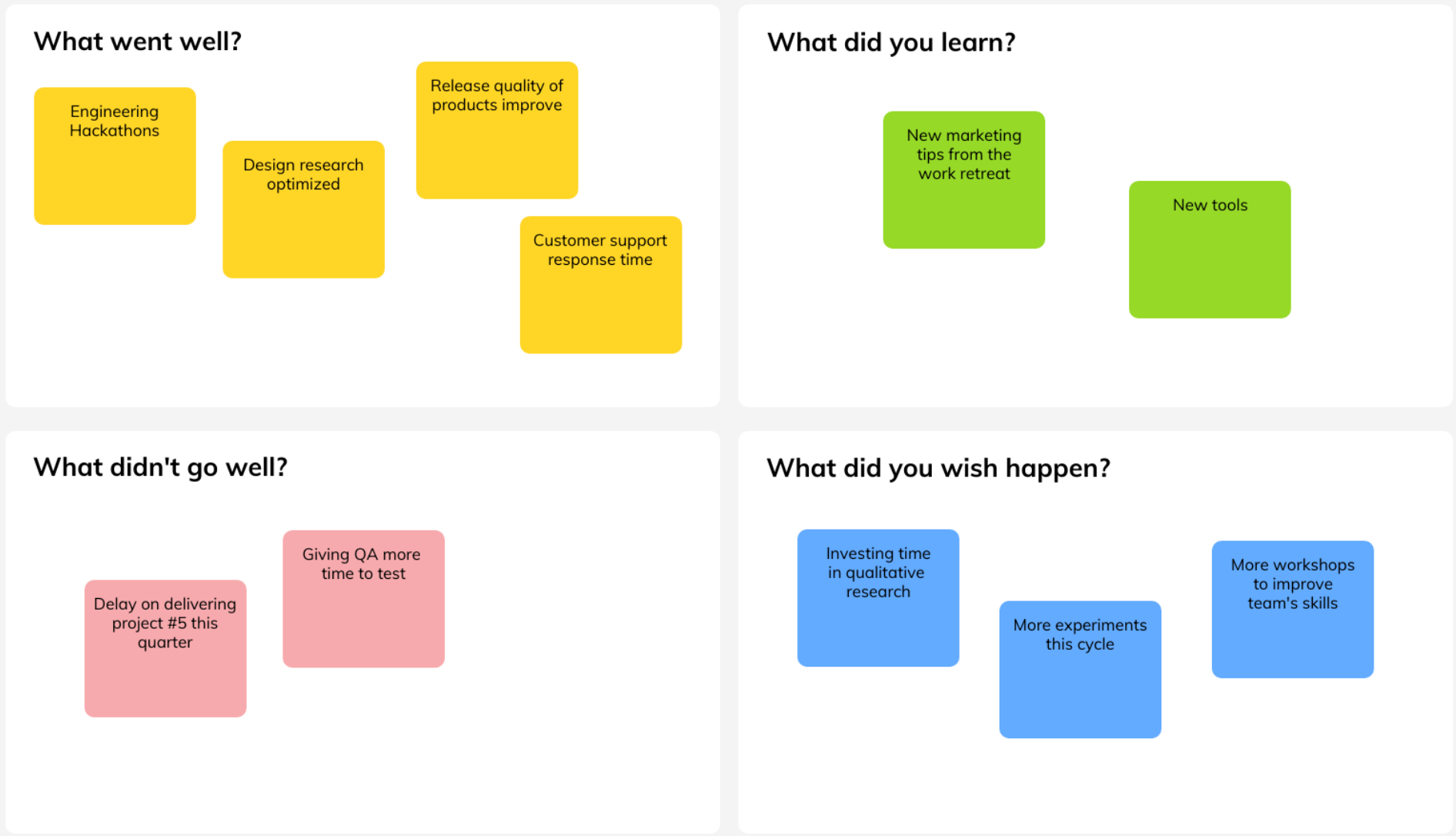
Sprint Retrospective (2)

- The Scrum Team reflects on the work done during the sprint
- The Focus is on the Team, not on the Product Increment
- Held immediately after the Sprint Review
- Typical Questions to be asked
  - What did go well?
  - What did not go well / What should be changed?
  - Unclear Areas?
  - New Suggestions / New ideas?
- What is the purpose and how does it help?

Sprint Retrospective (3)

- Team identifies potential problems/barriers
- Team identifies potential improvements
- Specific actions are being assigned
- Sprint Retrospective is not an isolated event
- Who runs the Retrospectives?
  - The Scrum Master facilitates the meeting
  - The PO and the Dev Team participate actively

Sprint Retrospective – Example (1)



Sprint Retrospective – Example (2)

Software University

How To Do A Retrospective

What went well	What to improve	Action items
Good Collaboration	Before picking any story, each story should have clear acceptance criteria.	Try to eliminate unnecessary meetings
Willingness to work on new tech stack	Fewer meetings more coding, Period!	Find more accurate ways to estimate our stories
New team member on-boarding going well	The team should pick more non-functional requirements to improve code base	Guarantee all stories have clear acceptance criteria before picking them up
Have meeting mid-sprint to discuss future stories	Each story should be divided & people assigned to it – not everyone on same story	
Good pairing work was done despite remote work.	Use a more accurate way of estimating story size – what we use isn't accurate.	
E2E automation has really improved things	Need to have more team building events	

Scrum Events – Sum Up

- Sprint planning
  - This event is about **preparing** the team for the upcoming iteration
  - **Aligning** the entire team for the Sprint
- Daily Scrum
  - Facilitates **Communication**
  - The event is to ensure timely, quick, regular interlock between team members
  - It allows the team to stay Agile and **react quickly** to any potential threats
- Sprint review
  - **Present** what has been produced during the Sprint
  - **Keeps all stakeholders in the loop**. There is no gap between expectations and deliverables
- Sprint retrospective
  - The event is there to support a constant, gradual, incremental **learning** curve
  - We learn best from **previous experience**

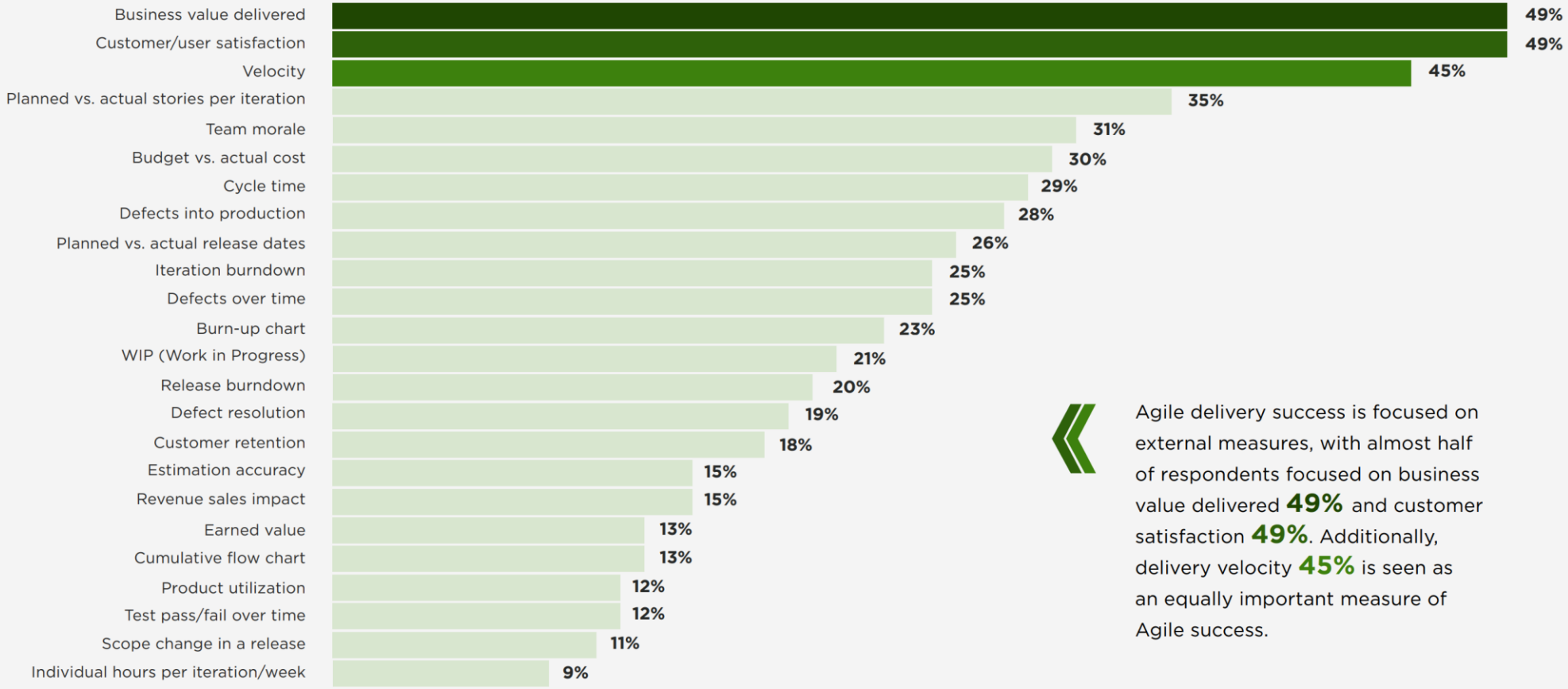
Running the Sprint

- Understanding the 3 key components of Scrum is the backbone of the framework
  - Scrum Actors
  - Scrum Artefacts
  - Scrum Events
- There are more practices that an experienced Scrum Master should be aware of
  - Think of Risk
  - Think of Innovation
  - Think of waste reduction
  - Think of measurements
  - Think of leadership and management
  - And much more…

How Do You Track Success?

Agile Adoption

How does your organization measure the success of agile delivery?



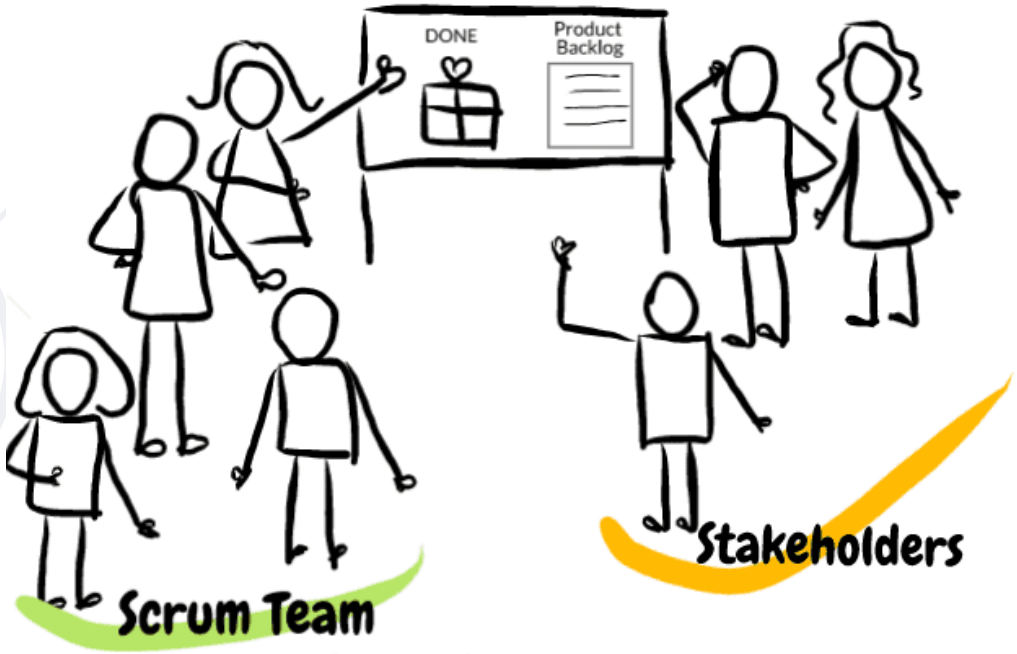
Agile delivery success is focused on external measures, with almost half of respondents focused on business value delivered **49%** and customer satisfaction **49%**. Additionally, delivery velocity **45%** is seen as an equally important measure of Agile success.

Tracking Team Performance

- Agile teams monitor their progress and performance with key tools and metrics also commonly used for planning and estimating

The Primary Metric for Agile Development

- Whether or not working software exists and whether it is demonstrably suitable for use in its intended purpose
  - In Scrum this key indicator is determined empirically, by a demonstration at the end of every single Sprint



Scrum Board – Status Reporting

- **Status Reports** usually include
  - Recent accomplishments
  - Upcoming activities
  - Risks
  - Schedule update
  - Budget update

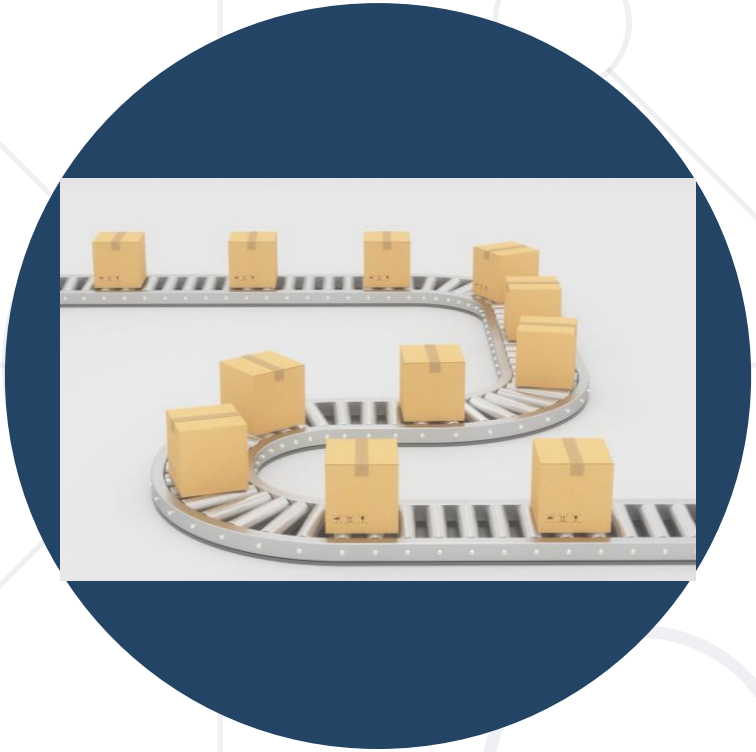
Goodhart's Law

- That's because it's possible to manipulate the things we measure
- Sprint velocity is a good example of this. Because velocity is directly derived from the team's estimates, if you assess team members based on their velocity, you are begging them to give you inflated estimates

```
let law = Goddhart's();
// "When a measure becomes the target,
it ceases to be a good measure"
```

Scrum Board – Release Burndown Chart

- **Visual Summary** of
  - **Projected Delivery** of product
  - **Effort Remaining** until completion
  - **Current Trend** to completion
  - **Scope Changes** on the timeline

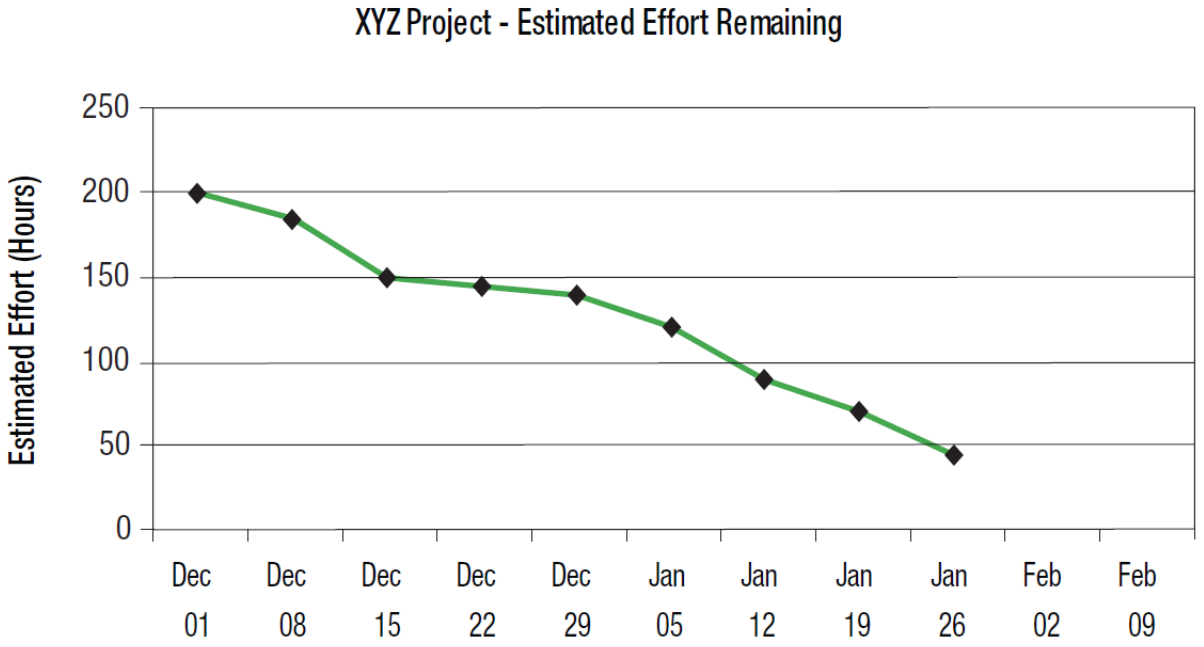


Key Metrics

Burndown Charts

- Tracks the work that remains to be done on a project
- Most commonly used for measuring the team's progress
- Can make it hard to separate the impact of scope creep from the team's progress

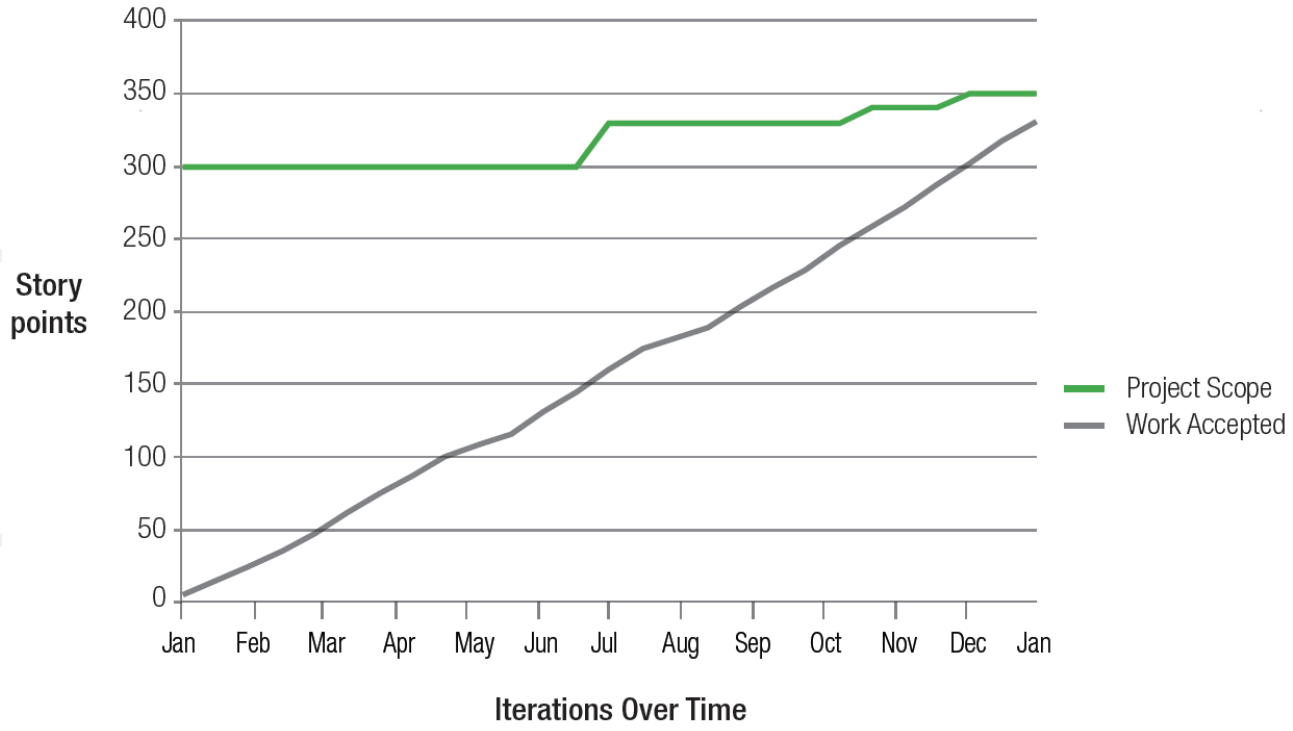
Burndown chart using time to measure progress



Burnup Charts

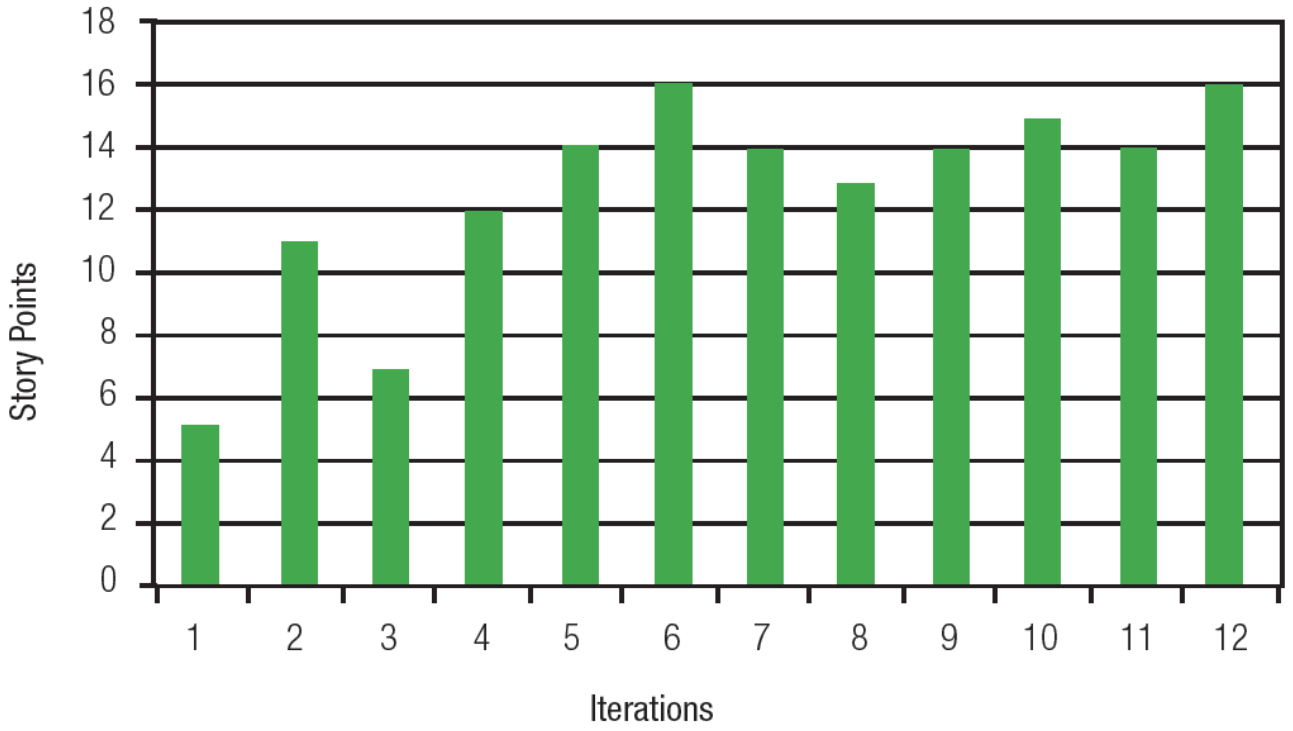
- Tracks the work that has been completed (accepted)
- Can show changes in scope and make the impact of changes visible

Sample burnup chart using points to measure progress



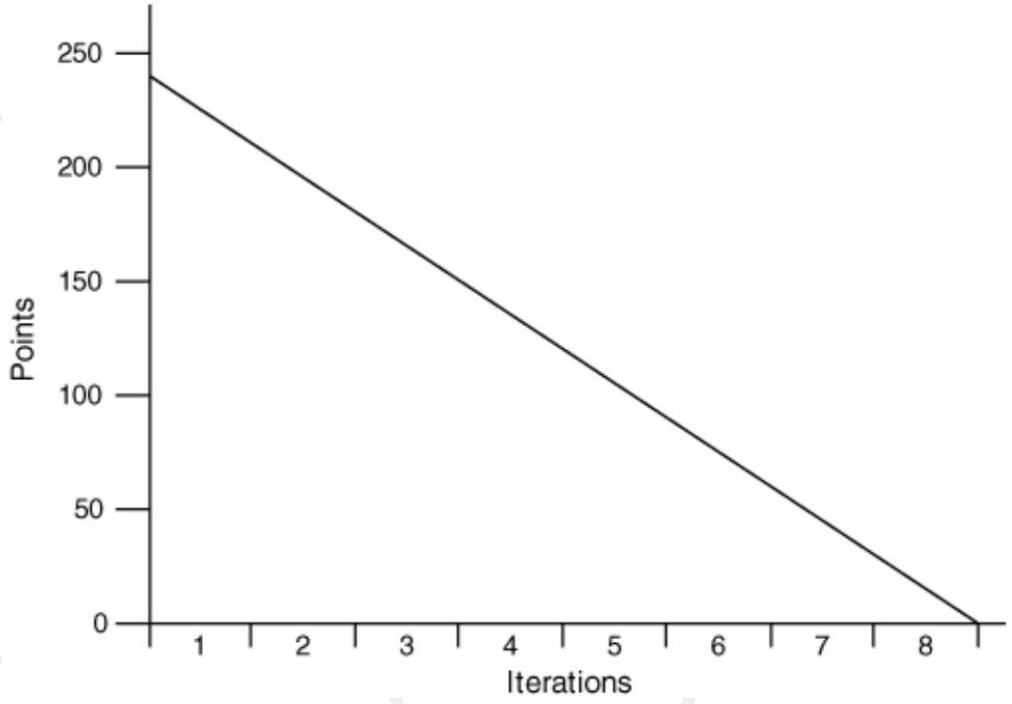
Velocity

- The measure of a team's capacity for work per iteration
- Predicts how much work the team will be able to do in future iterations, based on the amount of work they completed in past iterations
- Velocity is measured in the same units the team uses to plan work (i.e., story points, user stories, hours, days, jelly beans, etc.)

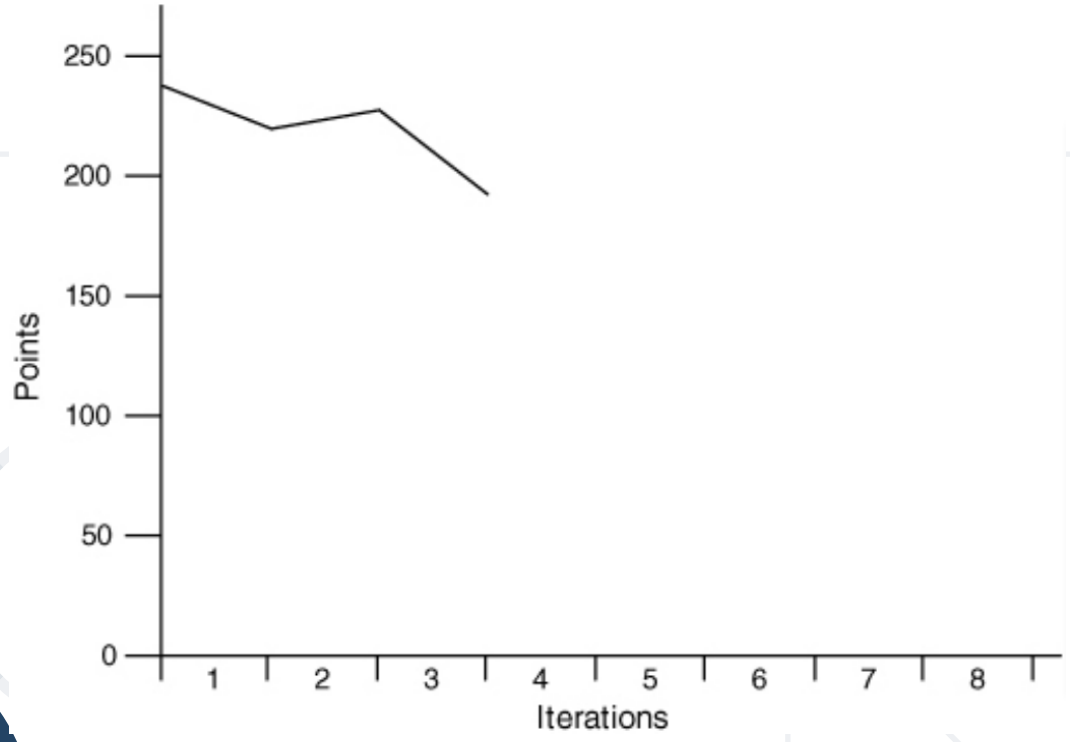


Backlog of 240 points

- The perfect burndown



- Realistic burndown



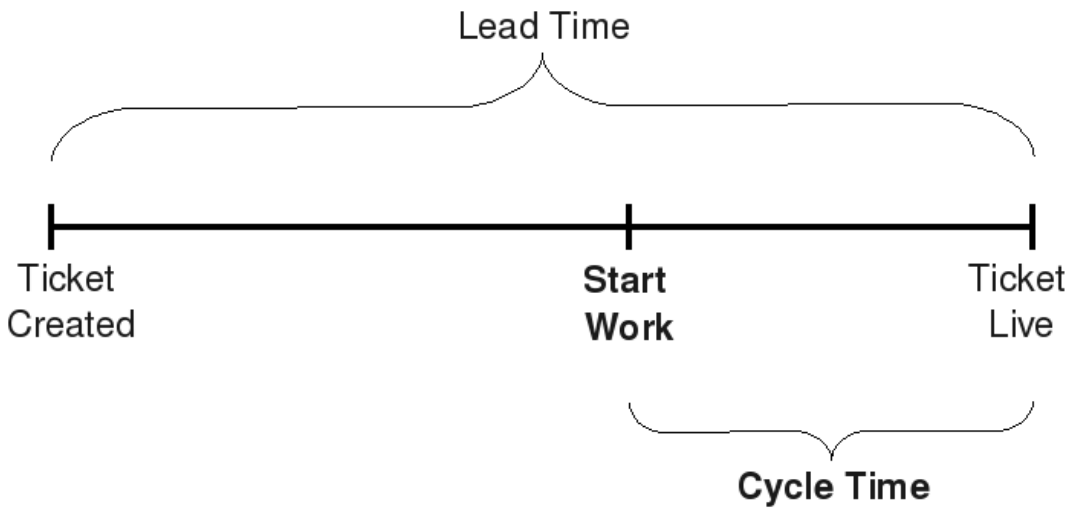
Velocity Anti-Patterns

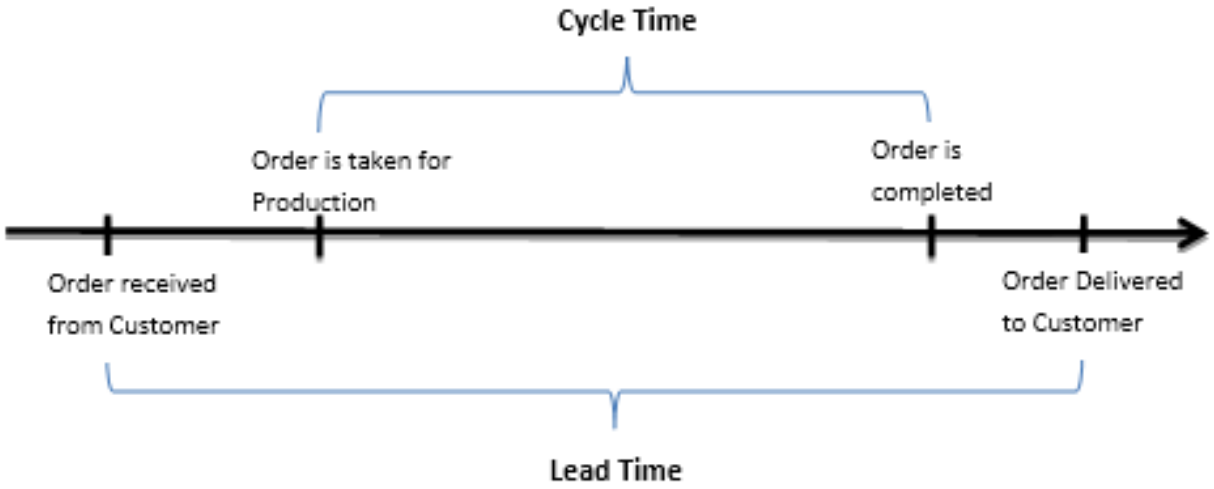
Software University

- Setting **targets** for velocity
- Setting velocity as a **percentage complete** to fit a traditional understanding (mindset) of a scope
- Expecting a team to **immediately** reach its maximum velocity
- Using velocity to **compare teams**
- Using velocity for the **mixed type of work** – value (capitalized) and enhancements (expense)

Additional Metrics

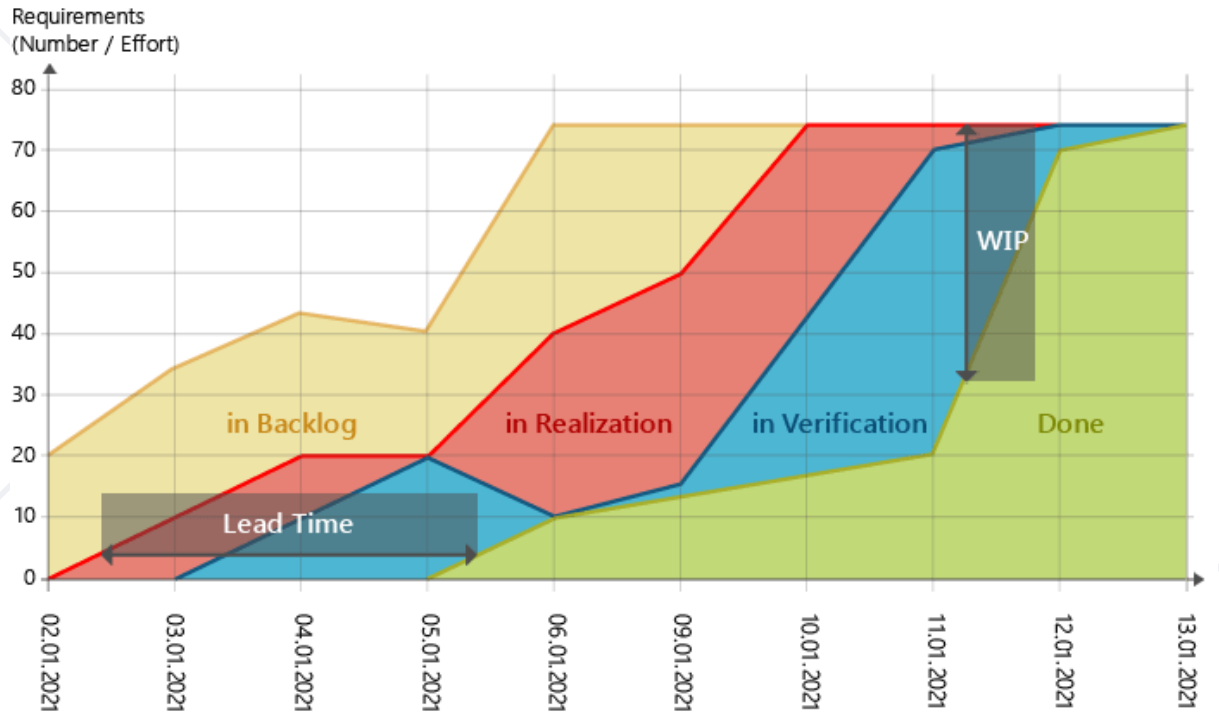
- Lead time – measures the time it takes something to go through the entire process – e.g., from requirements gathering to deployment
- Cycle time – measures the time it takes something to go through part of the process – e.g., from coding to testing





Cumulative Flow Diagrams

- Cumulative flow diagrams
  - Help find the bottlenecks in the process
  - Points out the points where there is a buildup of Work in Process (WIP)
  - Exposes overall throughput, cycle time, and queue depth, lead time, and location of bottlenecks



Traditional Metrics

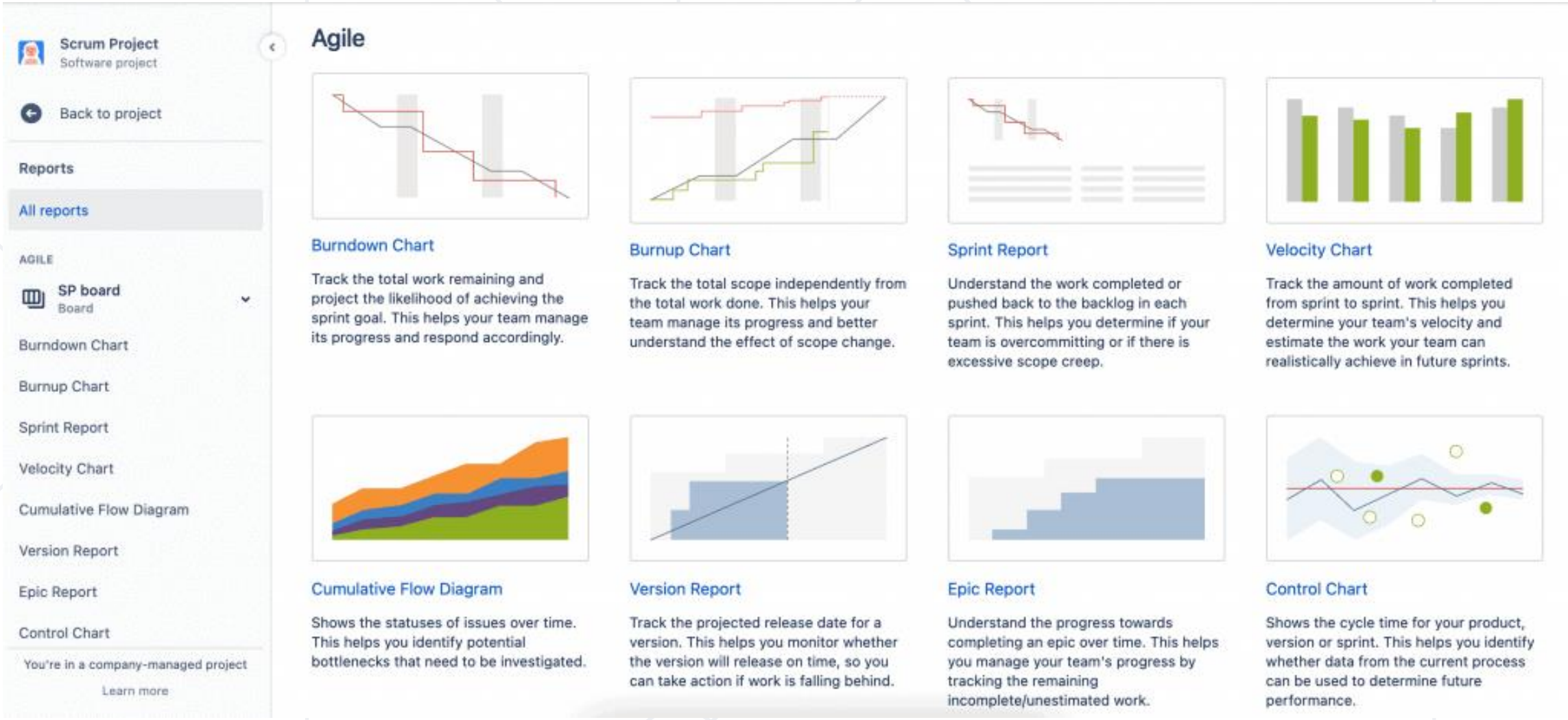
- Traditional process and project metrics
  - Defect count
  - Percentage of code with unit test coverage,
  - Percentage of code covered by automatic regression tests

Value Metrics

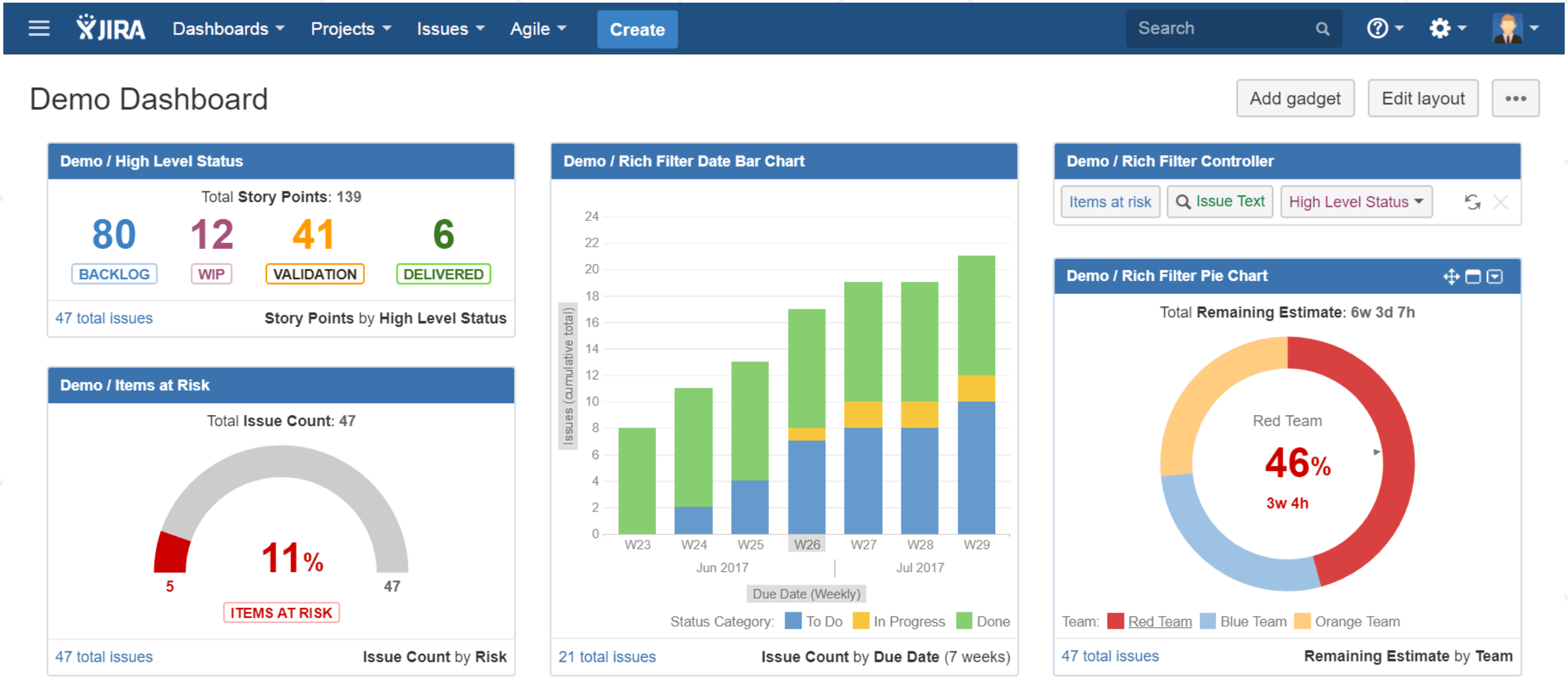
- Net Promoter Score (NPS)
- Revenue or profitability per customer
- Repeat customer business
- Reduction in the total cost of ownership
- Improved conversion rates
- Growth in the number of customers/users
- Customer referrals
- Market share
- Aggregate revenue or profit
- Cost to obtain a new customer
- Time saved for the customer to achieve a goal
- Frequency of feature usage
- Duration of feature usage
- Number of a customer using a feature

Scrum Board – No More Reports (1)

Software University



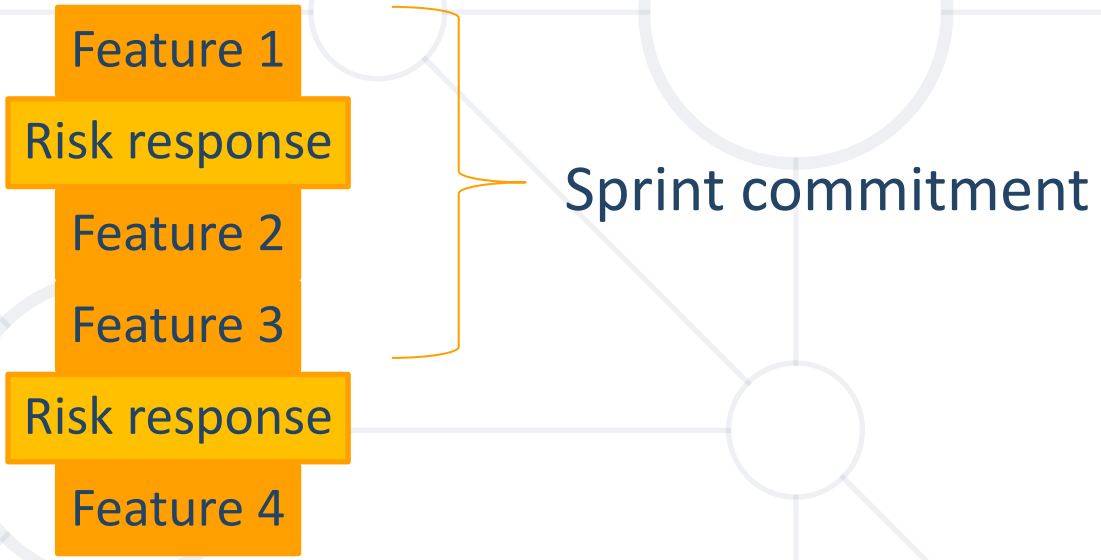
Scrum Board – No More Reports (2)



Managing Risk

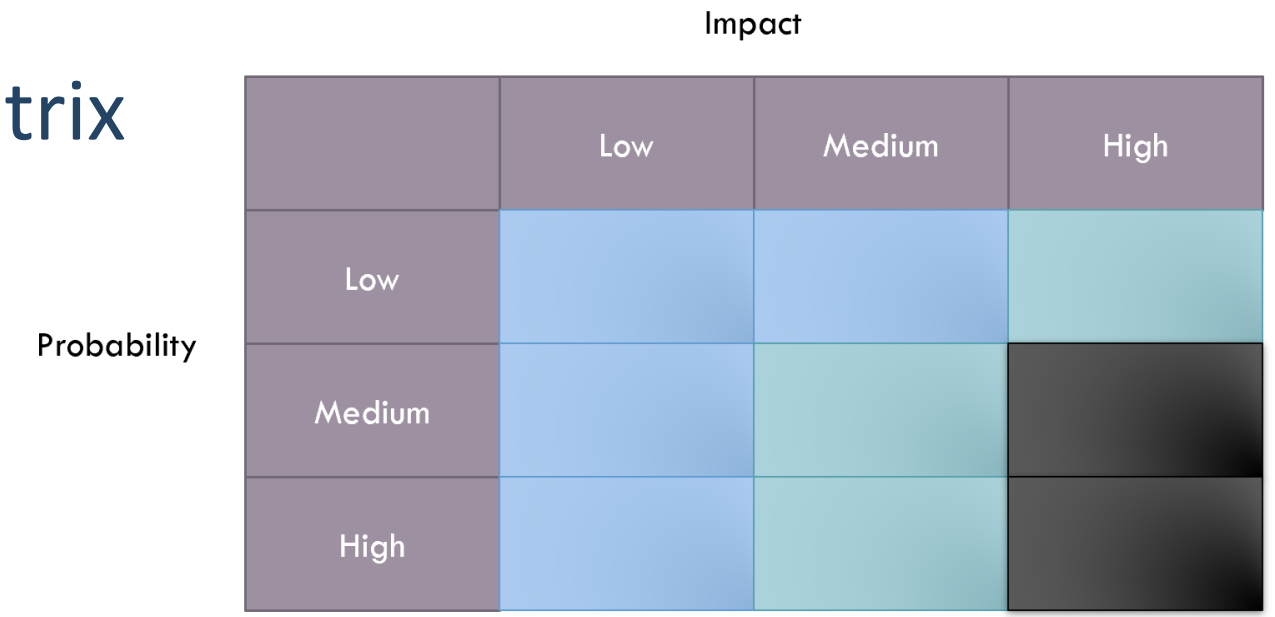
Value Metrics

- Agile backlogs are prioritized not only for features but also for risk responses to identified risks
- When planning an iteration, the team balances the delivery of high value features with the mitigation of the biggest project risks



Risk Severity

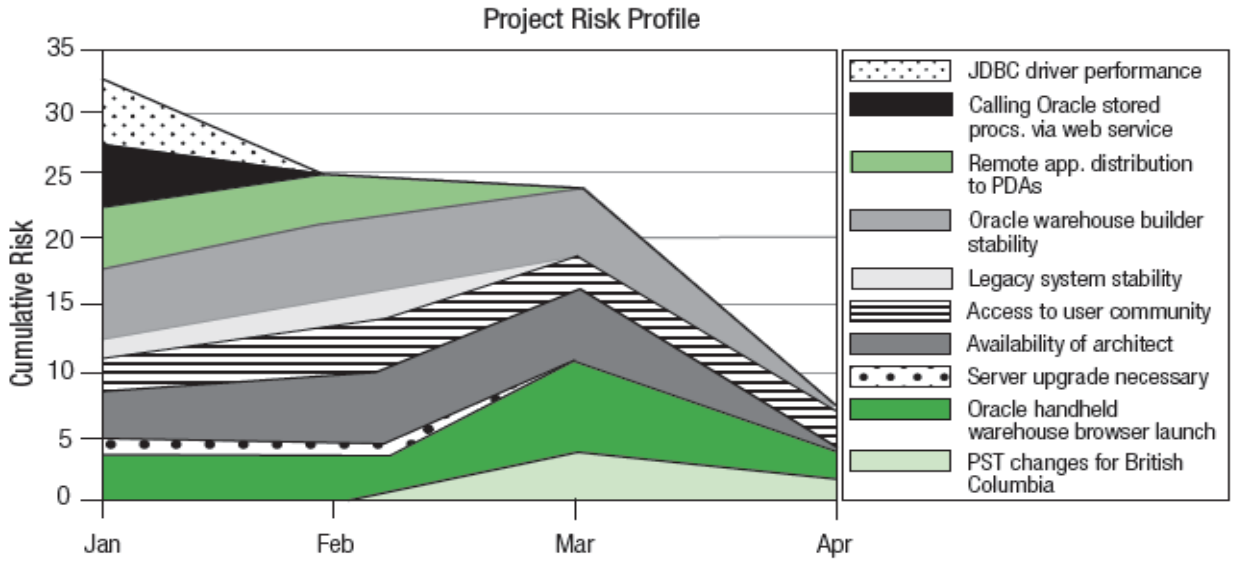
- Risks are assessed via two measures
  - **Risk probability**: A measure of how likely a risk is to occur
  - **Risk impact**: A measure of the consequence to the project should the risk actually occur
- Use Probability and Impact Matrix

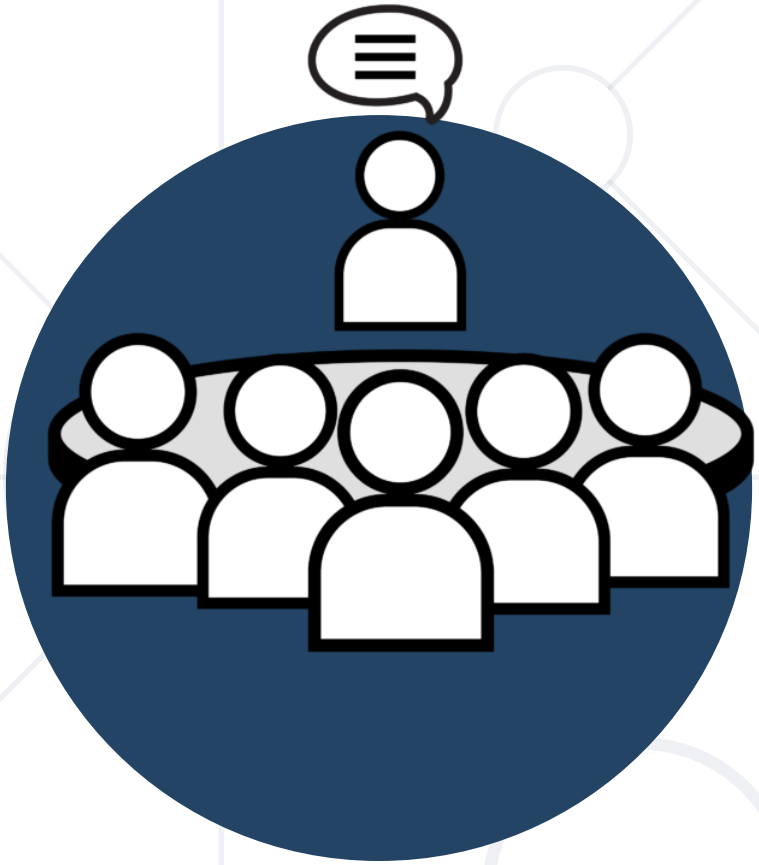


Risk Burndown Graph

- Stacked area graph of cumulative project risk severity
- Quickly inform stakeholders with an easy-to-interpret graph, whether the risks are moving in the right direction (downward), or if they are escalating

Risk burndown graph



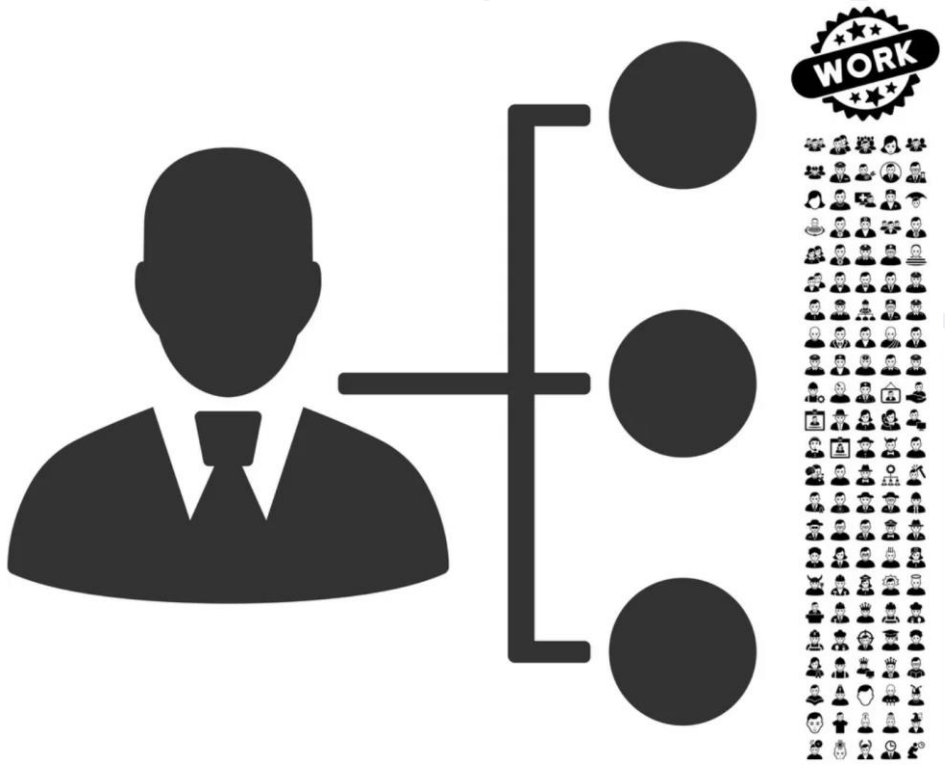


Agile Leadership

Statement

- Often heard the statement
  - "I don't even know what my job is anymore. I was promoted to the role of a manager because I was good at organizing and leading the work. Now that we're 'agile', I'm not supposed to be an expert on anything. I'm apparently just supposed to ask questions. But that's not much of a job."

Creating the Risk-Adjusted Backlog

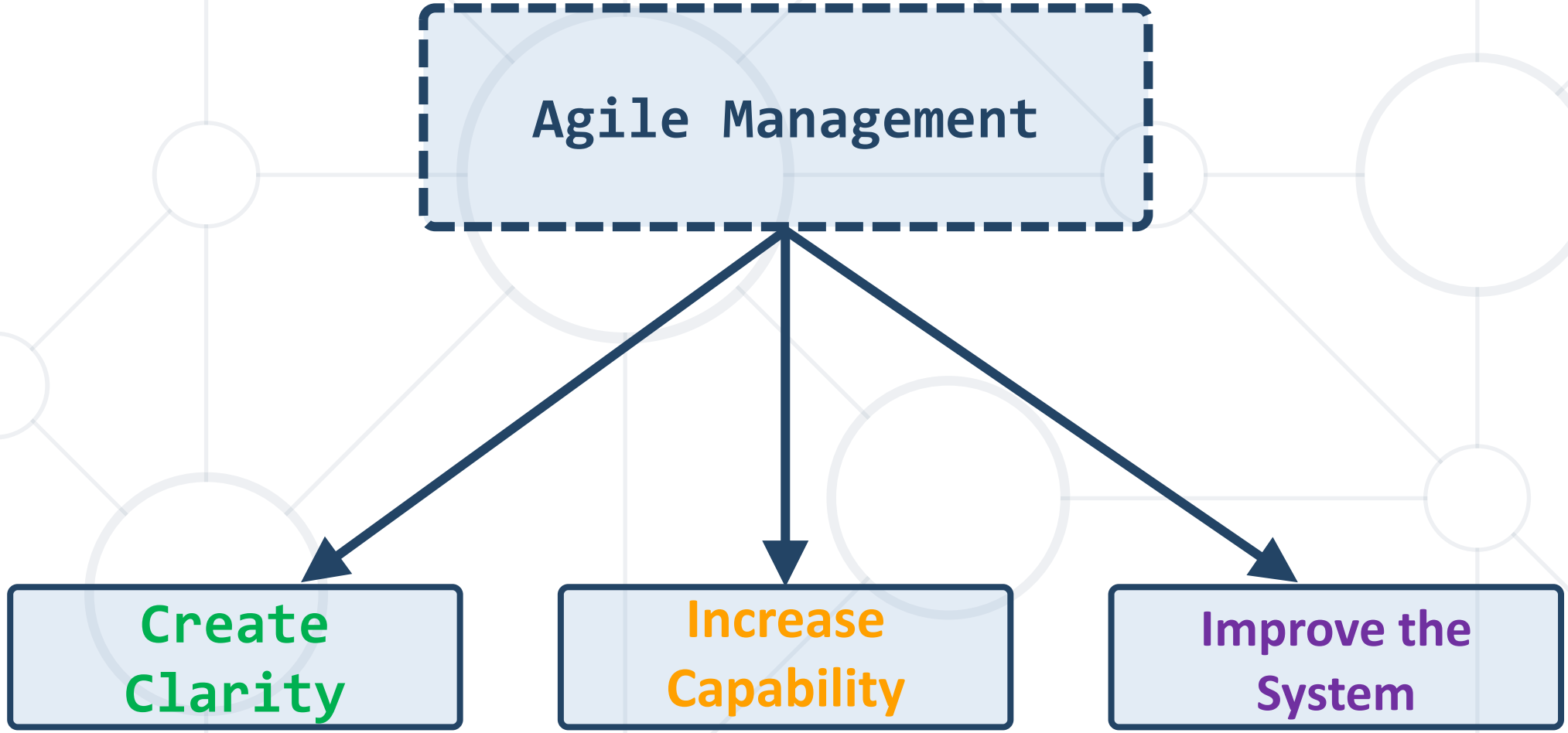


- Scrum, like most Agile methods, distributes much of the traditional manager's job across several other roles. This is an important part of the learning process
  - The Product Owner organizes and prioritizes work
  - The ScrumMaster facilitates team coordination and improvement
  - Developers (i.e., the people who build the product) do much of the planning and technical decision-making

"Best" Advice to Managers

- One manager directing the work of a whole team was inefficient
- Moving decisions about the work closer to the work itself makes sense
- Teams can leverage the experience, knowledge, and skill of everyone on the team
- But we shouldn't make the mistake of thinking there's nothing valuable for the manager to do now
- The manager in an agile context is freed to contribute in new ways

The Three Functions of Agile Management



SCRUM Myth or Fact

Software University

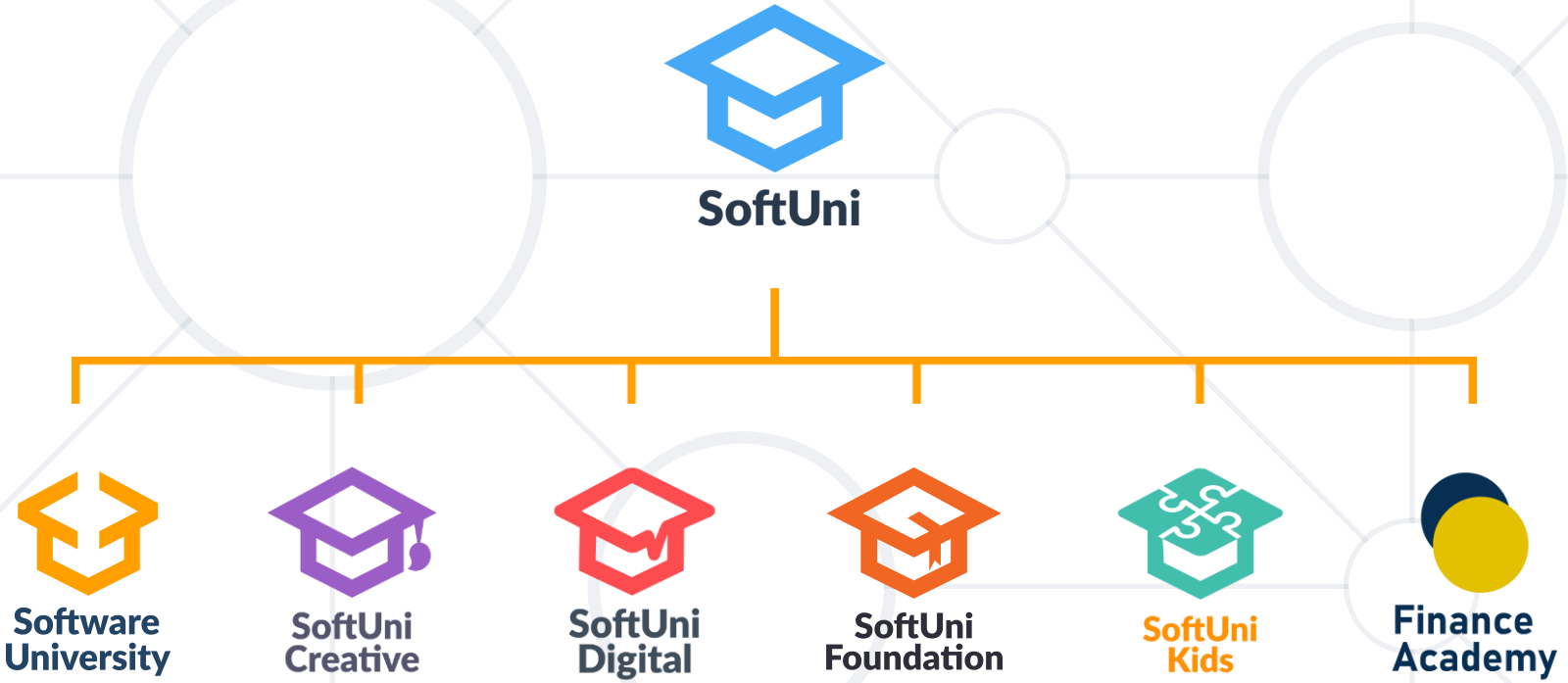
Homework and Additional Reading

- Agile Velocity
- Burndown charts
- Agile Risk Management
- Discussion topic – Should agile teams stay together?

Summary

- Define key metrics and monitor them
- Analyze results with the team
- Manage risk in the product backlog
- Act as a servant leader

Questions?



SoftUni Diamond Partners

Software University

SUPER HOSTING .BG

Coca-Cola HBC Bulgaria

Flutter™ International

INDEAVR
Serving the high achievers

AMBITIONED

DRAFT KINGS

SOFTWARE GROUP

BOSCH

Postbank

Решения за твоето утре

PHAR VISION

SmartIT

DXC TECHNOLOGY

createX

License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – https://about.softuni.bg/
- © Software University – https://softuni.bg

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg