Introduction to Scrum

Based on:

Certified Scrum Master - Ken Schwaber
Certified Scrum Master - Bas Vodde
Scaling Agile & Lean - Bas Vodde, Craig Larman

Overview

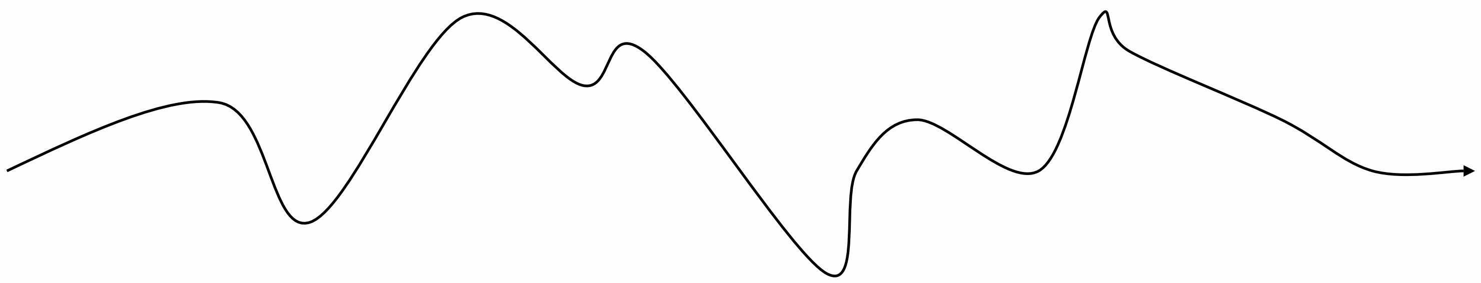
Predictive

Start with
Plan and all
requirements

End with all
requirements
completed

Scrum - Empirical

Start with
Goals and
some priority
requirements



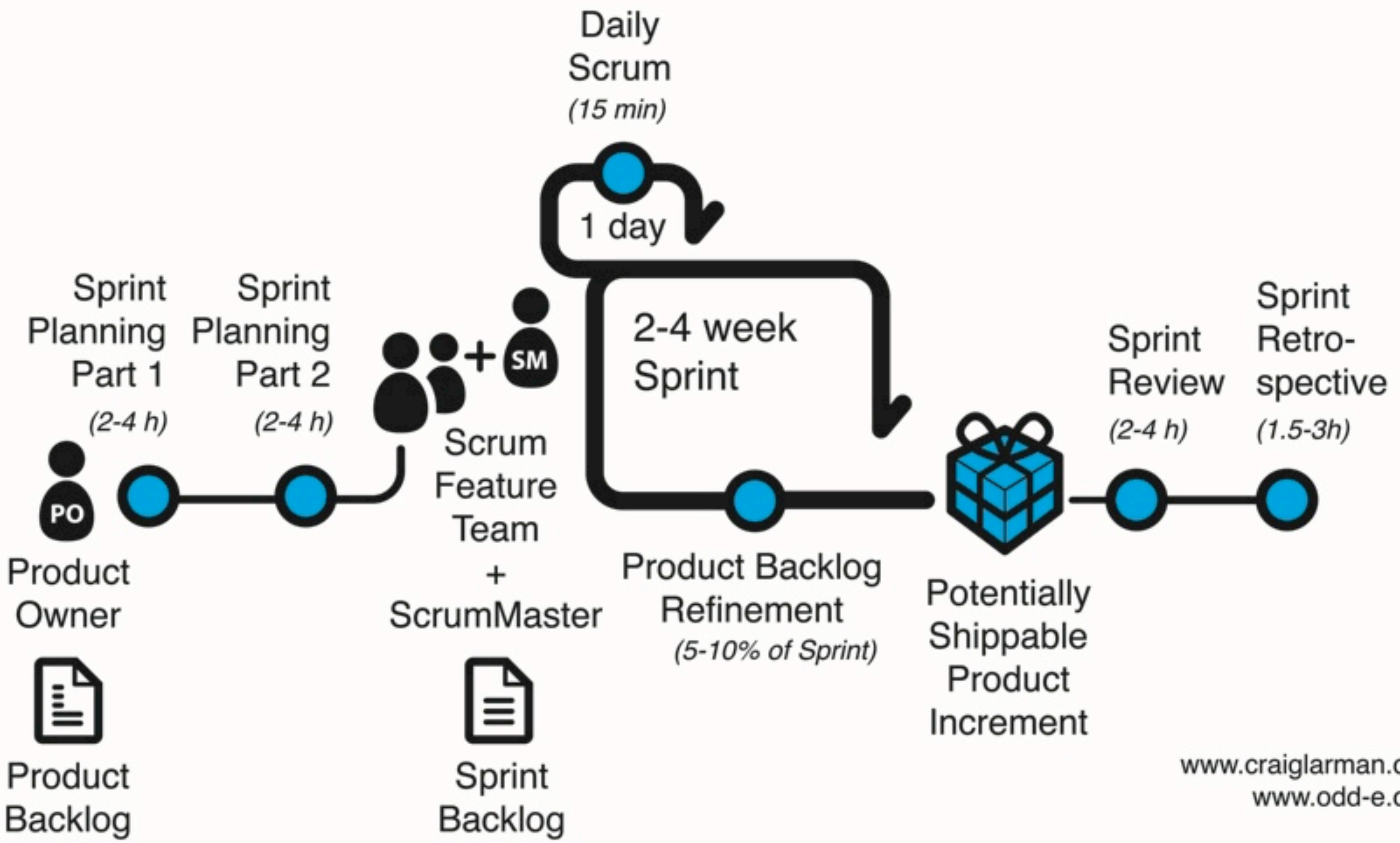
End with
Goals met

Odd-e

Empirical Process Control

- Uses Inspection and subsequent adaptation to optimize realization of goals.
- Transparency is required for inspection and adaptation.
- Transparency requires courage and change in reward systems.

HOW'S YOUR PROJECT COMING ALONG?
IT'S A STEAMING PILE OF FAILURE.
IT'S LIKE FIFTEEN DRUNKEN MONKEYS WITH A JIGSAW PUZZLE.
HOW'S YOUR PROJECT COMING ALONG?
FINE.
scottadams@aol.com
www.dilbert.com
3-15-07 © 2007 Scott Adams, Inc./Dist. by UFS, Inc.
© Scott Adams, Inc./Dist. by UFS, Inc.



Daily Scrum (15 min)
1 day
Sprint Planning Part 1 (2-4 h)
Sprint Planning Part 2 (2-4 h)
2-4 week Sprint
Sprint Review (2-4 h)
Sprint Retro-spective (1.5-3h)
PO
Product Owner
SM
Scrum Feature Team + ScrumMaster
Product Backlog Refinement (5-10% of Sprint)
Potentially Shippable Product Increment
Product Backlog
Sprint Backlog

www.craiglarman.com
www.odd-e.com

Roles

Scrum Roles

Product Owner

- ► Defines the features of the product, decides on release date and content
- ► Is responsible for the profitability of the product (ROI)
- ► Prioritizes features according to market value
- ► Can change features and priority every 30 days
- ► Accepts or rejects work results

ScrumMaster

- ► Ensures that the team is fully functional and productive
- ► Enables close cooperation across all roles and functions and removes barriers
- ► Shields the team from external interferences
- ► Ensures that the process is followed. Invites to daily scrum, iteration review and planning meetings

Team

- ► Cross-functional, seven plus/minus two members
- ► Selects the iteration goal and specifies work results
- ► Has the right to do everything within the boundaries of the project guidelines to reach the iteration goal
- ► Organizes itself and its work
- ► Demos work results to the Product Owner

Product Owner

- Should have a vision for the product.
- Defines the features of the product, decides on release date and content.
- Is responsible for the profitability of the product (ROI).
- Prioritizes features according to market value.
- Can change features and priority every sprint.

Change!!!

- Product Owner is used to “throwing the project over the wall” and holding engineering/development responsible for meeting needs. Scrum puts this responsibility back on the Product Owner and customers through the inspect and adapt and the Sprint Review.
- Make decisions regarding ROI every Sprint end.
- The single wringable neck!

Scrum Team

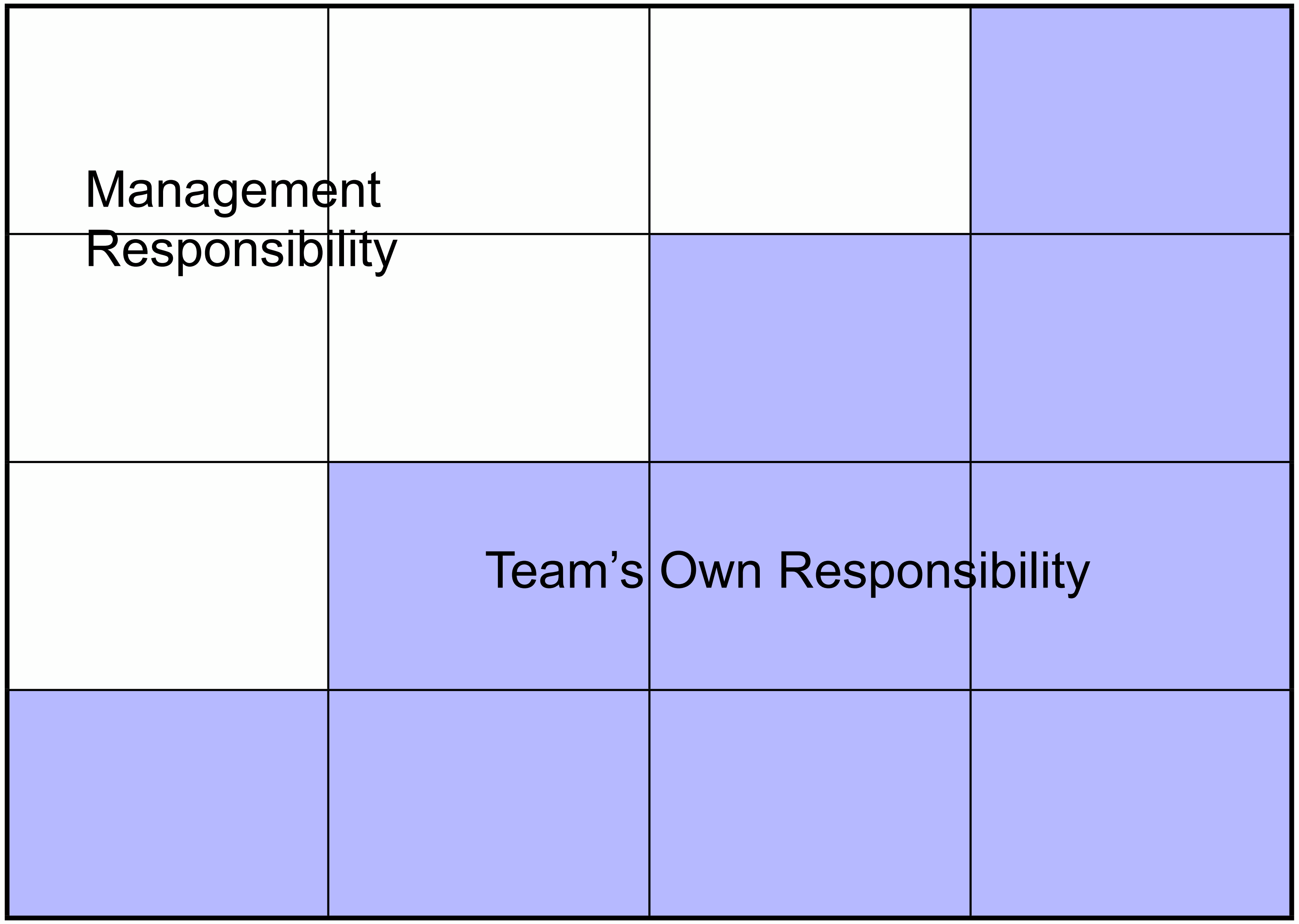
- Self-organizing;
- Cross-functional with no roles;
- Seven plus or minus two;
- Responsible for committing to work;
- Authority to do whatever is needed to meet commitment;
- Open, collocated space;
- Resolution of conflicts;
- Working agreements:

Working agreements

- Examples:
  - Time of daily scrum
  - Penalty for being late
  - Everyone integrates daily, not just before they leave
  - When you see ugly code -> refactor
  - Whenever you are unsure, ask someone
  - Pair programming and tdd rules

- Related
  - Coding standards
  - Definition of “done”

Authority Matrix



	Manager-led teams	Self-Managing teams	Self-Designing teams	Self-Governing teams
Setting overall direction				🟦
Designing the team and its organizational context			🟦	🟦
Monitoring and managing work process and progress		🟦	🟦	🟦
Executing the team task	🟦	🟦	🟦	🟦

Text from: "Leading teams"
By Richard Hackman

As a ScrumMaster, you are responsible for:

- Removing the barriers between development and the customer so the customer directly drives development;
- Teaching the product owner how to maximize ROI and meet their objectives through Scrum;
- Improving the lives of the development team by facilitating creativity and empowerment;
- Improving the productivity of the development team in any way possible; and,
- Improving the engineering practices and tools so each increment of functionality is potentially shippable.

Leader and Facilitator

ScrumMaster

- Is **not** a project manager
  - – Team manages itself
    - Organizes it's internal work
    - Coordinates with other teams and people
  - – Product Owner makes decisions on content/schedule
  - – Scrum Master removes impediments when team asks
- Does **not** have authority -> team makes decisions
- Challenges the organization, key-role in the change
  - – However, a dead scrum master is a useless scrum master

Traditional managers

- Tell people what to do and then make sure they do it properly
- Maintain the right to authorize decision
- Limit the information or resources available to them

The change from traditional manager to scrum master is very hard!

Old habits die hard.

Day of a ScrumMaster (1)

Find what to improve by asking:

How is my Product Owner doing?

- Is the product backlog in good shape?
- Does he understand his benefits of Scrum?

How is my team doing?

- Are your team members working together?
- Is there conflict in the team, are they resolving that?
- Is the team making decisions?

Day of a ScrumMaster (2)

How are our engineering practices doing?

- Is your team caring and improving them?
- How is the test automation?
- Is the team expanding done?

How my organization doing?

- Is there inter-team coordination?
- What organizational impediments are in the way?
- HR practices?

Full list at:
http://www.danube.com/blog/michaeljames/a_scrummasters_checklist

Ask the team!

- When you don't know what to do, ask the team:
- Example:
  - I noticed <situation> what shall we do?
  - I observed <situation> is that important?
  - I feel <feeling> do you share that?
  - Shall we try to find out why <situation>?
  - What do you think we should do?
  - Who has any idea about …?
  - Is this useful?
  - What have you decided?
  - What should I do?

Meetings

Sprint Planning Meeting

Product Backlog

Team Capabilities

Business Conditions

Technology Stability

Executable Product Increment

→ Review, Consider, Organize →

Next Sprint Goal

Product Backlog

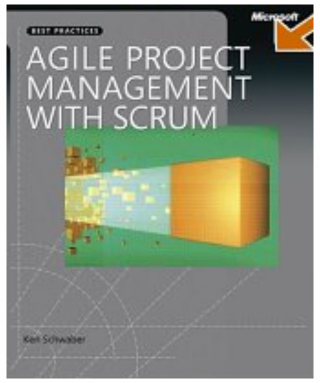
Sprint Backlog

2 – 4 hours meetings … time boxed

Doing Sprint Planning

- Sprint planning part 1
  - Clarification on backlog item
  - Tentatively select backlog items for sprint
- Sprint planning part 2
  - Hours based sprint planning
    - Don’t forget backlog refinement
    - Maintenance?
  - Commit to the items, get back to PO

Sprint Planning Part 1 Rules

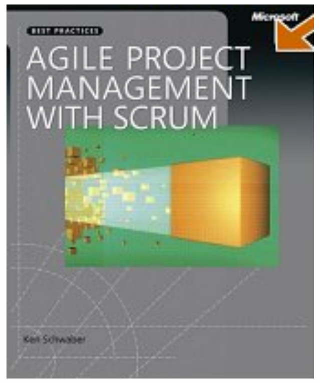


- Attendees: ScrumMaster, Team, PO
- Preparation: PO must prepare the backlog prior to meeting

- Goal: Select backlog items it believes it can commit to
- Goal: Analyze the Product Backlog
- Timebox: 4 hours

- Team: Make suggestions for items
- PO: Final decision on what items to work on
- Team: Final decision on how much items to work on

Sprint Planning Part 2 Rules



- Attendees: ScrumMaster, Team
- PO: Must be available

- Goal: Figure out how to convert the backlog items to implemented working functionality
- Timebox: 4 hours

- Output: Sprint backlog
- Output: Final commitment

Daily Scrum

- Daily 15 minute status meeting;
- Same place and time every day;
- Chickens and pigs;
- Three questions;
  - What have you done since last meeting?
  - What will you do before next meeting?
  - What is in your way?
- Impediments; and
- Decisions

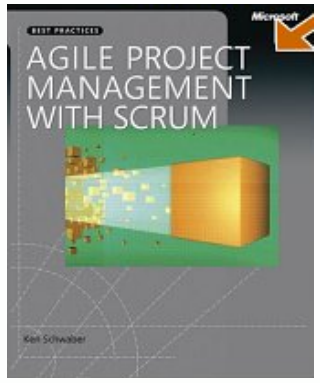
Usefulness

- Keep asking yourself:

  "Does the team find the daily scrum useful?"

- If the answer is no, find out why!
  - Do they manage themselves?
  - Do they share their work?
  - Do they report unclearly?
  - Are the tasks too big?
  - Does it take too long?
  - etc

Daily Scrum Rules

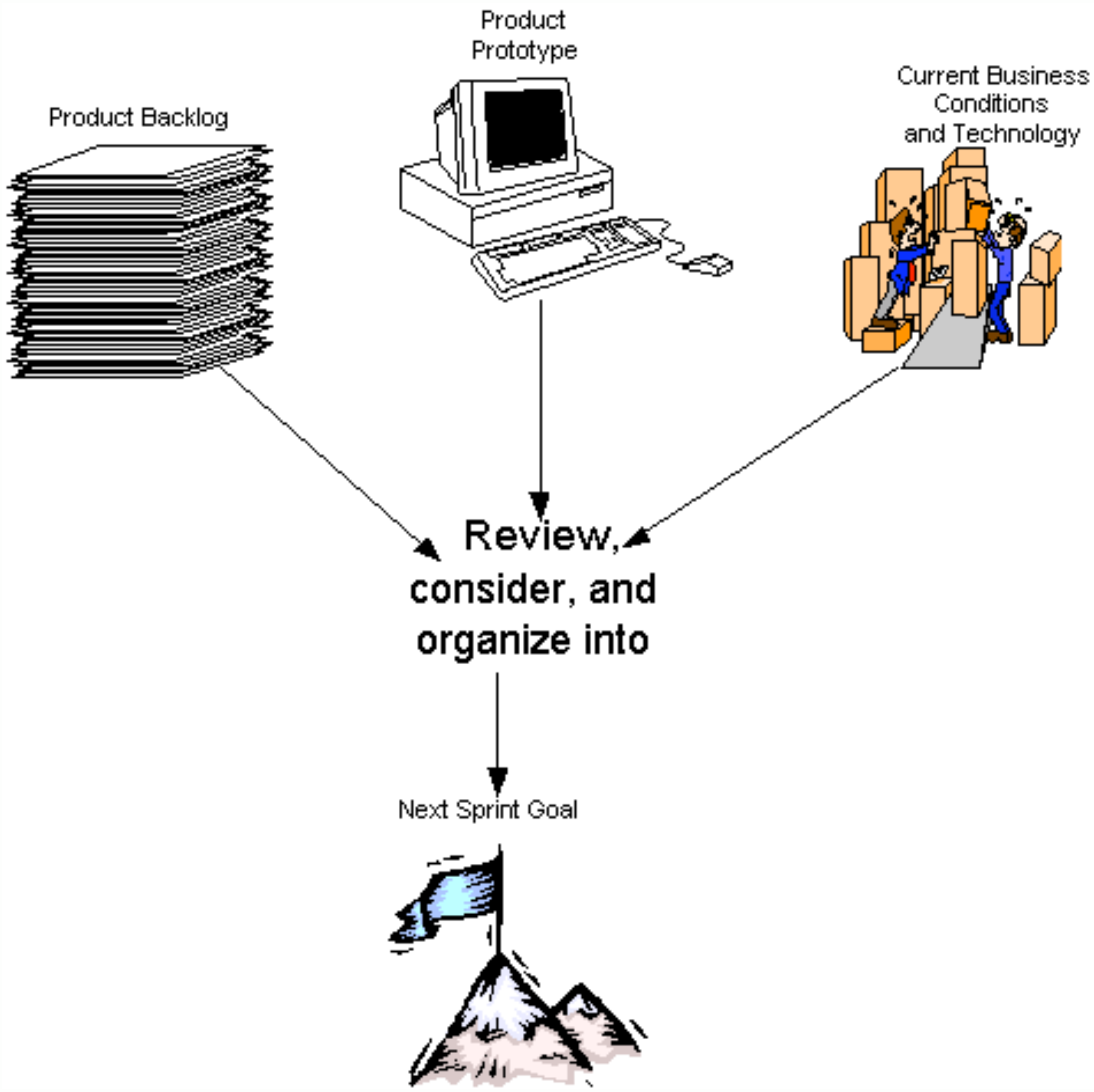


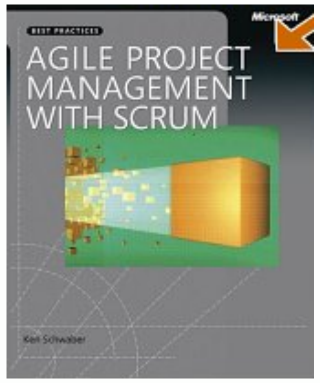
- Attendees: ScrumMaster, Team

- Goal: Synchronize work
- Timebox: 15 minutes

- Questions:
  - What have you done since last meeting?
  - What will you do before next meeting?
  - What is in your way?
- No discussion. Discussion after the daily Scrum
- Penalty for being late

Sprint review


Product Backlog
Product Prototype
Current Business Conditions and Technology
Review, consider, and organize into
Next Sprint Goal



Sprint Review Rules

- Attendees: ScrumMaster, Team, PO, Stakeholders
- Preparation: No more than 1 hour

- Goal: Present the functionality and review it
- Timebox: 4 hours

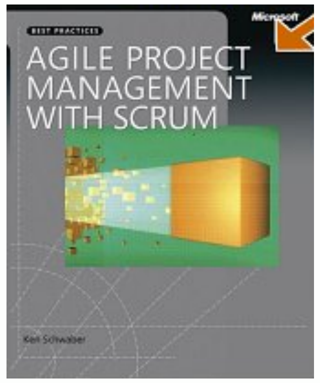
- Team presents “done” functionality and discusses with PO and stakeholders.
- Team cannot present functionality that is not “done”
- Product Owner prioritizes backlog accordingly

Things to check in Retrospectives

- Actions:
  - Do we only have a few actions?
  - Are they considered useful?
  - Are they implemented?

- Is “done” extended?
- Updating our working agreements?
- Do we require:
  - Tasks in the Sprint backlog?
  - Items in the Product backlog?

Sprint Retrospective Rules



- Attendees: Team, ScrumMaster. PO is optional
- Timebox: 3 hours

- Goal: Reflect and create actions for improvements

- Questions:
  - What went well during last sprint
  - What could be improved during last spint

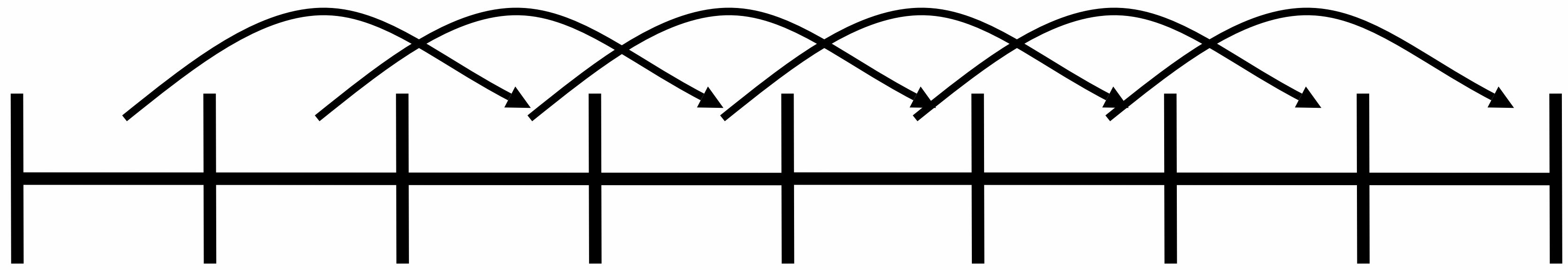
- ScrumMaster facilitates
- Improvements needing action will be added to next sprint

Product Backlog Refinement

1. Allocate 5-10% of every Sprint for working with the Product Owner, which could be compartmentalized to minimize interruption; and

2. Work on the Product Backlog for next Sprints – Split , clarify and estimate probable items;

3. Never allow the Product Owner to go into the Sprint Planning meeting with an inadequate Product Backlog.



Artifacts

Product Backlog

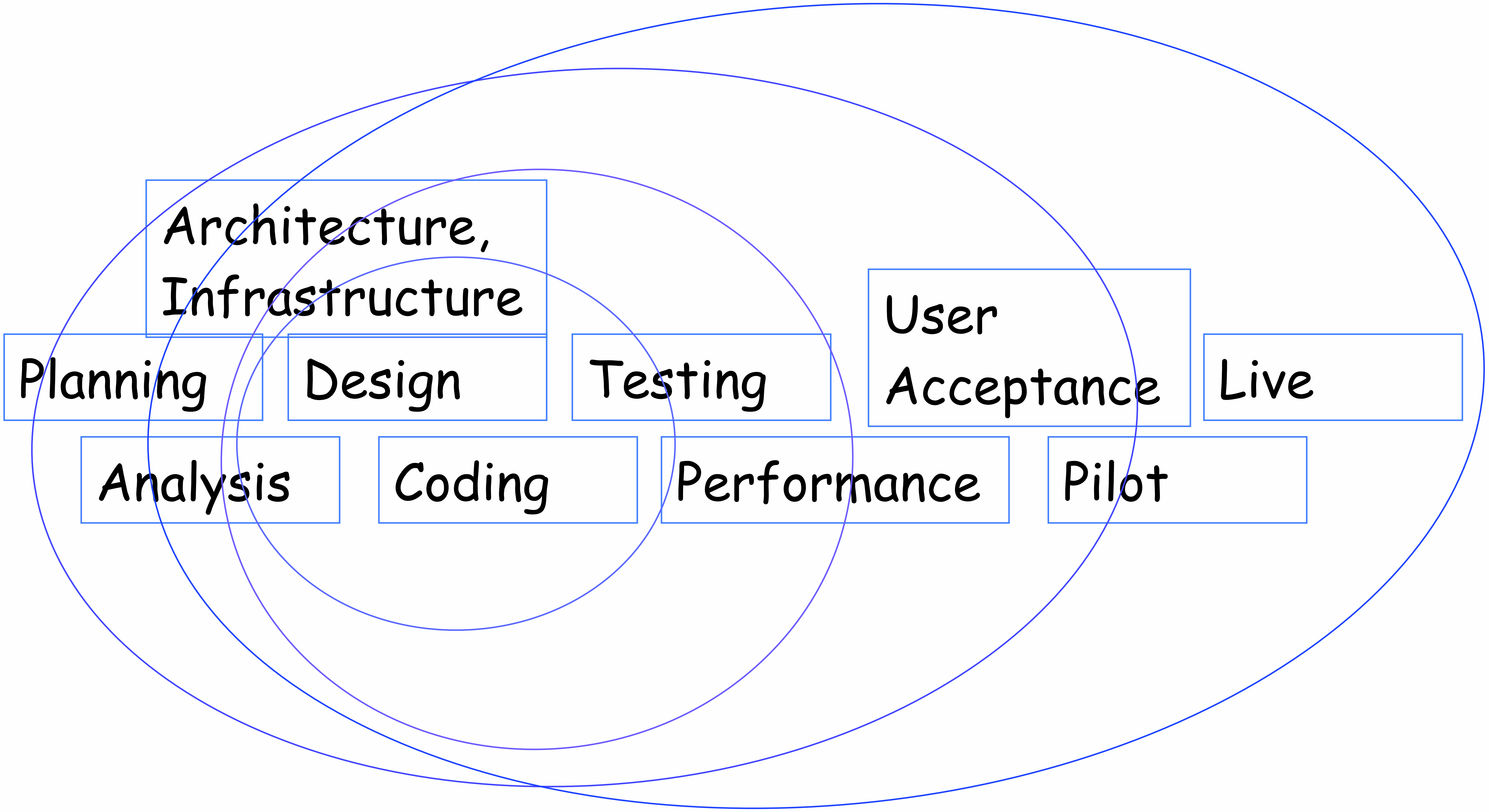
- One list per product
- List of functionality, technology, issues
- Emergent, prioritized, estimated
- More detail on higher priority items
- One list for multiple teams
- Product Owner responsible for priority
- Anyone can contribute
- Maintained and posted visibly
- Derived from Business Plan or Vision Statement, which sometimes have to be created with customer

Definition of "Done"

- Team together with product owner defines what "done" means.
- Examples of "done"
  - Design, coding, unit testing, integrated
  - Static analysis, refactored
  - Acceptance tested, deployable

- Done defines the current technical capability of the team.
  - Over time Done should include everything needed before deployment.
  - Not done backlog items should not be reviewed

Extending "done"



Sprint Backlog

- Tasks to turn product backlog into working product functionality
- Tasks are estimated in hours, usually 1-16
- Tasks with more than 16 hours are broken down later
- Team members sign up for tasks, they aren't assigned .
- Estimated work remaining is updated daily
- Any team member can add, delete or change the Sprint Backlog (theirs or new)
- Work for the Sprint emerges
- If work is unclear, define a Sprint Backlog with a larger amount of time … break it down later.
- Update work remaining as more is known, as items are worked

Sprint Backlog Estimates

- Work remaining reporting during a Sprint updates the estimated number of hours required to complete a task.
- **This should not be confused with time reporting, which is not part of Scrum.**
- There are no mechanisms in Scrum for tracking the amount of time that a team works.
- **Teams are measured by meeting goals, not by how many hours they take to meet the goal. Scrum is results oriented, not effort driven.**

Techniques

Good Engineering Practices

- Refactoring
- Test Automation
- Test Driven Development
  - Unit-TDD
  - Acceptance-TDD
- Continuous integration

- Slice of functionality

Continuous Integration

Continuous Integration is a **developer practice**
with the goal to always keep a **working system**
by making **small changes**, slowly growing the system
and **integrating** them at least **daily**
on the **mainline**
typically supported by a **CI system**
with lots of **automated tests**

- Increases transparency
- Increases cooperation and communication
- Enables people to work on same code

Splitting backlog items

- Different scenarios
- Stubbing out external dependencies

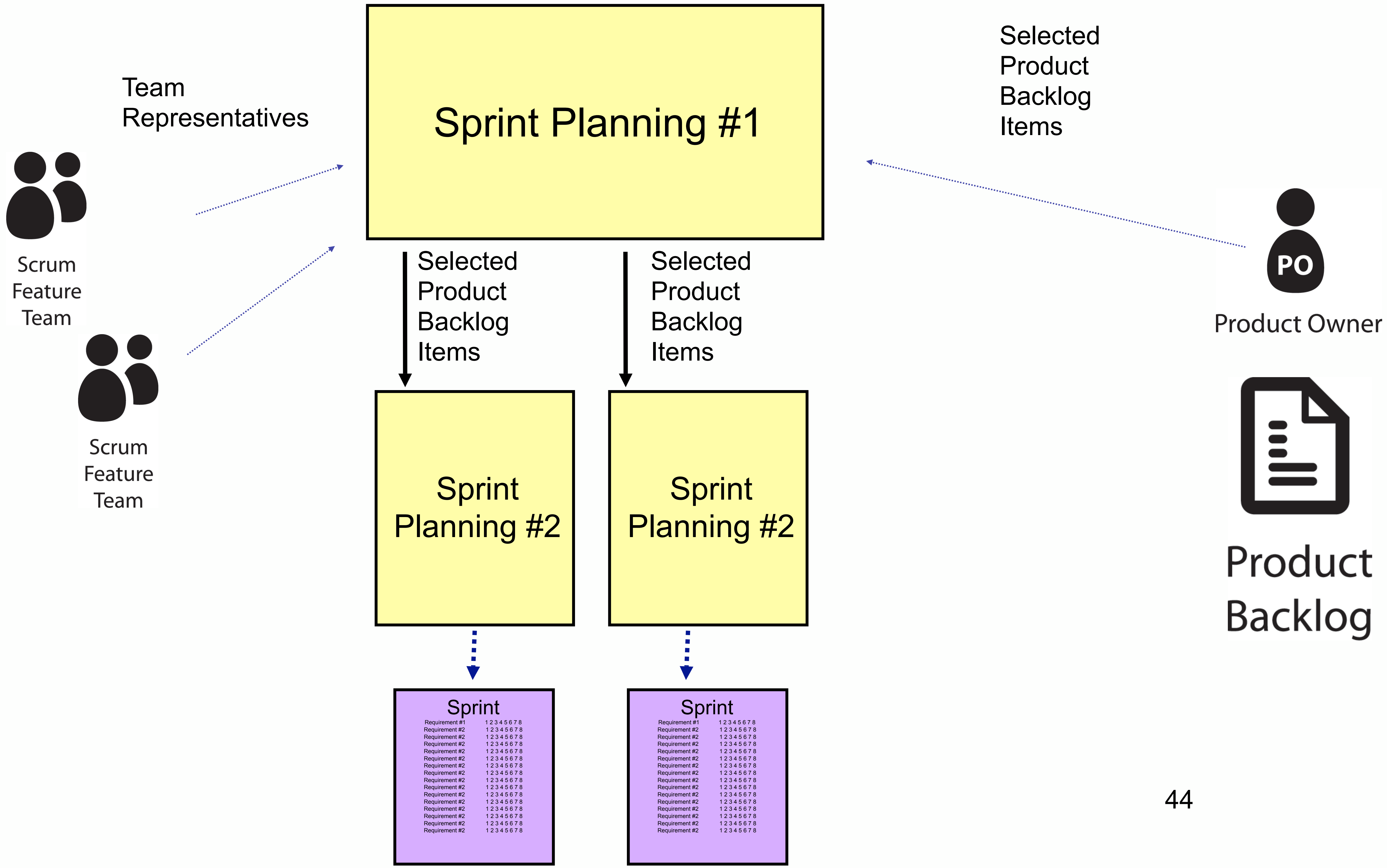
- Splitting across supported data
  - Different configuration options
- Splitting on operations performed
  - Different protocol messages
- Splitting on CRUD operations
- Separating cross-cutting concerns
  - E.g. security, logging
- Splitting functional and non-functional

Planning Poker

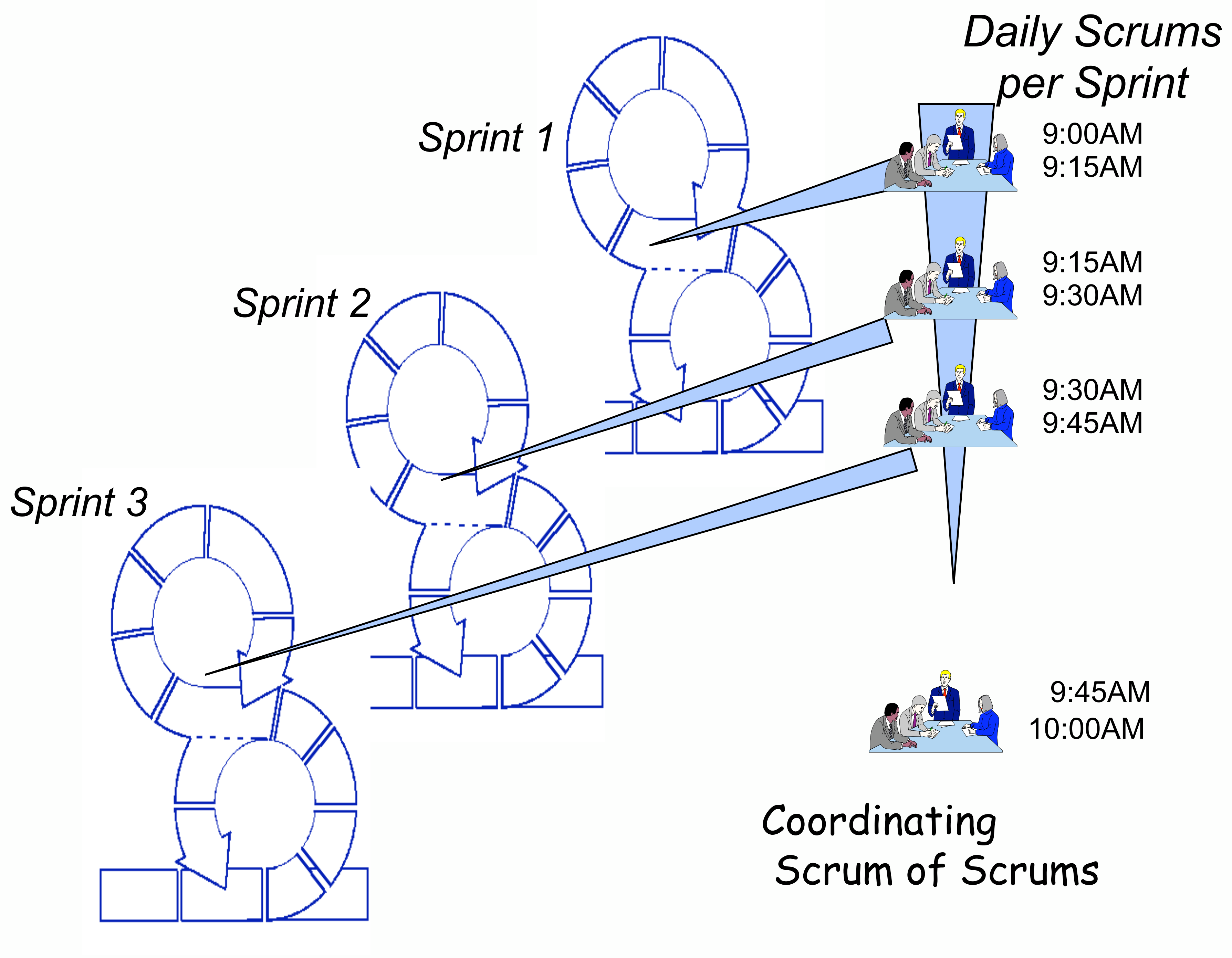
- Fibonacci – Less argument
- Clarification on the difference
  - Not on the agreements
- Everybody involved
  - Not dictated by most-knowledgeable person
- Quick
- Reliable
- Fun

Scaling

Scaling Sprint Planning

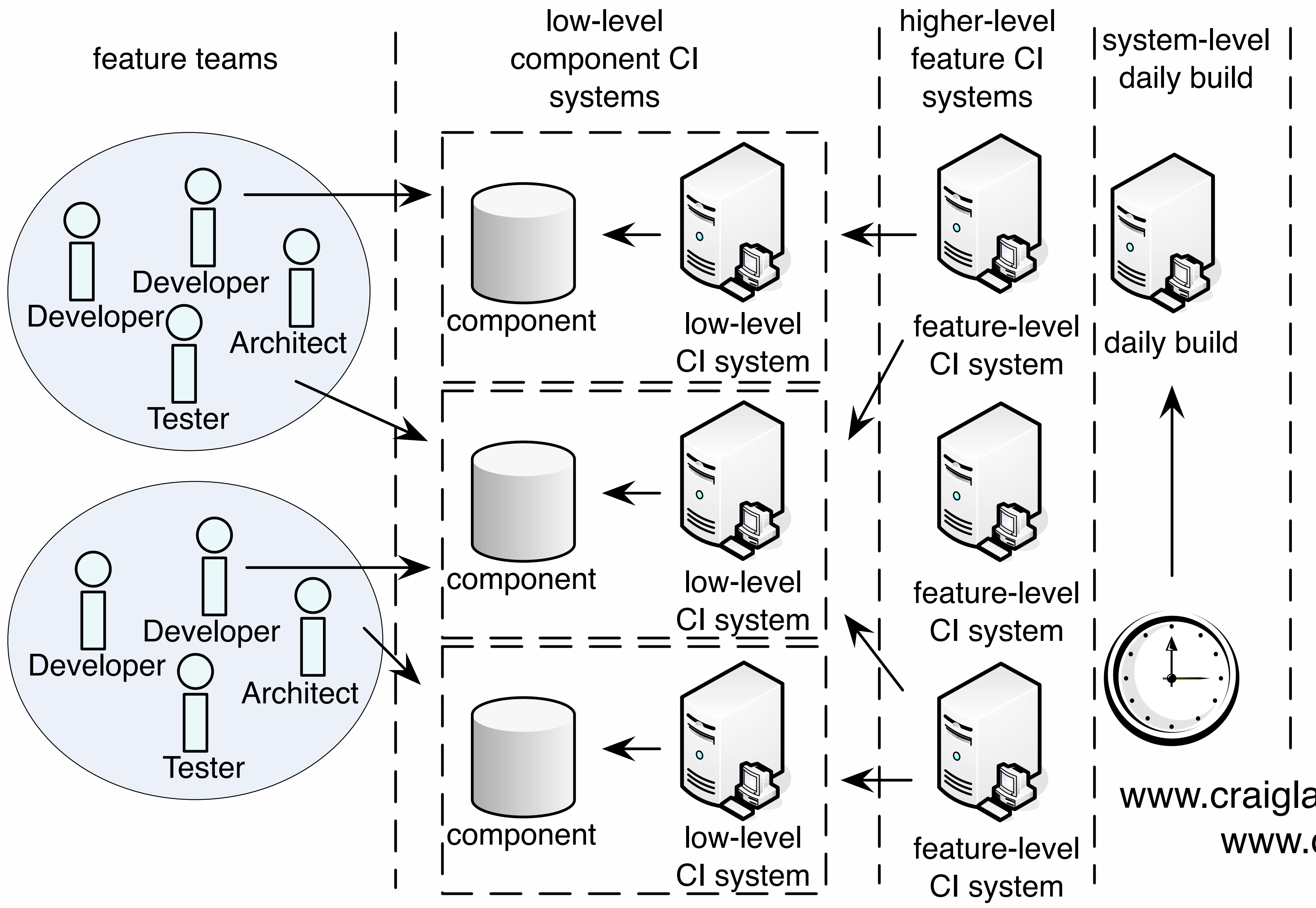


Team Synchronization



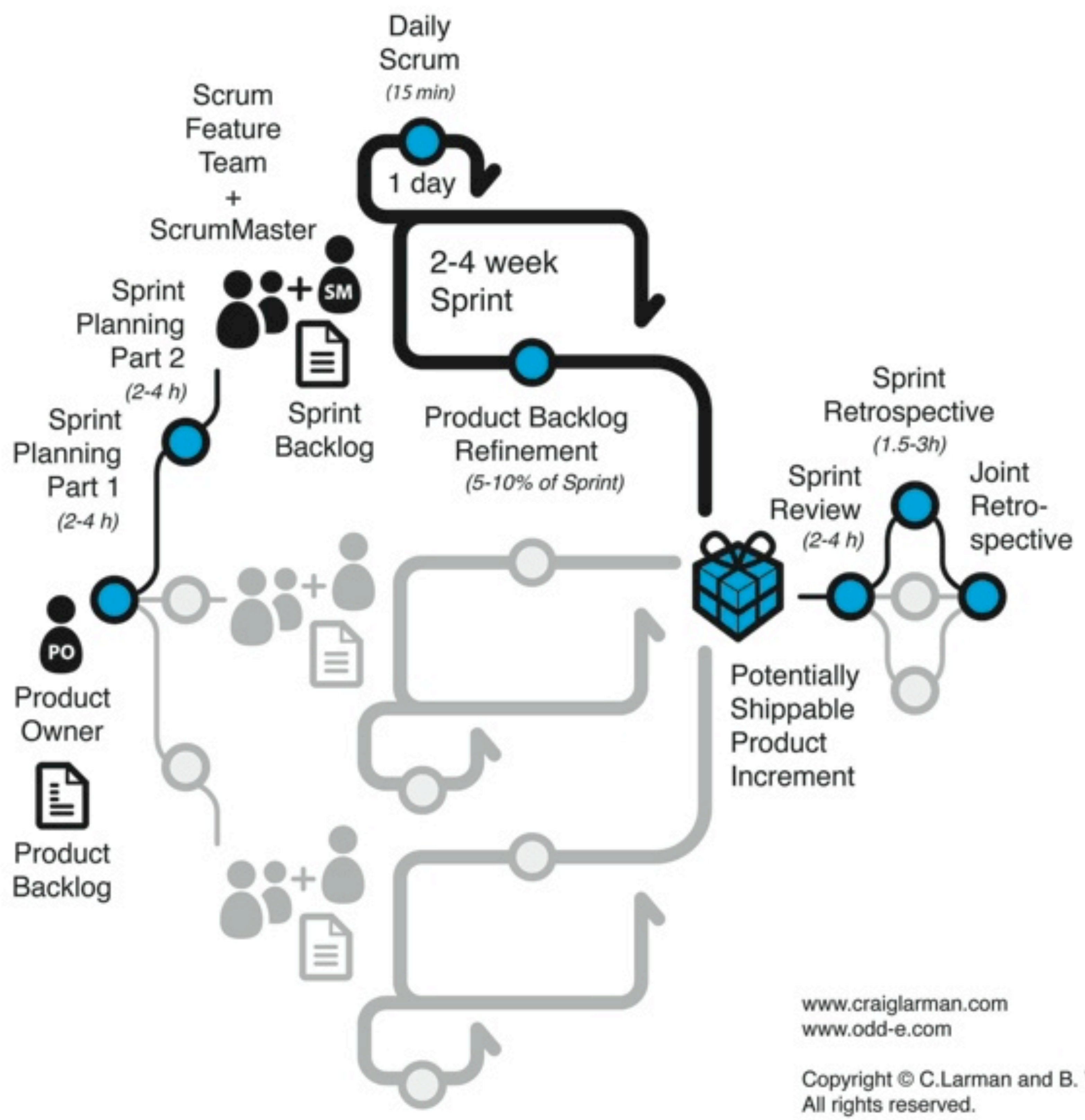
Coordinating Scrum of Scrums

Scaling CI - Example



www.craiglarman.com
www.odd-e.com

Large-scale Scrum for up to 10 teams with one Product Owner



Daily Scrum (15 min)
1 day
Scrum Feature Team + ScrumMaster
SM
2-4 week Sprint
Sprint Planning Part 2 (2-4 h)
Sprint Planning Part 1 (2-4 h)
Sprint Backlog
Product Backlog Refinement (5-10% of Sprint)
Sprint Retrospective (1.5-3h)
Sprint Review (2-4 h)
Joint Retro-spective
PO
Product Owner
Product Backlog
Potentially Shippable Product Increment

www.craiglarman.com
www.odd-e.com

Feature Teams

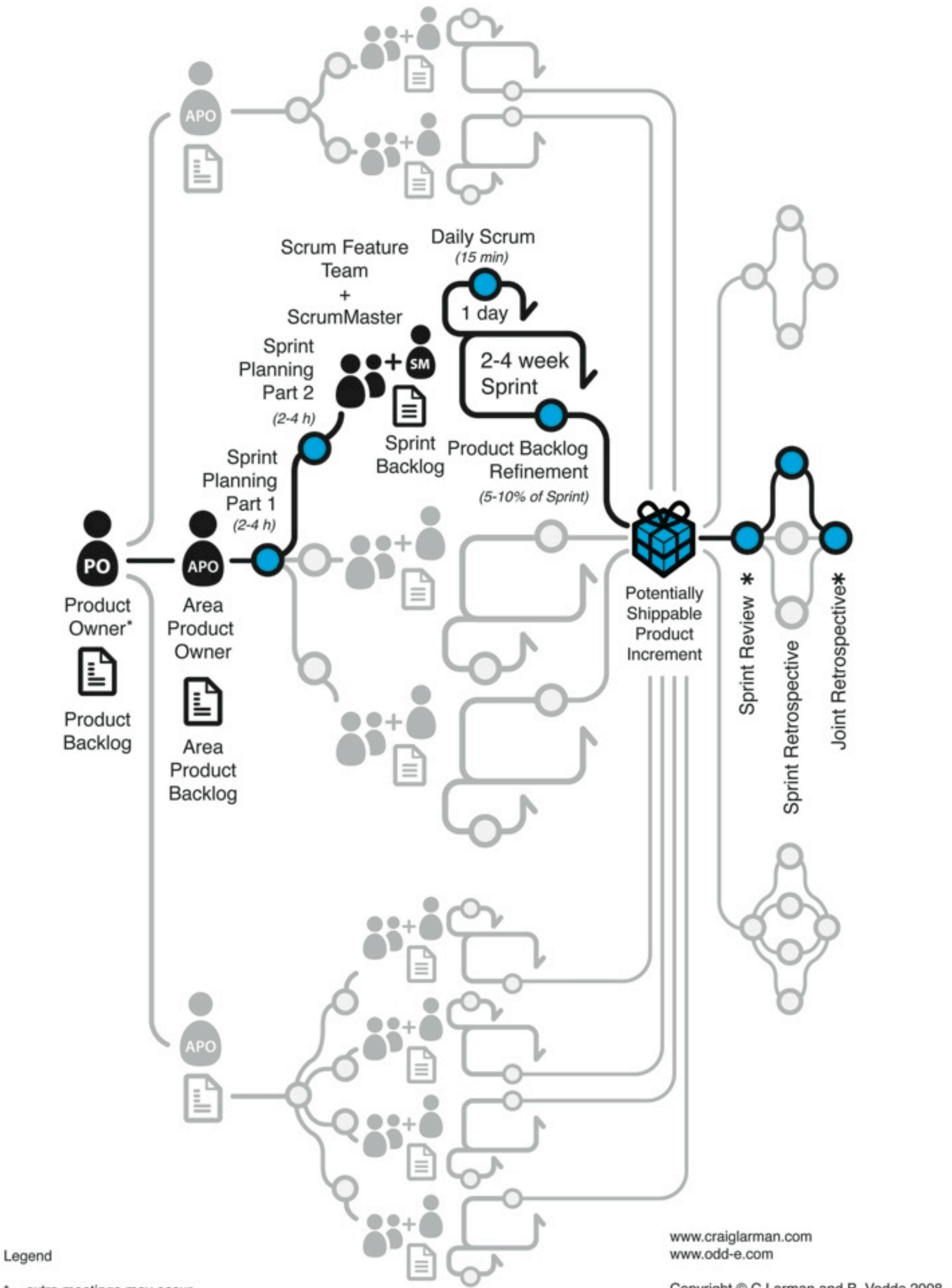
- long-lived—the team stays together so they can ‘jell’ for higher performance; they take on new features over time
- cross-functional and co-located
- work on a complete customer-centric feature, across all components and disciplines
- composed of generalizing specialists

Component vs. Feature teams

- Component teams
  - Traditional way of organizing teams
  - Focus on more and more specialization
  - Will lead to waterfall development
  - Will lead to unimportant features being implemented
- Feature teams
  - Cross-functional teams
  - Focus more on multiple specializations
  - Well suitable for iterative development
  - Gives flexibility in development

  - Much more difficult to implement!

Large-scale Scrum when "many" teams: One Product Owner and Area Product Owners



Legend

* = extra meetings may occur not shown on this diagram

www.craiglarman.com
www.odd-e.com

Organization

Organizational roles

- Scrum will be conflicting with lots of traditional organization roles and responsibilities:
  - Project manager will disappear?
  - Line manager will change.
    - Impediment backlog
  - Other roles will change.
- Always remember, this is a personal change in some persons future and career!
  - It's difficult and sometimes painful.

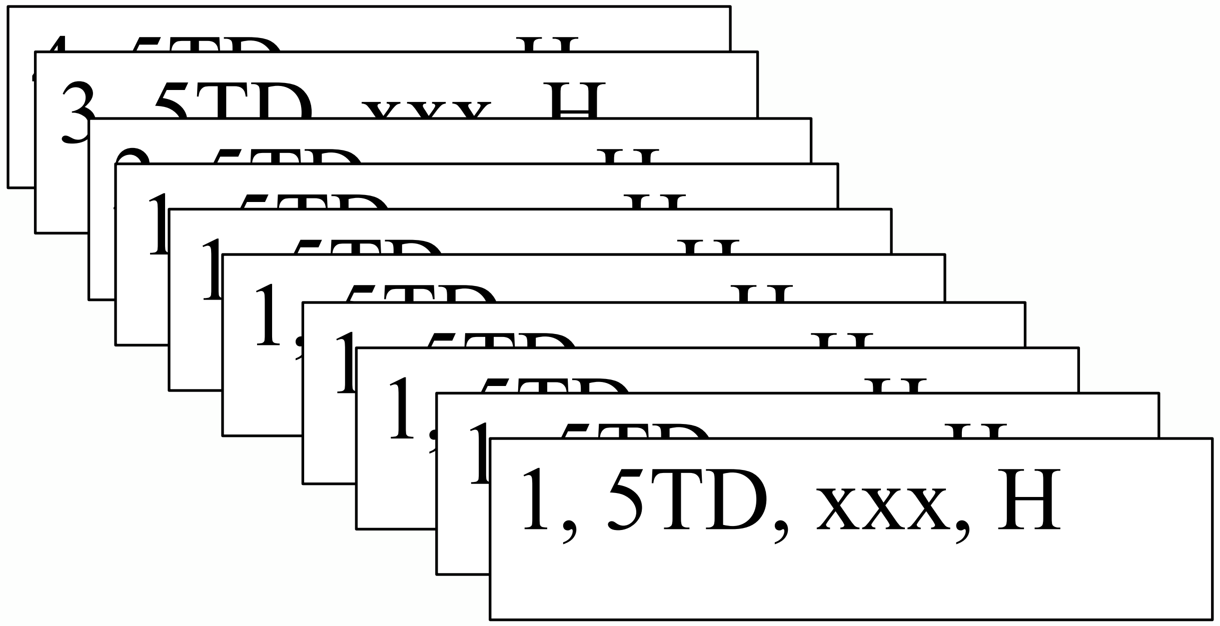
Odd-e

First Improve Yourself!

- You need to gain the trust of the organization before they will be changing.
- Therefore, always focus on changing and **improving** yourself instead of **blaming** the organization!

Why would the organization believe any team that is not:
Delivering working software without bugs every month!

Impediment backlog



Organizational impediments to optimized software construction and delivery are uncovered during the Sprints.

Management's job it to prioritize and systematically remove these impediments. This is very hard work.