Agile Planning

TECHNIQUES FOR SHARPENING THE SAW

David Starr
PRINCIPLE SOLUTIONS ARCHITECT

@ElegantCoder www.elegantcode.com

Overview

Traditional Plans

Agile Planning

Planning a Release

Techniques of Product Backlog Ownership

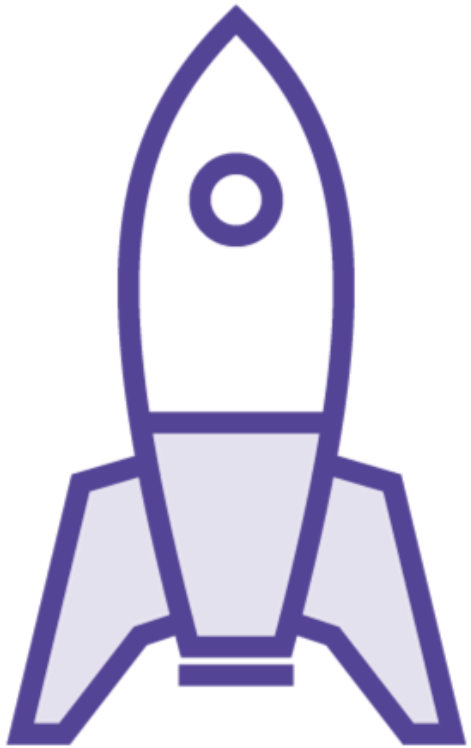
Techniques of Iteration Planning

The Daily Plan

The Winchester House

Perfect execution of a clear vision with no plan

The Mars Climate Orbiter



Lockheed Martin used Imperial units instead of metric units as specified by NASA

The spacecraft was destroyed by atmospheric stresses and friction during entry

Total project cost was 327.6 million

Traditional Plans

Because they work so well

A Good Plan Is

Clear

Reliable

Used

Available

Traditional Plans

Plan activities, not deliverables

Rely on strict sequencing

Time over runs are passed to next phase

Are developed for systems instead of features

Assert that the end result is known

Why We Need Plans

Reduce Risk

Make Informed Decisions

Reduce Uncertainty

Establish Trust

Convey a tangible vision

So customers can depend on you

Building a traditional plan

When the business side dominates

- Functionality and dates are mandated

- Little regard for reality or whether the developers understand the requirements

- Lengthy upfront requirements and signoff process

- Features are aggressively dropped as deadline approaches

When technologists dominate

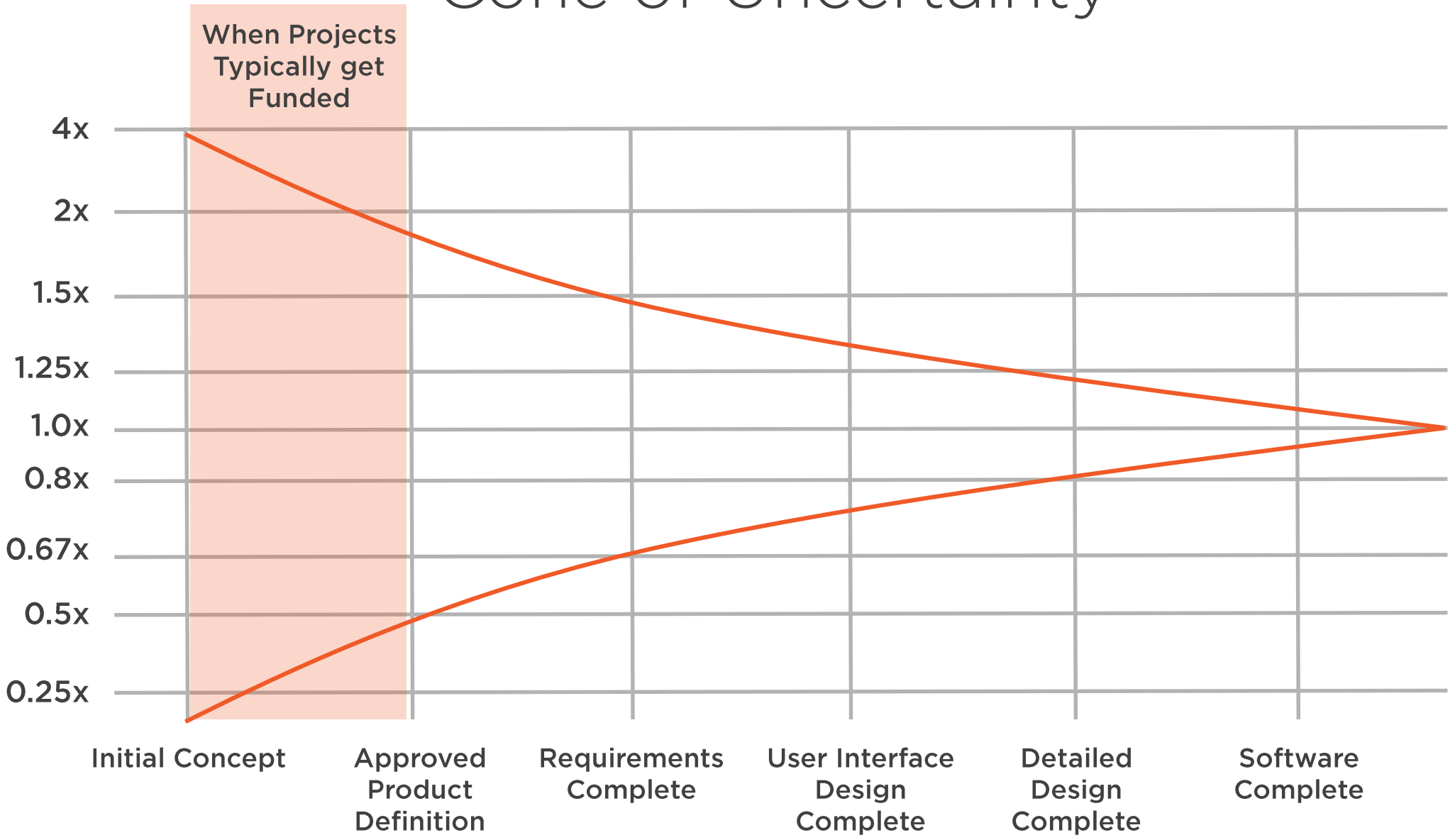
- Technical jargon replaces the language of the business and developers lose the opportunity to learn from listening

- May trade quality for additional features

- May only partially implement a feature

- May make decisions without feedback from the business

Cone of Uncertainty



The Unspoken Reality

We cannot perfectly predict a software schedule

Too many intangibles

Developers have a notoriously hard time estimating

The Unspoken Reality

We can't accurately say what will be delivered

As users see the software they come up with new ideas

Scope should change as new information is uncovered

Agile Planning

A Better Way

Agile Planning

Plan constantly, not just in the beginning

Planning is an activity, not a document

Don’t try to control change, encourage it

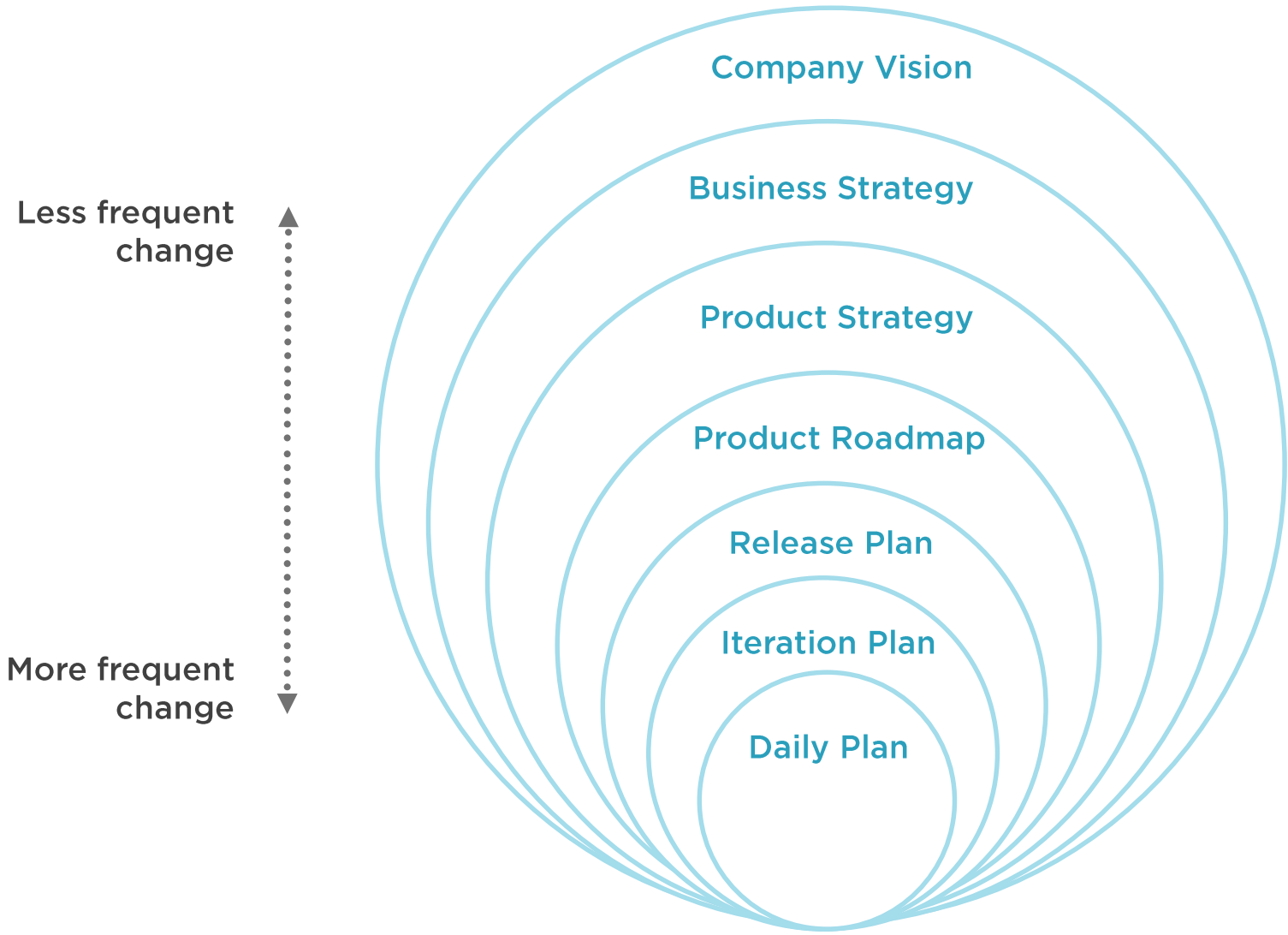
Be constantly transparent

Focus on historical performance, not hyper-optimal scenarios

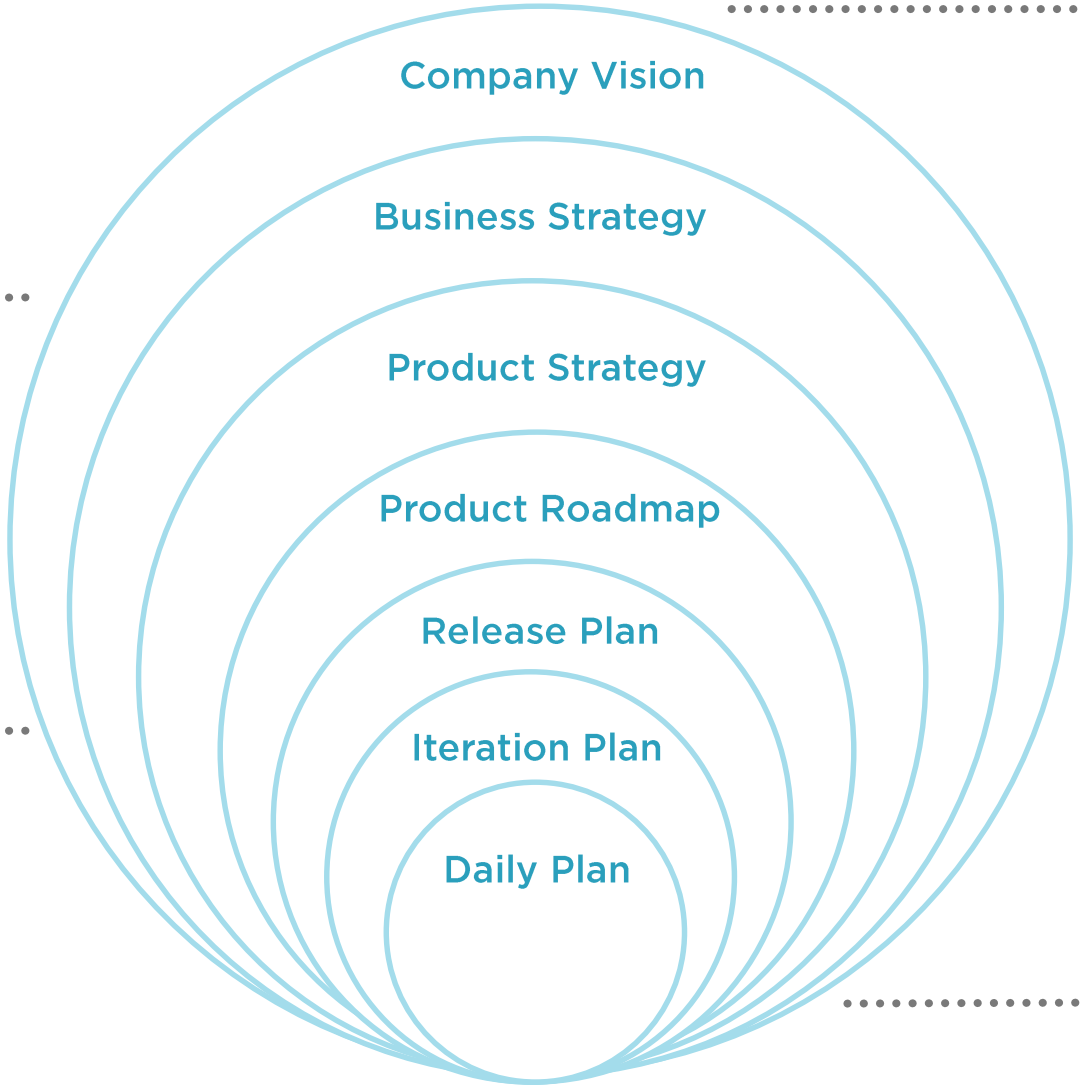
Changing the plan doesn’t mean changing timing

Levels of Agile Planning

Frequency of Change



Levels of Accountability

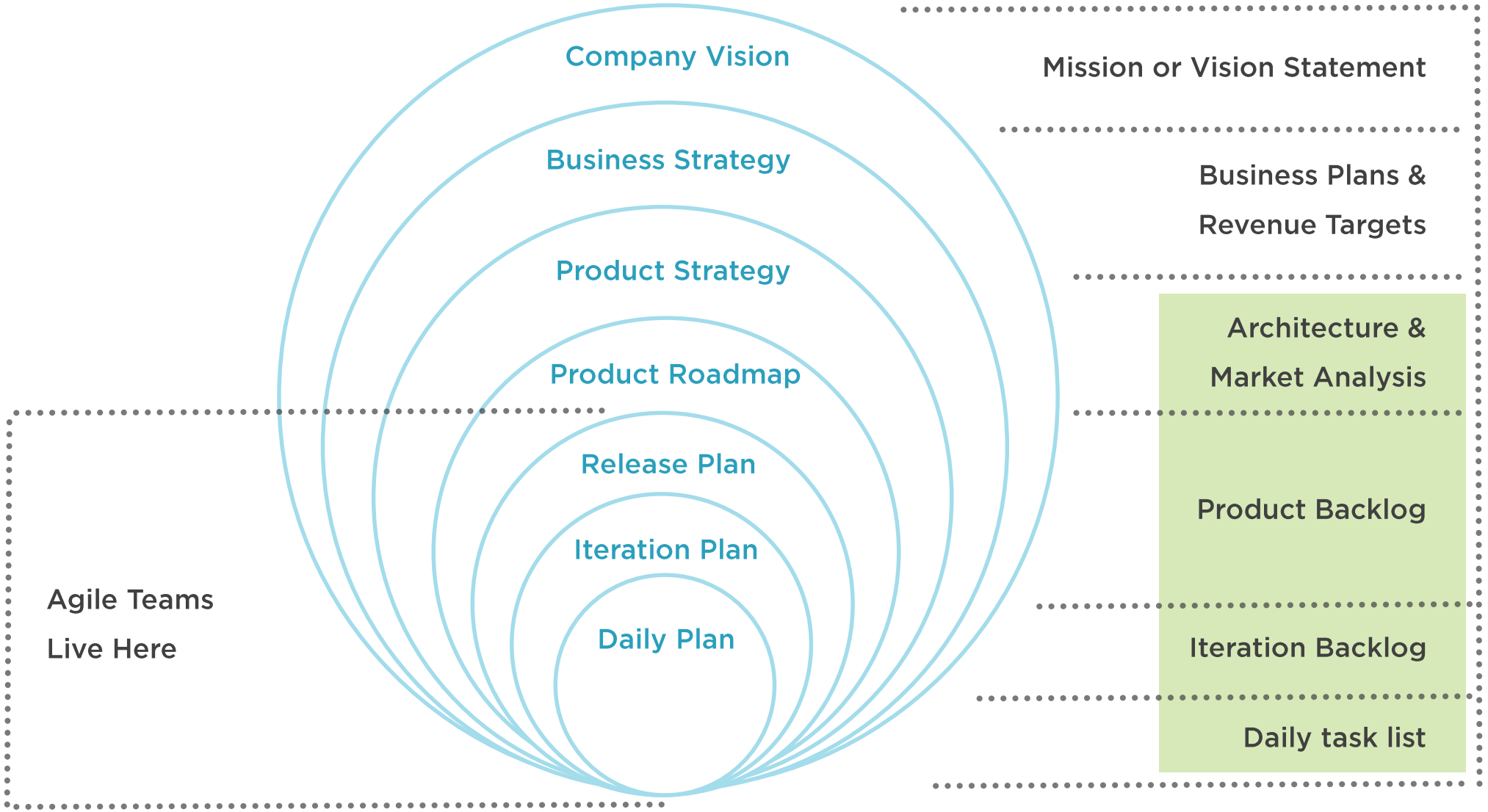


Executives
Departmental Leaders
Architecture

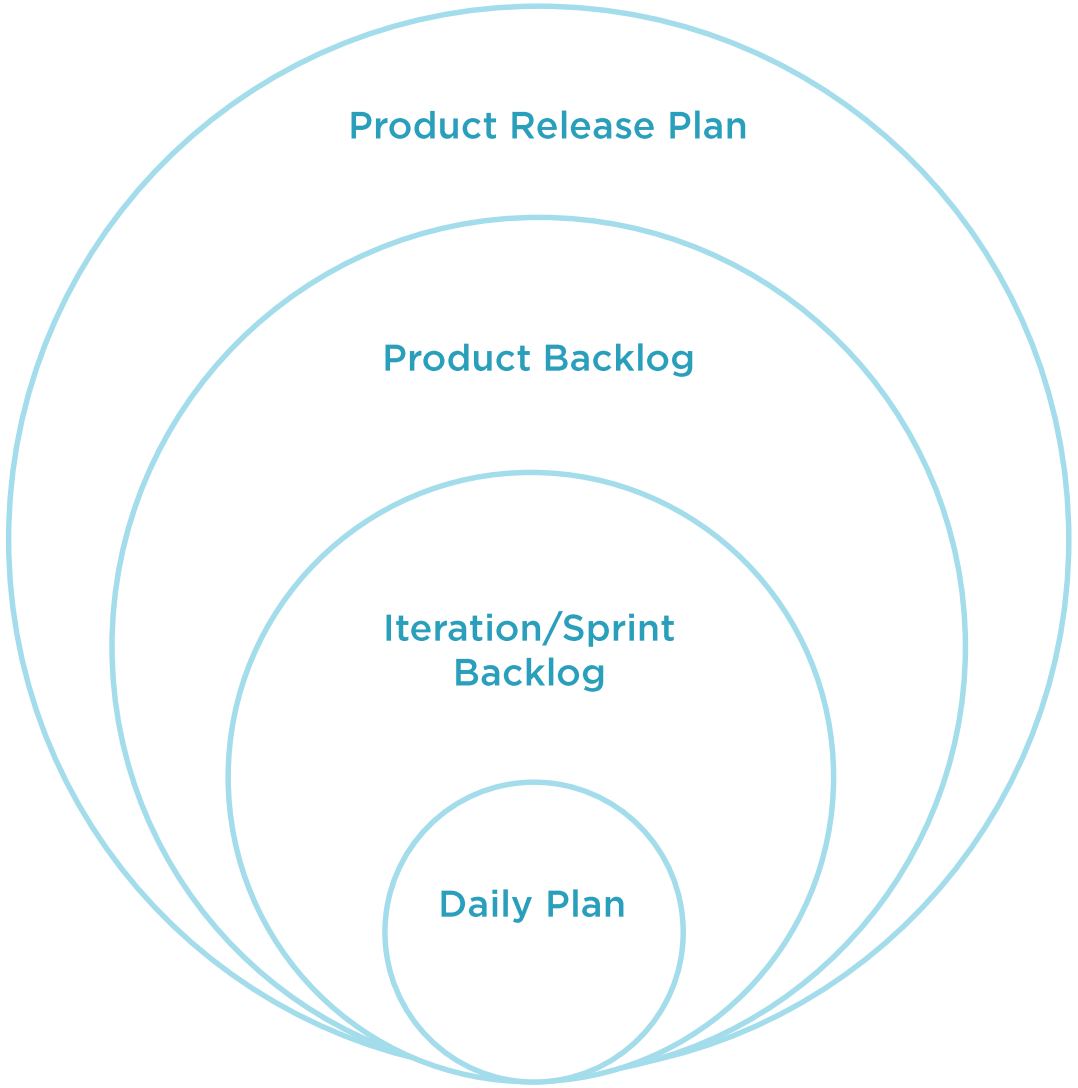
Customer Advocate
Product Management
Partners
Product Owner

Implementation Team

Artifacts of Agile Planning



Our Focus


Product Release Plan
Product Backlog
Iteration/Sprint Backlog
Daily Plan

Planning a Release

Actually shipping software

2 Basic Types of Release Planning

Date Target Planning

- The product will release on a specific date

Feature Target Planning

- The product will release when features A, B, and C are ready

“We do both” is not realistic.

One or the other will win
in the end.

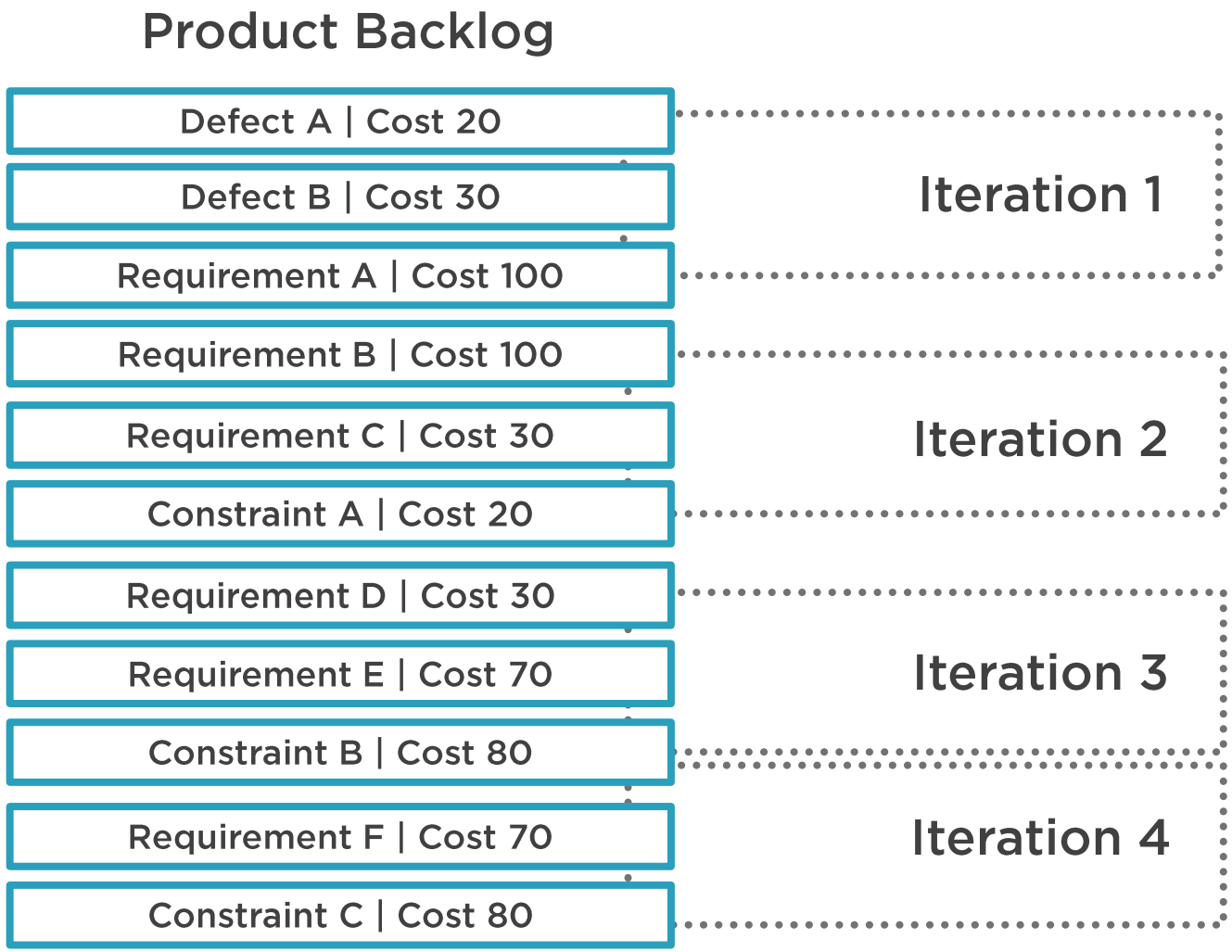
Rule 1

An accurate release plan requires a prioritized and estimated backlog.

Rule 2

An accurate release plan requires known velocity.

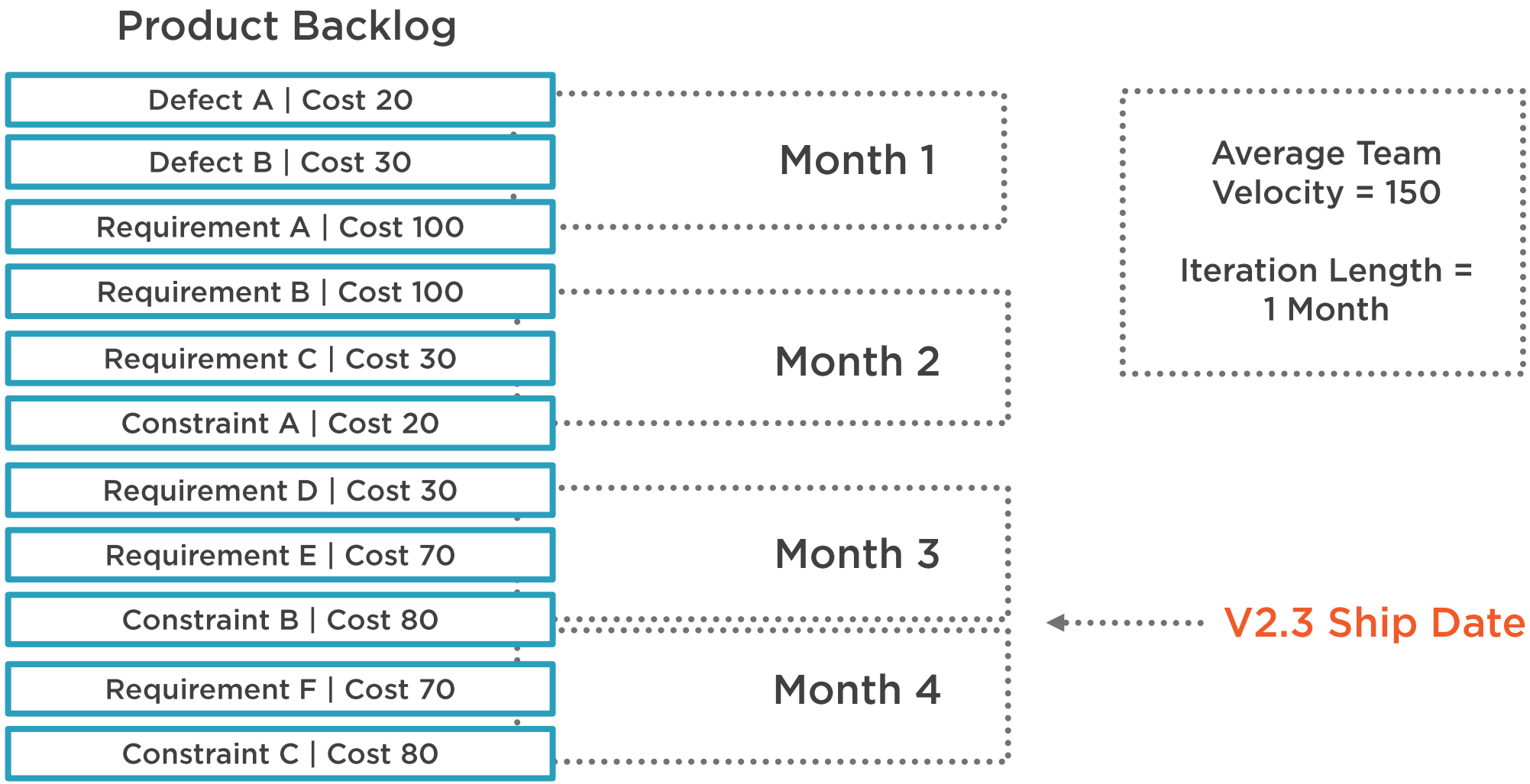
When Will Requirement F Likely Ship?



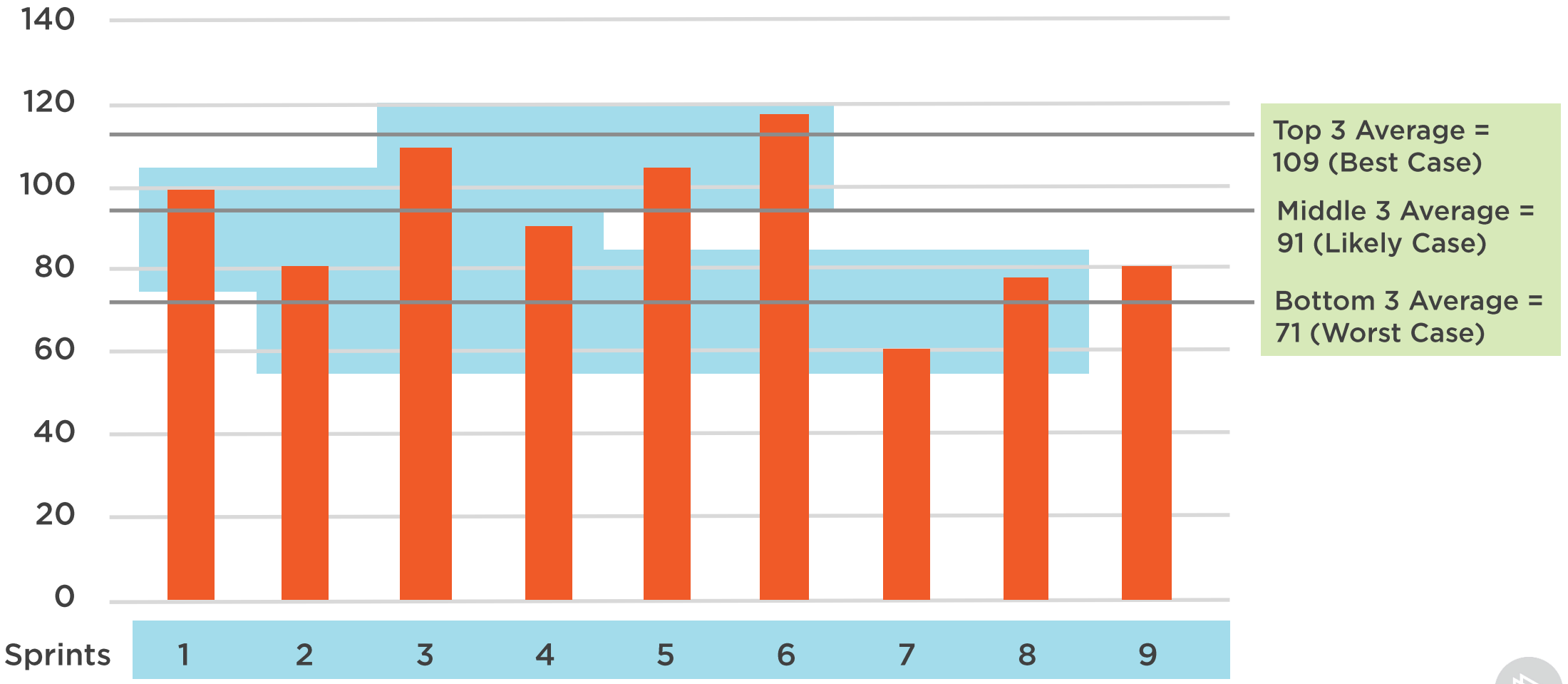
Average Team Velocity = 150

Iteration Length = 1 Month

What Will Ship In V2.3?



Analyzing Velocity



What Will Be Ready In 3 Months?

Product Backlog

- Defect A | Cost 20
- Defect B | Cost 30
- Requirement A | Cost 100
- Requirement B | Cost 100
- Requirement C | Cost 30
- Constraint A | Cost 20
- Requirement D | Cost 30
- Requirement E | Cost 70
- Constraint B | Cost 80
- Requirement F | Cost 70
- Constraint C | Cost 80

Worst case 3 x 71 = 213

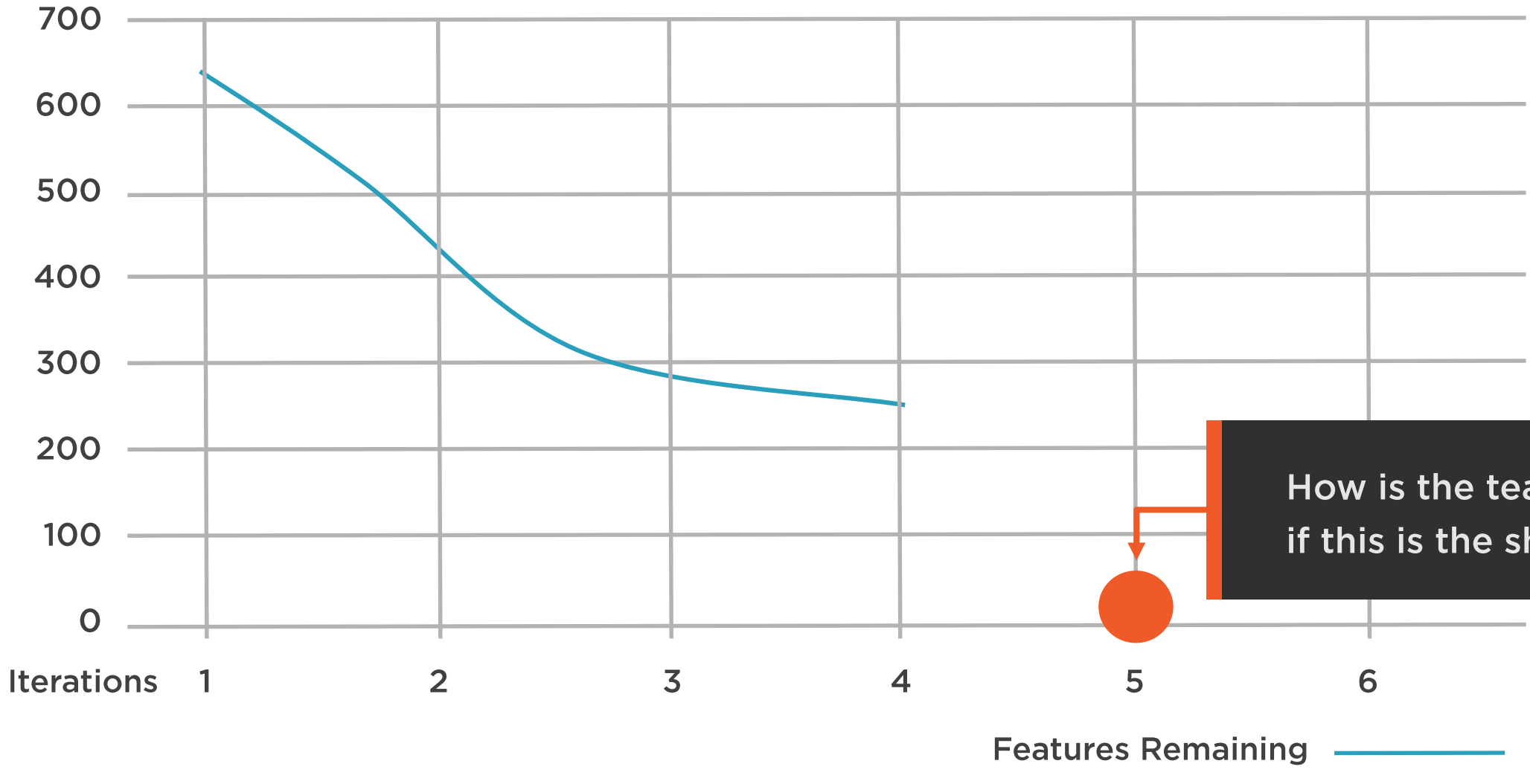
Likely case 3 x 91 = 273

Best case 3 x 109 = 327

Average Team Velocity = 150

Iteration Length = 1 Month

Release Burnout Chart



Product Roadmap vs. Release Plan

Product Roadmap

Communicate the big picture

Determine and communicate when releases are needed

Determine what functionality is sufficient for each release

Focus on business value derived from the releases

Product Release Plan

Predicts to what extent we are poised to deliver on the Product Roadmap

Provides tangible targets of functionality and dates backed by the reality of the Product Backlog

Used to make reality-based decisions

A Typical Release Plan

V1, Q2 2009	V1.1, Q3 2009	V2, Q4 2009	V2.1, Q4 2010
Theme: Framework	Theme: UI Enhancements	Theme: Administration Tools Enhancements	Theme: New Browser Support
Feature A	Feature A.1	Feature C.1	Feature C.2
Feature B	Feature C	Feature D.1	Feature E
	Feature D	Feature F	Feature G
	Feature E		

Techniques of Product Backlog Ownership

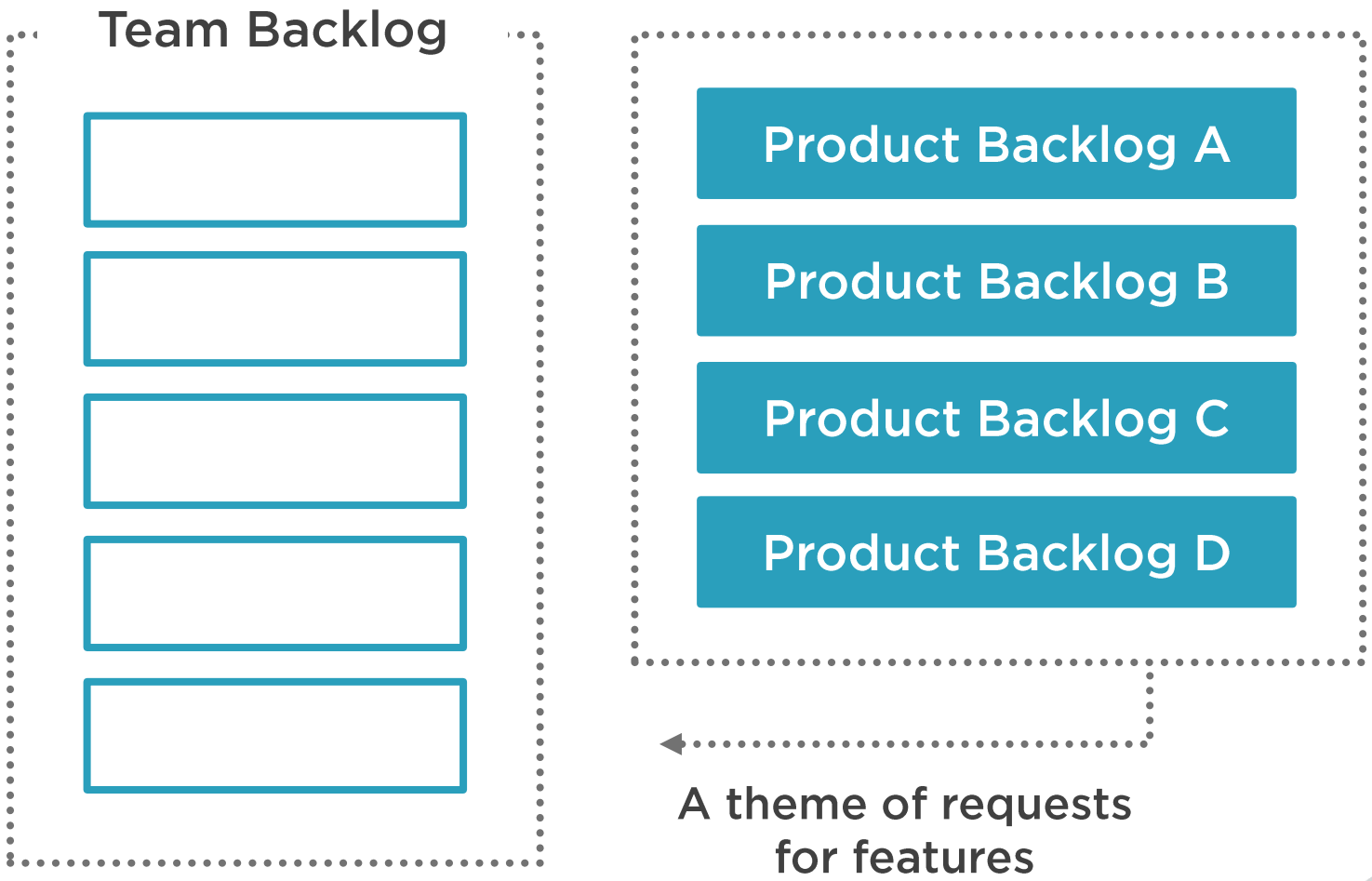
Making Informed Decisions

Many Products Sharing Themes

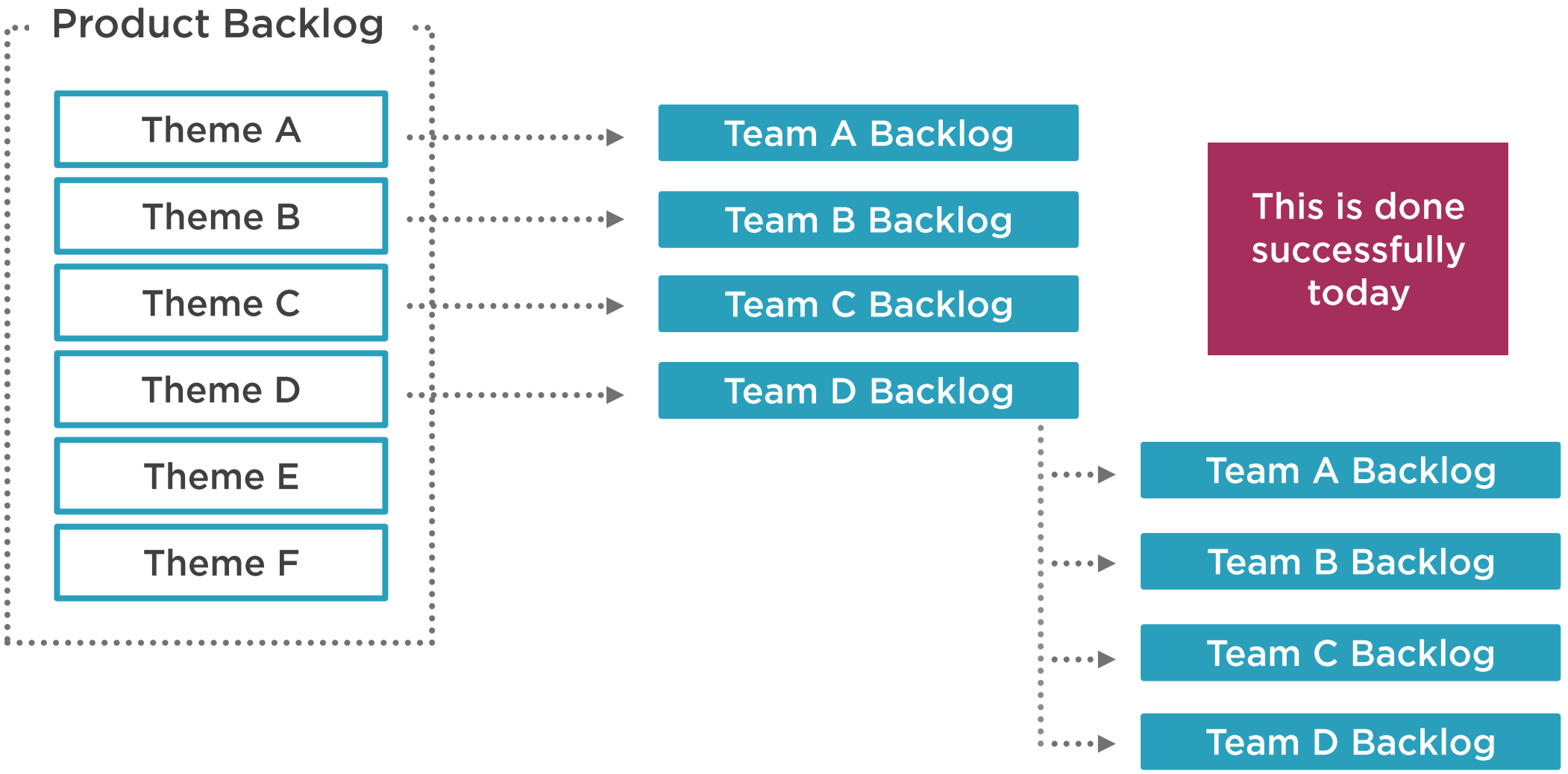
Themes	Products			
	MS Word	MS Excel	MS PowerPoint	MS Outlook
Smart Art	✓	✓	✓	
Spell Checking	✓		✓	✓
New Colors and Fonts	✓		✓	✓
Menu Ribbon Bars	✓	✓	✓	✓
Flashy Animation			✓	

1 Team, Many Products

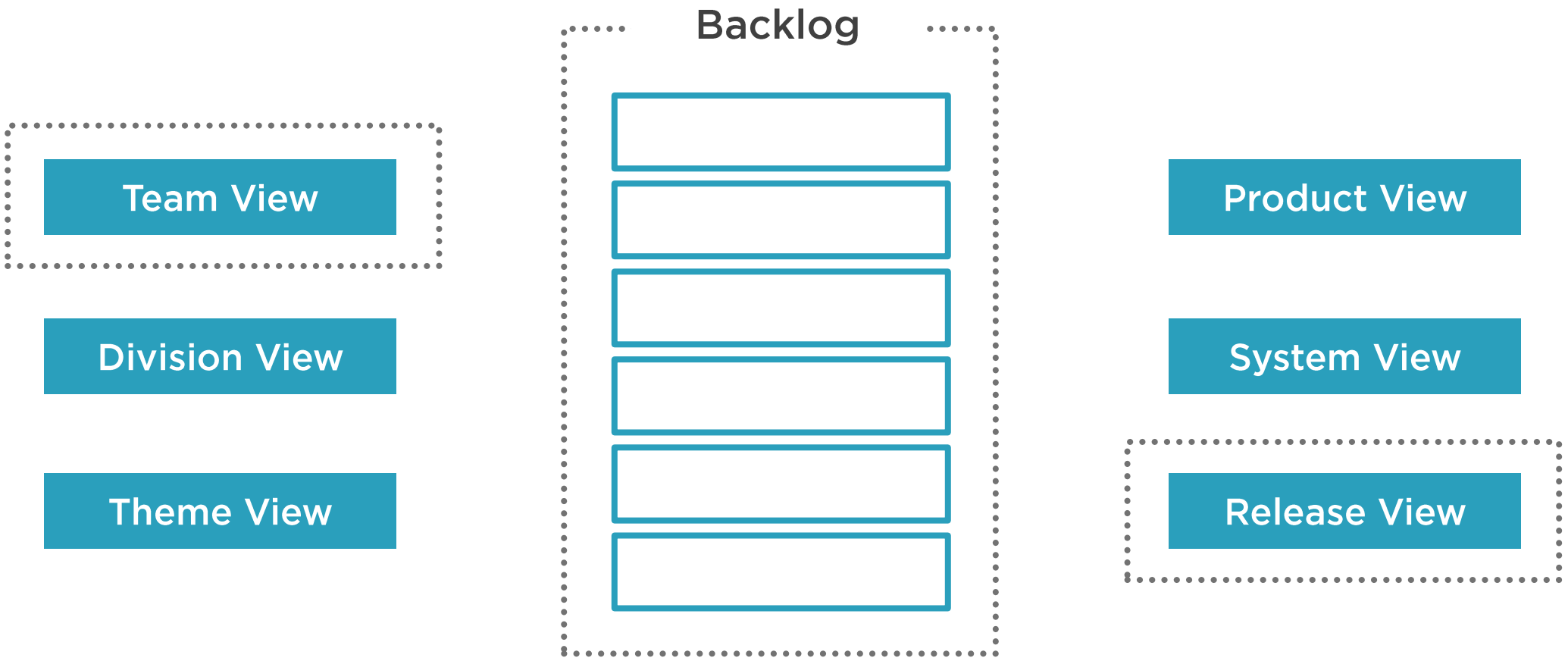
This Requires a Chief Production Owner



Many, Many Teams, One Huge Product



Which View of the Backlog Is Real?



These Views Are Special

Team backlog view

- The team uses this to plan the next iteration of work

- If you are a theme owner and your work items aren't showing up in the Team View, you're in trouble

Release Backlog View

- The absolute reality of what clients will get in the next release

Techniques of Iteration Planning

Getting ready to go fast

Iteration Planning Meeting

This is a pre-game meeting

First day of new iteration

Time Boxed

Everyone involved in the work is present

The point is to plan the next iteration only

The goal is to make a to-do list for the upcoming iteration

Velocity vs. Commitment Based Planning

Velocity Based

Uses average velocity over time or uses velocity of last iteration

Most Useful with long historical record

Unreliable in what will be accomplished

Assumes conditions are constant across iterations

Commitment Based

Team commits based on what they believe to be true right now

Likely to lead to realistic expectations

Uncovers future impediments now

Forces team to be deliberate in their thinking

Commitment Based Iteration Planning

Discuss the highest priority item on the product backlog

Decompose it into tasks

Whole team estimates each task in ideal time

Team answers “Can we commit to this?”

If yes, see If we can add another backlog item

If not, remove this item but see if we can add another smaller one

Ideal Time

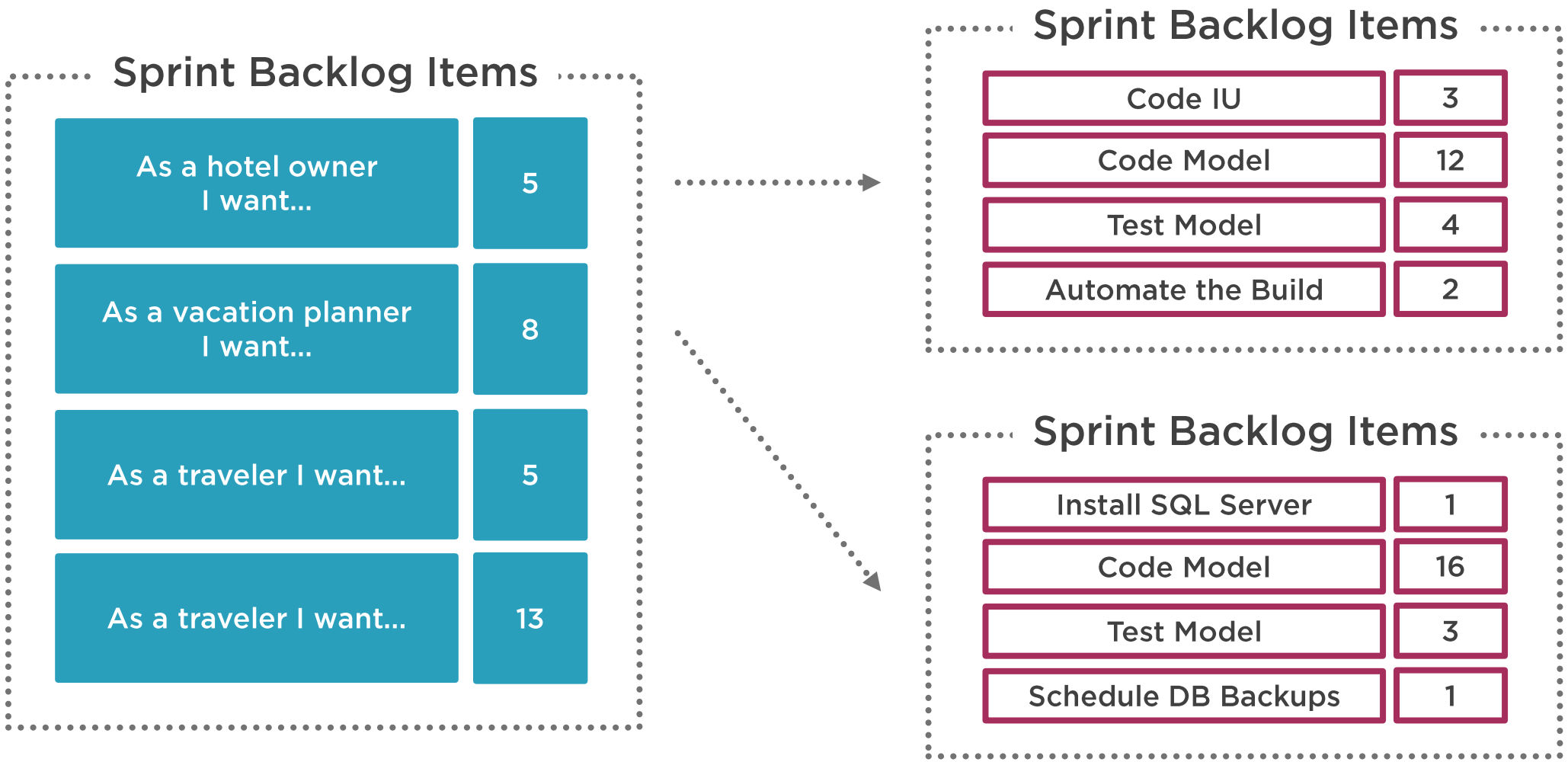
How long something would take if

- It's all you worked on
- You had no interruptions
- Everything you need is available

The ideal time of a football game is 60 minutes

- Four 15-minute Quarters
- The elapsed time is much longer (3+ hours?)

Task Decomposition



About Tasks

The real work is in the PBI

Tasks don't typically need a lot of detail

These items represent a conversation

Simply meant to be a to-do item

Keep it simple

The Daily Plan

Staying focused

The Daily (Scrum/Standup/Planning Session)

Sharing Commitment

Communicate daily and plans to the team and any observers

Identify impediments

Set direction and focus

Regularly rallying the team builds a stronger team

Tips for Staying Effective

- Limit to 15 minutes
- Good stand-ups will feel supportive and respectful
- All team members participate, everyone is heard
- It’s all pig, no chickens
- Everyone walks away with actionable commitments
- Co-locate the meetings with information radiators

Information Radiators

A large display of critical team information

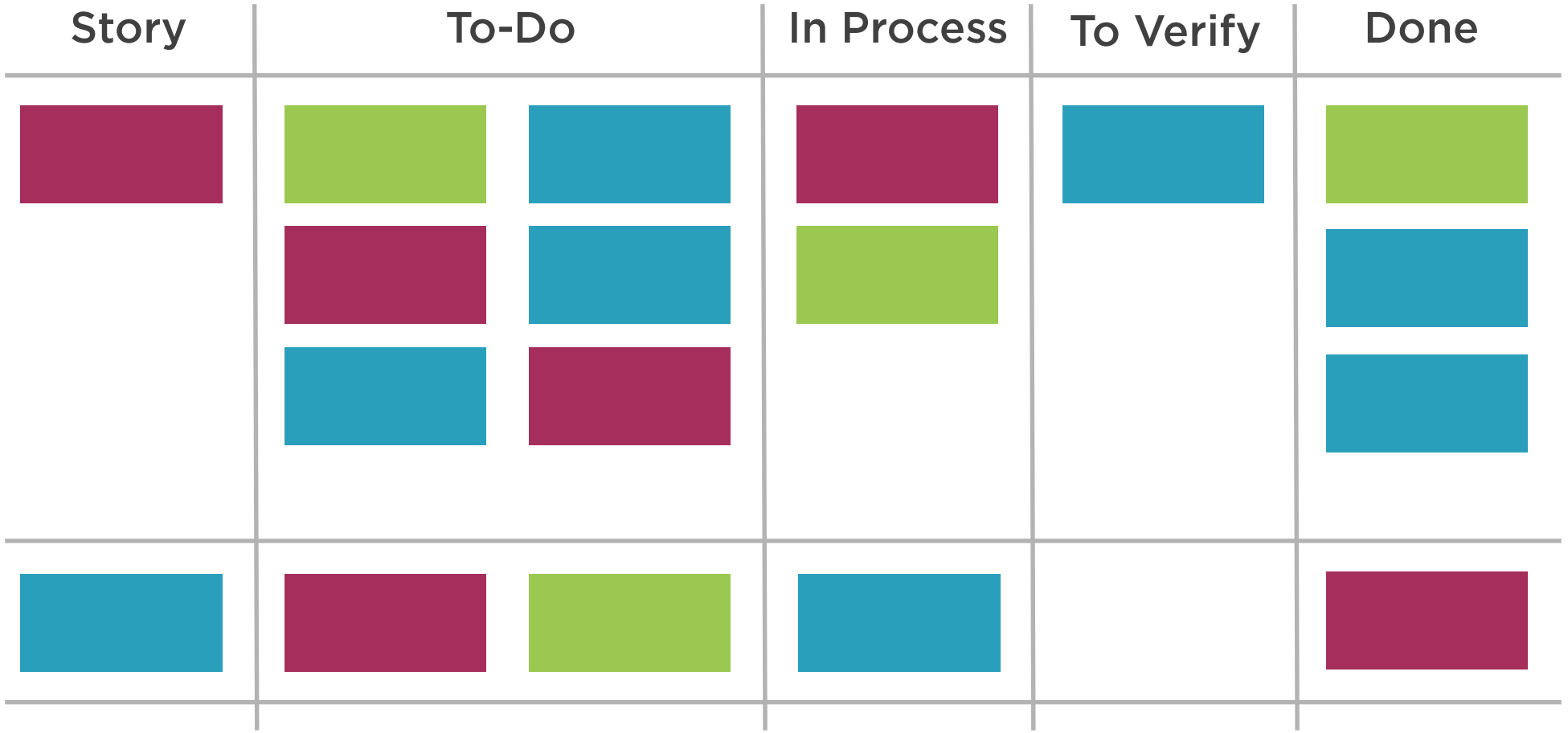
Continuously updated

Located where the team can see it constantly

Perfect Items for the Team Information Radiator

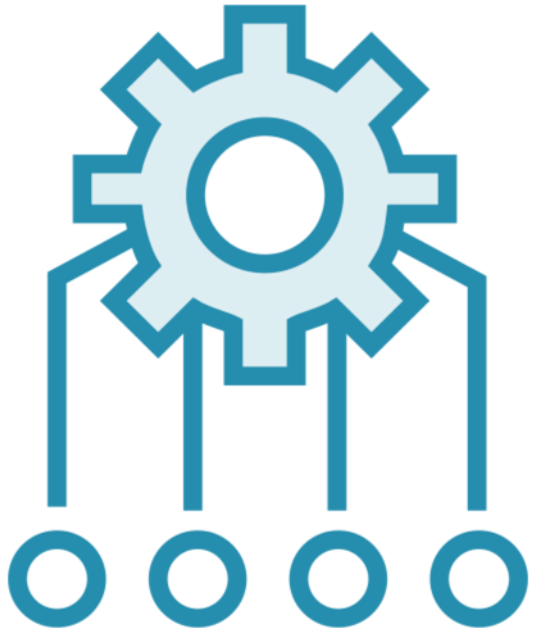
1. Task Board
2. A Burnout Chart
3. Historical View of Team Velocity
4. Current Build Status
5. Number of Current outstanding defects
6. Number of passing tests
7. Current code coverage
8. Release Plan

The Task Board



Typical Task Board

More Tips to Stay Effective



Focus on the backlog

Create a parking lot for the following up later

- Problem solving
- Story telling
- Impediments

Signal the end

Time the meeting

Keeping it fun and interesting

More Tips to Stay Effective

Last person to arrive starts the meeting

Bring food

Fine latecomers – the money is spent on the ship party

Create a “Standup Duration Chart”

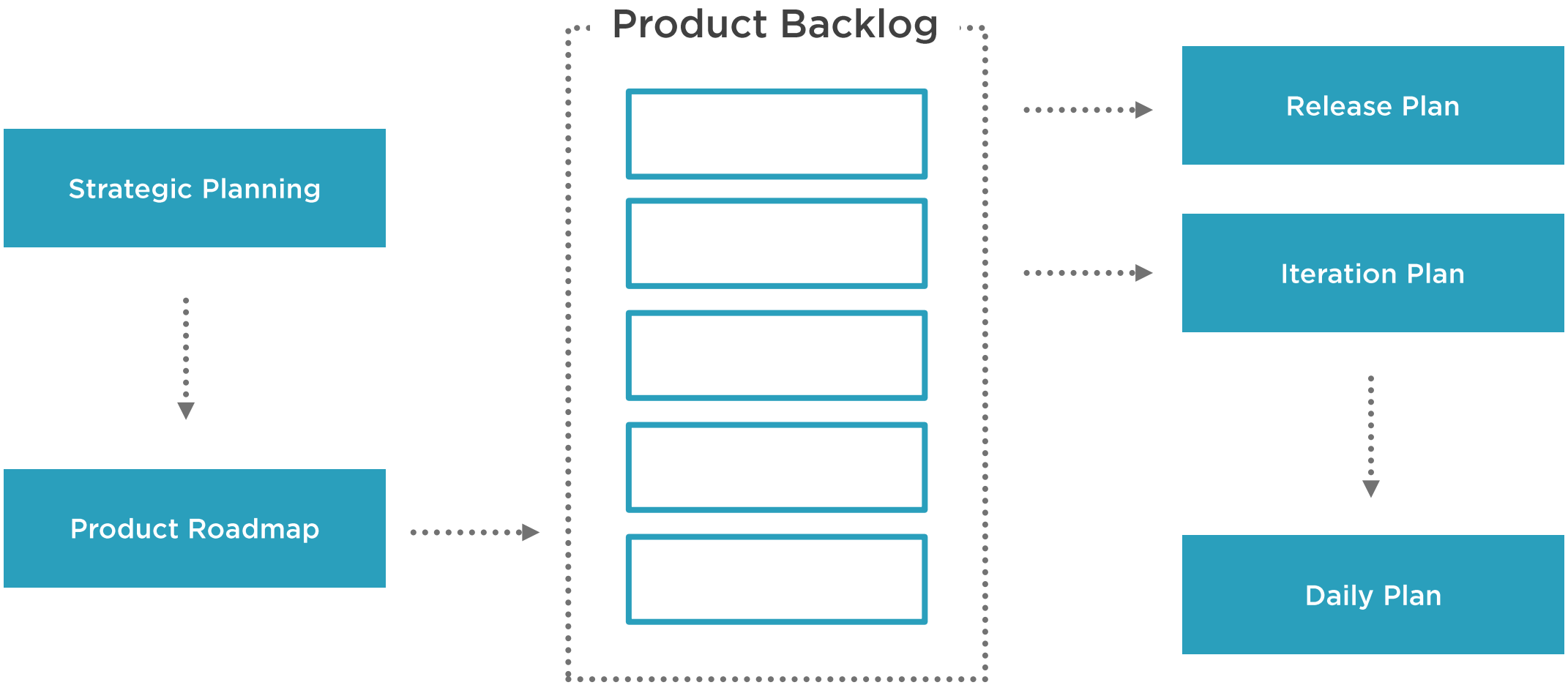
Changing up order

- Draw cards
- Round robin
- Pass the token

Perfect Items for the Team Information Radiator

Starting late	Socializing
Missing pigs	Gloom and doom
The meeting overload	Impediments aren't raised
Squawking chickens	Impediments aren't resolved
The storyteller	

Summary


Strategic Planning
Product Roadmap
Product Backlog
Release Plan
Iteration Plan
Daily Plan

References

Agile Estimating and Planning, Mike Cohn

Ford Drops Oracle-based Purchasing System, InfoWeek, August 2004

It’s Not Just Standing Up: Patterns of Daily Stand-up meetings
http://martinfowler.com/articles/itsNotJustStandingUp.html