

Deliver faster with
AGILE

SARA COPE

What is Agile?

Agile vs. Waterfall

Agile methodologies

Agile life cycle phases

Overview

The Agile Manifesto

"...we have come to value:

Individuals and interactions over processes and tools,
Working software over comprehensive documentation,
Customer collaboration over contract negotiation,
Responding to change over following a plan,

That is, while there is value in the items on the right, we value the items on the left more." — [The Agile Manifesto](#)

Agile Principles

1. Satisfy the customer through early and continuous delivery of valuable software
2. Welcome changing requirements
3. Deliver working software frequently
4. Daily collaboration between business people and developers throughout the project
5. Build projects around motivated individuals
6. Face-to-face conversation when possible
7. Working software is the primary measure of progress
8. Promote sustainable development
9. Continuous attention to technical excellence and good design
10. Simplicity--the art of maximizing the amount of work not done--is essential
11. The best architectures, requirements, and designs emerge from self-organizing teams
12. Reflect and adjust regularly

Waterfall

Predictive Process

Agile

Adaptive Process

Fixed

Requirements

Resources

Time



Estimated

Resources

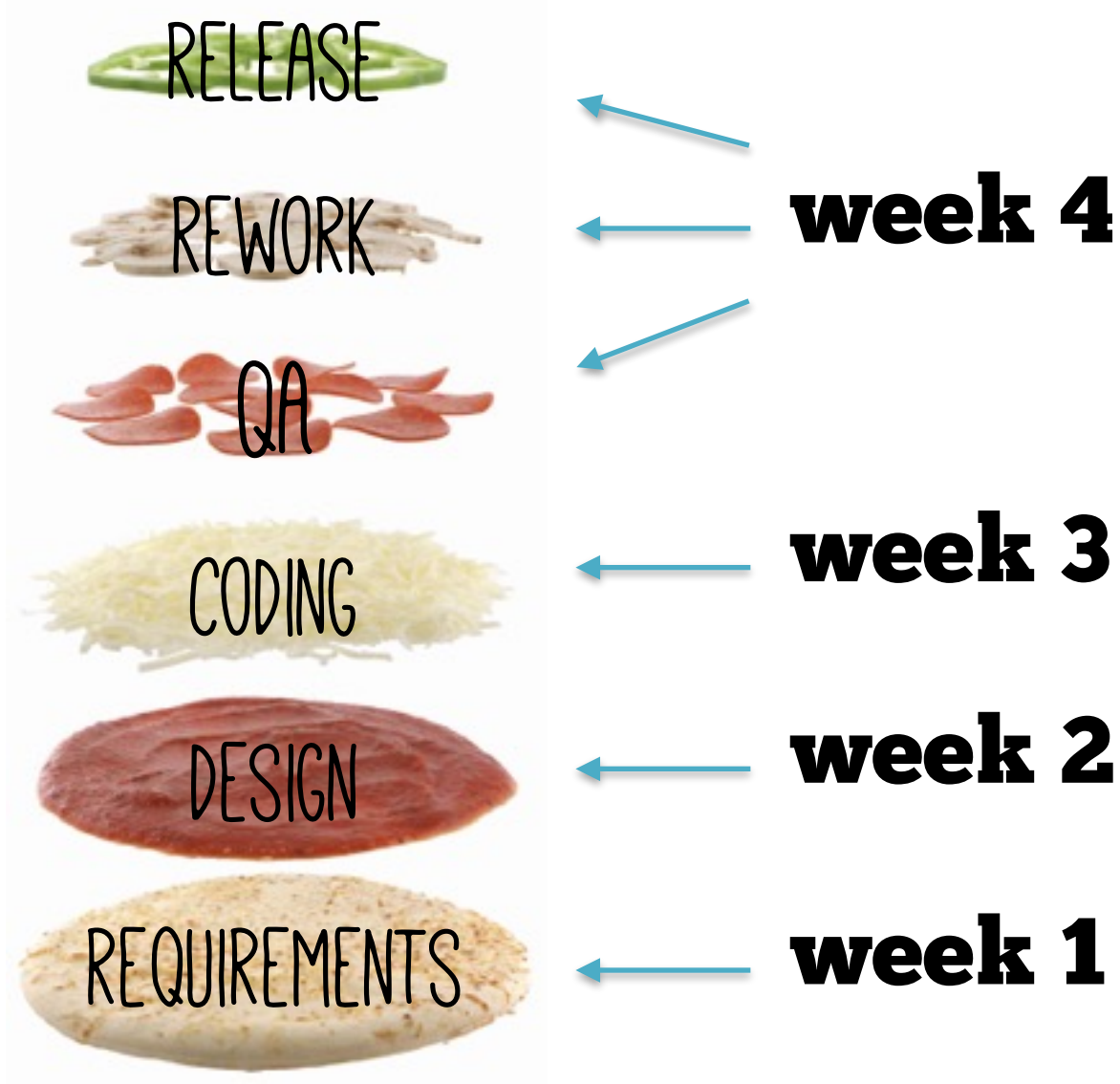
Time

Features

The plan creates cost and schedule estimates

The vision creates feature estimates

Waterfall Development



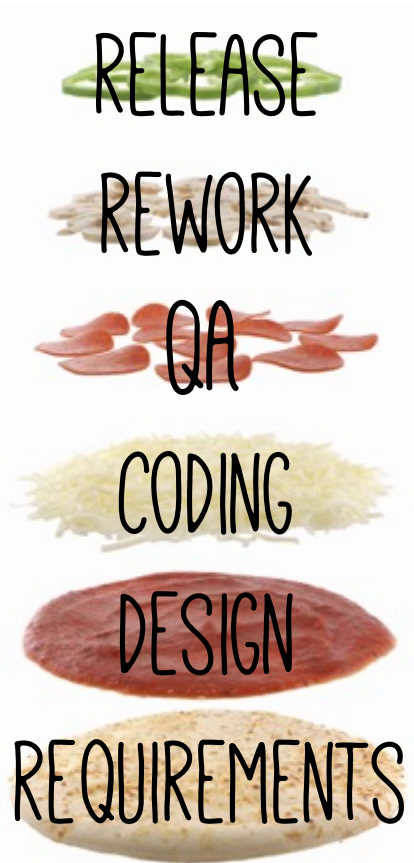
Agile Development

week 1

week 2

week 3

week 4

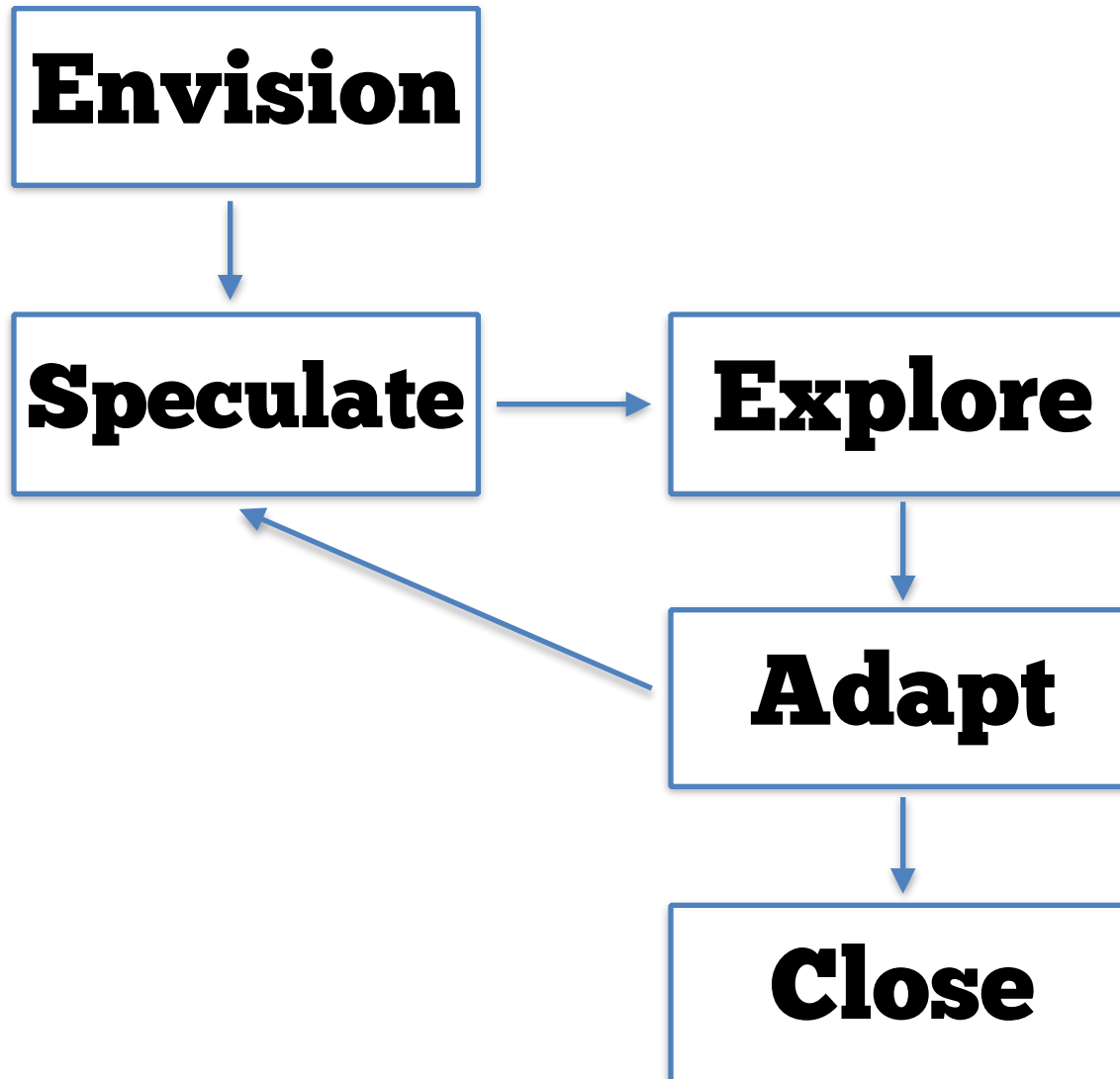


Agile Myths

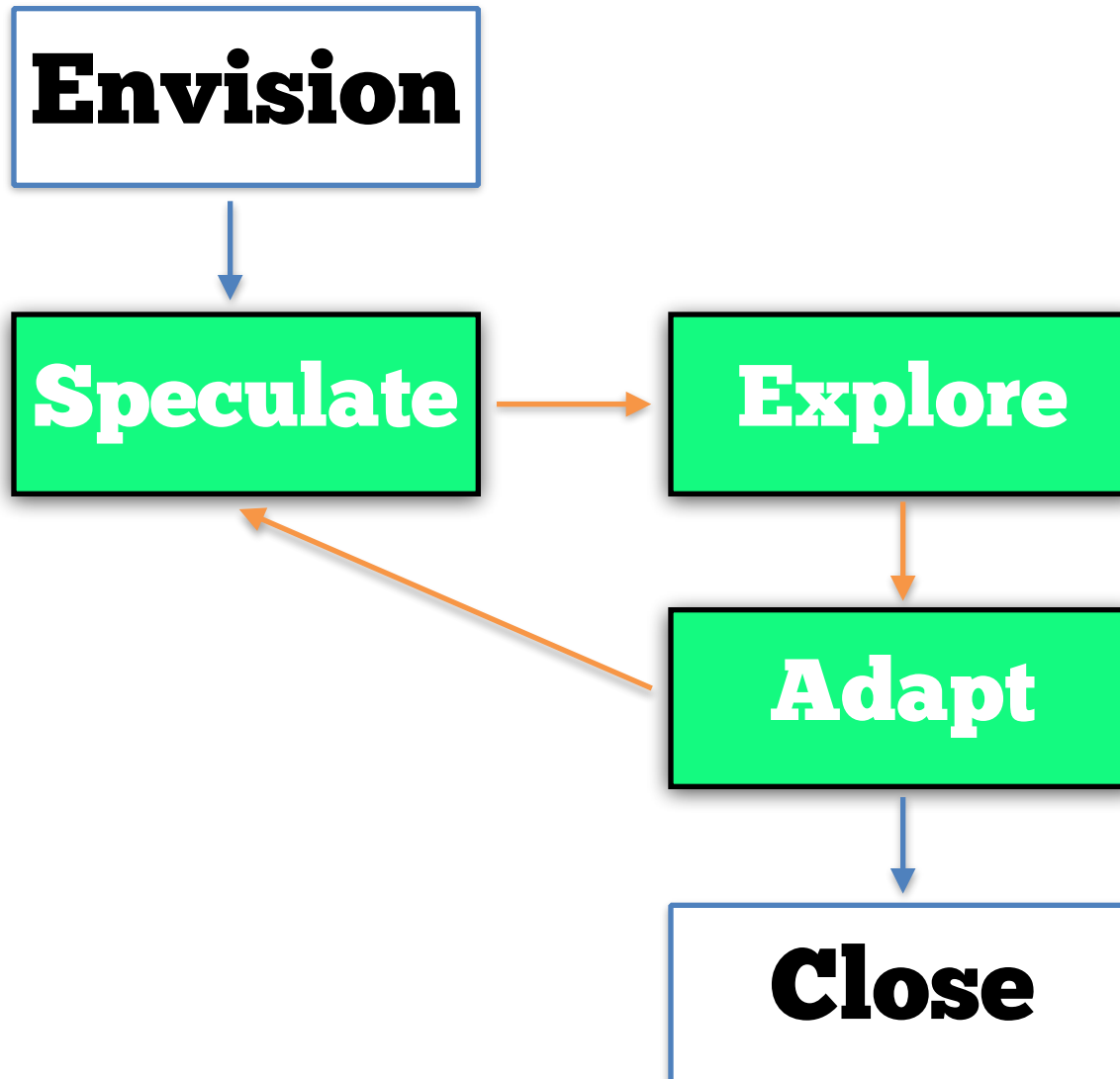


© Scott Adams, Inc./Dist. by UFS, Inc.

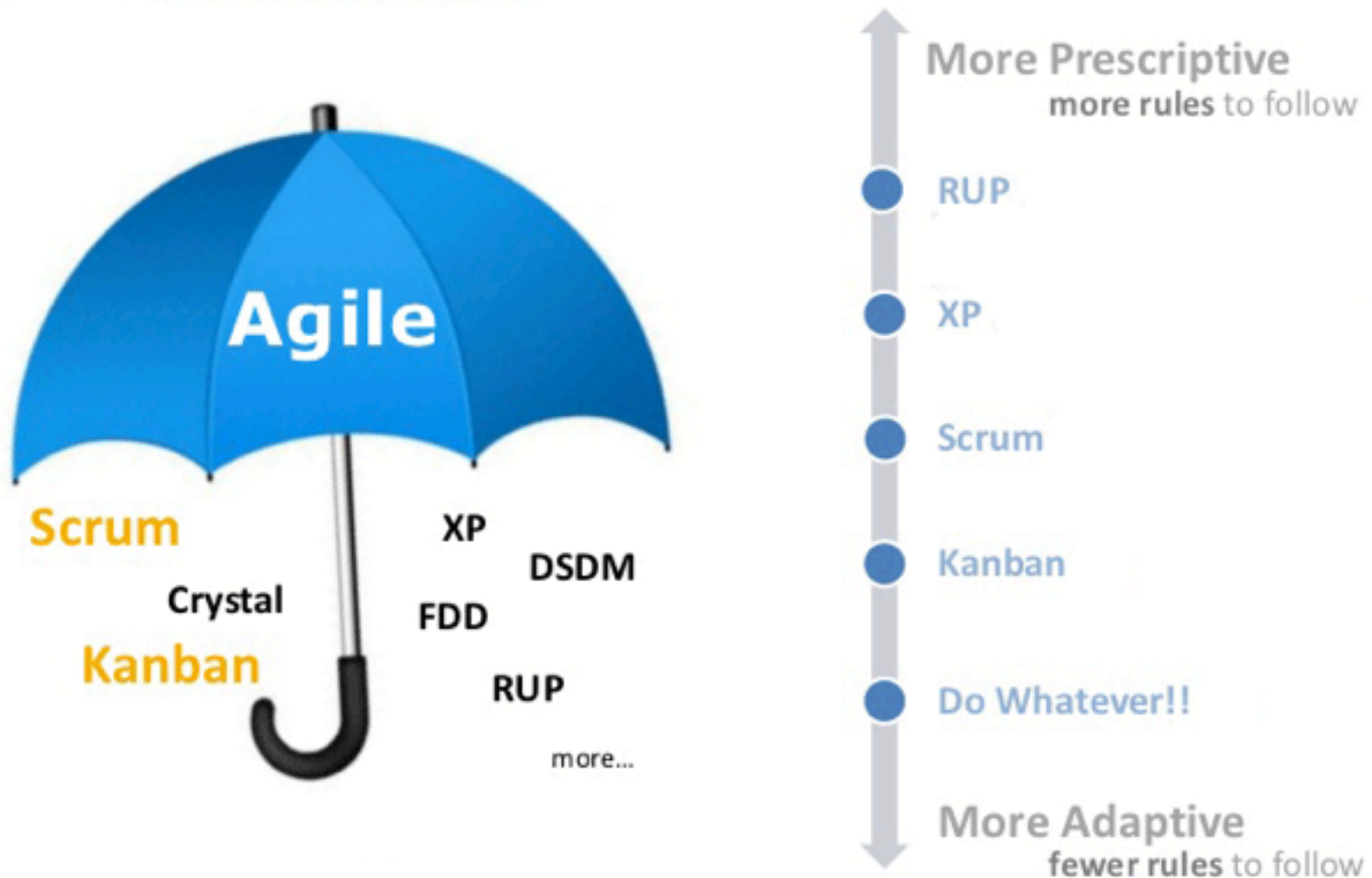
Agile Life Cycle



Agile Life Cycle



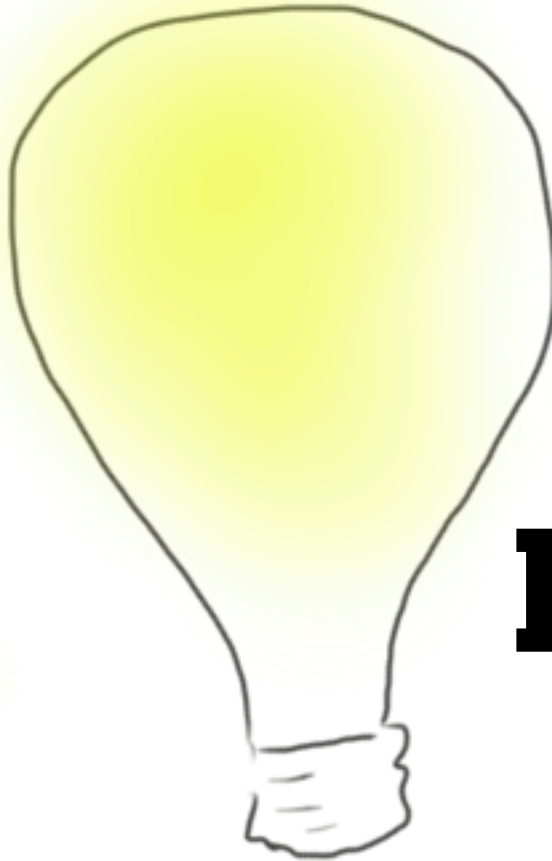
Agile Methodologies



Envision

WHAT

WHO



HOW

Product Vision

Project Scope

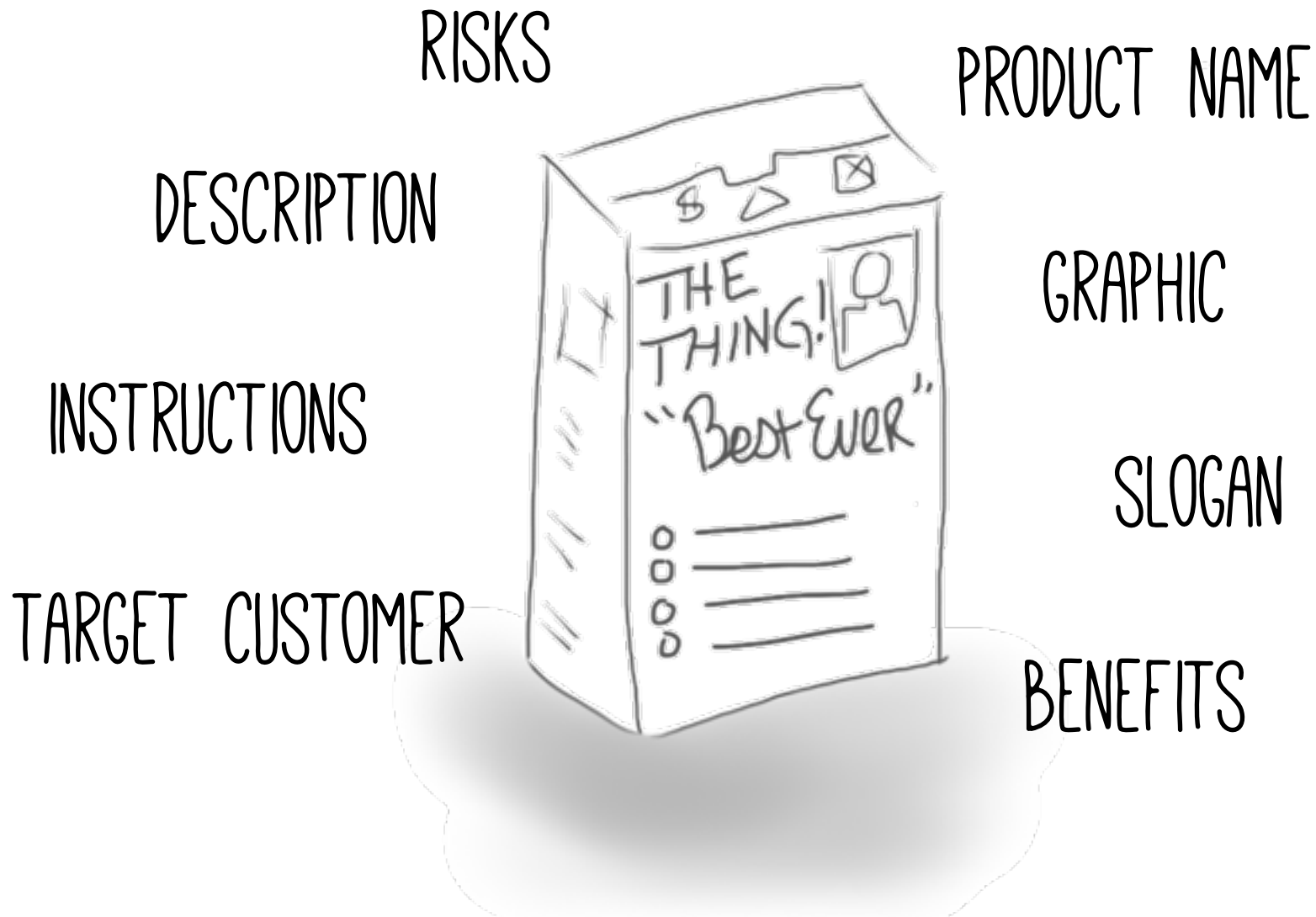
Key Features

Constraints



The
What

Vision Box



User

Product Owner

Project Manager

Development Team

Other Stakeholders



The
Who



Small, 5-9 people

Cross-functionality diverse

Domain experience

Focused and committed to the teams goal

Team players who can check their ego
at the door

Ground Rules

Tools



*The
How*

Daily scrum meeting at
10am

Being on-time for
meetings means getting
there at least 1 minute
early

No talking over team
members

Stay out of email

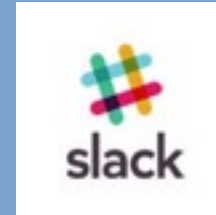
Ground Rules

Project hub

Team collaboration

Code version control
and hosting

Task management

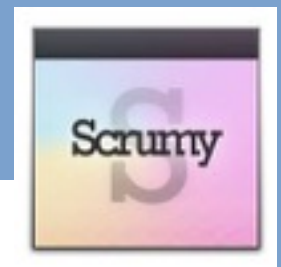
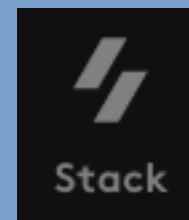
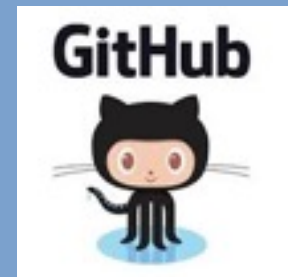


git

Team



Tools

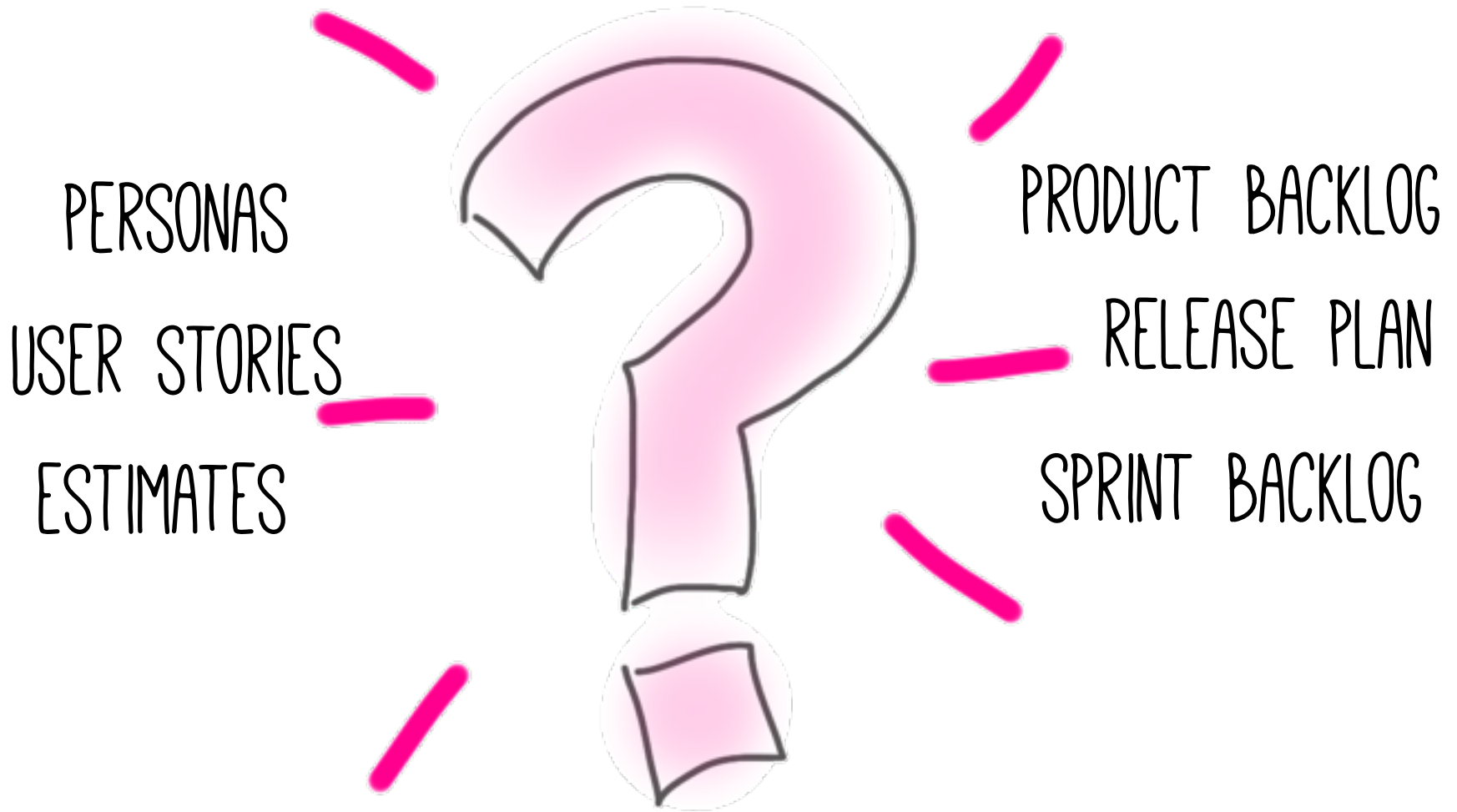


Project Charter

Team Established

*Outputs for
Envision
Phase*

Speculate



What is a user story?

A written description of a piece of functionality that is of **value to the user**

Written in the **voice of the user**
Gains detail **over time**

A placeholder for a **conversation** with the customer

Story Card Template

Story Name _____

As a _____

I want _____

so that _____

☒ User Story ☐ Spike ☐ Foundation

Size: **0**

Acceptance Criteria:

- Packet contains the following:
- 100 Story Card Templates
- Instructions & Ideas for using the cards

COMMENTS:

Story Card Template ©2013 Davis Consulting, LLC

Davis Consulting

As a [user role]
I want to [goal]
So that [benefit]

Independent

Negotiable

Valuable

Estimable

Small

Testable

*How do we know
when we have
good stories?*

Release Plan

| Product Backlog | Sprint 1 | Sprint 2 | Sprint 3 | Sprint 4 |
|-----------------|------------|------------|------------|------------|
| User Story | User Story | User Story | User Story | User Story |
| User Story | User Story | User Story | User Story | User Story |
| User Story | User Story | User Story | User Story | |
| User Story | | User Story | User Story | |
| User Story | | User Story | User Story | |
| | | User Story | | |

Sufficiently detailed
user stories

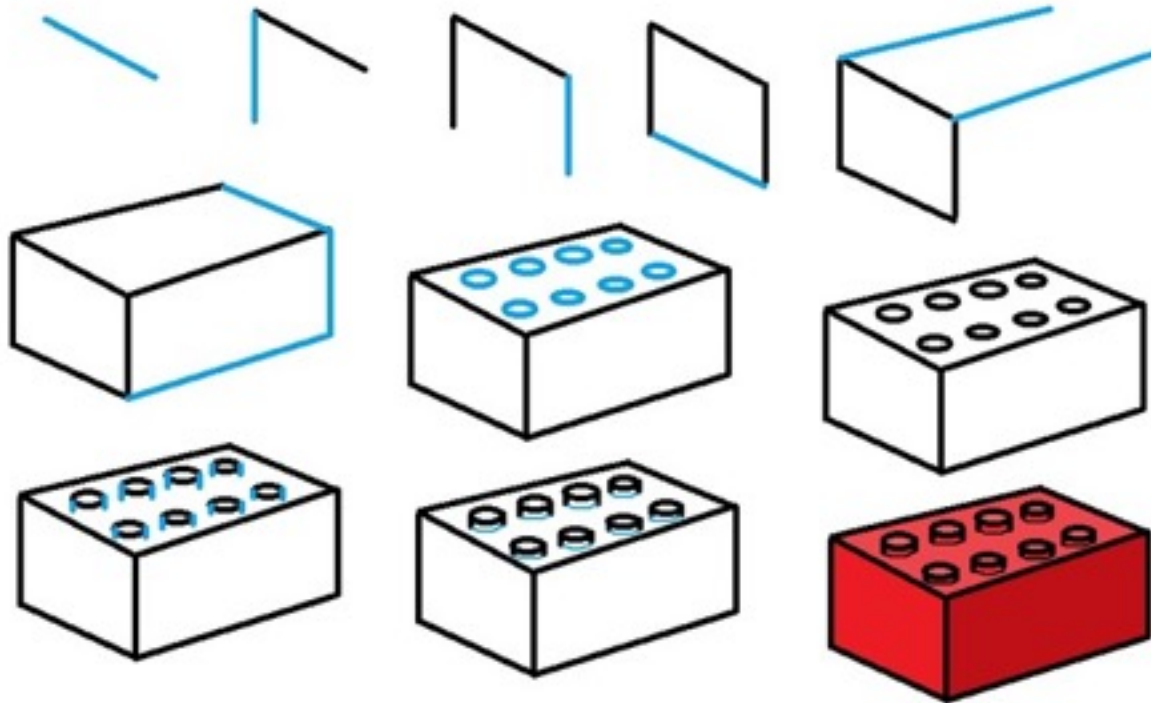
Refined product
backlog

Updated release plan

Identified sprint
backlog

*Outputs for
Speculate
Phase*

Explore



Build!

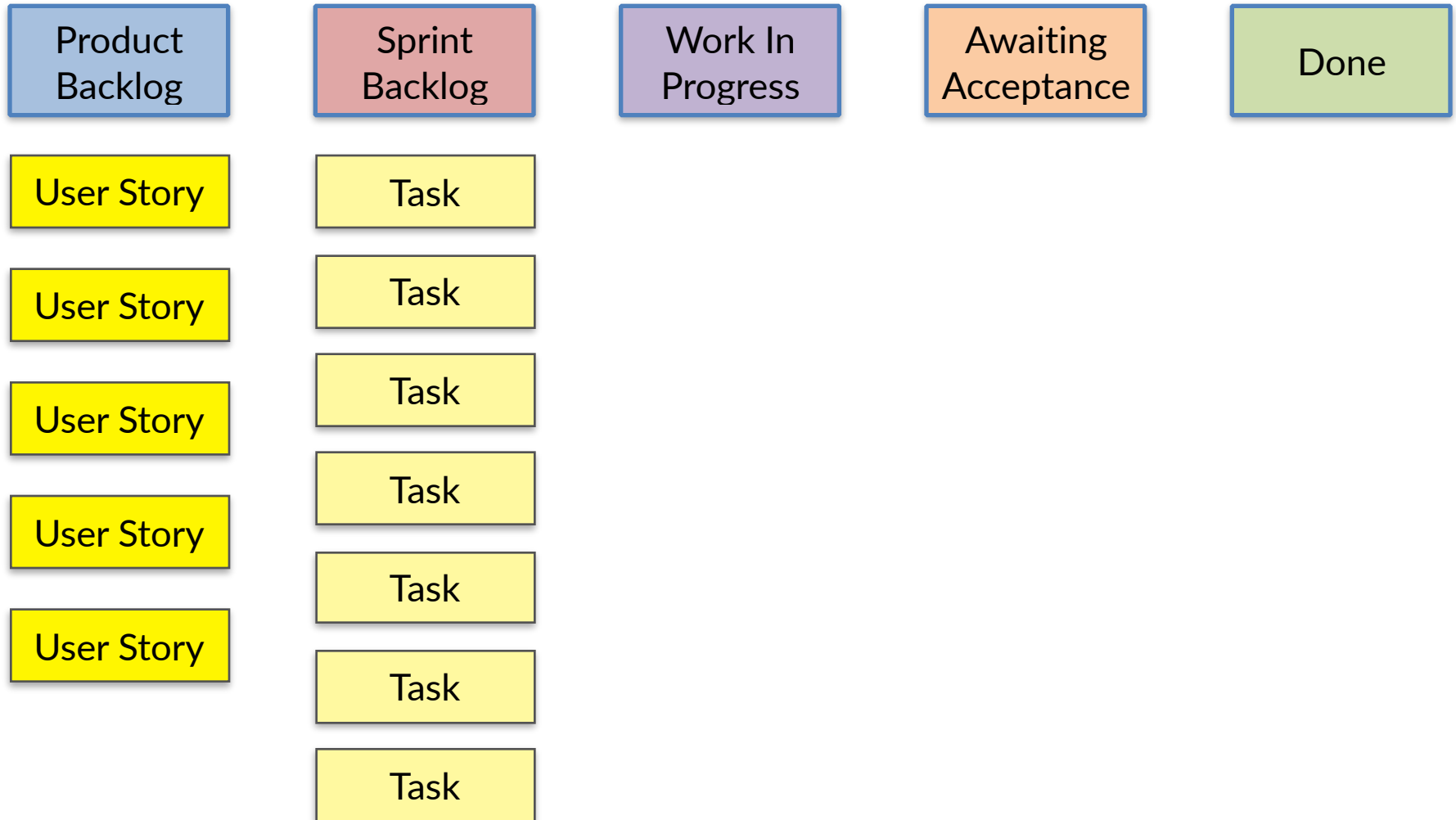
Daily stand-up
meetings

Task selection

Daily interaction with
the product team

*Explore
Activities*

Visual Taskboard



Thin stories aggressively
during early sprints to
build all essential
functionality early

Build up functionality only
after all necessities are in
place

Protect time in the final
sprints for product
refinement

*Tips for
releasing
on time*

Review & Adapt



PLAN



DO



CHECK



ADJUST

Final review with the
customer

Confirm if working as
expected

Review

Compare actual work
to planned work

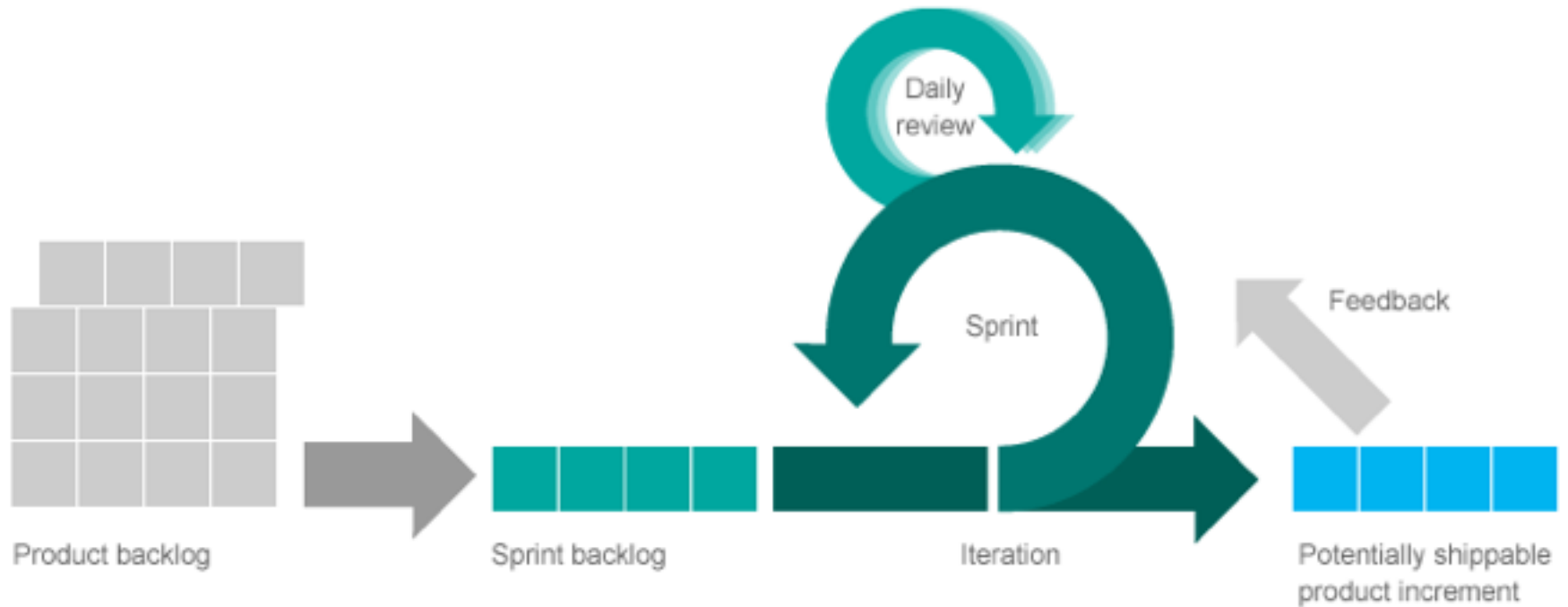
Talk about what is and
isn't working

Brainstorm to resolve
issues

Capture lessons
learned

Team Retrospective

Rinse & Repeat



Close



Project retrospective

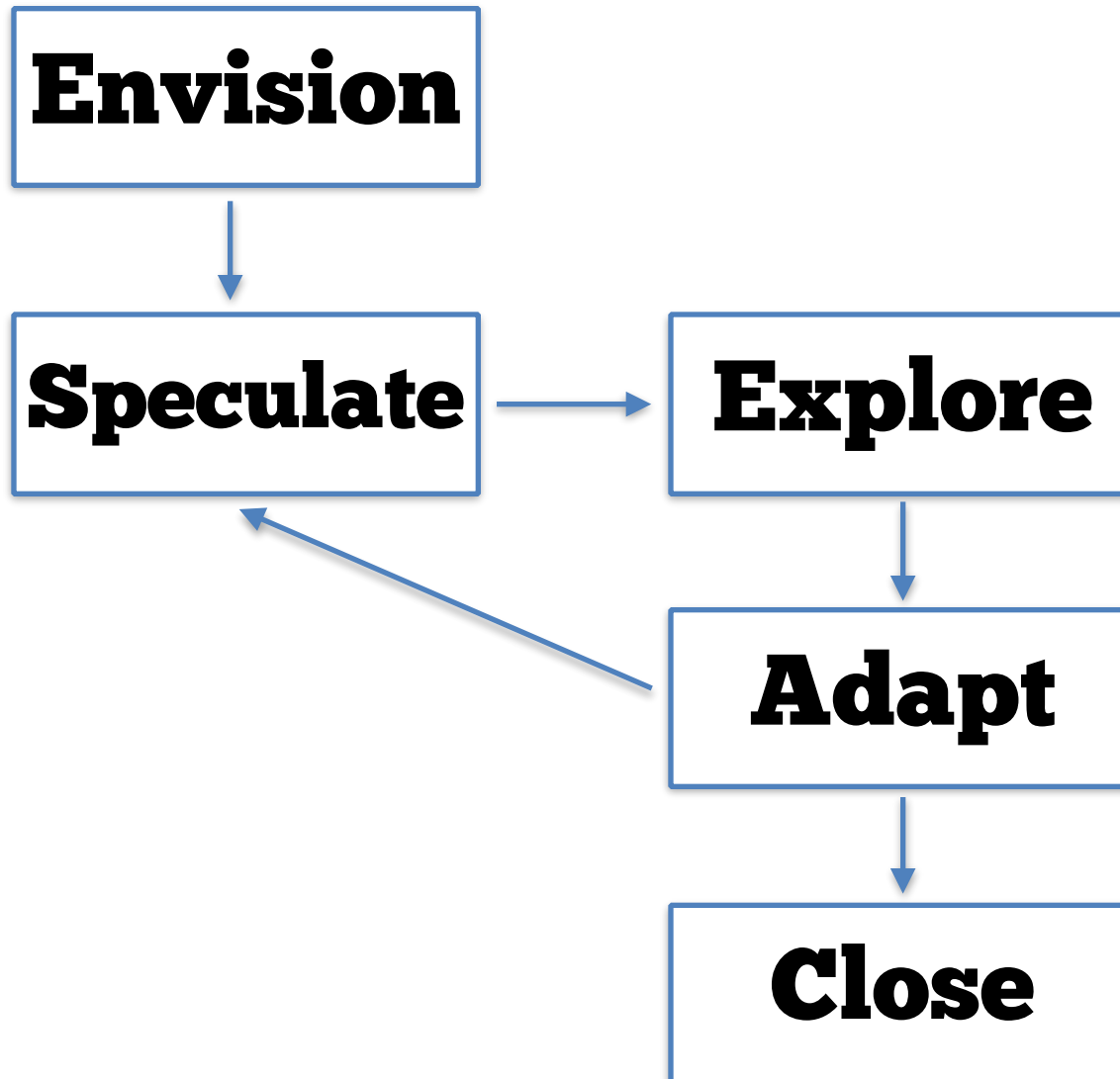
Communicate overall
results to stakeholders

Transfer backlog to
product owners

Recognize
achievements

Closeout

Agile Life Cycle



In the end remember,

If it works, do it.

If it doesn't, don't.

Thank You!



SARA COPE
@SARASSASSIN