Abbreviated Scrum Product Owner Workshop

Leo Primero

Certified ScrumMaster | Certified Scrum Product Owner | Certified SAFe Agilist

Legal Notices

Copyright

Protected Health Information (PHI)

The content and images included in this product documentation do not contain and PHI.

You Will Learn

- Why Agile?
- Scrum Values & Practices
- Roles in Scrum
- Levels of Planning
- Sprint Planning and Tracking
- Scrum Meetings
- Product Backlog & User Stories
- Estimation & Release Planning
- Communicating Plans and Progress

cognizant®

Why Agile? Are agile practices still relevant?

Why Agile? Are agile practices still relevant?

- It’s been 21 years
- Manifesto for Agile Development
- Scrum, DevOps, and Kanban.

- Customer collaboration over contract negotiation
- Individuals and interactions over tools and processes
- Quality software instead of exhaustive documentation

- The following are the key principles of Agile:
  1. Individuals and interactions over processes and tools
  2. Working software over comprehensive documentation
  3. Customer collaboration over contract negotiation
  4. Responding to change over following a plan

Why Agile? Are agile practices still relevant?

- Agile methodologies have become popular because they provide the following benefits:
    1. Faster time-to-market
    2. Higher customer satisfaction
    3. Better quality software
    4. Greater transparency
    5. More effective teamwork
    6. Increased employee engagement

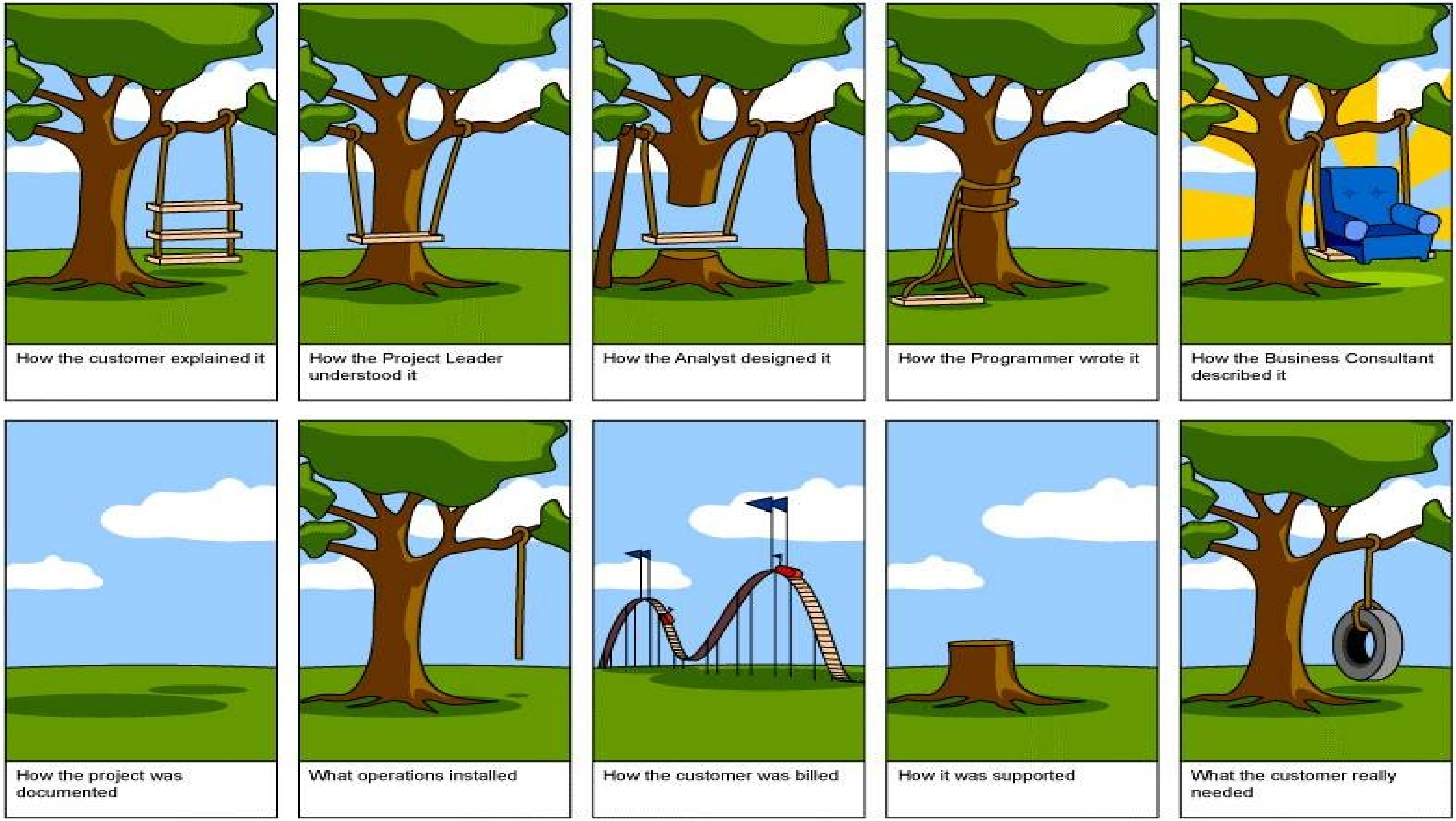
- Current State of Agile Adoption. Agile adoption is still rising, and more businesses and software development teams are adopting Agile methodologies. According to the 15th Annual State of Agile Report, 97% of organizations practice Agile in some form.

- However, not all Agile implementations are successful.

Why Agile? Are agile practices still relevant?

- The following are some of the challenges of adopting Agile:
  1. Resistance to change
  2. Lack of buy-in from stakeholders
  3. Poorly defined processes
  4. Inadequate training and coaching
  5. Lack of trust and collaboration

- To overcome these challenges:
  1. Businesses and software development teams must adopt a growth mindset and be open to learning and experimentation.
  2. Companies should invest in training and coaching to ensure everyone understands the Agile principles and practices.



How the customer explained it

How the Project Leader understood it

How the Analyst designed it

How the Programmer wrote it

How the Business Consultant described it

How the project was documented

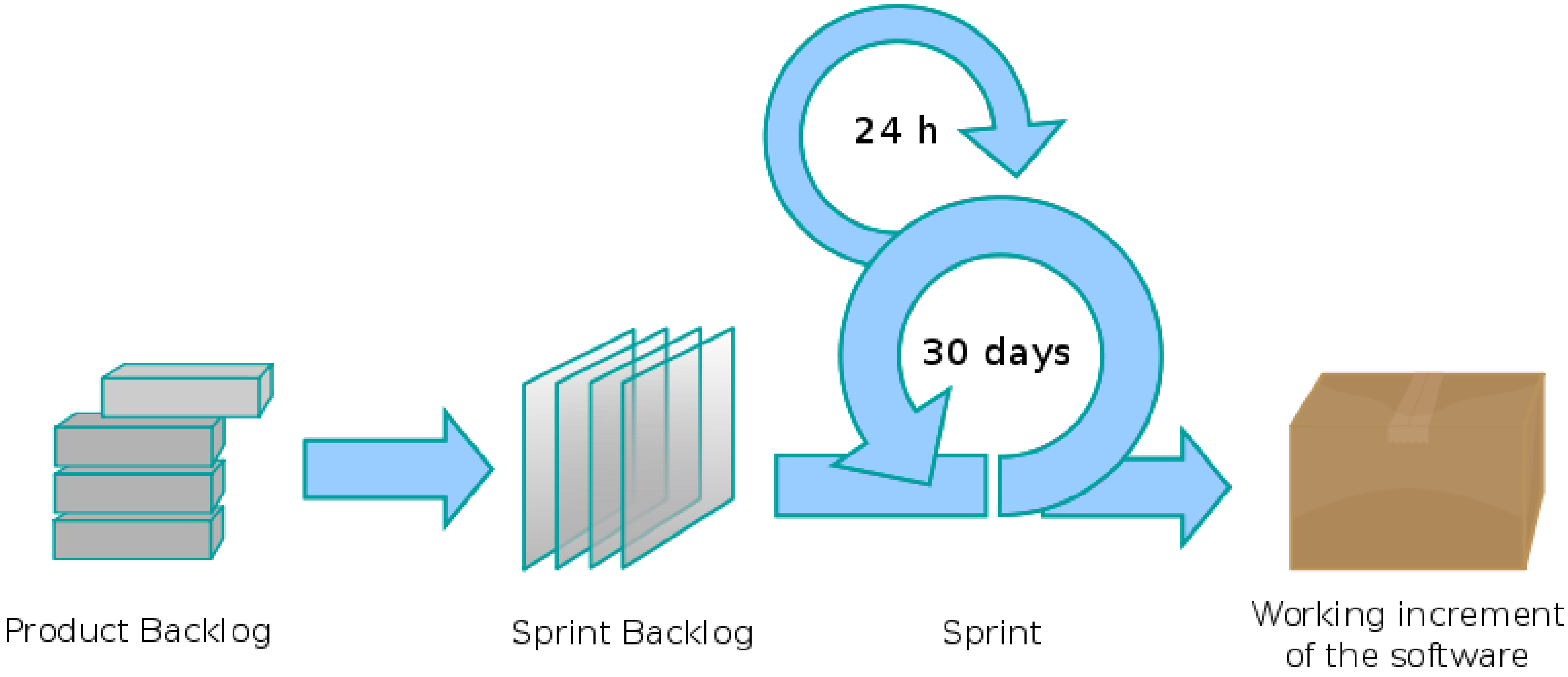
What operations installed

How the customer was billed

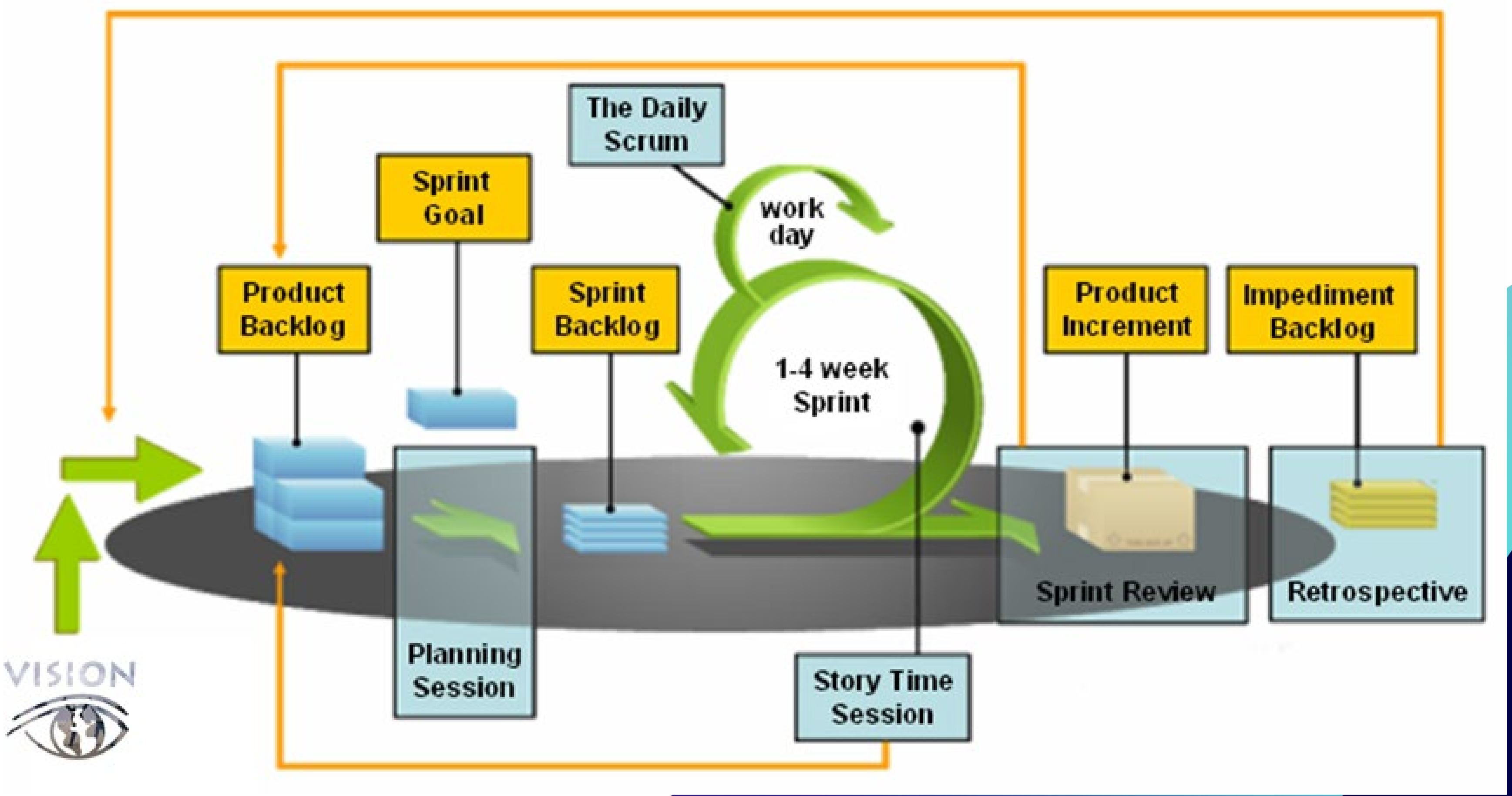
How it was supported

What the customer really needed

Scrum Practices



Product Owner Involvement



cognizant

Scrum Values

Commitment

Openness

Focus

Respect

Courage

Scrum Values Defined

Commitment

Be willing to commit to a goal. Scrum provides people with all the authority they need to meet their commitments.

Focus

Do your job. Focus all your efforts and skills on doing the work that you’ve committed to doing. Don’t worry about anything else.

Openness

Scrum keeps everything about a project visible to everyone.

Scrum Values Defined

Respect

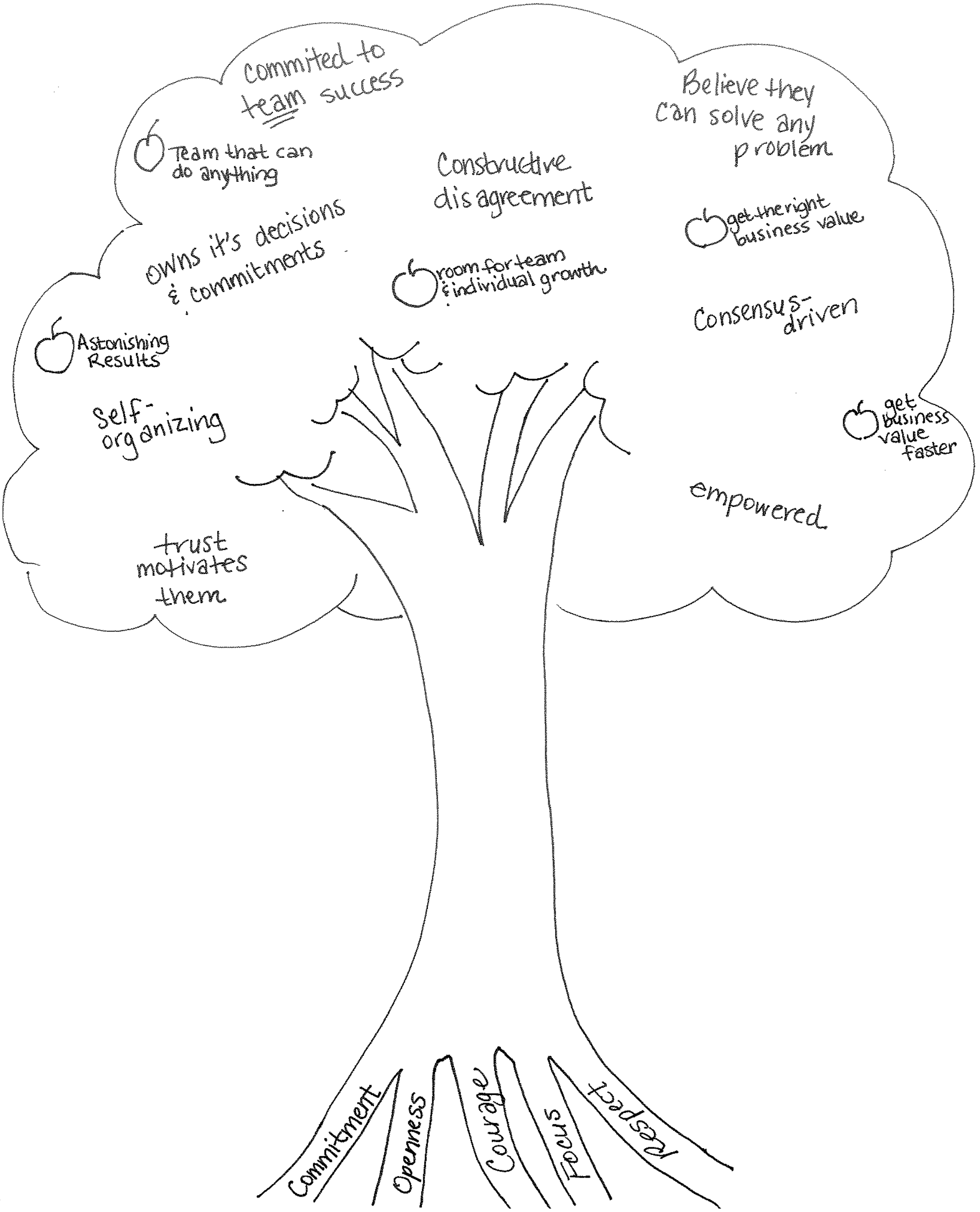
Individuals are shaped by their background and their experiences. It is important to respect the different people who comprise a team.

Courage

Have the courage to commit, to act, to be open and to expect respect.

The fruits of high performanc

- **Faster** business value
- **Right** business value
- **Astonishing** results
- Team that can do **anything**
- Room for team and personal **growth**



commited to team success
Team that can do anything
Constructive disagreement
Believe they can solve any problem
owns it's decisions & commitments
room for team & individual growth
get the right business value
Astonishing Results
Consensus-driven
Self-organizing
get business value faster
empowered
trust motivates them
Commitment
Openness
Courage
Focus
Respect

12 Agile Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment to support their needs, and trust them to get the job done.
6. Face-to-face conversation is the most efficient and effective method of conveying information to and within a development team.

12 Agile Principles

7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
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. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Role of the Product Owner

BUSINESS-VALUE-DRIVER

Daily Decision Maker

Vision Keeper

Heat Shield

Single throat to choke

To be a great Product Owner:

- Move from schedule-driven to business-value-driven
- Cultivate business-value-driven thinking in all interactions
- Be of one mind with the stakeholders
- Ask for more (or less) rather than micromanaging
- Hold the team to their commitments
- Leverage critical moments

What is a ScrumMaster?

Bulldozer

Shepherd

Servant Leader

Guardian of quality and performance

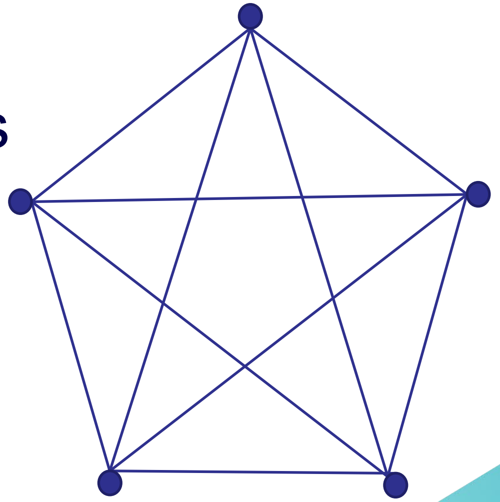
What a ScrumMaster Does/Does Not

DOES	DOES NOT
Guide and facilitate	Direct or drive
Keep everyone focused on delivering business value	Stick to deadlines and approaches that no longer work
Have a keen interest in the team’s overall performance	Become attached to specific outcomes from the team
Coach the team for high-performance	Get involved in task-level direction
Promote the skills and growth of every team member	Become the only voice of the team

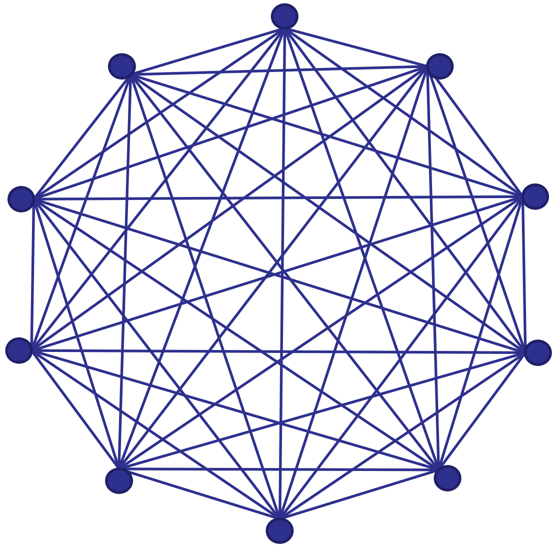
Team Members

- Smallest number of people possible who cover the skills and subject matter expertise needed to get the job done.
- 7 +/- 2
- Cross-functional: SEs, QAs, BAs, designers, etc.
- Shared commitments

Team Members



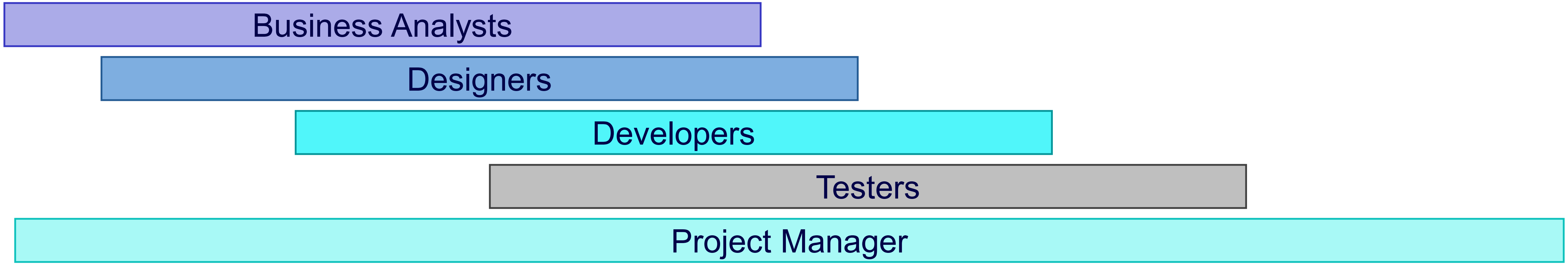
5 team members
=
10 relationships



10 team members
=
??? relationships

Everyone Starts Together

Defined approach (waterfall)



Business Analysts

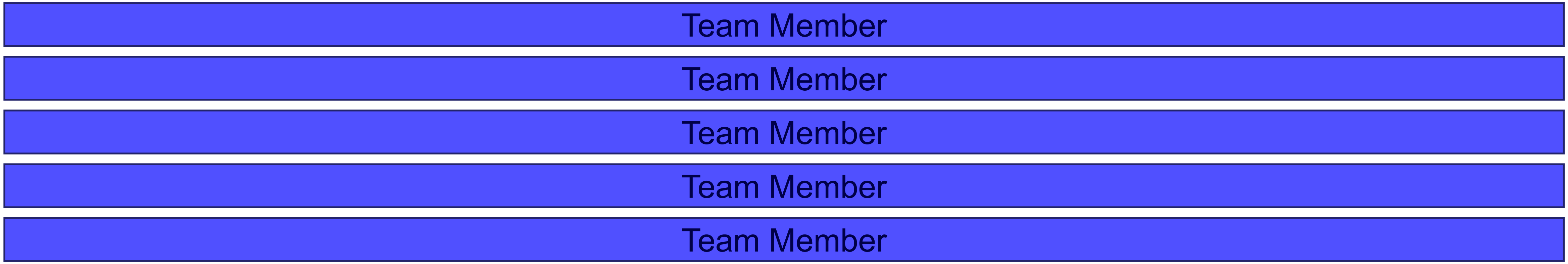
Designers

Developers

Testers

Project Manager

Empirical approach (Scrum)



Team Member

Team Member

Team Member

Team Member

Team Member

Outside People

Customer, Stakeholders, and Managers: anyone with an interest

What should you expect from them?

- They will add items to the Product Backlog through the Product Owner
- They will direct questions to the ScrumMaster or Product Owner
- They will not speak during standups
- They will attend sprint reviews to give direct feedback
- They will remove impediments **when asked**
- They will leave the team alone to get their work done

What is a Product Vision?

SHORT

Clear

Compelling

What is a product vision?

- A compelling vision is one of the biggest contributors to teams that produce products that matter.
- The lack of a compelling vision is one of the biggest contributors to a team’s failure or lackluster performance.

"Envisioning exciting possibilities and enlisting others in a shared view of the future is the ONE attribute that most distinguishes leaders from non-leaders."

Creating a Product Vision

- Use Scrum sprints to create the product vision.
- You may also need to do:

1. Market research (focus groups, Innovation Games®, data research)
2. Lots of internal alignment meetings
3. Financial analyses

Vision Questions

1. Who is going to buy the product? Who is the target customer?
2. Which customer needs will the product address?
3. Which product attributes are critical to satisfy the needs selected and, therefore, for the product's success?
4. How does the product compare against existing products from competitors and the same company? What are the product's unique selling points?
5. What is the target timeframe and budget to develop and launch the product?

What is a Product Backlog?

List

Open

THEMES

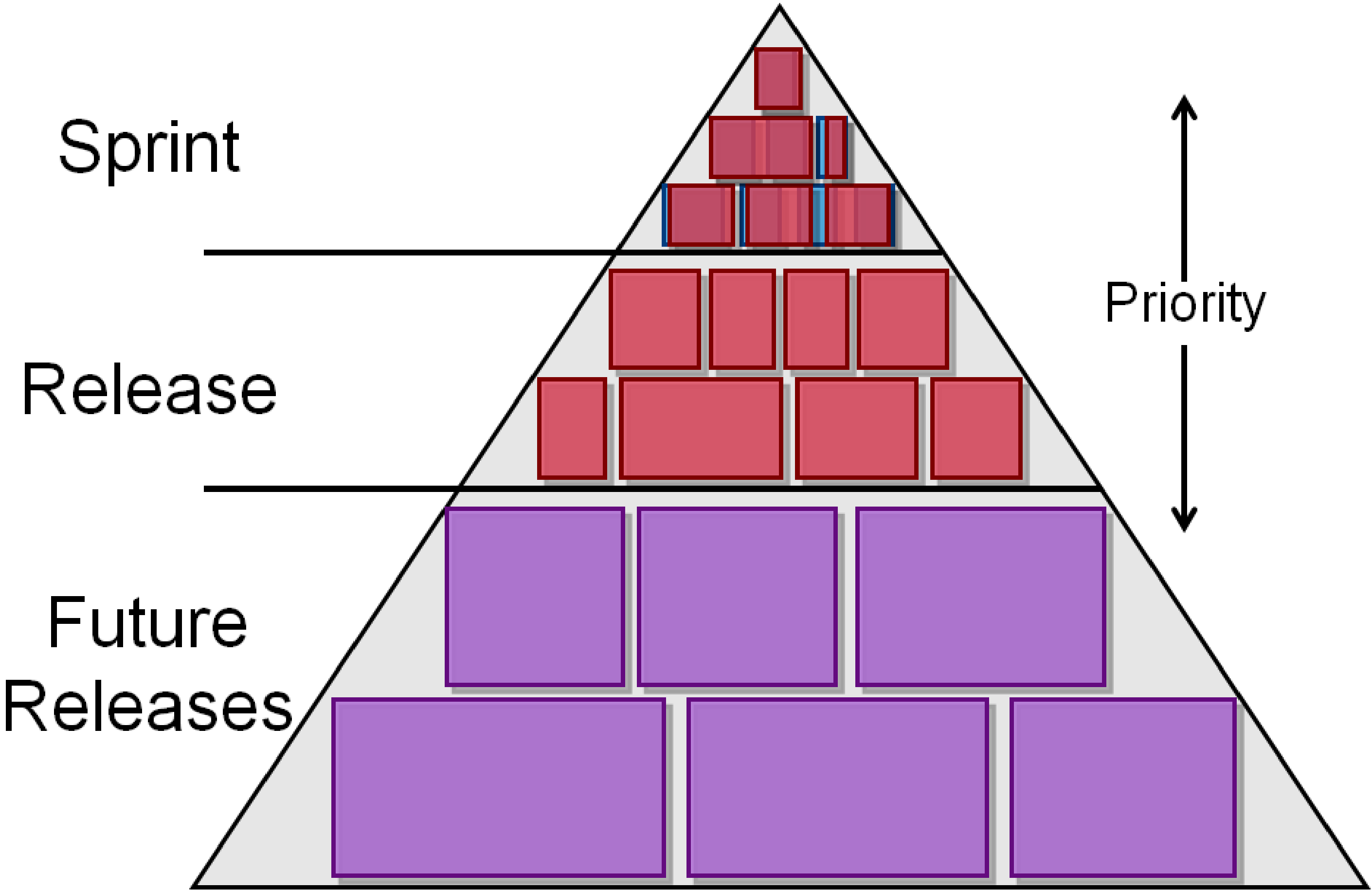
User Stories

Prioritized

A Product Backlog is

- A list of what needs to be done
- What we know right now
- Open for all to see
- May contain epics, themes
- Will contain user stories
- Prioritized according to highest business value going down to the least.

The Product Backlog Iceberg



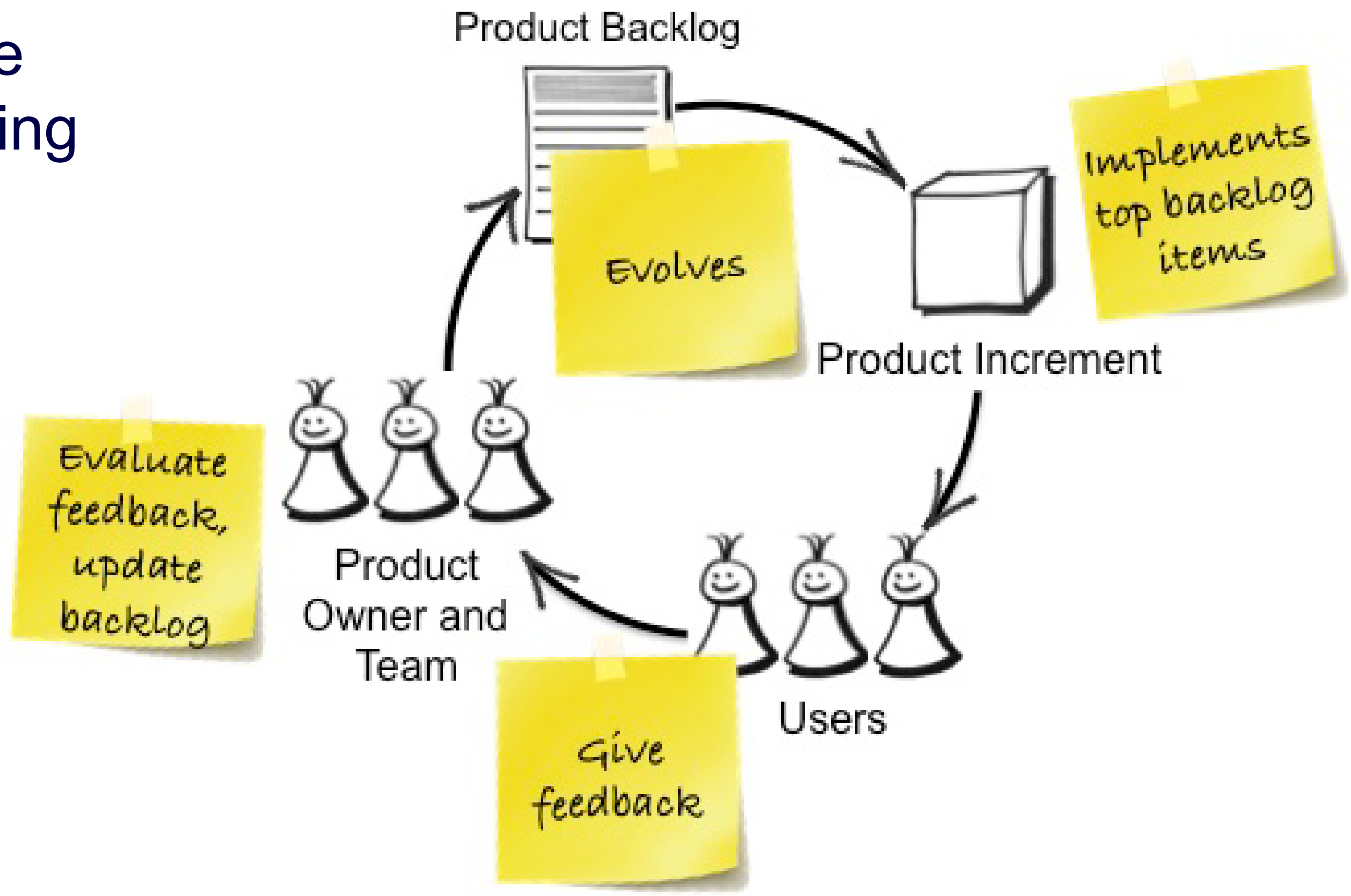
cognizant

What is a Well-Groomed Product Backlog?

D	Detailed appropriately
E	Estimated
E	Emergent
P	Prioritized

A product backlog constantly changes:

Change should be based on the learning obtained from developing the software and exposing it to customers, users, and other stakeholders through usability testing or design reviews.



Prioritization Criteria

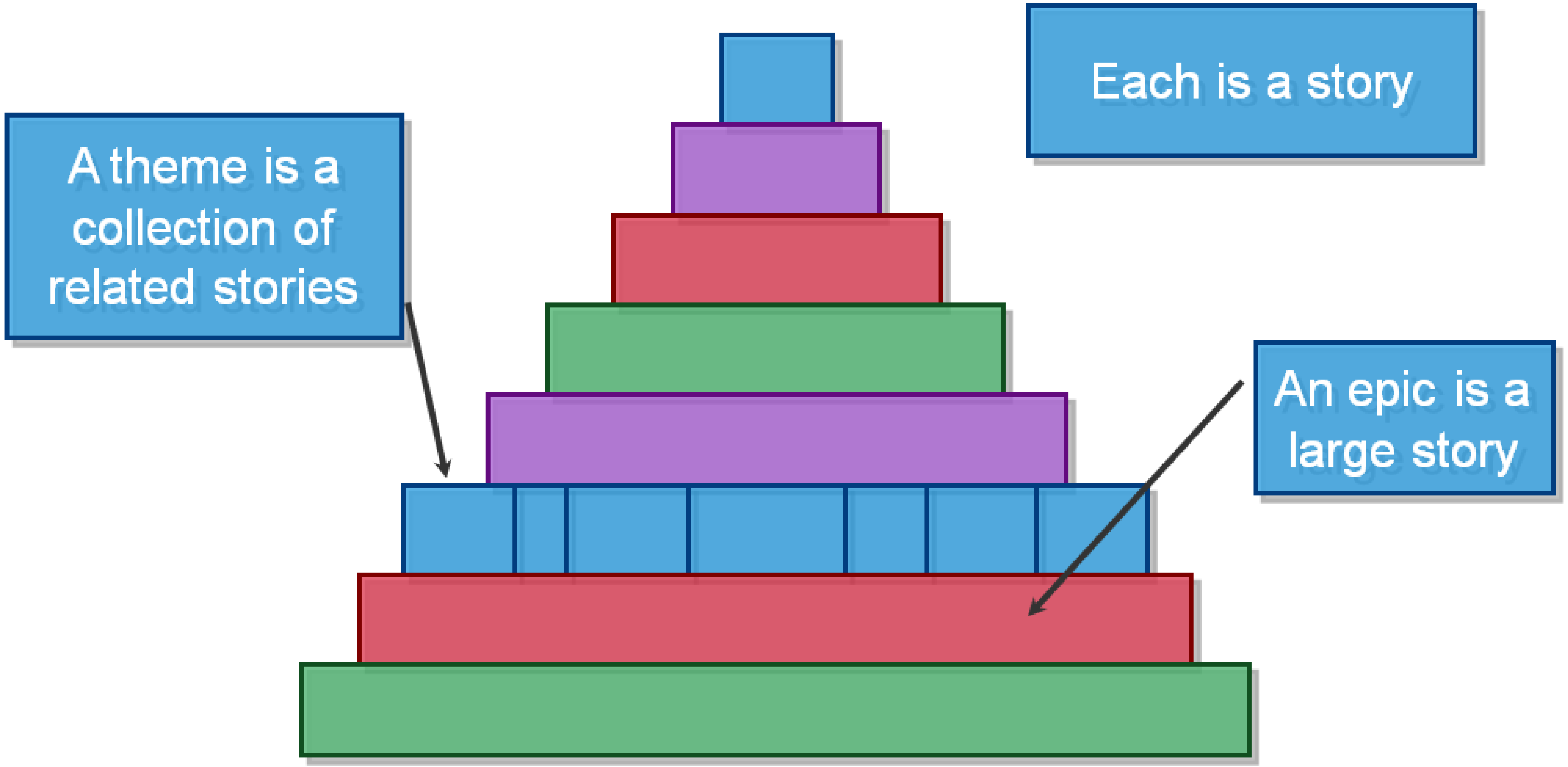
IN

- Return on investment
- Profit
- Risk
- Knowledge to be gained
- First mover advantage

OUT

- What feature I like best
- Embarrassment at changing the delivery expectation
- My personal performance evaluation criteria
- Which external groups are "ready" for the team
- "Hard" dependencies

The Epic, Theme, Story



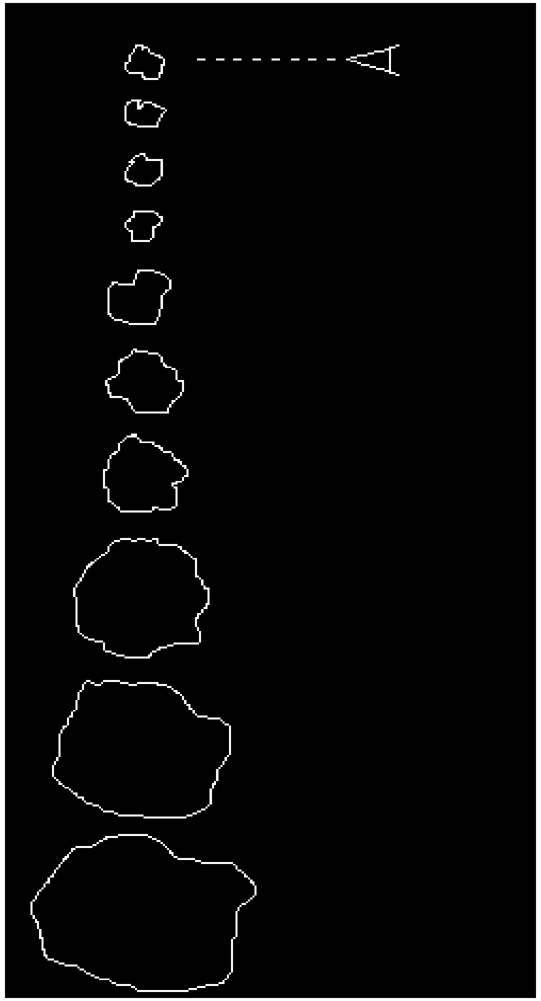
A theme is a collection of related stories

Each is a story

An epic is a large story

It’s a game of Asteroids

- Working through a product backlog is like playing a game of Asteroids.
- Large rocks (epics) get repeatedly broken down into smaller rocks (user stories) until they are small enough to be eliminated (easily developed and then delivered).



cognizant

What is a User Story?

- It is one of the **primary** development artifacts for Scrum teams.
- A user story describes the functionality of a system that will be valuable to an end user of a system or software.
- It is oriented to reflect the desires of the end user, which in turn will help developers remain focused on the customer.
- Product Owners write user stories.

Components of a User Story

It uses a simple template:

As an **<actor>**, I want to **<action>** so that **<achievement>**.

Actor: The one who would "act" the user story or the end user. It's very easy to end up using the name "user" for the actor but it is recommended to be more specific.

Action: What the actor wants to do. You can also differentiate between mandatory actions and optional actions. This can be done using the "want" or "must" keywords before the action.

Achievement: What the actor wants to achieve by performing the action. This results from executing the action seen from the actor's point of view.

INVEST in good stories

I	Independent
N	Negotiable
V	Valuable
E	Estimatable
S	Small
T	Testable

Sample User Stories

- Two actors:
- Project manager – the person responsible for the project
- Project member – the person working on a project

- Project Manager Story:
- "As a project manager, I want to add members to my project so that the project members can register hours on the project."

- Project Member Story:
- "As a project member, I want to register hours on a project so that my manager will know how much time I have spent on the project."

The right way to do it wrong:

1. *"As a user I want to be able to manage ads, so that I can remove expired and erroneous ads."*
2. *"As a Product Owner I want the system to have possibility of deleting ads, so that users have possibility of deleting ads."*
3. *"As a developer I want to replaced the folder widget, so that I have maintained folder widget."*
4. *"As a commercial advertiser I want to have filtering option."*
5. *"As a web user, I want to be able to filter the search by clicking the submit button in the result page using the Rhythmical Engine ARX2012, so that I can see the specific result I am looking for."*

User Story Writing Exercise

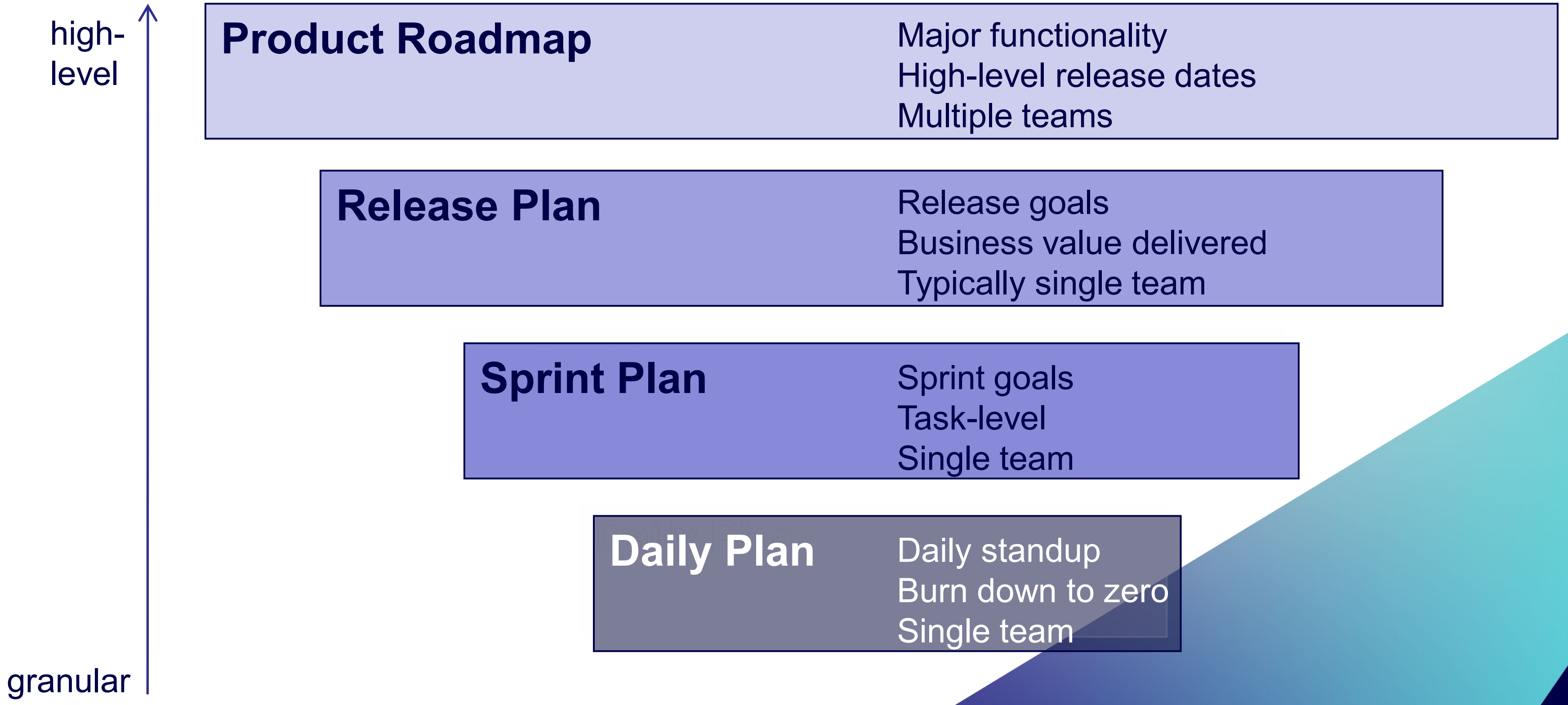
Project: Your son’s 10th birthday party

Timeline: The party is in 2 weeks

Instructions:

1. Form into one group.
2. Collaborate to write a vision.
3. Each writes three stories related to the project that you think you can do in two weeks’ time.
4. Time Box is 10 minutes.

Agile Planning Levels



high-level

granular

Product Roadmap
Major functionality
High-level release dates
Multiple teams

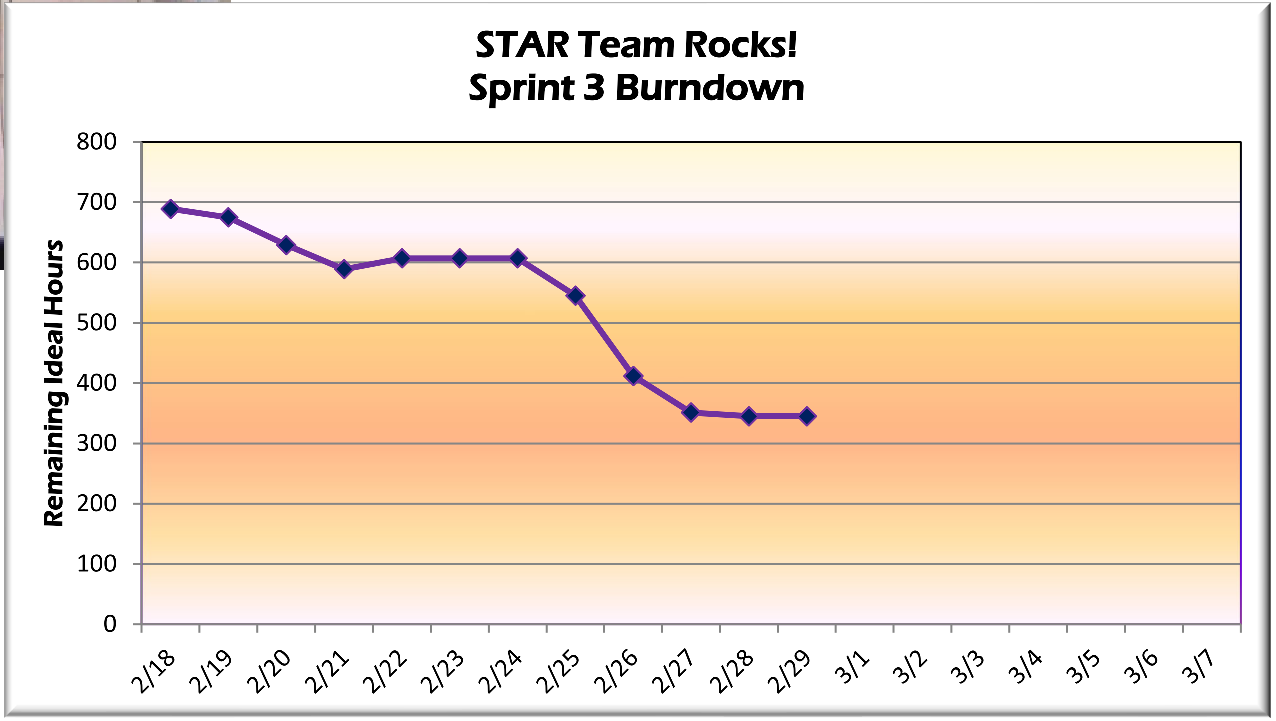
Release Plan
Release goals
Business value delivered
Typically single team

Sprint Plan
Sprint goals
Task-level
Single team

Daily Plan
Daily standup
Burn down to zero
Single team

Planning is just-in-time, just-enough and done frequently to adjust to reality.

Visual Tracking

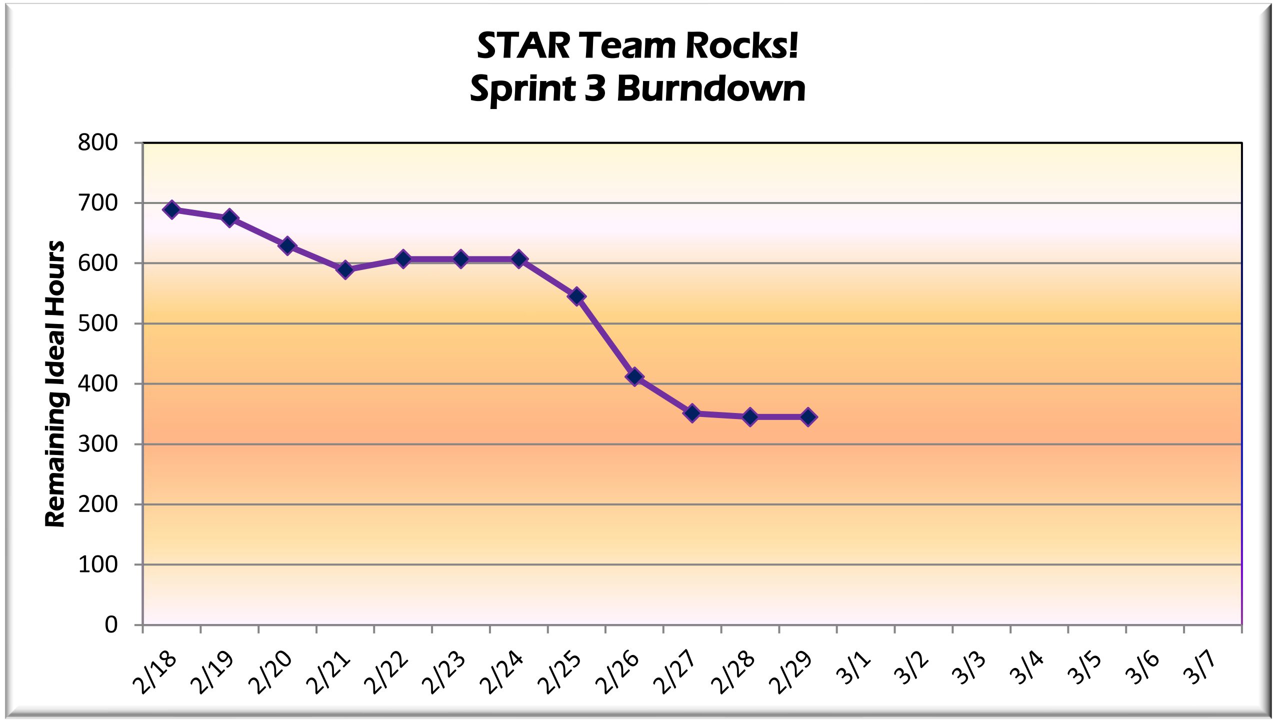


Burndown Chart

Shows work left to do in the sprint

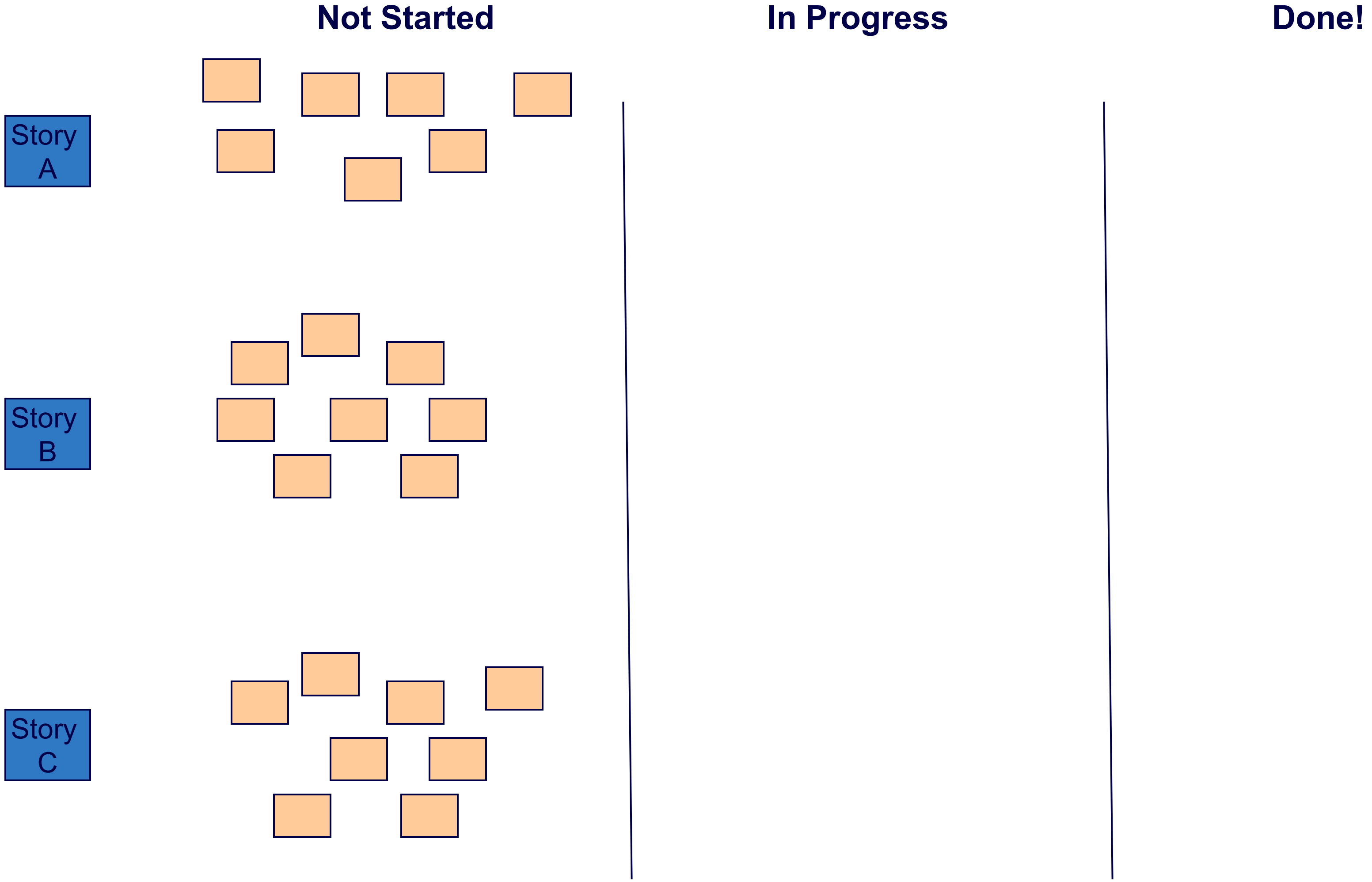
Not a justification of the original estimate

Meant to spark a conversation about "how do we get to zero?"



Story Board

Day 1 of 10



Not Started

In Progress

Done!

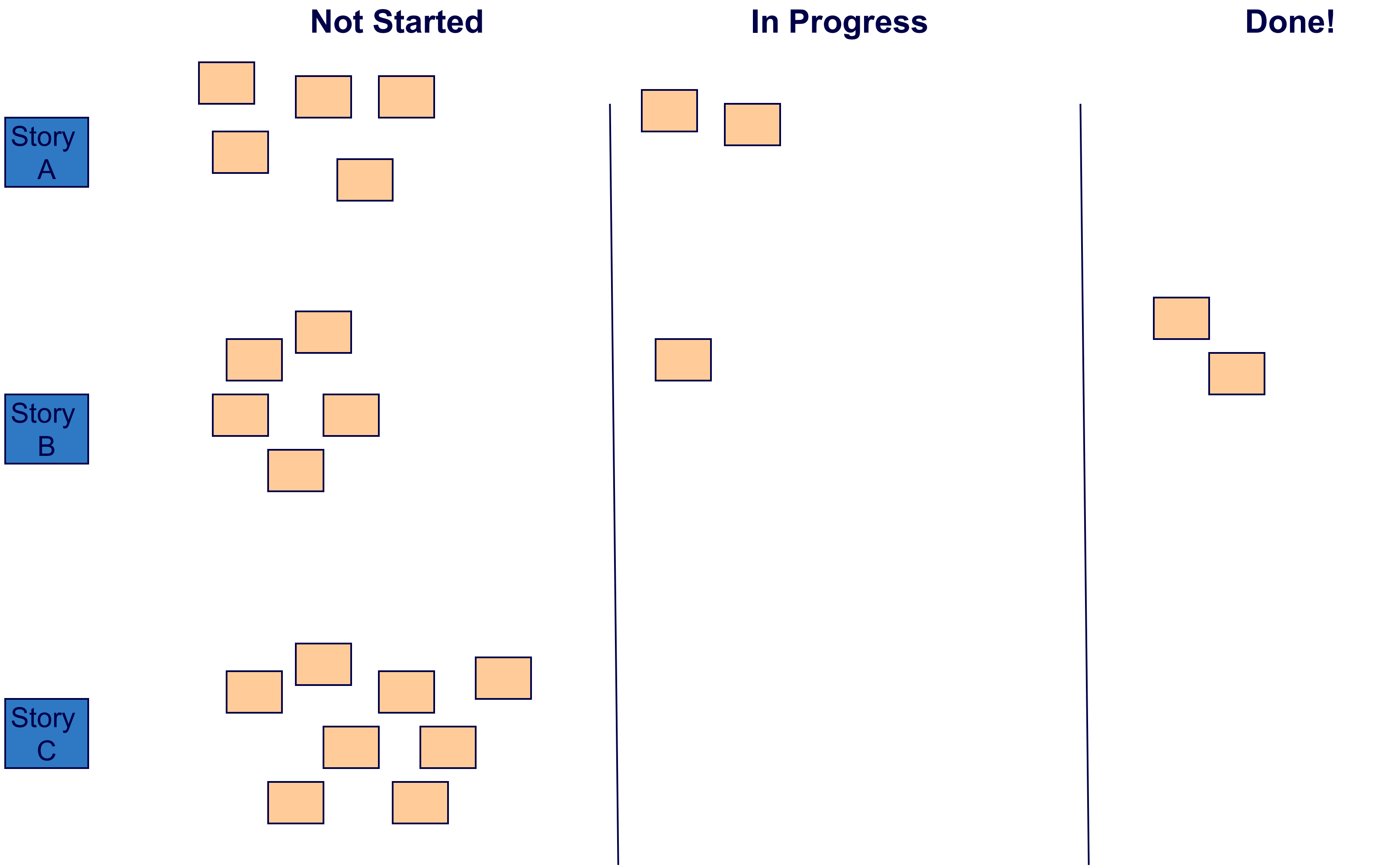
Story A

Story B

Story C

Story Board

Day 2 of 10



Not Started

In Progress

Done!

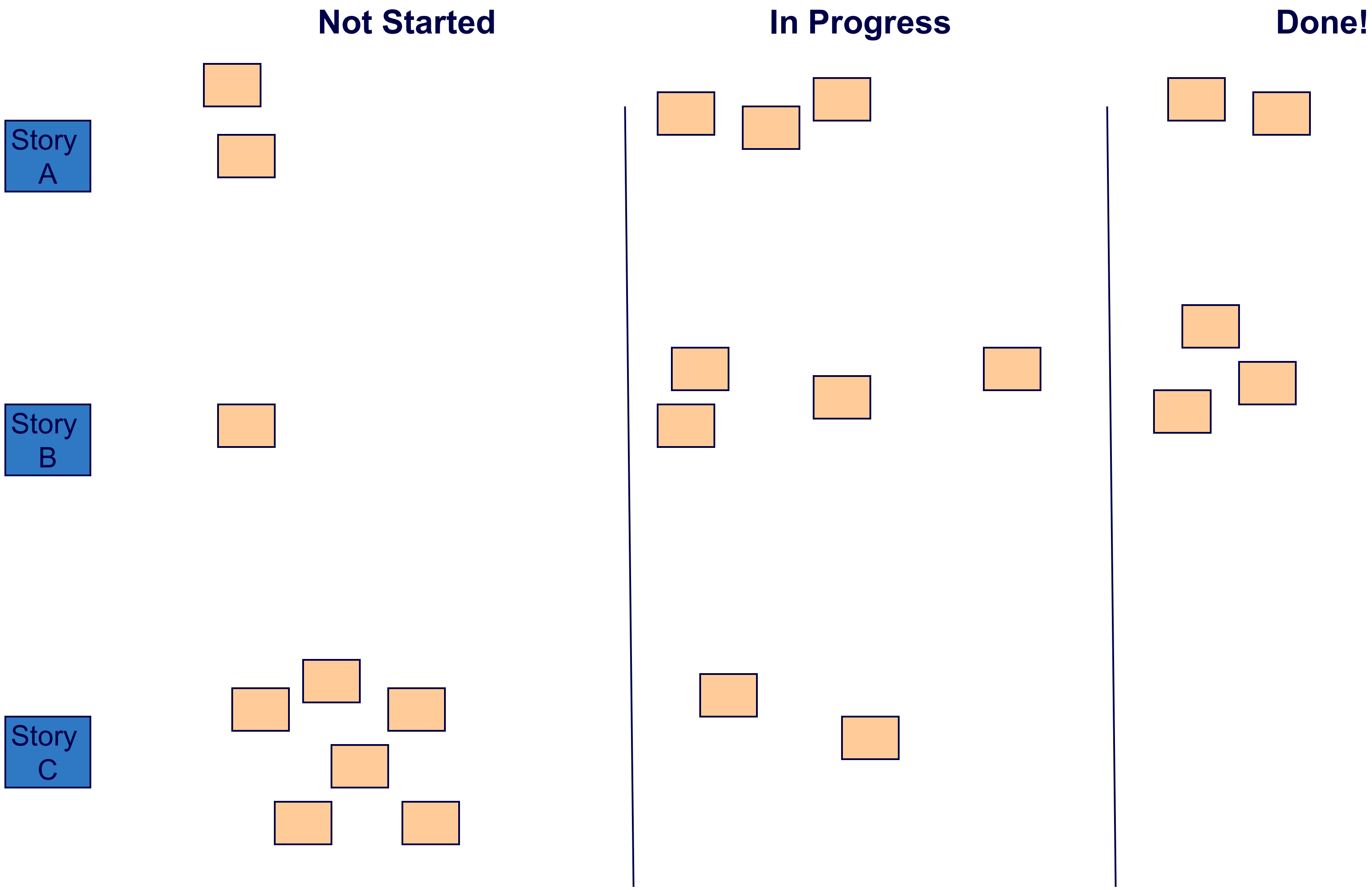
Story A

Story B

Story C

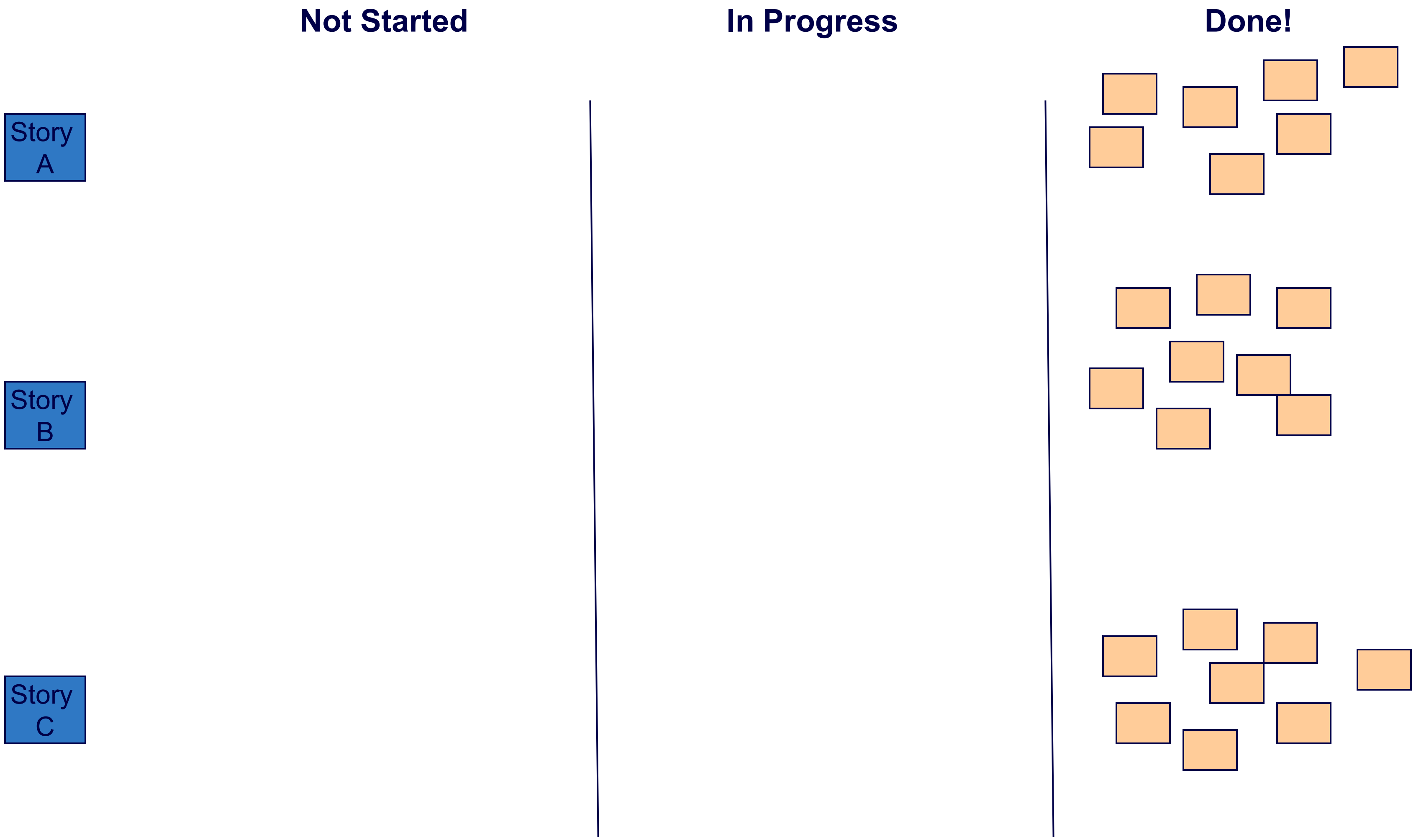
Story Board

Day 5 of 10



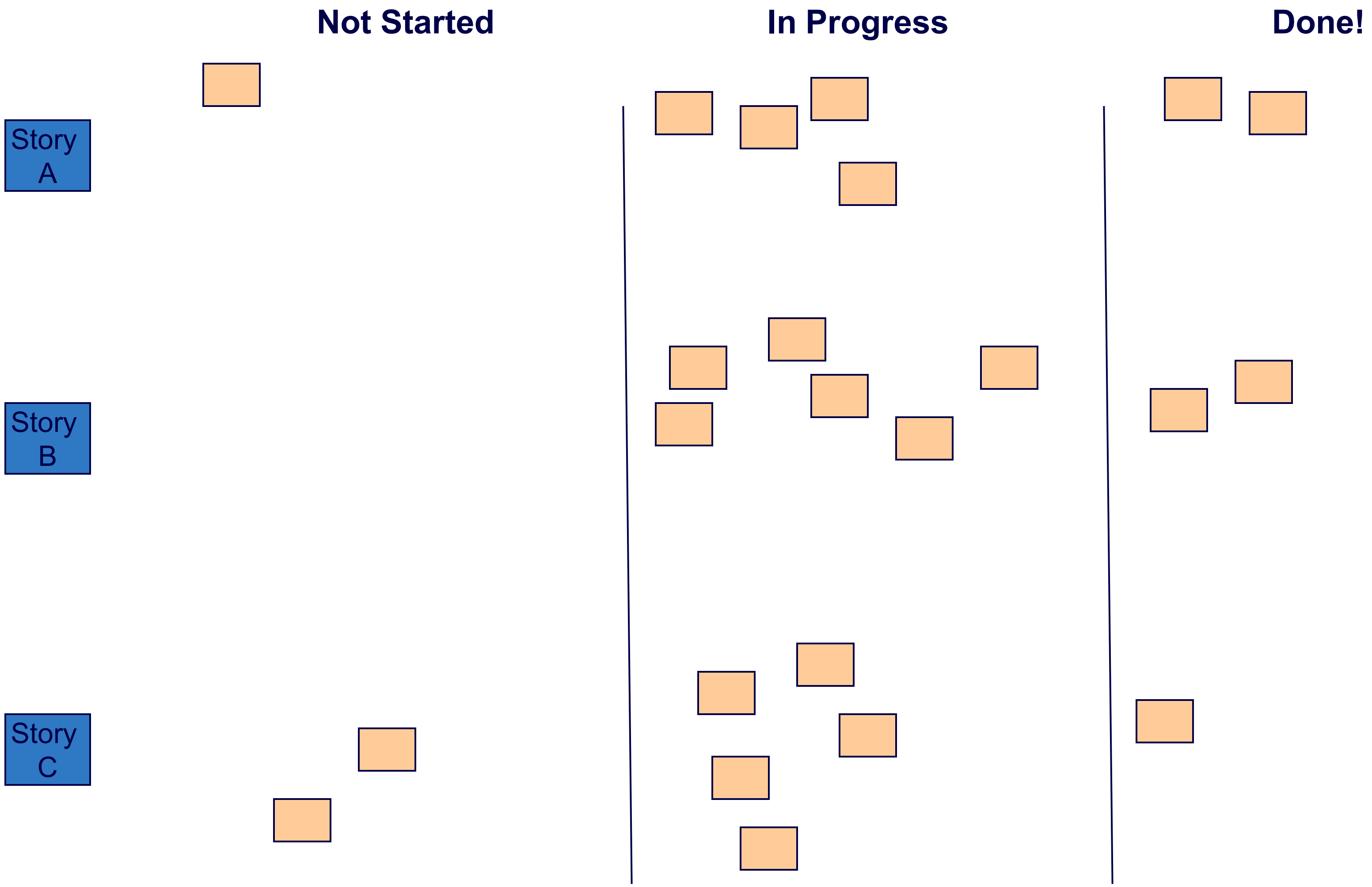
Story Board

Day 10 of 10



What’s happening to this team?

Day 5 of 10



Not Started

In Progress

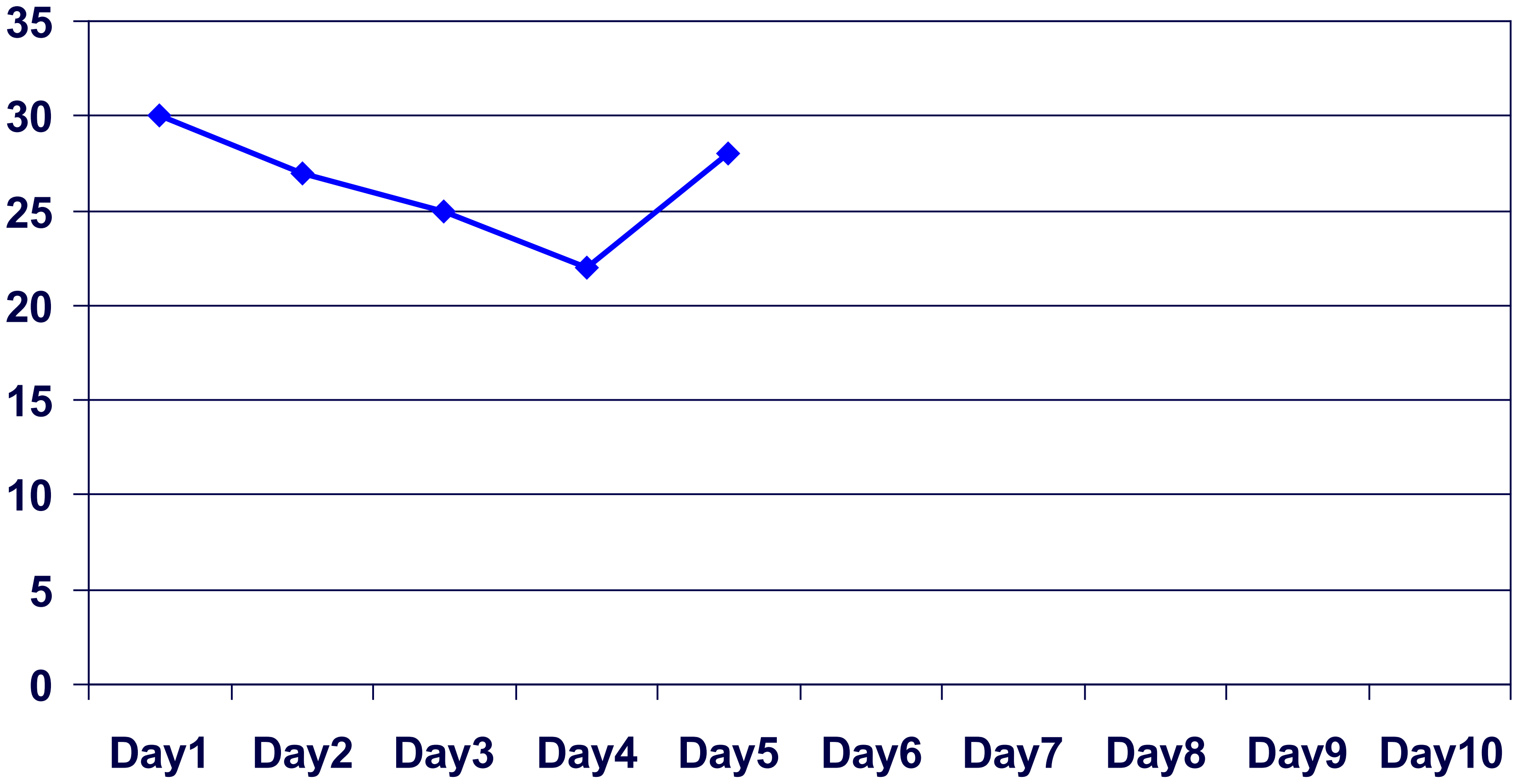
Done!

Story A

Story B

Story C

Does seeing the Burndown help?



Bottom line and perceived effects

The Burndown chart provides daily empirical data to get the team talking about getting to zero.

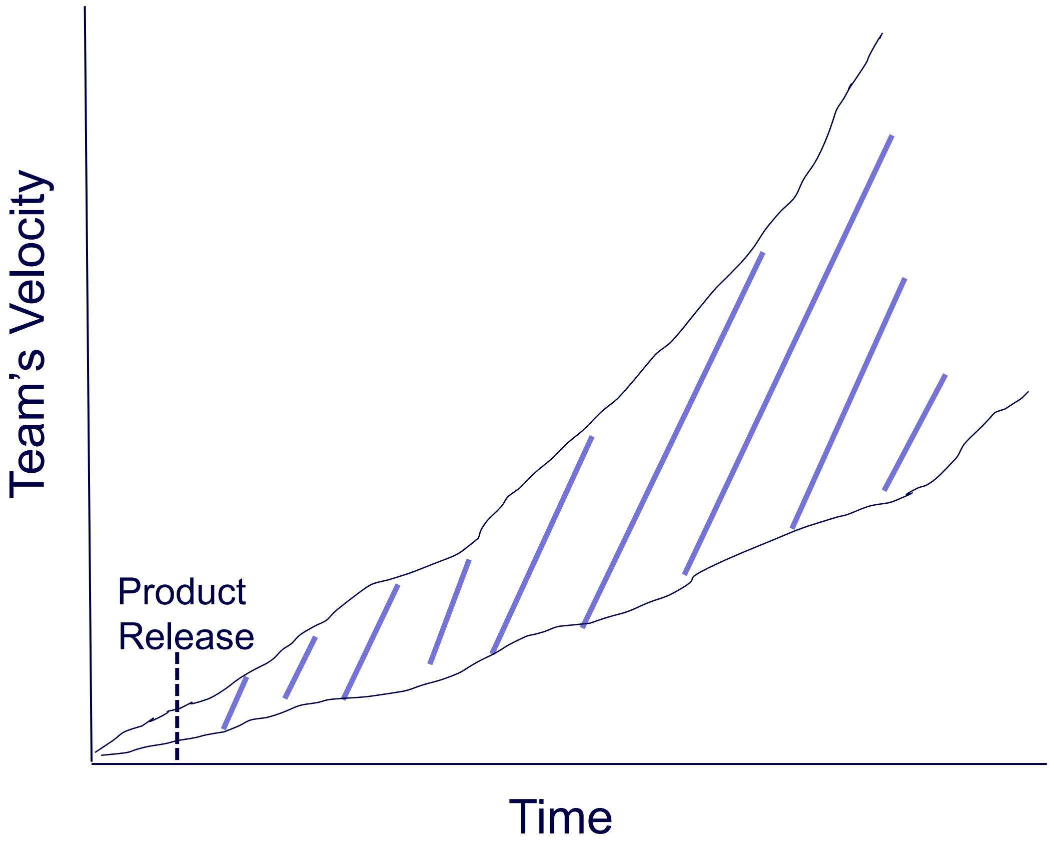
It is not for outside consumption:

- Not a reason to press on the team.
- Let the ScrumMaster know your thoughts.
- Leave the team alone to do their job.

If something comes up that needs to be addressed as important, place it on the top list for the next sprint.

However, if something comes up as a **show-stopper**, you have the prerogative to call the sprint down and start anew.

What is a Technical Debt?



Technical Debt

weak

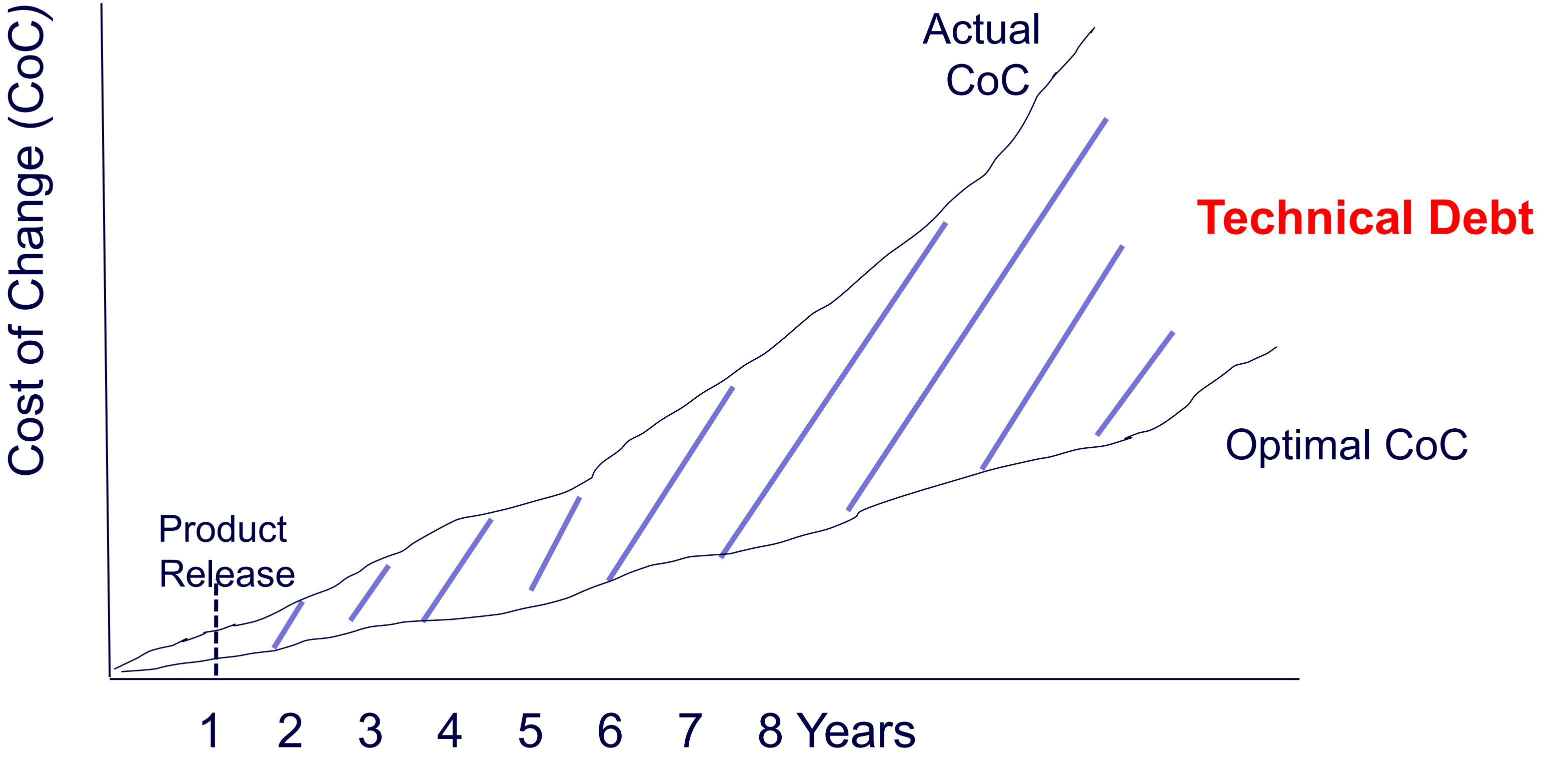
definition
of done

© 2023 Cognizant

Where does technical debt come from?

- Business pressures
- Lack of process or understanding
- Lack of building loosely coupled
- Lack of test suite
- Lack of documentation
- Parallel Development
- Delayed Refactoring
- Laziness

A technical debt due to hurrying



What will happen to this team’s software over time?

How do you avoid Technical Debt?

1. Achieve a robust definition of "done."
2. Ensure quality is held high and constant – no cutting corners
3. Build in non-functional requirements as user stories or acceptance criteria (scaling, performance, etc.)
4. Use engineering practices to avoid debt
    - Simple design
    - Continuous integration: "OK, who broke the build just now?"
    - Automated testing
    - Pair programming – coding is *thinking*, not typing
    - Code as documentation
    - Refactoring as a way of life
    - Coding standards – have them use them

cognizant®

Purpose of the Daily Standup

PEER PRESSURE

Fine-grain coordination

Daily Commitment

Raise impediments

Focus on the **few**

What is a Stand Up?

- This is short (15 minutes or less).
- Everyone on the team attends.
- Everybody stands.
- Not a status meeting to report to a manager.
- Other people not members of the team may observe but cannot join in the conversation.
- Each member answers three questions:

  (1) What did you do since the last standup?

  (2) What are you planning to do by the next standup?

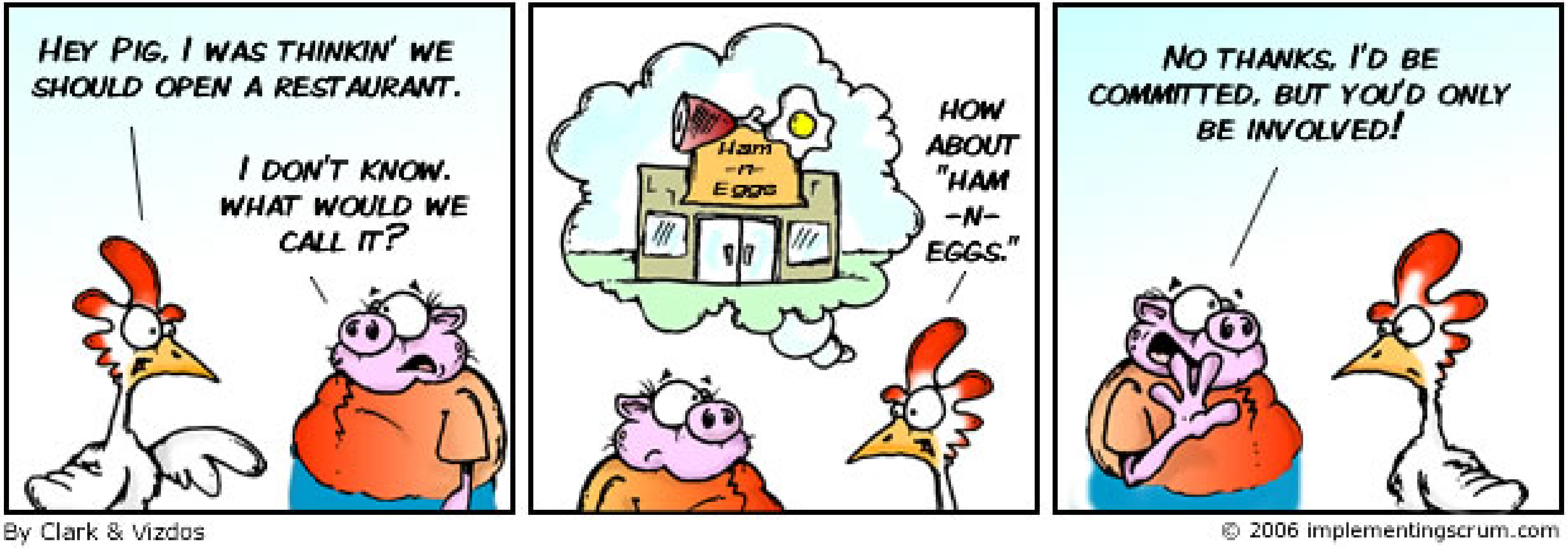
  (3) Any blocks or impediments that are in your way?

Sample Impediments

- My PC is being re-imaged today.
- I still haven’t got the software I ordered a month ago.
- I need help debugging a problem with _______.
- I’m struggling to learn _______ and would like to pair with someone who’s done it before.
- I can’t get the vendor’s tech support group to call me back.
- Our new contractor can’t start because no one is here to sign her contract.
- I can’t get the _____ group to give me any time, and I need to meet with them.
- The department VP has asked me to work on something else "for a day or two."

Tidbits about Daily Standup

- Members of Scrum Team are known as **Pigs** because they are committed to delivering Sprint Goal.
- People who are involved but not dedicated to the project are known as **Chickens** - they attend Scrum meetings as observers.


HEY PIG, I WAS THINKIN' WE SHOULD OPEN A RESTAURANT.
I DON'T KNOW. WHAT WOULD WE CALL IT?
Ham -n- Eggs
HOW ABOUT "HAM -N- EGGS."
NO THANKS, I'D BE COMMITTED, BUT YOU'D ONLY BE INVOLVED!
By Clark & Vizdos
© 2006 implementingscrum.com

Purpose of the Sprint Review

Get direct feedback

Show and tell

Share insights

Ask for help

True-up

What is a Sprint Review?

- At the end of the Sprint, the Product Owner, Team, ScrumMaster, and Stakeholders come together and see a demo of what the team has produced.
- The Product Owner gathers feedback from everyone on ways to improve what’s been built.
- This feedback is then incorporated into the Product Backlog.

Sprint Review Preparation

- As little as possible
- Low tech, high impact
- No PowerPoint – this is not about slideware; this is about showing real stuff
- Train team in "value first" talking
- Be ready to ask for help (if needed)
- Developers present their work

Purpose of the Retrospective

INSPECT AND ADAPT

Do even better next time

Look back at the HOW, *not what?*

Retrospective Format

Well-facilitated activities and discussions that get at:

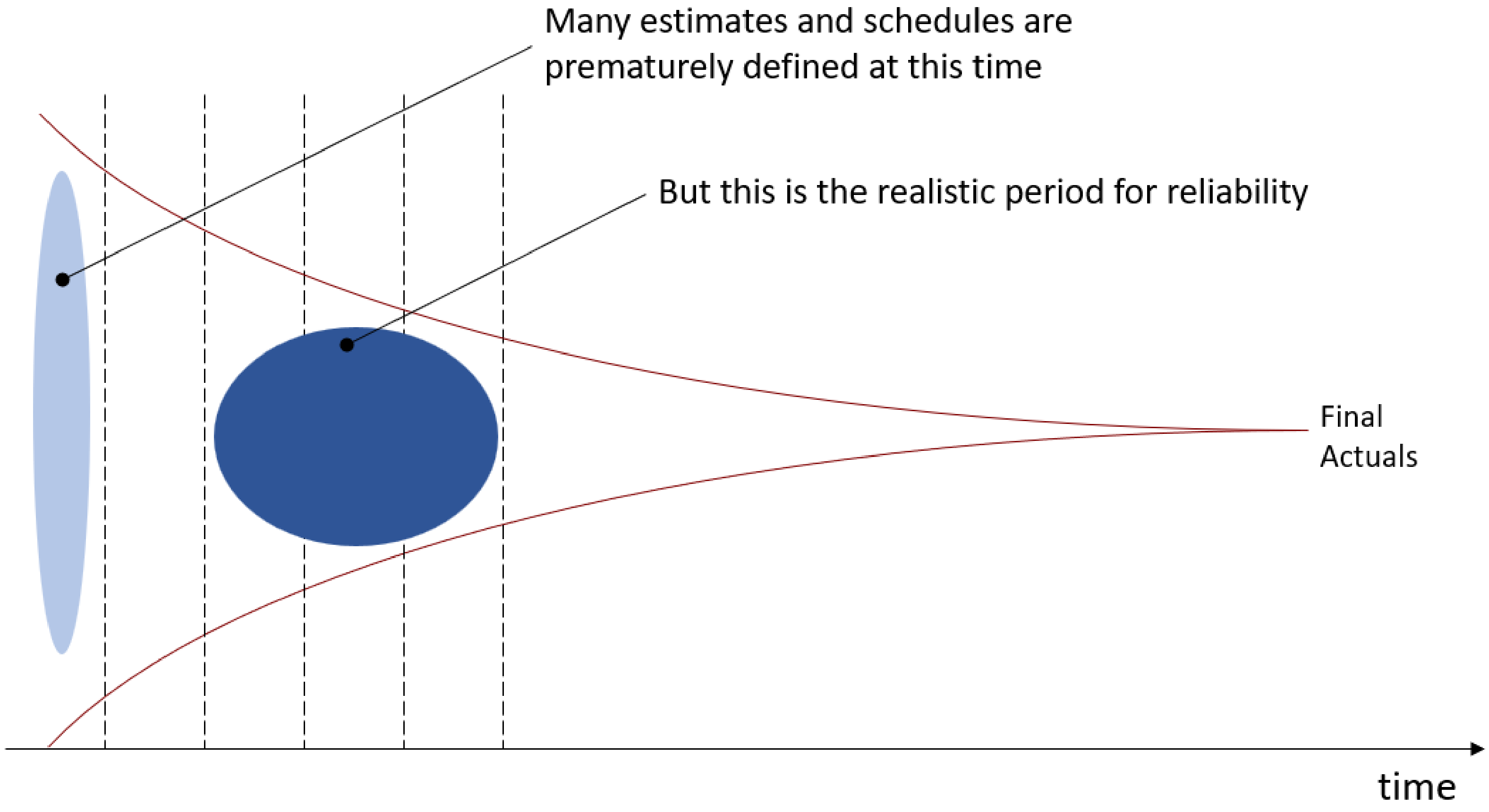
- Team’s progress
- Improvement ideas
- Agreements on what to improve

Facilitated by the ScrumMaster (the Guardian of Quality and Performance), to initially start the ball rolling.

Unfortunately, only team members sit in here for often this is when they vent their problems and, as a team find possible solutions to their frustrations to improve.

Estimation & Release Planning

In the beginning, a guess is as good as a plan.



Many estimates and schedules are prematurely defined at this time

But this is the realistic period for reliability

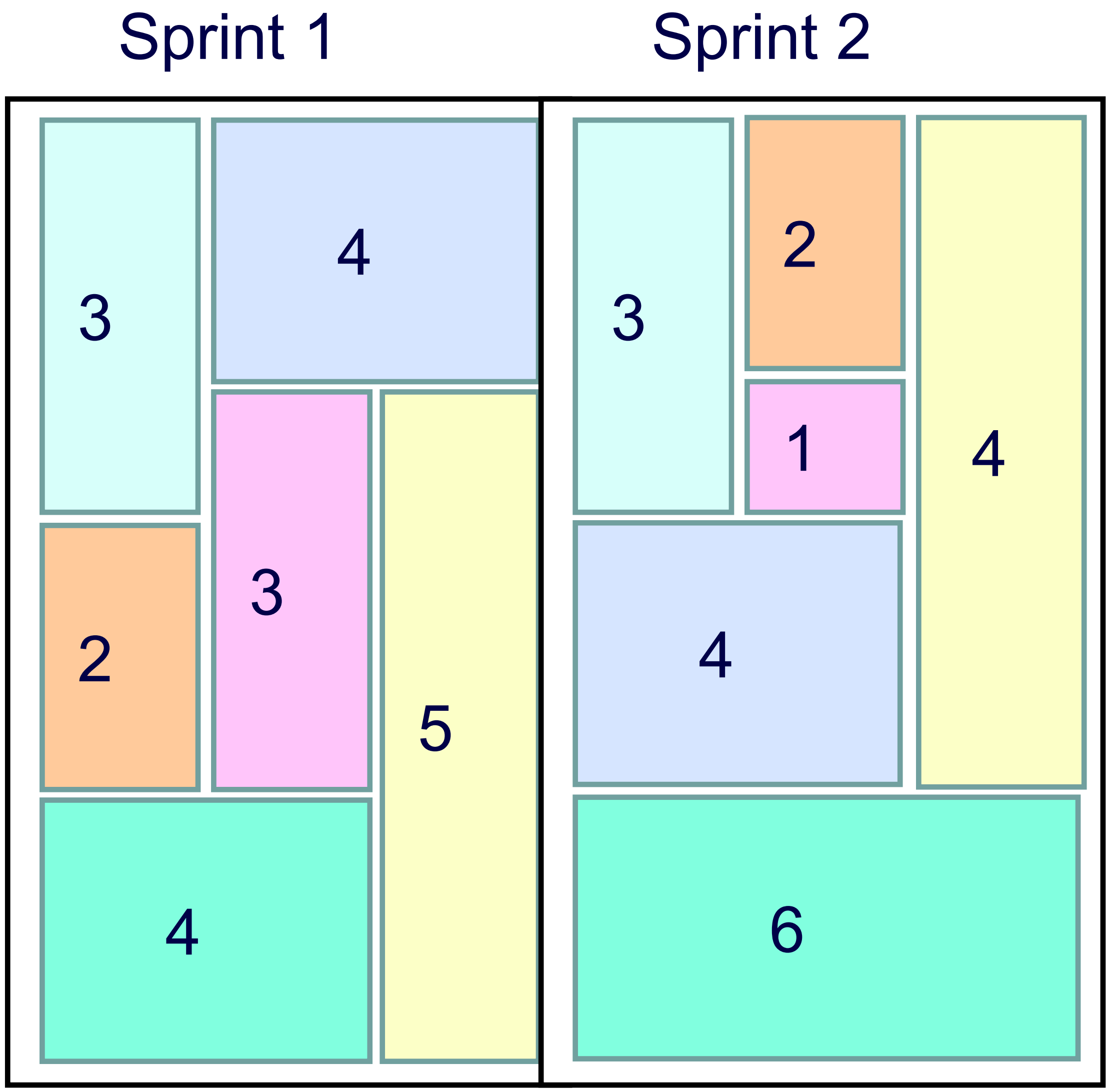
Release Planning

A **release** comprises multiple iterations or sprints.

Each **sprint/iteration** can be thought of as a same-sized box.

Stories are put into each box (iteration) until it's full.

The size of the box is the planned **velocity**.



What Release Planning does

- Sizes all the work in the Product Backlog
- Slots the remaining work into Sprints
  - Based on the team’s actual velocity
  - Based on the team’s sizes
- Creates releases – units of business value that will be delivered
- Gives the team a longer vision to shoot for and adjust after every sprint
- Gives stakeholders a view of where the team is going and when along the way they expect to deliver

Release Planning vs. Sprint Planning

- Sprint Planning is about planning what’s included in the next iteration.
- While Release Planning is about planning multiple Sprints, in order to predict when a release (or releases) might be delivered.
- In release planning the team can choose between "ideal days" and "story points." Regardless of which they choose, they still do sprint planning in "hours".

Purpose of the Sprint Planning

Work

Choose

Understand

Task

Volunteer

Fresh start

Commit to shared goals

Sprint Planning Components

Sprint **Goal** – The "headline" of what the team will accomplish in the sprint. This is a way for the team to "true back" as tasks, and even stories, change.

Sprint Backlog gets further refined into:

1. **User Stories** – The body of work the team has committed to complete.
2. **Tasks** – All the tasks needed to get each user story done.

How to do a Sprint Planning Part 1

- The Product Owner and the team negotiate on stories to tackle for the iteration
- Suggested time box is 2 hours for a 2-week iteration with 5-9 team members.
- When the team agrees to tackle on the work, the Product Owner adds the corresponding stories into the sprint backlog.

How to do a Sprint Planning Part 2

- The Product Owner is typically asked to leave at this time but encouraged to be on call.
- Suggested time box is 2 hours for a 2-week iteration
- ScrumMaster helps the team decompose the sprint backlog items into tasks.
- The ScrumMaster records the tasks and assign owners respectively.
- Once defined, the team commits to the tasks on hand and then disperses to get to work.

Mechanics of Sprint Planning

Planning Units

Ideal hours or days

Each card = points (a point usually equals a day)

Sprint Rules

» No changes during

» No change to sprint duration

» Team takes work until they're full

Purpose of Estimates

Not for tracking time

Scrum is results-oriented, not effort-driven

Velocity

» How much work does the team actually accomplish in a sprint

Sprint Planning Duration

To choose the sprint duration, consider these questions:

- How long can the business "sit still"?
- How many chances at learning do we want to get?
- How short does the sprint duration need to be to break us of the habits from plan-driven thinking?

Communicating Plans and Progress

- What's the difference between a traditional and an agile report?
- What is measured by Agile?
- What are sample scrum visual reports and progress panels?

Traditional Status Report

Project Cash – Monthly Status

Project Manager: Jack Greenley

Phase: Design

Customer: Jennifer Carter

Delivery Date: July 2008

Executive Status Summary

Approximately 45% complete with project. Requirements finalized and team is 65% through design.

SCOPE | SCHED. | BUDGET

OVERALL

Key Milestones	
Final Requirements	1/15 Completed
Vendor SOW	2/15 Completed
Final Design	4/21 See Issue
Start Development	4/24
Final Test Plan	3/14 On Track
SARBOX Checklist	5/1

Key Issues/Risks & Mitigation Plans

Design Resource Issue:

- **– 65% through design when 2 key resources became unavailable, one due to serious illness, one to reassignment**
- **–Pursuing contractor replacement and/or vendor proposal from specialized staffing firm**
- **–Impact to final design deliverables if not resolved in 2 weeks**

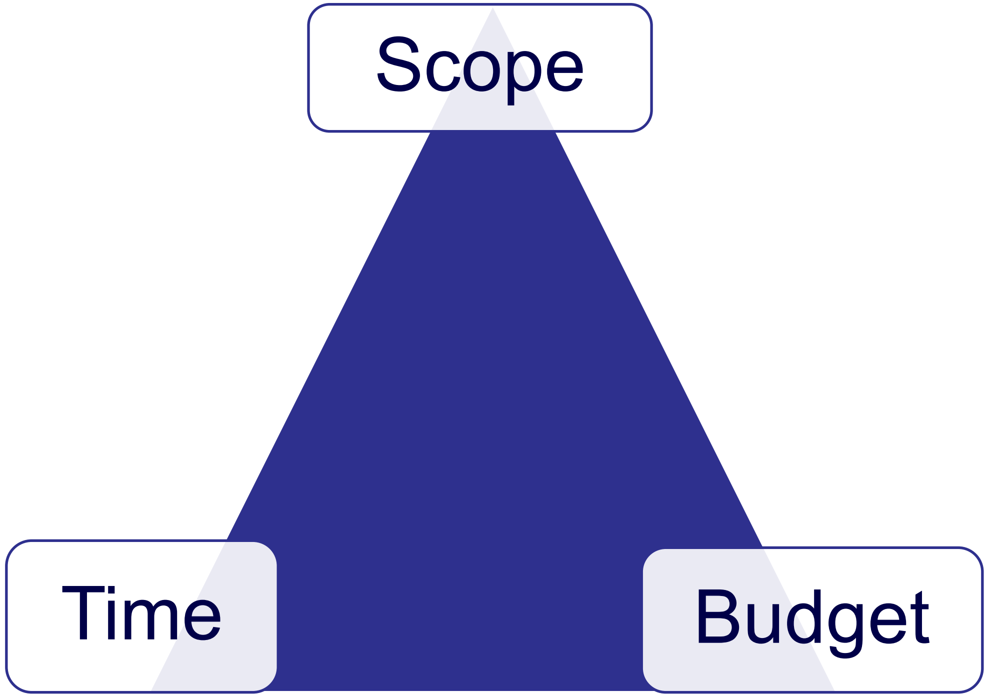
Scope Management:

- **–Change Request #3 with vendor to be approved by 3/11**
- **–Bill Pay Group's need for audit reports could cause significant schedule impact. Change request under review.**

Project Cost Information

Actual Billable Labor Cost	Actual Non-Labor Cost	Total Billable Actuals	Current Approved Budget	Project Budget Expended
$832,900	$320,000	**$1.15M**	$2.1MM	**55%**

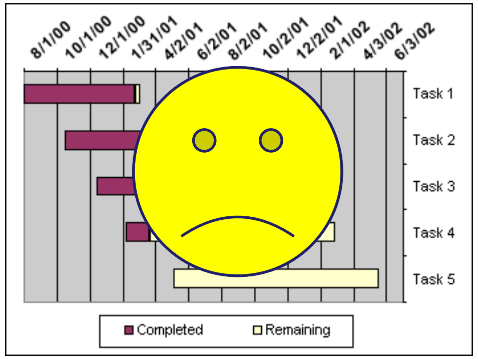
What is being measured


Scope

Time

Budget

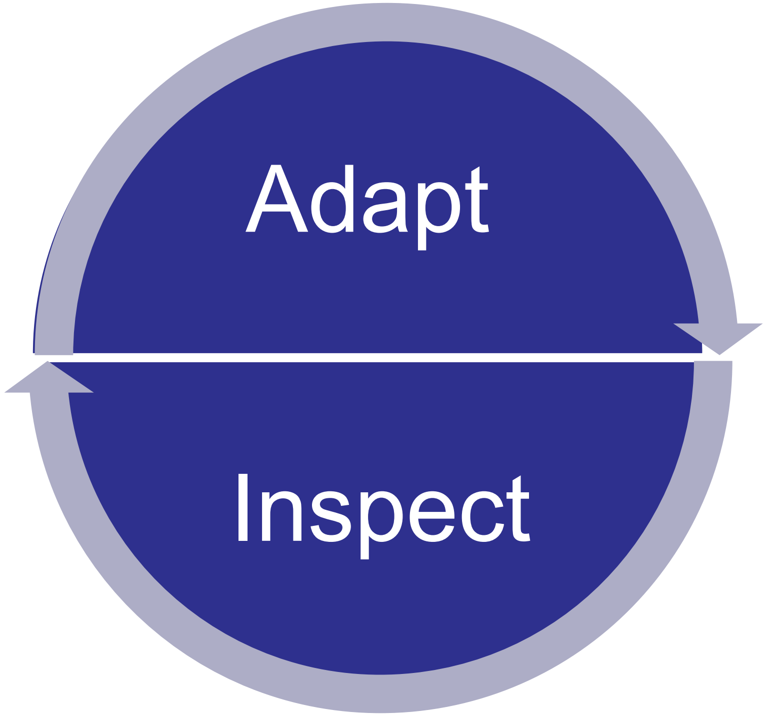
Conformance to Plan



Plan-driven vs. Agile approach

Status Indicator	Traditional	Agile
Scope	Scope inflexible. Change requests must be approved.	Scope flexible. The Product Owner is free to reorder priorities each sprint.
Schedule	Assumes prediction will be correct. Manage to milestones, % tasks done.	Assumes change. Rigid time box. Manage to sprint goals. Prediction accuracy better each sprint.
Cost	Managed to original budget + change orders. Measure = % spent.	Managed to original budget. Measure = % spent compared to business value delivered.
Risks/Issues	Managed by PM in lists, updated periodically.	Managed by Team daily as impediments.

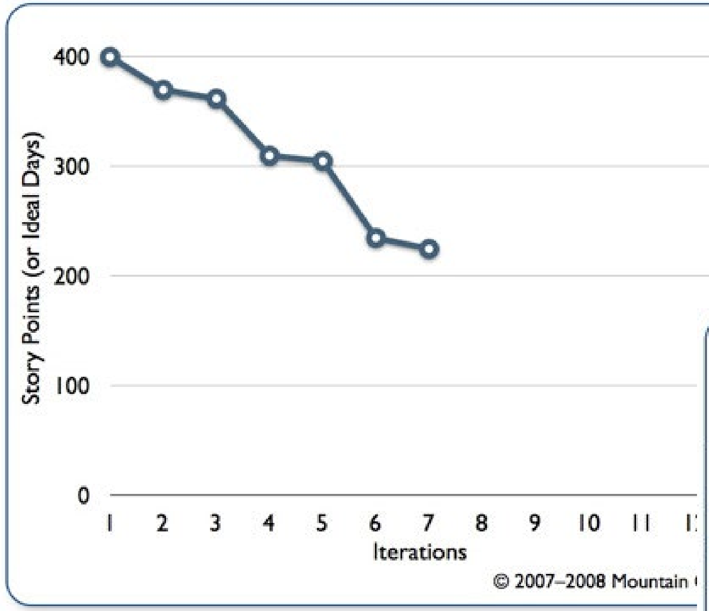
What is Agile measuring?



Conformance to Business Value

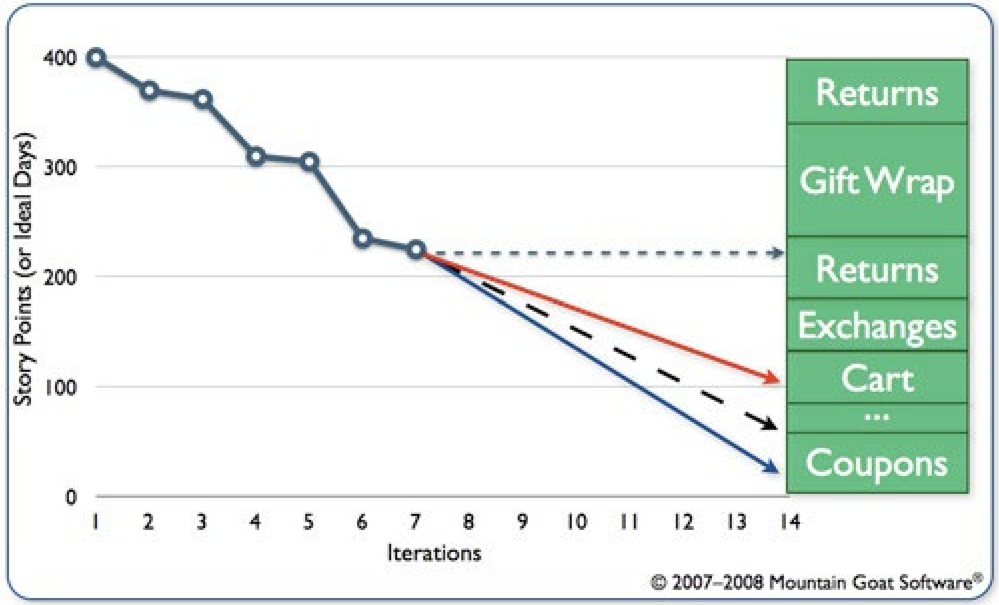
Product Burndown examples

Basic product Burndown



© 2007–2008 Mountain

Predictive product Burndown



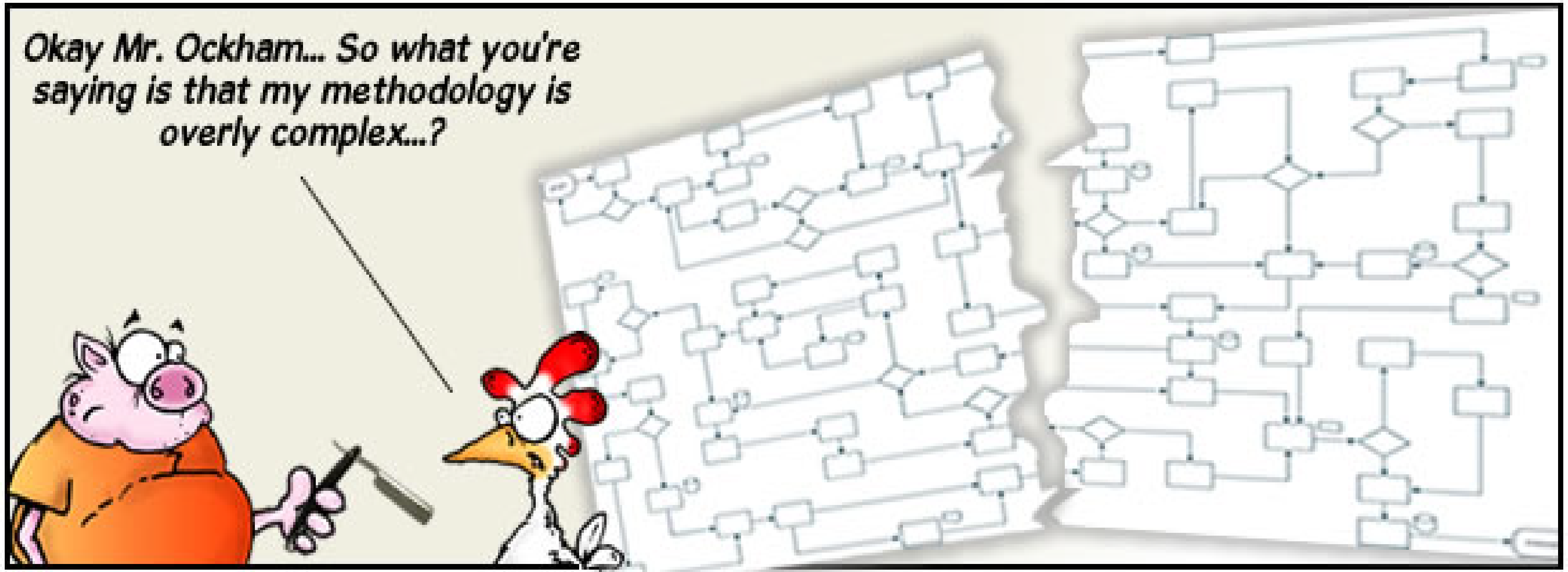
© 2007–2008 Mountain Goat Software®

That’s pretty much what a Product Owner in Agile does.

We have learned much this session.

Practice the Agile/Scrum theory for retention.

Agile is meant to be simple; don’t complicate it.


Okay Mr. Ockham... So what you're saying is that my methodology is overly complex...?

By Clark & Vizdos

© 2008 implementingscrum.com

cognizant

Thank you

Leo Primero

Certified ScrumMaster | Certified Scrum Product Owner | Certified SAFE Agilist

Email: 522033@cognizant.com