INTRODUCTION TO SCRUM FOR THE TEAM

BOOTCAMP

Instructor: Barb Waters, MBA, PMP

Class will begin at 11:00 am Eastern Time

AGILE METHODOLOGIES

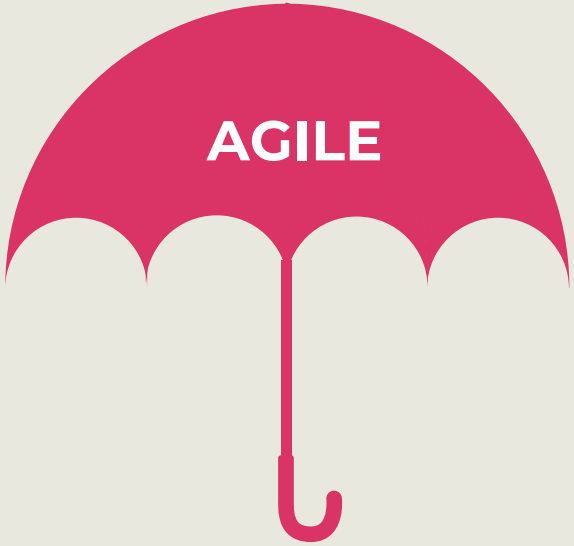
There are over a dozen agile methodologies

No single right way

Can be tailored once a team is experienced

Most common

- Scrum (really a framework)
- Lean product development
- Kanban
- Disciplined Agile
- Extreme Programming (XP)
- Feature-driven development (FDD)
- Dynamic Systems Development Method (DSDM)
- Crystal



SCRUM

Framework rather than a methodology

Scrum is Agile

Agile is *not only* Scrum

Employs various techniques

High-performing cross functional teams

Iterative, incremental approach

Iterations are known as “Sprints”

The term “Scrum” comes from rugby.

A scrum (short for scrummage) is a method of restarting play. The players pack closely together with their heads down and attempt to gain possession of the ball.

Scrum Pillars and Values

Pillars

Transparency

- Shared vision among stakeholders

Inspection

- Facts and observation define performance reporting

Adaptation

- Welcome change and quickly realign performance

Values

- Focus
- Courage
- Openness
- Commitment
- Respect

THE SCRUM TEAM

Includes:

- Developers
- Scrum Master
- Product Owner

Developers

Scrum Master

Product Owner

PROBLEM WITH DUAL ROLES

- Product owner promotes value delivery and new functionality
- Developers sets realistic expectations and perform the work
- Scrum Master protects the team and promotes governance

Can someone serve as both the Scrum Master and Product Owner?

What if one of these roles is not represented?

Simple tools

Tactile and visible

Collocated

33 ft or 10 m

Small team size

3-9 members

PRODUCT OWNER

- Develops product vision
- Serves as voice of the stakeholders (liaison)
- Collects and prioritizes requirements
- Controls budget
- Oversees return on investment
- Determines value of features
- Validates product quality

Product Owner

Stakeholders

DEVELOPERS

Self-managing

- Acts like an intelligent system
- Simple constraints cause the team to self-organize into a hyper-productive state

Builds the product increments during each Sprint

Estimates the work

Decides what can be done during each Sprint

Cross-functional

- Every necessary skillset is represented

SCRUM MASTER

Leader who serves the Scrum Team

- (not usually a developer)

Ensures adherence to Scrum framework

Facilitates meetings

Removes impediments (roadblocks, blockers)

Serves as a buffer to prevent interruptions

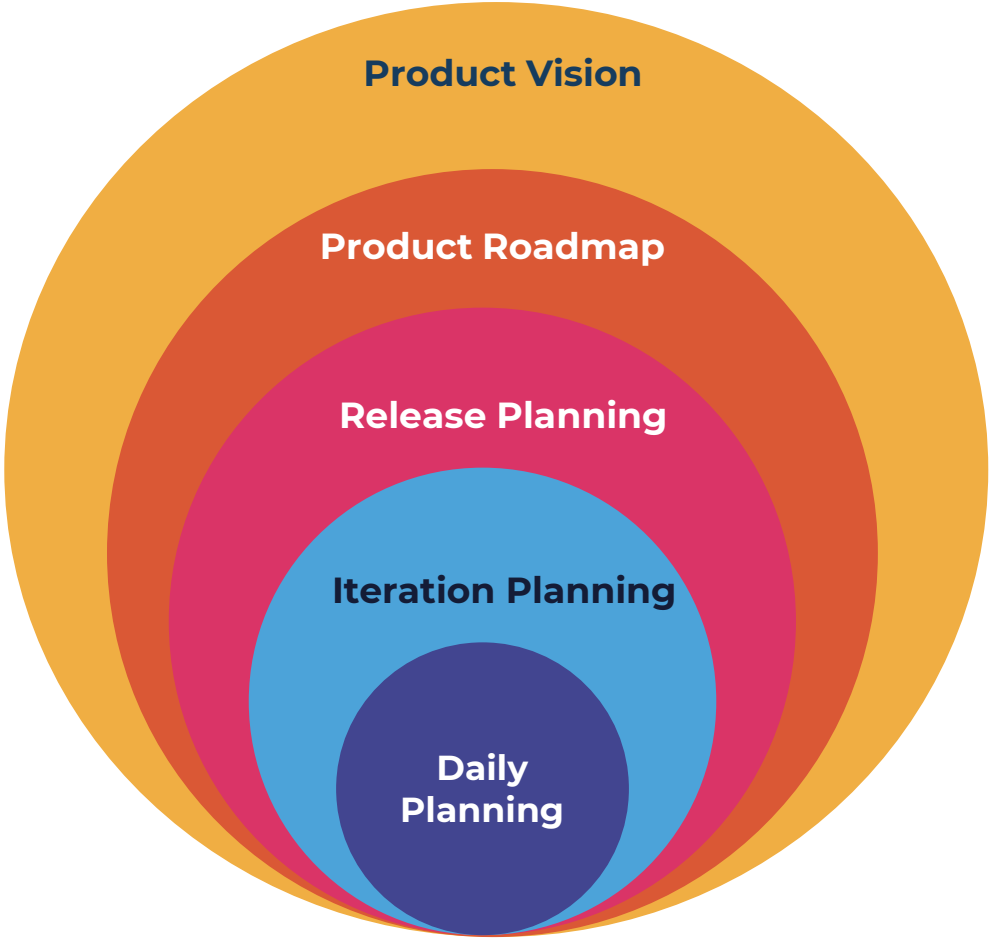
Provides essential resources

- Tools
- Resources
- Rewards
- Encouragement

Coaches team members

Assists Product Owner with managing backlog

Serves as Scrum “ambassador” to the organization



LEVEL OF DETAIL

CREATING THE **PRODUCT VISION**

- **Interview stakeholders**
- **Focus on how a product adds value**
- **Motivates Developers**

PRODUCT VISION

Why you're building a product

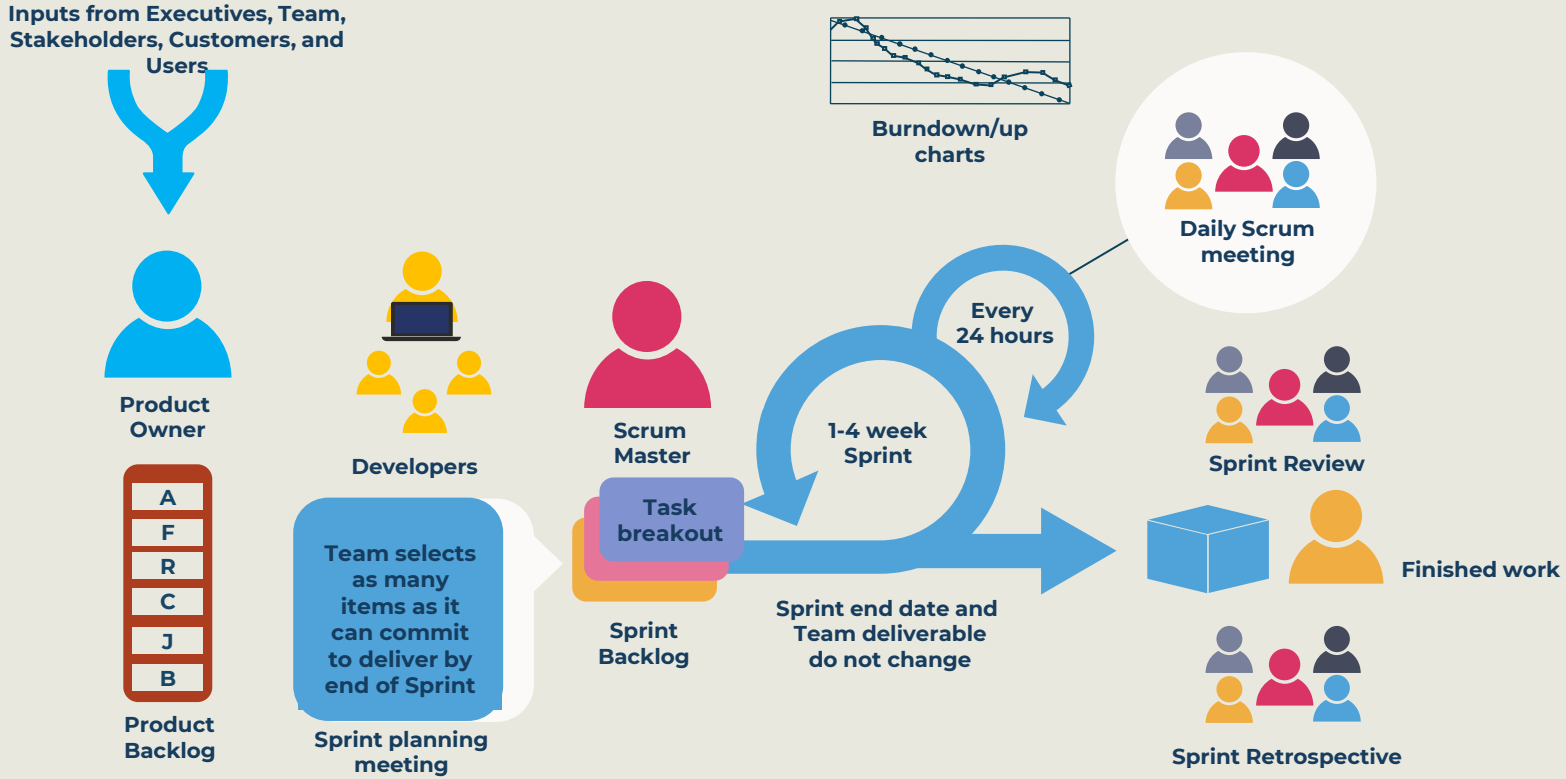
Benefits of product

Who you're building it for

Why you are positioned to develop it

Since scope is evolving it is important to share an understanding of what is being created

SCRUM FRAMEWORK



USER STORIES

Short, simple descriptions of a feature

Told from the user's perspective

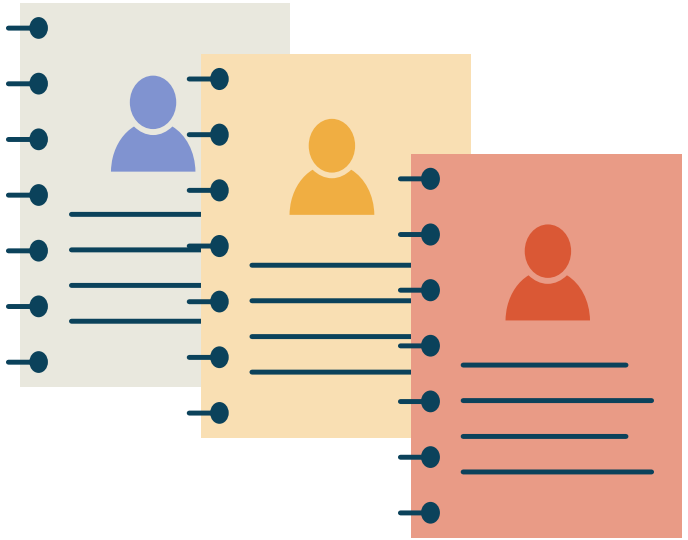
When large or complex, can be called "epics"

Sentence structure:

"As a *role*, I want *functionality*, so that *business benefit*."

Example:

"As a customer, I want my credit card information to be stored, so that I save time when checking out."



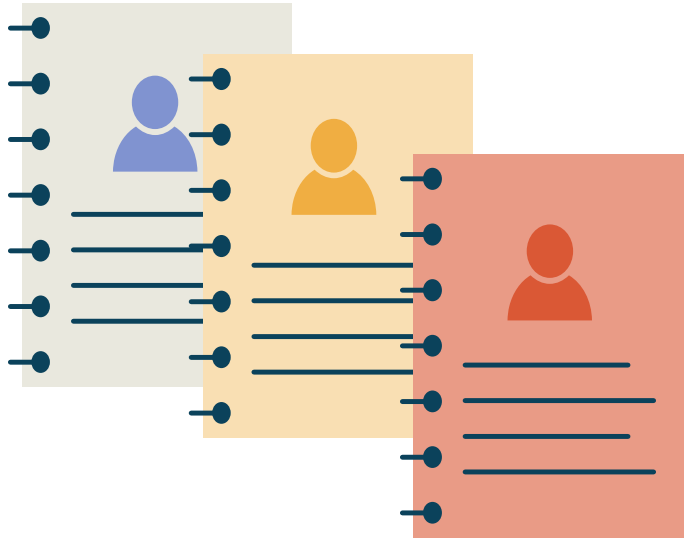
FORMATTING USER STORIES

Keep them simple

- **Gather feedback**
- **Experiment**
- **Use storyboards**
- **Use annotations**
- **Provide explanations**

INVEST criteria

- Independent
- Negotiable
- Valuable
- Estimable
- Small
- Testable



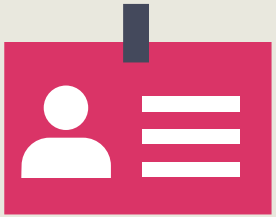
PERSONAS: BEST PRACTICES

- **Based on reality**
- **Specific**
- **Have defined goals**
- **Include negative personas**
- **The end user you are *not* designing for**
  - **Example: the user who always cancels after the free trial**
- **Ideally 3 or less, or the product may become confusing**

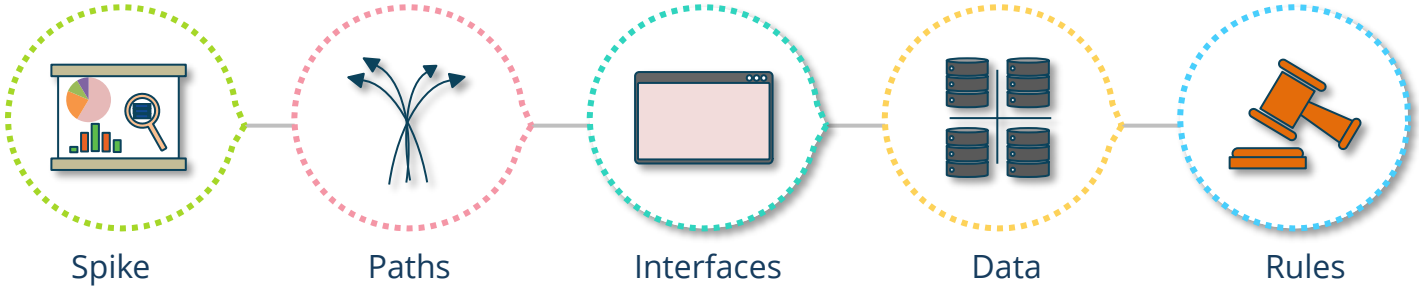
PERSONAS

- Quick reminder of stakeholder needs
- Composite
- Represents majority of actual users
- Not a replacement for stakeholders
- Focus on value and priorities

Mario is an employee of ABC Company. He must use his ID badge to access his work computer. For security reasons, the computer automatically logs users out after 5 minutes of inactivity. Mario would like to remain logged in while he is sitting at his desk.



SPIDR Approach to Splitting Stories



"Five Simple but Powerful Ways to Split User Stories" mountaingoatsoftware.com. Accessed April 2020.
https://www.mountaingoatsoftware.com/blog/five-simple-but-powerful-ways-to-split-user-stories

PRODUCT BACKLOG

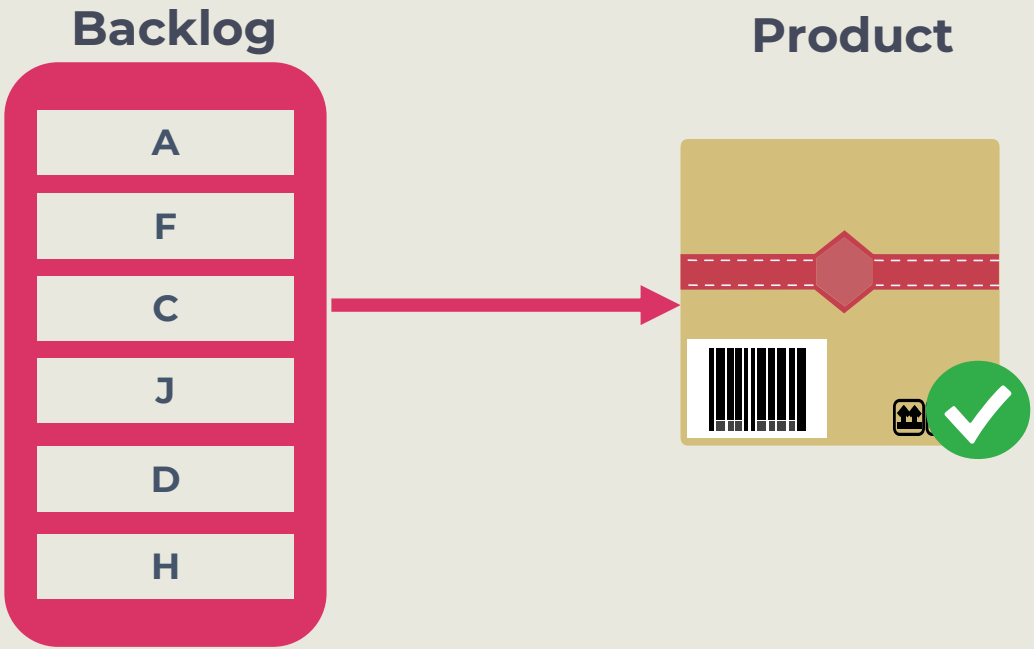
Prioritized list of everything that is needed in the product

Single source of product requirements

Always changing

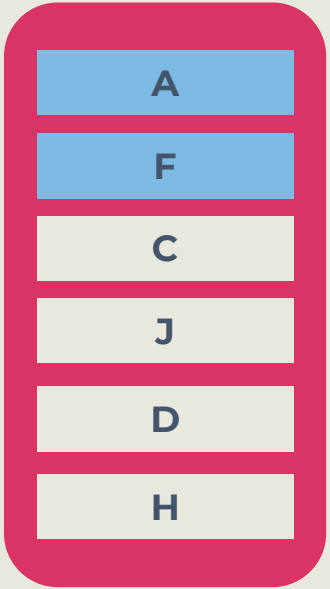
Items are added, dropped, and reprioritized based on value

The product is built incrementally based on work selected from the backlog



PRODUCT INCREMENT

- The result of the latest sprint
- Demo during sprint review
- Must meet the “definition of done” established during planning

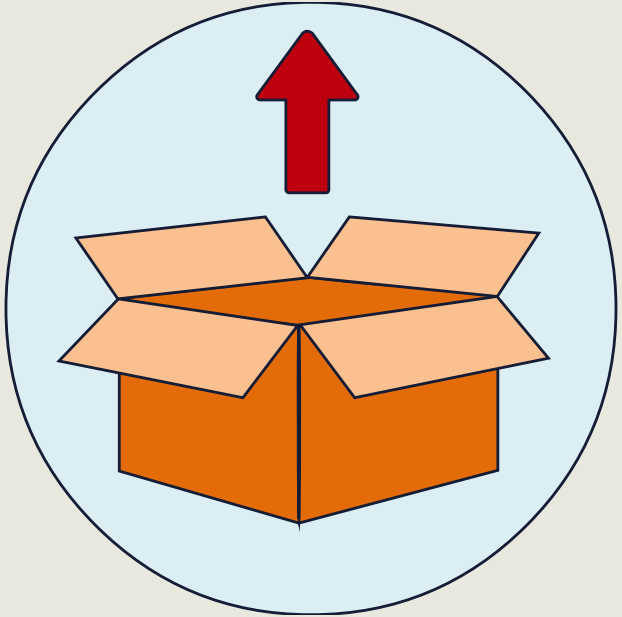


SMALL RELEASES

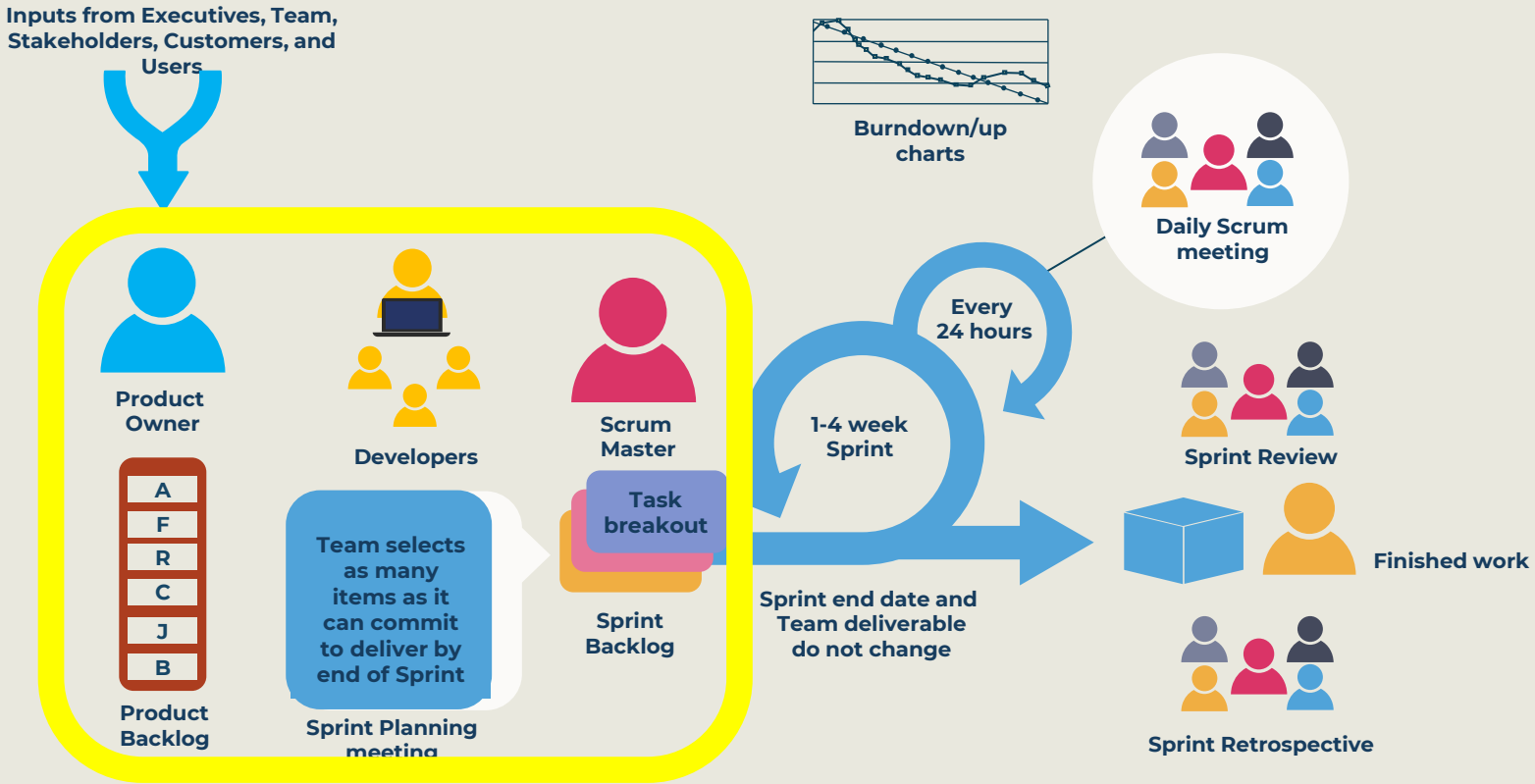
Demonstrates progress

Increases visibility to the customer

Smaller increments means rapid deployments



SPRINT PLANNING



SPRINT PLANNING

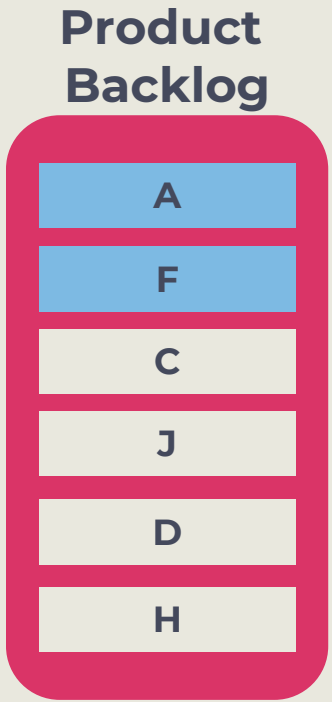
Participants

- Everyone

Actions

- Product Owner presents the updated backlog
- Developers estimates the work
- Team members pull from the product backlog
- Team commits to a set of deliverables for the Sprint
- Establish “definition of done”

Typically 2 hours per week of Sprint



SPRINT GOAL

- Start with high level goal for functionality needed
- Select product backlog items that align with the goal
- Some unrelated but priority backlog items may be included
- Developers will decide if the amount of work is realistic
- The goal is fixed, the Sprint backlog can evolve

Sprint Goal

“Why”

Product Backlog Items

“What”

Action Plan

“How”

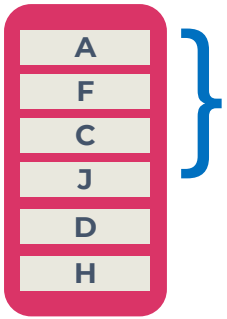
WHAT IS IN THE SPRINT BACKLOG?

DECOMPOSING THE WORK

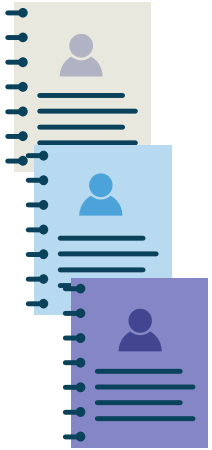
- Only decompose the work for this iteration.
- “Just in Time” (JIT)

Prioritized Features
“User Story Backlog”

Product backlog



User stories



Tasks

One feature may equal one or more user stories.

AGILE ESTIMATING TECHNIQUES

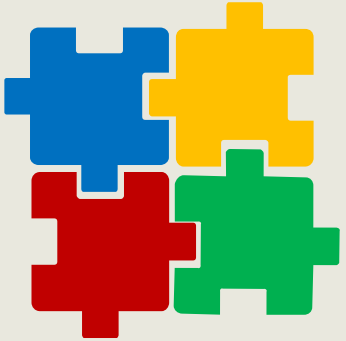
Story points

- Relative estimation
- Arbitrary measure
- Usually used by scrum teams
- Express effort required to implement a story
- 3 items taken into consideration: level of complexity, level of unknowns, effort to implement.

FACTORS IN ESTIMATING

Volume of work
How much effort?

Uncertainty
How risky is the work?



Complexity
How complex is the work?

AGILE **ESTIMATING TECHNIQUES**

Ideal days

- Alternative to story points
- Units of time rather than arbitrary measure
- How long to build, test, and release functionality with ideal conditions

AGILE **GAMES**

- Collaboration activities
- Brainstorming activities
- Retrospectives
- Drawing
- Storytelling

- Foster collaboration, communication, innovation
- Used to teach, demonstrate, improve
- Help model complex processes
- Facilitate issue resolution
- Drive good behaviors
- Overcome destructive behaviors

STORY **POINTS**

Relative sizing

- We aren't good at absolute estimate
- We are better at relative estimates

Not tied to days, hours, or dates

- Removes pressure or emotion

Based on quantity of work, not speed
Unique to a team

- Not comparable to the work of other teams
- Removes competition between teams

Reference for future estimates

Reserves and buffers are not necessary

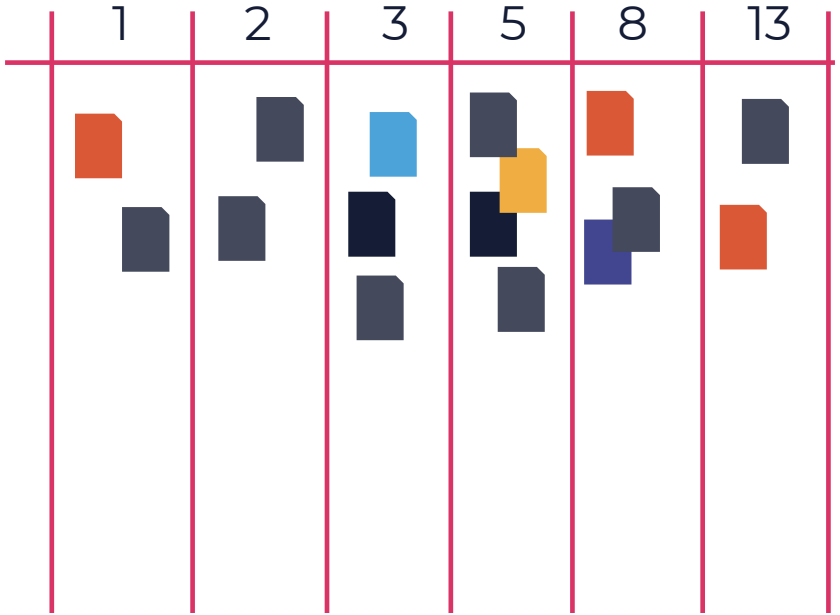
While story points is the most commonly used metric, teams may choose any unit to represent work.

PLANNING **POKER**

Uses Fibonacci sequence

Each player receives a deck of cards

- Facilitator reads a user story
- On the count of 3, everyone shows their estimate
- Purpose is to build consensus
- close to consensus, move on and round to higher number
- Scattered estimates, discuss and estimate again
- Estimates are approximates



AFFINITY ESTIMATES

Quick and easy method

Group User Stories

- Follows estimating
- Similar sizes are grouped together

Prevents story point inflation

RELATIVE SIZING

Quick and easy technique

Absolute value not considered

T-shirt sizing

Similar to Fibonacci

- Sizes instead of numbers

TEAM VELOCITY

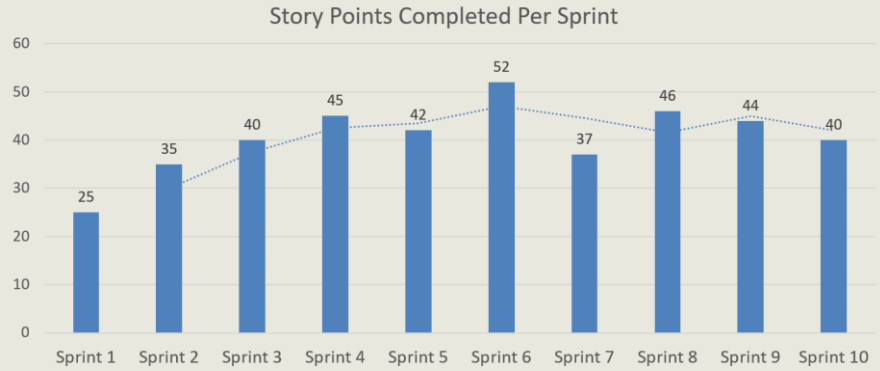
Velocity:

- Actual amount of development work completed per a certain amount of time or time-box
- Usually measured using a sprint as the time-box
- Used to estimate how quickly a certain amount of work can be completed
- Expressed as points (typically)
- Useful for forecasting

Use historical velocity data and take an average

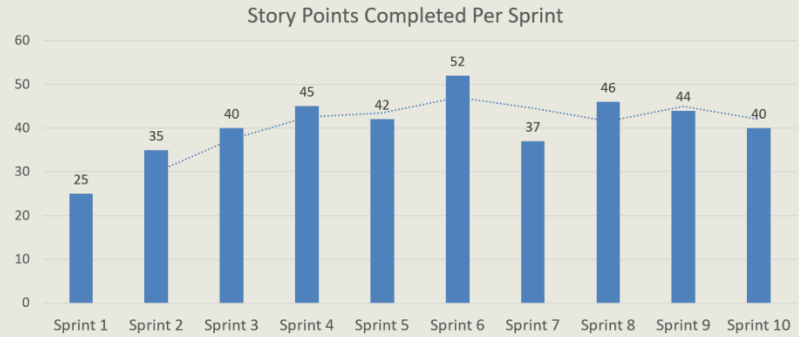
If first time:

- Historical value from other projects
- Run a few iterations for a baseline

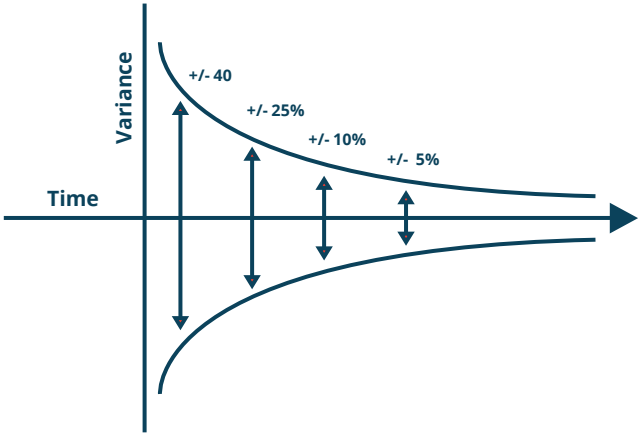


VELOCITY VS CAPACITY

Velocity is based on story points achieved historically



Capacity is based on team's availability to do the work



Barry Boehm's Estimate Convergence Graph

Schedule Buffers

Not recommended

- May appear that team is padding the schedule
- If used, buffer the number of sprints needed rather than individual task estimates

Exceptions

- When customer insists on a firm estimate
- Must account for contingencies
- When team velocity is undetermined
- When there are regular interruptions

Estimate in ranges

Acceptance Criteria and Definition of Done

Acceptance Criteria	Definition of Done
Specific (user story)	Applies to all work
Meet user needs	Team's shared vision
Functional features for users	Includes non-functional and quality requirements

Acceptance Criteria and Definition of Done

- agreed to as a team
- updated as new learnings come to light
- testable
- clear
- concise

PROBLEMS WITH TOO MUCH DETAIL

- Excessive planning
- Rework
- Anti-agile

PROBLEMS WITH NOT ENOUGH DETAIL

Team is not familiar with the user stories

Multiple questions about the work reduce productivity

Delays while waiting for answers or clarification from the Product Owner/stakeholders

The story is vulnerable to inflation

SPRINT ZERO

Some suggest that it…

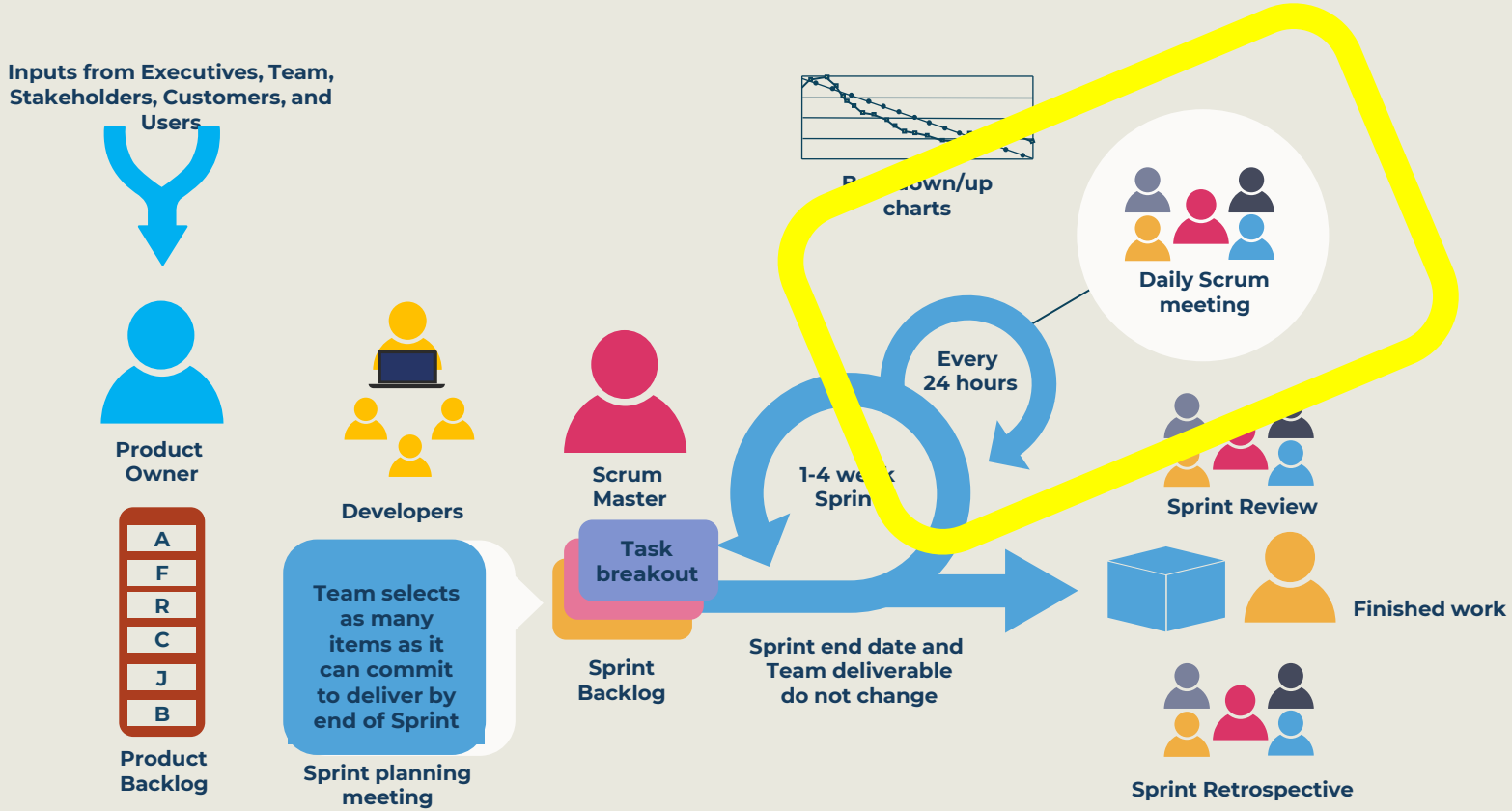
- allows a team to get a better idea of the work
- gives the team a chance to self-organize in order to perform better in the long run
- builds confidence within the team that they can handle the work
- provides clarity, which leads to the success of subsequent sprints
- opportunity to plan a framework for success

Scrum.org says…

- no to Sprint Zero or any specially named Sprints
- it is anti-Scrum because it creates no business value
- it sets a precedent that a sprint can have no potentially shippable product
- It is not favored by many founders of scrum
  - Alistair Cockburn – Agile Manifesto
  - Ken Schwaber – Scrum.org
  - Mike Cohn – Mountain Goat Software

Even during the first Sprint, the Developers should create at least one functional, potentially releasable product increment.

DAILY SCRUM OR STAND-UP



DAILY SCRUM OR STAND-UP

Occurs during the Sprint
“What did I do yesterday?”
“What will I do today?”
“What are my roadblocks or impediments?”

Alternate Questions
“Why is this Sprint valuable?”
“What can be done this Sprint?”
“How will the chosen work get done?”

Participants
- Developers
- Scrum Master
- Product Owner should be present

Typically 15 minutes or less
- Reserve off-topic subjects for a separate discussion

SCRUM OF SCRUMS

Used to scale Agile

- When teams are >12 members
- Each team selects an ambassador

Report on

- Completions
- Next steps
- Impediments

Resolve coordination challenges between teams

Scrum of Scrums has its own backlog of these items

May meet a few times per week

Team A

Team B

Team C

Ambassador

Ambassador

Ambassador

There is also a Scrum of Scrum of Scrums!

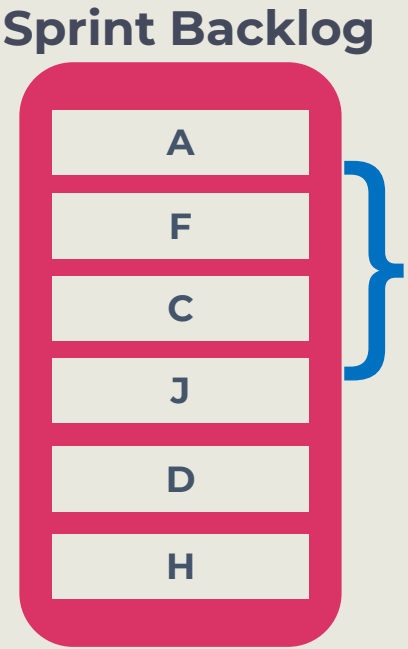
SPRINT BACKLOG

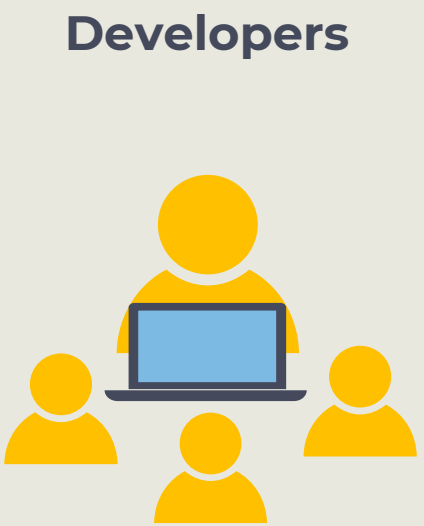
Belongs to Developers

Subset of the product backlog

Goal for the current Sprint

Highly detailed and visible



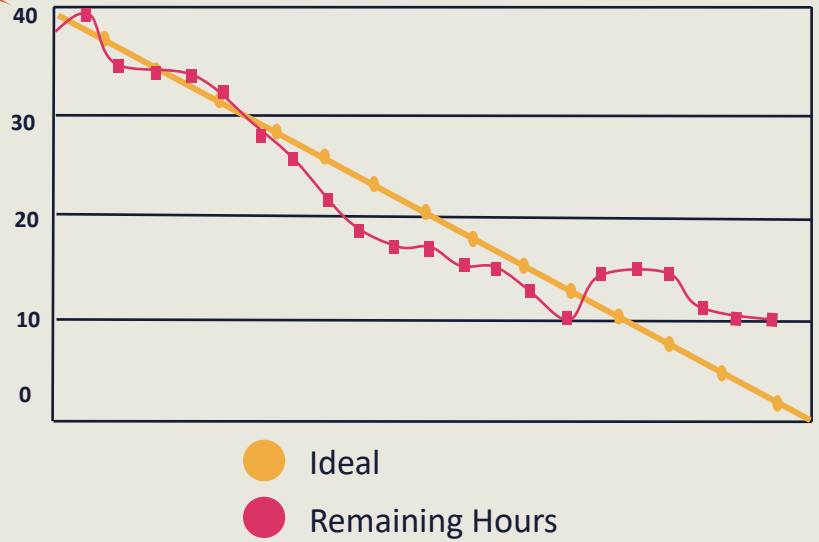


PERFORMANCE TRACKING: BURN CHARTS

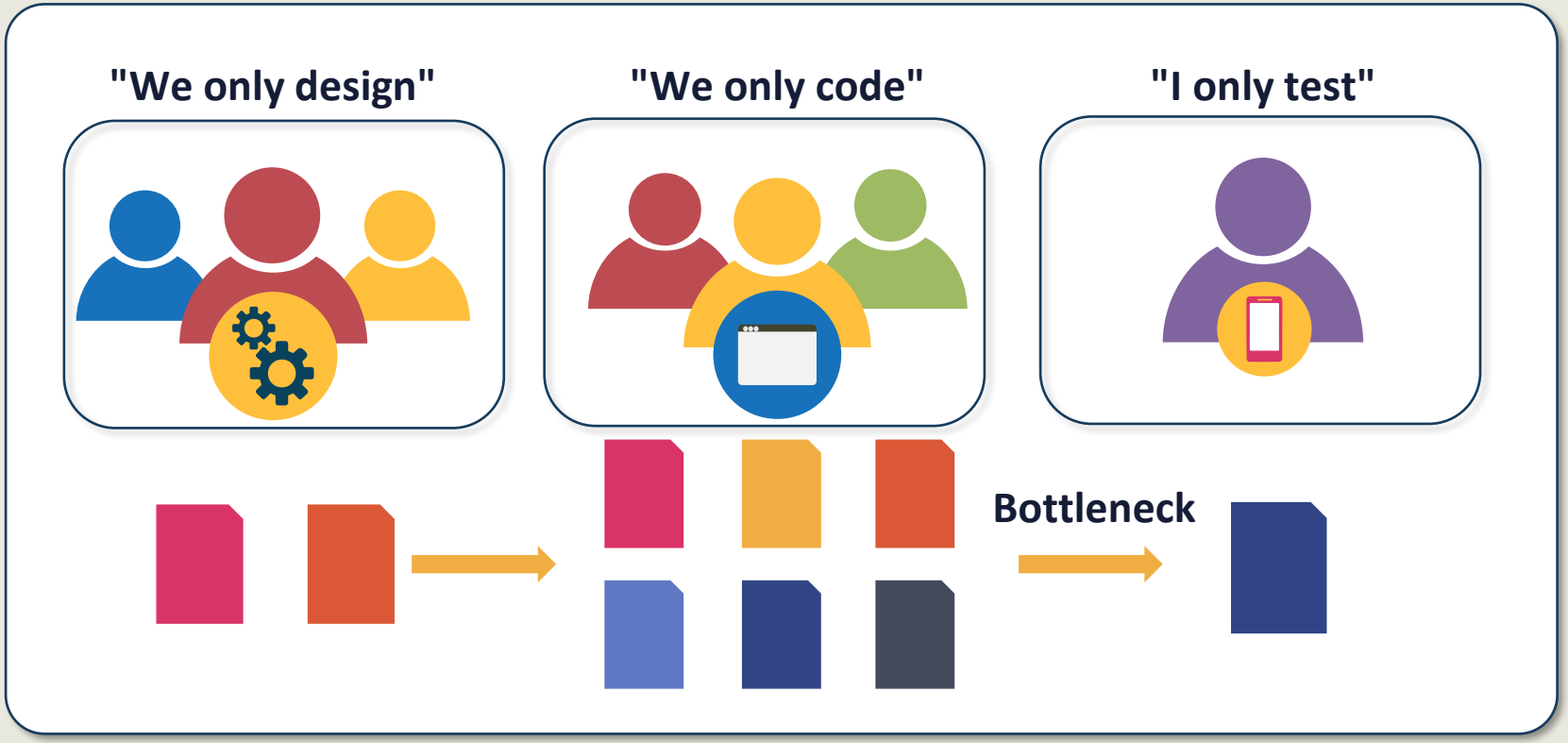
Burndown and burnup charts

“Information Radiators”

- Generic term for a highly visible information display
- Graphs, charts, data dashboard
- Communication tool
- Shows remaining work for the Sprint
- Trend line shows the running average, and what will likely happen if progress continues at this rate



SPECIALISTS (I-SHAPED)



GENERALIZED SPECIALISTS (T-SHAPED)

"We only design"

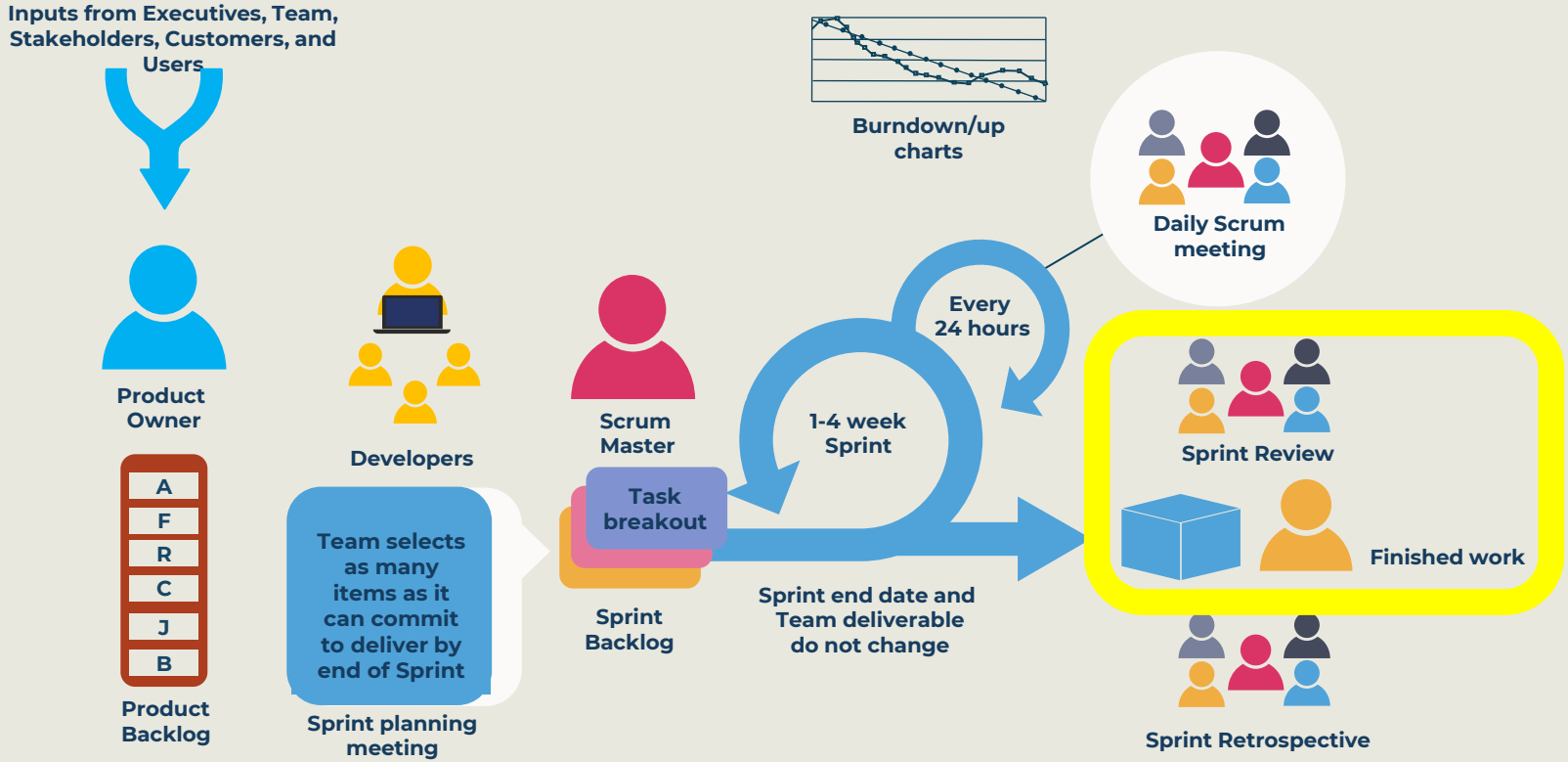
"We can code *and* test"

"I can code *and* test "

Ready for release

TASK SWITCHING

SPRINT REVIEW



SPRINT REVIEW

Occurs at the end of a Sprint

Participants

- Developers
- Scrum Master
- Product Owner
- Stakeholders (invited by Product Owner)

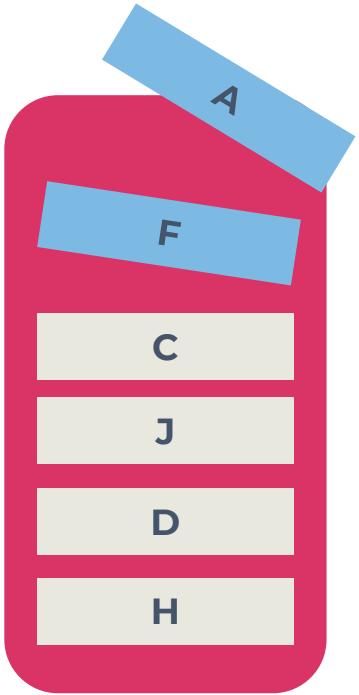
Developers demos the product to Product Owner and possibly stakeholders

Product Owner inspects deliverables

Elicit feedback and foster collaboration

Team and Product Owner adapt product backlog if necessary

Typically 1 hour per week of Sprint



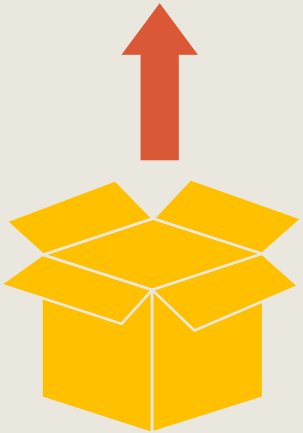
PRODUCT INCREMENT

The result of the latest Sprint

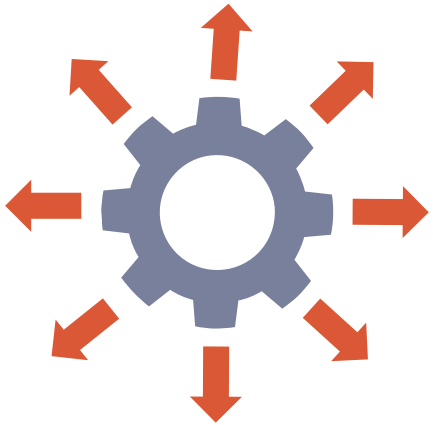
Demo during Sprint review

Must meet the “definition of done” established during planning

PRODUCT SCOPE EVOLVES



With each release, product becomes more robust



SPRINT REVIEW BEST PRACTICES

- Put the item in the hands of the customers
- Observe their interaction with the product
- Ask probing questions
- Maintain a friendly and informal environment
- Thank the stakeholders for their time and feedback

Eliciting Feedback

Does it solve a problem?

Does it serve a purpose?

Does it add value?

“POTENTIALLY” RELEASABLE PRODUCT INCREMENT

Complete

Acceptance Criteria

Definition of Done

Tested

Avoid escaped defects

Deliverable Now

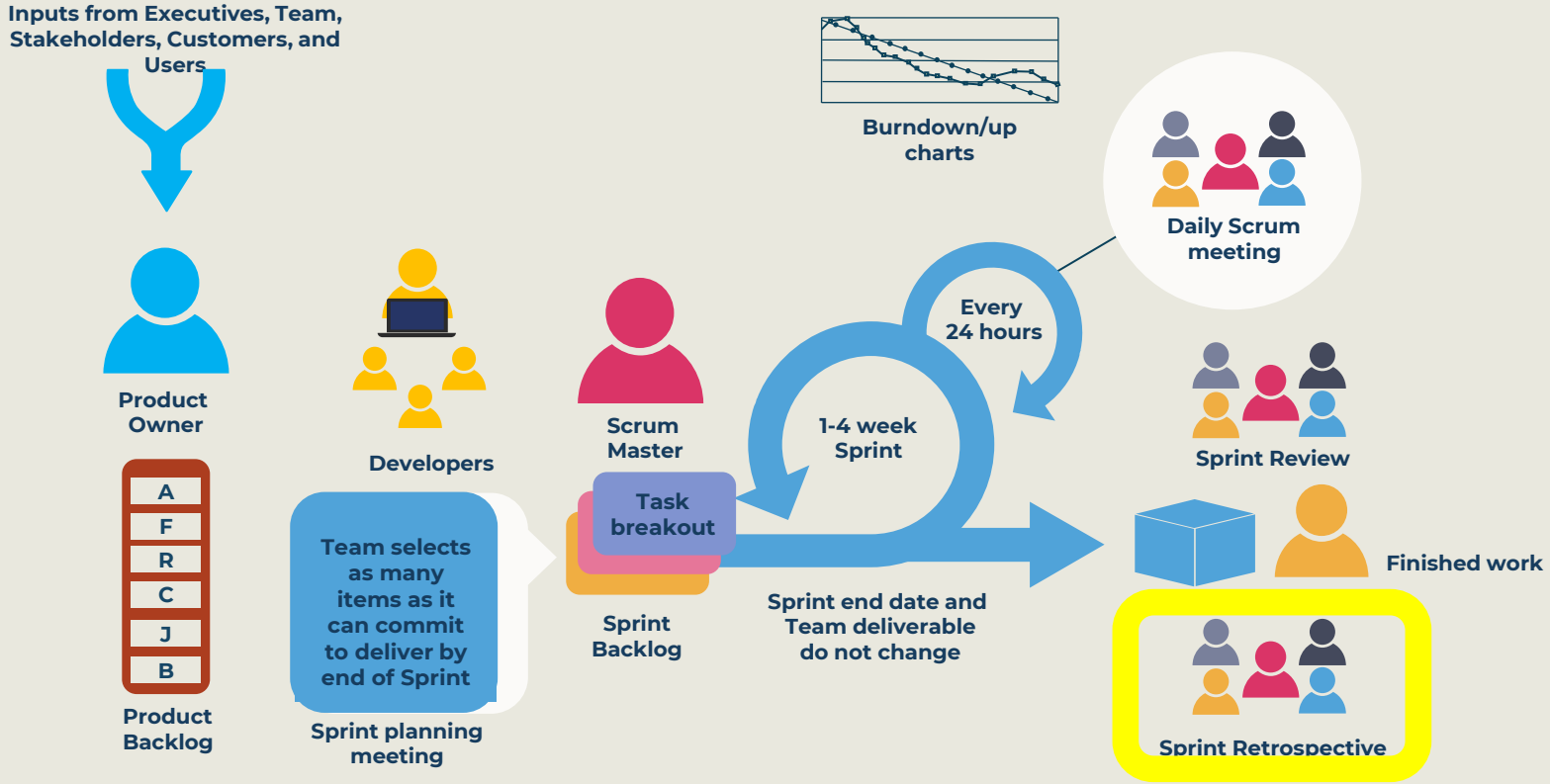
No remaining work, including user instructions, etc.

Reasons the Product Owner might delay release

Costs associated with release

- **Marketing expenses**
- **Additional customer support**
- **Customer’s willingness to adapt**
- **Inadequate Definition of Done**

SPRINT RETROSPECTIVE



SPRINT RETROSPECTIVE

Participants

- The Scrum Team
  - Developers
  - Scrum Master
  - Product Owner

Evaluate the last Sprint

- People
- Processes
- Tools

Plan improvements for next iteration

Typically .75 hours per week of Sprint

SPRINT RETROSPECTIVE TOPICS

What went well

Best practices

What could be better

Areas of opportunity

Next sprint

Team commitments

LESSONS LEARNED

Small steps for incremental improvements

DO WE REALLY NEED THE RETROSPECTIVE?

- The Scrum Team is high-performing
- Everyone is getting along well
- The Retrospective takes time
- Can we skip it?

MAINTAINING THE SCRUM FRAMEWORK

- Meeting is already on the calendar
- There is an appropriate time for discussion
- Allows the Developers to focus on work
- Retrospective is part of inspecting and adapting

MAD, SAD, GLAD EXERCISE

Mad

Fix for the future

Sad

It's in the past

Glad

Keep doing this

FORCE FIELD ANALYSIS

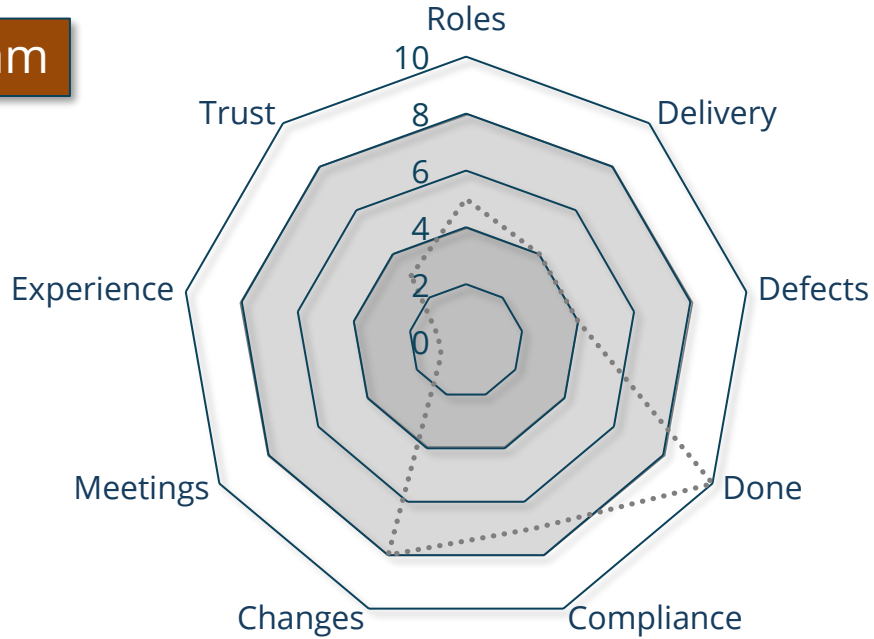
Mountain climbing

Car racing

Sailing a boat

Team Radar

Team



Product

Process

SAFE ENVIRONMENT

- Ask for help
- Share problems
- Admit mistakes
- Understand barriers to lessons learned
- Kaizen
  - Small, achievable steps
  - Identify and remove impediments
  - SMART goals

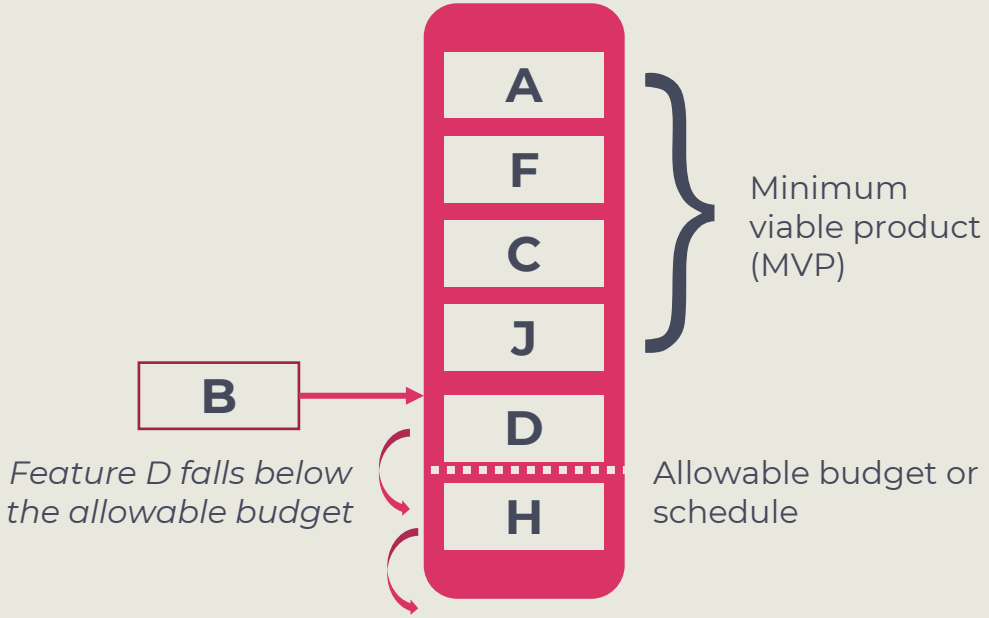
BACKLOG REFINEMENT

New features can be inserted into the priority list

- Developers estimates effort
- Product owner decides priority

All work should be included

- Bug fixes
- Changes
- Single, prioritized list

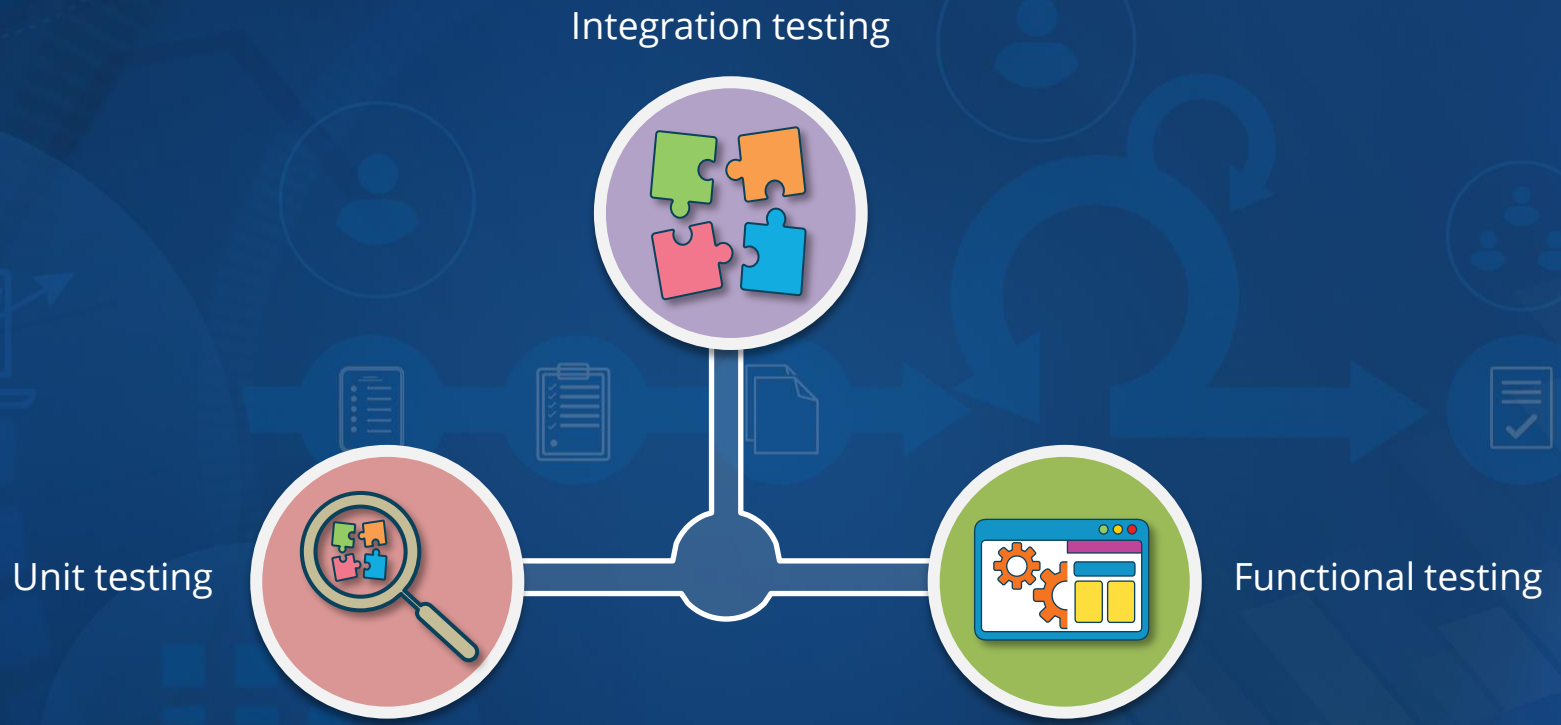


TIPS FOR A REALISTIC PRODUCT BACKLOG

- Set a target of 100 items
- Drop bottom-dwellers
- Consider a separate archive

Product Testing and Scrum

Common Types of Tests



Regression Tests

Detect new bugs

Test-driven Development

Test fails

Test passes

Productivity

As much as possible

Quality

As correct as possible

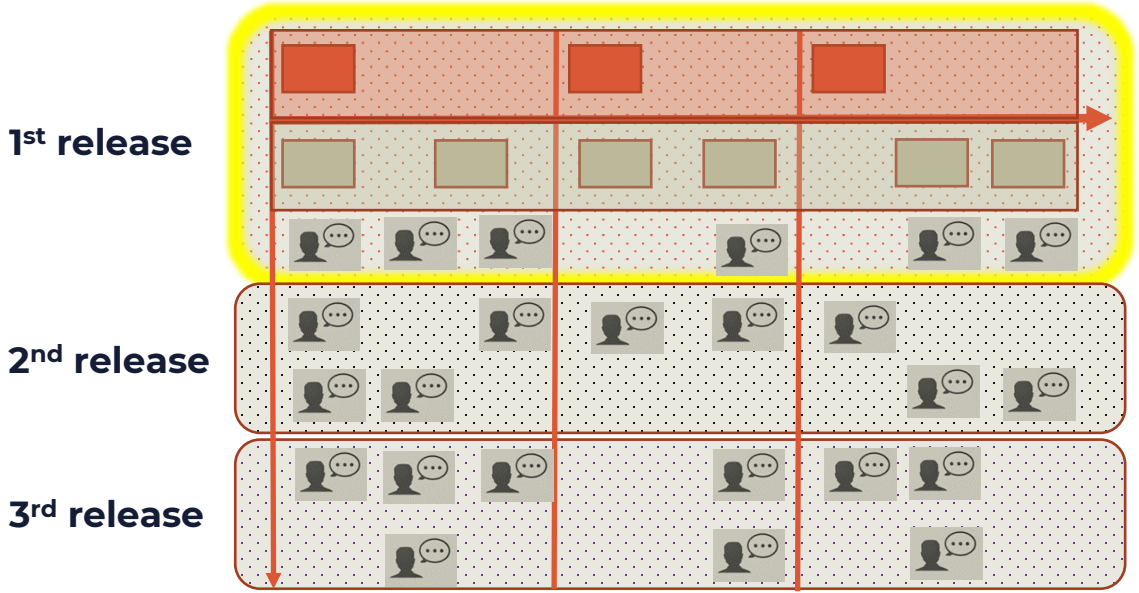
MINIMUM VIABLE PRODUCT

- Customer Journey
- Story Map
- End-to-end functionality
- Example: video streaming service

Log in | Perform search | Select a movie | Make payment | Watch movie

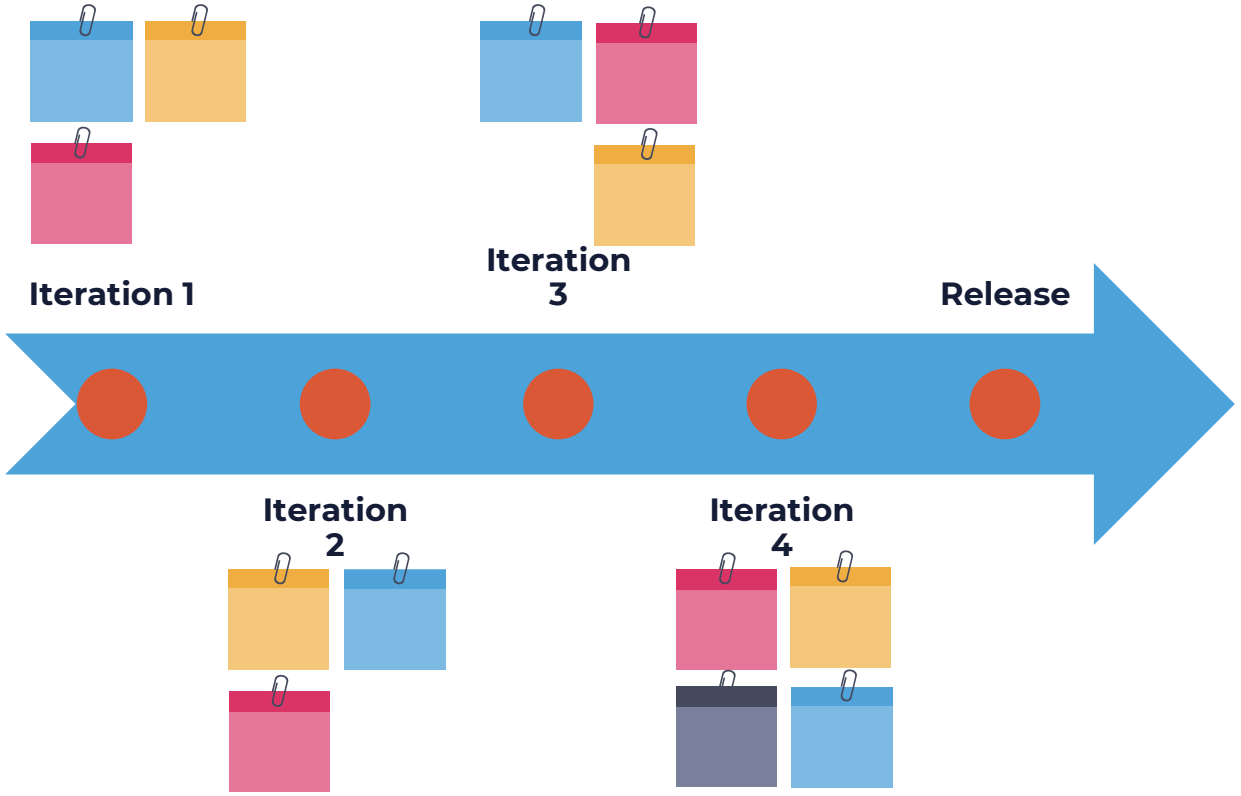
PRODUCT **ROADMAP**

- Story map with timing of deliverables
- Considers priorities against Developers velocity
- Subject to change as backlog is refined
- With each release the product becomes more robust



SAMPLE

RELEASE PLAN



DAILY BOOTCAMP SURVEY

Please share your thoughts.

At the end of each Bootcamp session please let us know how we are doing. Your feedback helps us to offer the best possible Bootcamp experience.

Thank you for attending this Bootcamp!