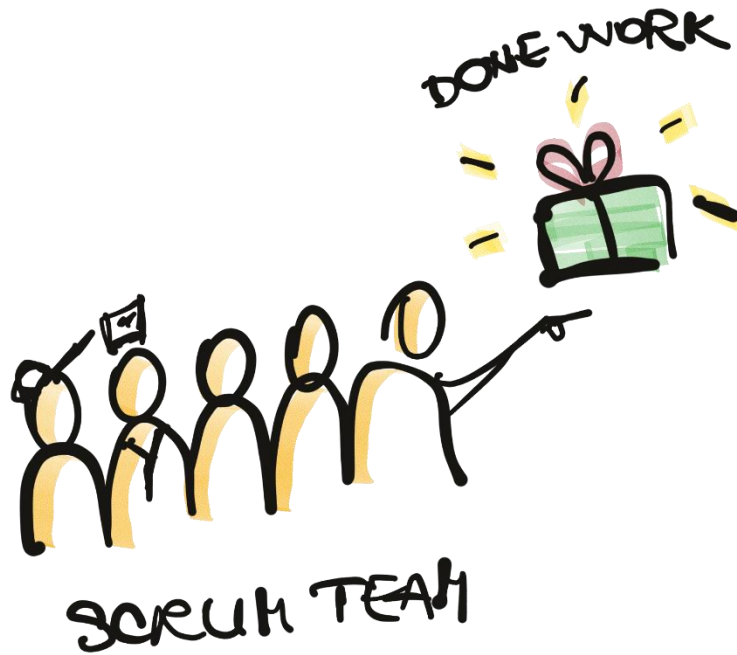


SPRINT REVIEW AND RETROSPECTIVE

Systems Analysis and Design
Sharif University of Technology
Fall 1400-1401

SPRINT REVIEW



Sprint Review

- **What?**

- An informal meeting with low ceremony and **high value** during which we **inspect** (and adapt) the **result of the work** (the potentially shippable product increment).
 - Provides a transparent look at the **current state** of the product, including any **inconvenient truths**.
 - Time to ask questions, make observations or suggestions, and have discussions about **how to best move** forward given current realities.
 - An important **learning loop** which allows for frequent course corrections to keep the development moving in the right direction.

- **When?**

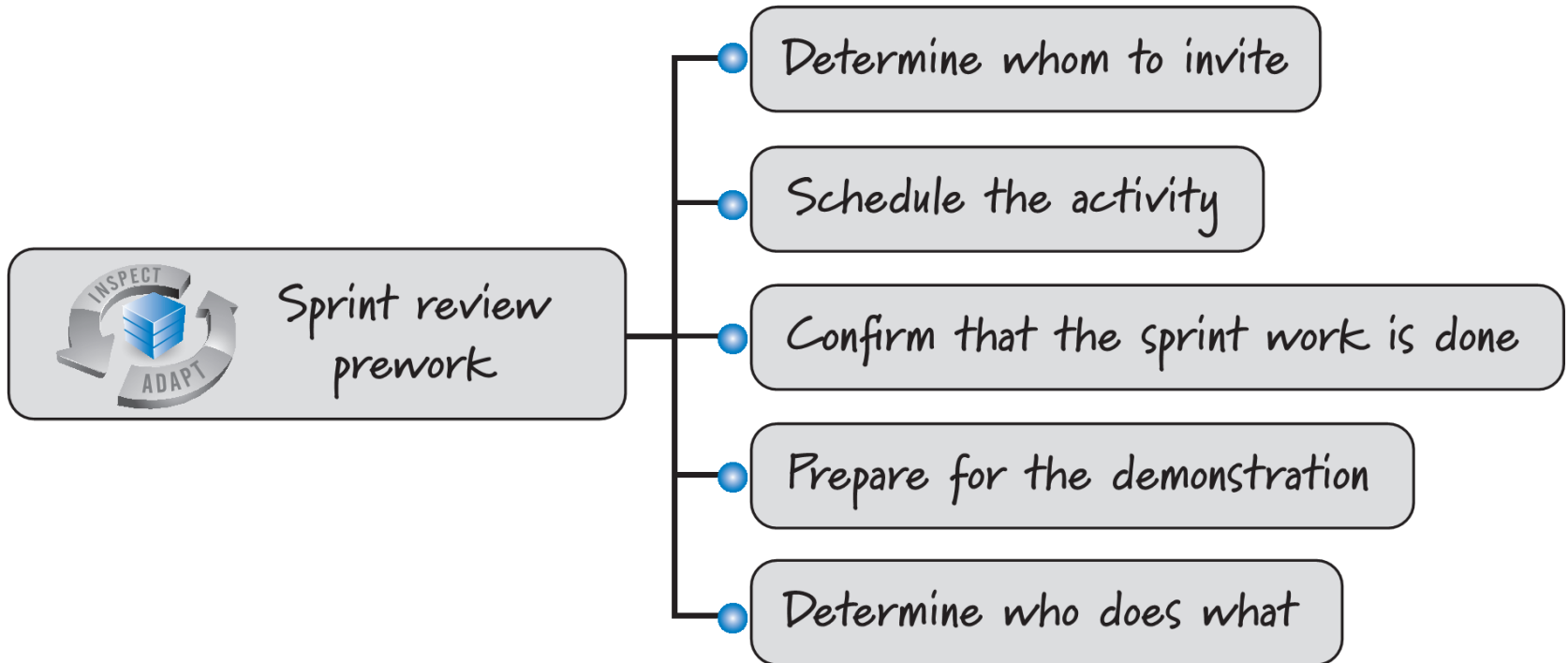
- Near the end of each sprint cycle, just after sprint execution and just before (or occasionally after) the sprint retrospective.

Sprint Review: Participants

Source	Description
Scrum team	The product owner, ScrumMaster, and development team should all be present so that they can all hear the same feedback and be able to answer questions regarding the sprint and the product increment.
Internal stakeholders	Business owners, executives, and managers should see the progress firsthand so that they can suggest course corrections. For internal product development, internal users, subject matter experts, and the operations manager of the business function to which the product relates should attend.
Other internal teams	Sales, marketing, support, legal, compliance, and other Scrum and non-Scrum development teams might want to attend sprint reviews to provide area-specific feedback or to sync their own groups' work with the Scrum team.
External stakeholders	External customers, users, and partners can provide valuable feedback to the Scrum team and other attendees.

[Rubin 2012]

Sprint Review: Prework



[Rubin 2012]

Sprint Review: Prework

1. Determine Whom to Invite

- The goal is to invite the **right set of people** to extract the highest value.
- The team might need to **constrain attendance** to a certain person or group whose input is **essential** to reviewing this sprint's work.
- General Approach: Identify a **core group** that should be invited to every review and then issue a separate invitation to others on a sprint-by-sprint basis.

2. Schedule the Activity (when, where, and how long)

3. Confirm That the Sprint Work Is Done

4. Prepare for the Demonstration

5. Determine Who Does What

Sprint Review: Prework

1. Determine Whom to Invite

2. Schedule the Activity (when, where, and how long)

- If we use consistent-duration sprints, we can schedule the sprint review meetings using a regular cadence (e.g., every second Friday at 2:00 p.m.).
- Sprint reviews vary in duration depending on several factors, including sprint length, team size, and whether multiple teams are participating in the review.
- Many teams have found the **one-hour-per-sprint-week rule** helpful; e.g., for a two-week sprint the review should take no more than two hours

3. Confirm That the Sprint Work Is Done

4. Prepare for the Demonstration

5. Determine Who Does What

Sprint Review: Prework

1. **Determine Whom to Invite**
2. **Schedule the Activity** (when, where, and how long)
3. **Confirm That the Sprint Work Is Done**
 - At the sprint review, the team is allowed to present **only completed work**—work that meets the agreed-upon definition of done.
 - It is the product owner's responsibility to determine the work which has been done.
4. **Prepare for the Demonstration**
5. **Determine Who Does What**

Sprint Review: Prework

1. Determine Whom to Invite

2. Schedule the Activity (when, where, and how long)

3. Confirm That the Sprint Work Is Done

4. Prepare for the Demonstration

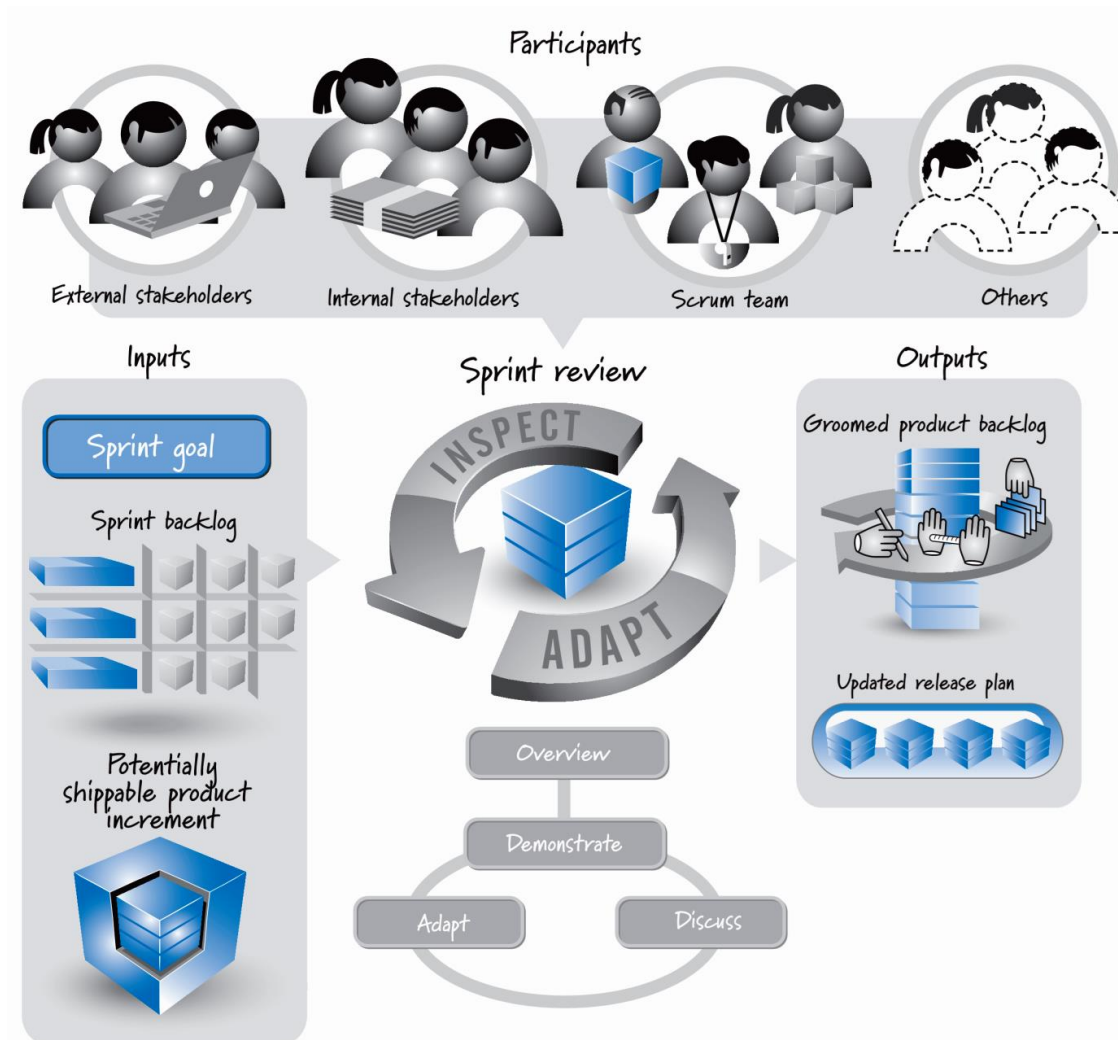
1. The goal is to provide transparency for inspecting and adapting the product, not to put on a showcase to create excitement.
 - Participants should see **working software**, not a PowerPoint presentation.
2. Rules of thumb:
 - Do not spend more than 30 minutes to an hour per week of sprint duration to prepare for the sprint review.
 - Only **show the artifacts** produced as a consequence of achieving the sprint goal.

5. Determine Who Does What

Sprint Review: Prework

1. **Determine Whom to Invite**
2. **Schedule the Activity** (when, where, and how long)
3. **Confirm That the Sprint Work Is Done**
4. **Prepare for the Demonstration**
5. **Determine Who Does What**
 - Prior to the sprint review, the team needs to **decide who** on the Scrum team is going to **facilitate** the review and who will **demonstrate** the completed work.
 - Typically:
 - The Scrum Master facilitates.
 - The product owner might kick things off by welcoming members of the stakeholder community and providing a synopsis of the sprint results.
 - Preferably, every member of the development team should have an opportunity at some sprint review to go hands-on and demonstrate.

Sprint Review: Process



[Rubin 2012]

Sprint Review: Process

- The sprint review starts with a Scrum team member (frequently the product owner) **presenting**:
 1. The sprint goal.
 2. The product backlog items associated with the sprint goal.
 3. An overview of the product increment that was actually achieved.
- This information provides a **summary** of how the sprint results **compare** with the sprint goal.
- If the results do not match, the Scrum team provides an **explanation**; it is important that the sprint review be a **blame-free environment**.
 - If the goal was not met, everyone participating should refrain from trying to assess blame.
 - The purpose is to describe what was accomplished and then to use the information to **determine the best course of action** for moving forward

Activities: Demonstrate

- One or more Scrum team members will **demonstrate** all relevant aspects of the product increment that was built during the sprint.
- Although a demonstration is quite helpful in the sprint review, the demo is not the aim of the sprint review:
 - The most important aspect of the sprint review is **in-depth conversation** among the participants to enable productive adaptations to surface and be exploited.
 - The demo aims at energizing this conversation around something concrete.
- In **certain organizations**, such as game studios, it can be even more effective to let the stakeholders actually **try the produced increment**.
- What if there is nothing to demo?
 - If the team did not get anything done, the sprint review will focus on **why** nothing got done and how the future work will be affected.
 - If the product is **hard to demo**, the team must at least **demo the tests** that it has used to verify that the work is done to the satisfaction of the product owner.

Activities: Discuss

- Observation, comments, and **reasonable discussion** on the product and direction are strongly encouraged.
 - Deep problem solving, however, should be deferred to another meeting.
- Benefits of discussion are two-way:
 - It allows the participants who are not on the Scrum team to **understand the state** of the product, and help guide its direction.
 - Scrum team members gain a **deeper understanding of the business** and marketing side of their product.

Activities: Adapt

- Through demonstration and discussion, the team is able to **ask and answer** questions, including the following:
 - Do the stakeholders like what they see?
 - Do they want to see changes?
 - Is what we're building still a good idea in the marketplace or to our internal customers?
 - Are we missing an important feature?
 - Are we overdeveloping/investing in a feature where we do not have to?
- Asking and answering these questions provides input on **how to adapt** the product backlog and release plans.
 - New PBIs are often created or existing PBIs are changed, reprioritized or deleted if they are no longer needed.
 - Changes may propagate to the **release plan**; e.g., we might decide to alter one of the key release-planning variables: Scope, Date, or Budget.

Sprint Review Issues

- **Sign-offs**

- Sprint reviews are not the proper venue to sign off on (approve) PBIs.
 - PBIs should have **already been “approved”** by the product owner before the sprint review starts.
- What if during the sprint review a stakeholder believes a PBI is not done?
 - The product owner can schedule a change to the feature by creating a new product backlog item to reflect the requested behavior.
 - He should also **investigate** to determine **why** he disconnected from the stakeholders on this PBI and **make adjustments** to prevent future mishaps.

- **Sporadic Attendance**

- Encourage the stakeholders to **attend** through **early and frequent** release of valuable and working software.

- **Large Development Efforts**

- If you have a larger development effort with multiple Scrum teams, it might make sense to consider doing a **joint sprint review**.



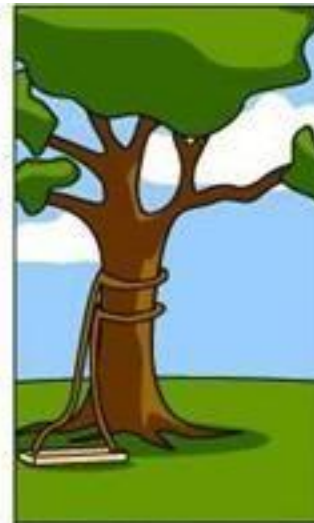
How the customer explained it



How the Project Leader understood it



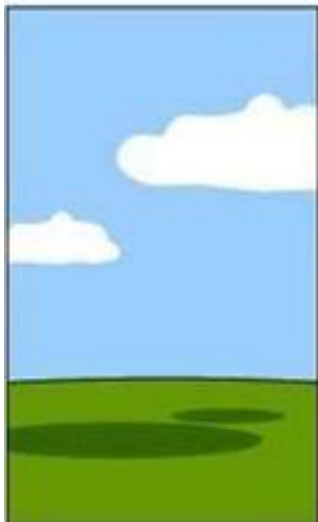
How the Analyst designed it



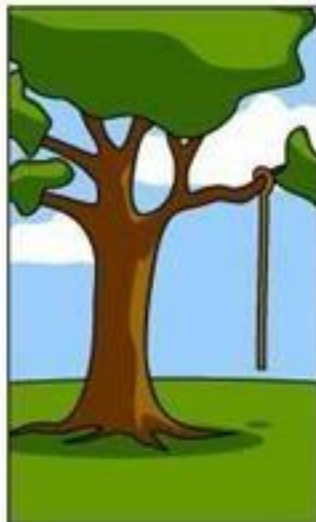
How the Programmer wrote it



How the Business Consultant described it



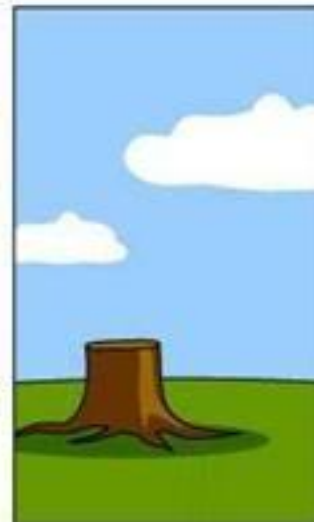
How the project was documented



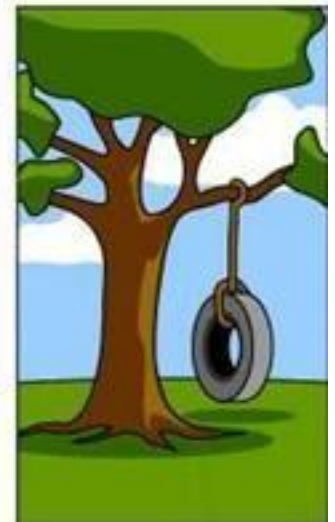
What operations installed



How the customer was billed

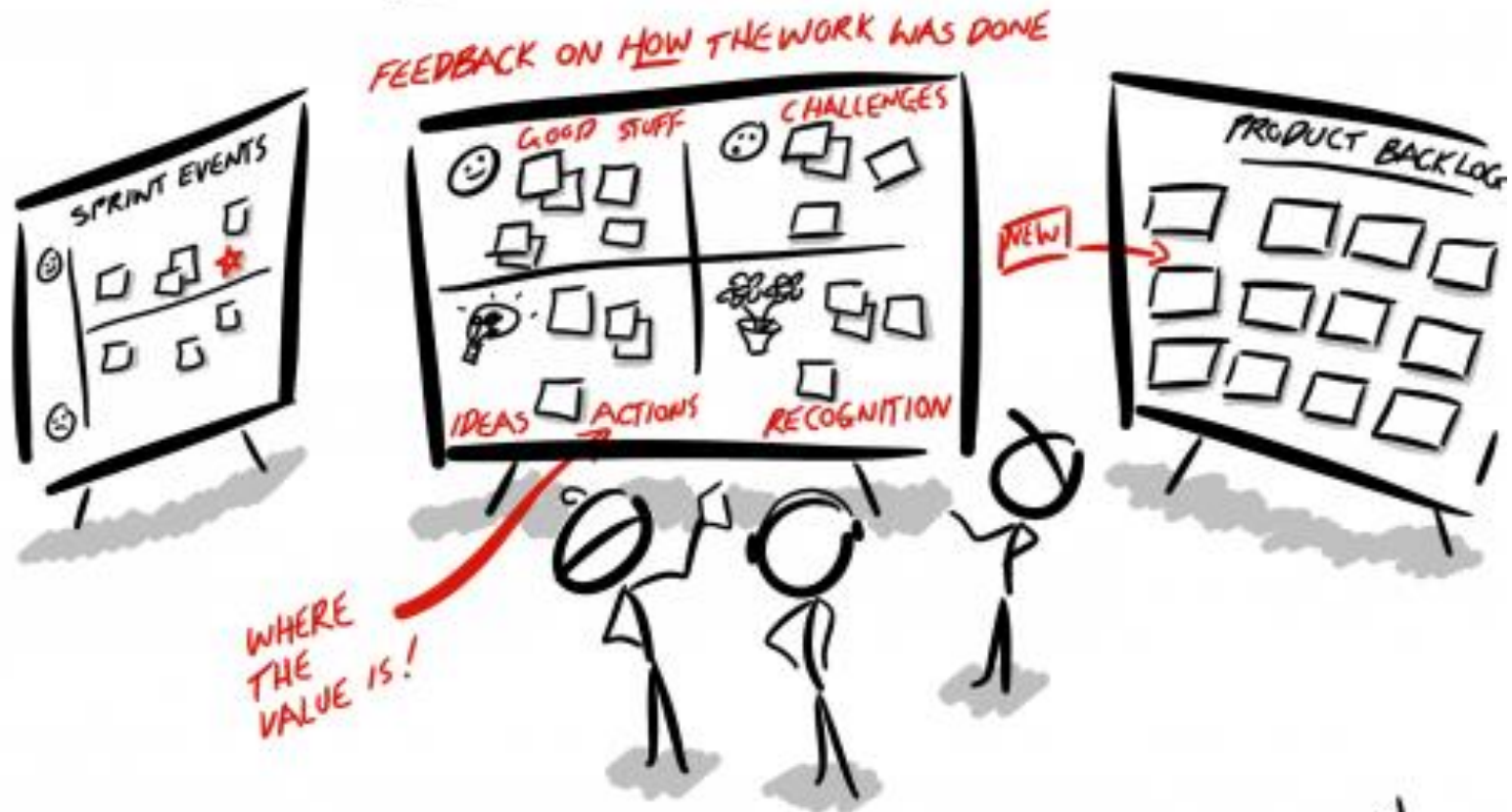


How it was supported



What the customer really needed

SPRINT RETROSPECTIVES



BALESY
'17
#33

Sprint Retrospective

- **What?**

- An informal meeting where teams **analyze the way they work**, identify ways to improve, and make plans to implement these improvements.
 - Anything that affects how the team creates the product is open to scrutiny: Processes, practices, communication, environment, artifacts, tools, and so on.
 - It can be as simple as the Scrum team members coming together to discuss questions such as:
 - What worked well this sprint that we want to continue doing?
 - What did not work well this sprint that we should stop doing?
 - What should we start doing or improve?

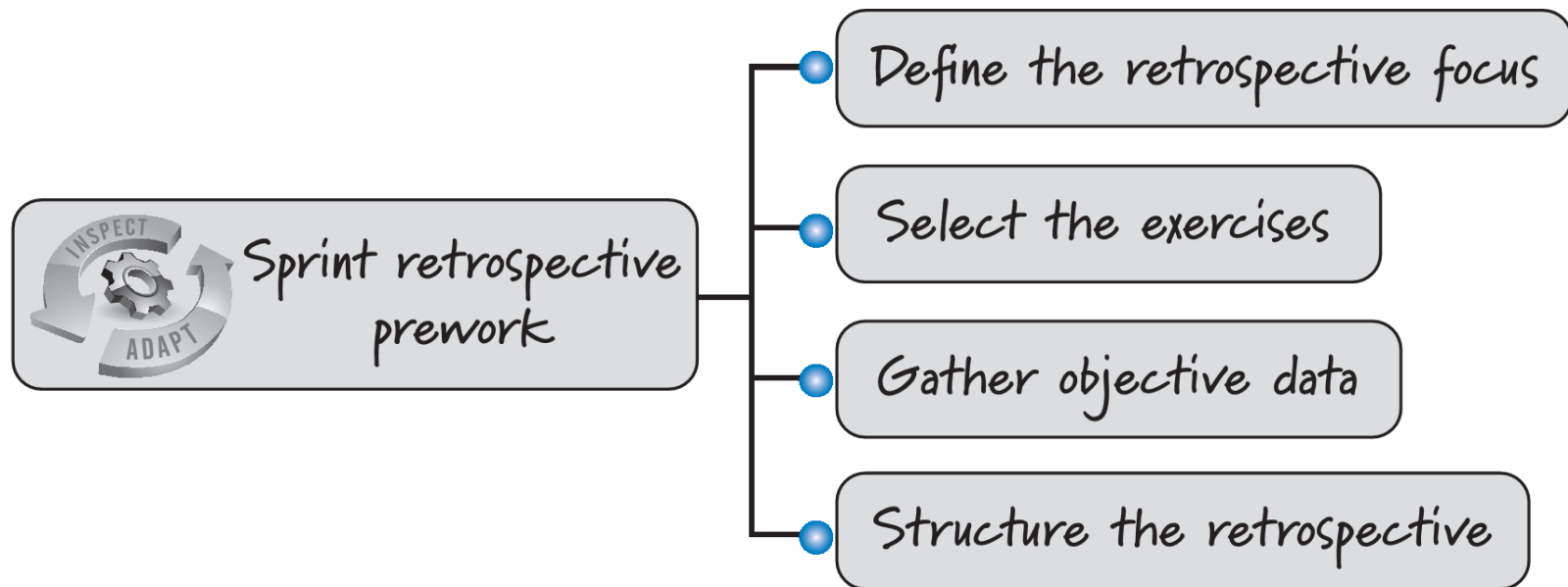
- **When?**

- At the end of each sprint cycle, typically after sprint review.

Sprint Retrospective: Participants

- **Development Team:** Their diverse perspectives are essential for **identifying** process **improvements** from multiple points of view.
- **Scrum Master:** The Scrum Master is an integral part of the process and the process authority for the Scrum team.
 - The Scrum Master does **not tell** the team how to change its process.
 - The Master **points out** where the team is not adhering to its own agreed-upon process and acts as a valuable source of knowledge and ideas for the team.
- **Product Owner:** The product owner is critical to achieving fast and flexible flow of business value, and is therefore a key element of the process.
 - If speaking candidly is not comfortable for team members, the product owner will not attend until the Scrum Master can create a **safer** environment!
- Due to safety considerations, stakeholders or managers who are not on the Scrum team will attend **only if invited** by the Scrum team.

Sprint Retrospective: Prework



[Rubin 2012]

Sprint Retrospective: Prework

1. Define the Retrospective Focus

1. The default focus is to **review all relevant aspects** of the process the Scrum team used during the current sprint.
2. However, a team might select a different retrospective focus based on what is **currently important** to the team and where improvement is required.
 - For example: Focus on how to improve our skills with TDD.
3. Establishing the focus allows the team to determine if **any non-Scrum team members** should be invited, and sets the stage for the rest of the prework.

2. Select the Exercises

3. Gather Objective Data

4. Structure the Retrospective

Sprint Retrospective: Prework

1. Define the Retrospective Focus

2. Select the Exercises

- **Exercises** help participants to engage, think, explore, and decide together; typical exercises include:
 1. Create and mine a sprint **event timeline**.
 2. **Brainstorm** insights.
 3. Group and **vote** on insights.

3. Gather Objective Data

4. Structure the Retrospective

Sprint Retrospective: Prework

1. Define the Retrospective Focus

2. Select the Exercises

3. Gather Objective Data

- Because a sprint retrospective is performed in a **focused, short period of time**, any legwork to collect needed data should be done **before** the retrospective begins.
- Objective data is **hard data** (not opinions), such as: What events happened and when, or the burnup chart for the sprint illustrating the flow of completed work.
- At this point we are **not organizing or analyzing** any data; we are **just collecting** it so that it is available during the retrospective.

4. Structure the Retrospective

Sprint Retrospective: Prework

1. Define the Retrospective Focus

2. Select the Exercises

3. Gather Objective Data

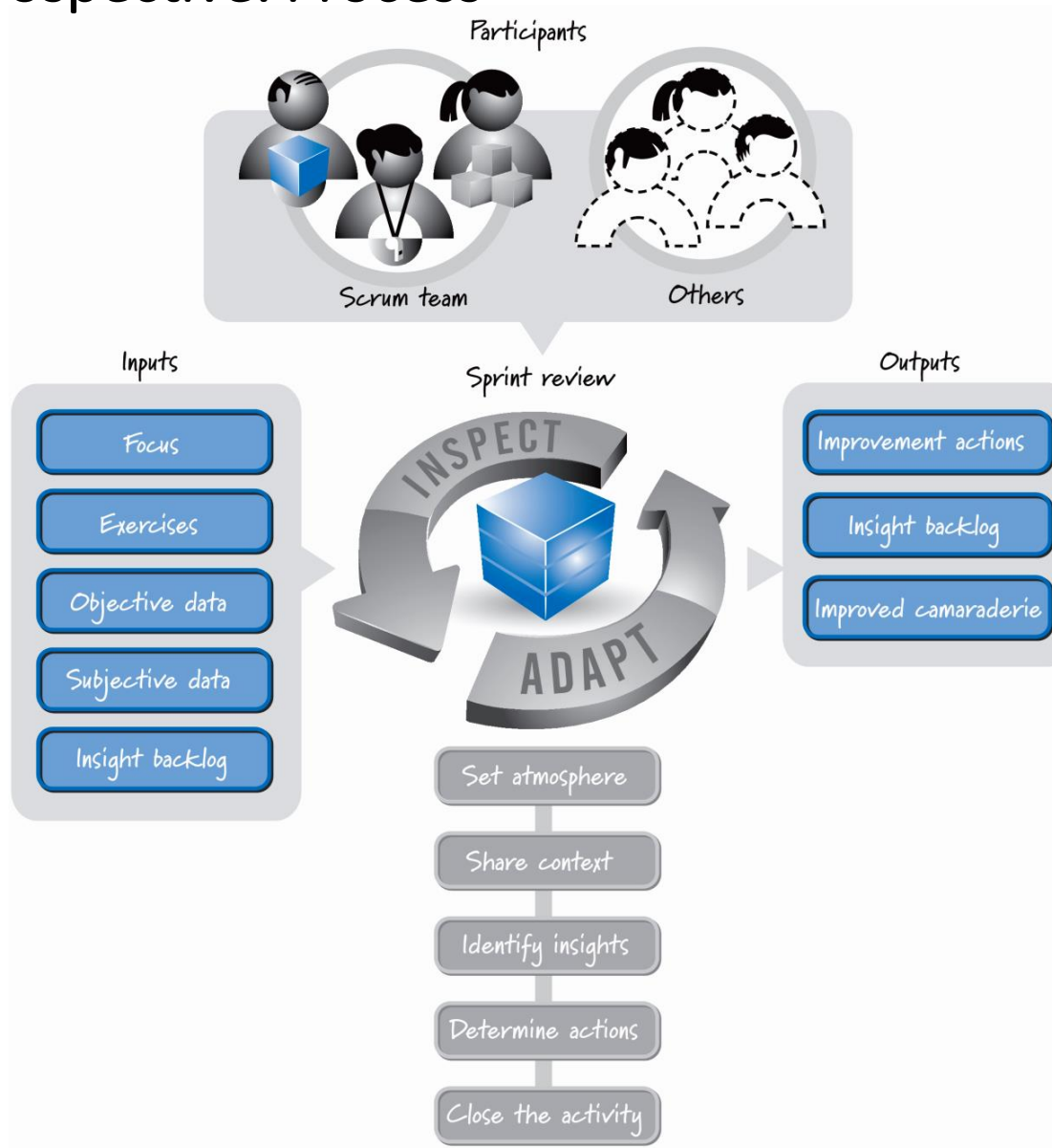
4. Structure the Retrospective

1. **Duration:** Influenced by factors such as **how many people** are on the team, how new the team is, whether any team members are located remotely, and so on.

- Budget about **1.5 hours** for the sprint retrospective when using **two-week sprints**, and proportionally more when using longer sprints.

2. **Facilitation:** The Scrum Master is usually assigned as the facilitator, but a skilled, neutral, outside facilitator can be assigned as well.

Sprint Retrospective: Process



[Rubin 2012]

Activities: Set the Atmosphere

- A good way to start the retrospective is to establish an atmosphere that makes people **feel comfortable participating**.
 - People must feel it is safe to express their opinions without fear of retribution.
 - It is helpful to make clear that the focus is on the organizational system and process, **not the individuals**, thus making it safe to explore what went wrong.
- There will be times when problems are people problems; the retrospective is not the place to solve them.
 - The retrospective is about improving the Scrum team's process, not about assigning **blame** or **reprimanding** individual behavior.
 - When setting the atmosphere, set ground rules which reinforce the concept of a blame-free environment.
- It is important to encourage active participation.

Activities: Share Context

- It is important to establish a **shared context**: The participants must align their diverse individual perspectives into a shared team perspective.
 - Team members can all experience the same event and yet interpret it differently.
 - If individual perspectives are allowed to dominate, the retrospective could degrade into a session of opinion debate rather than a focused session.
- When establishing a shared context, therefore, it's imperative that you first ground the retrospective in an **objective, big-picture** view of the sprint.
 - **Share objective data**, such as committed PBIs, PBIs completed, number of defects, and so on (based on the retrospective focus.)
 - Allow **relevant subjective data** to be exposed and discussed.
- **Two Common** exercises that can be used to develop a shared context of objective and subjective data: Event Timeline and Emotions Seismograph.

Share Context: Event Timeline

- The event timeline is a shared artifact that visually represents the flow of events during a sprint.
 - Events could include “Busted the build,” or “Interrupted to fix production failure,” or “Salina returned from holiday.”
- Draw a timeline on a wall or whiteboard and have the participants put cards (or sticky notes) on the timeline, representing meaningful events that occurred during the sprint.
- To help visually categorize events, use colored cards:
 - To represent different event types (for example, green is a technical event, yellow is an organizational event, red is a personal event); or
 - to represent feelings or energy levels (for example, green is a positive event, yellow is a neutral event, and pink or red is a negative event).

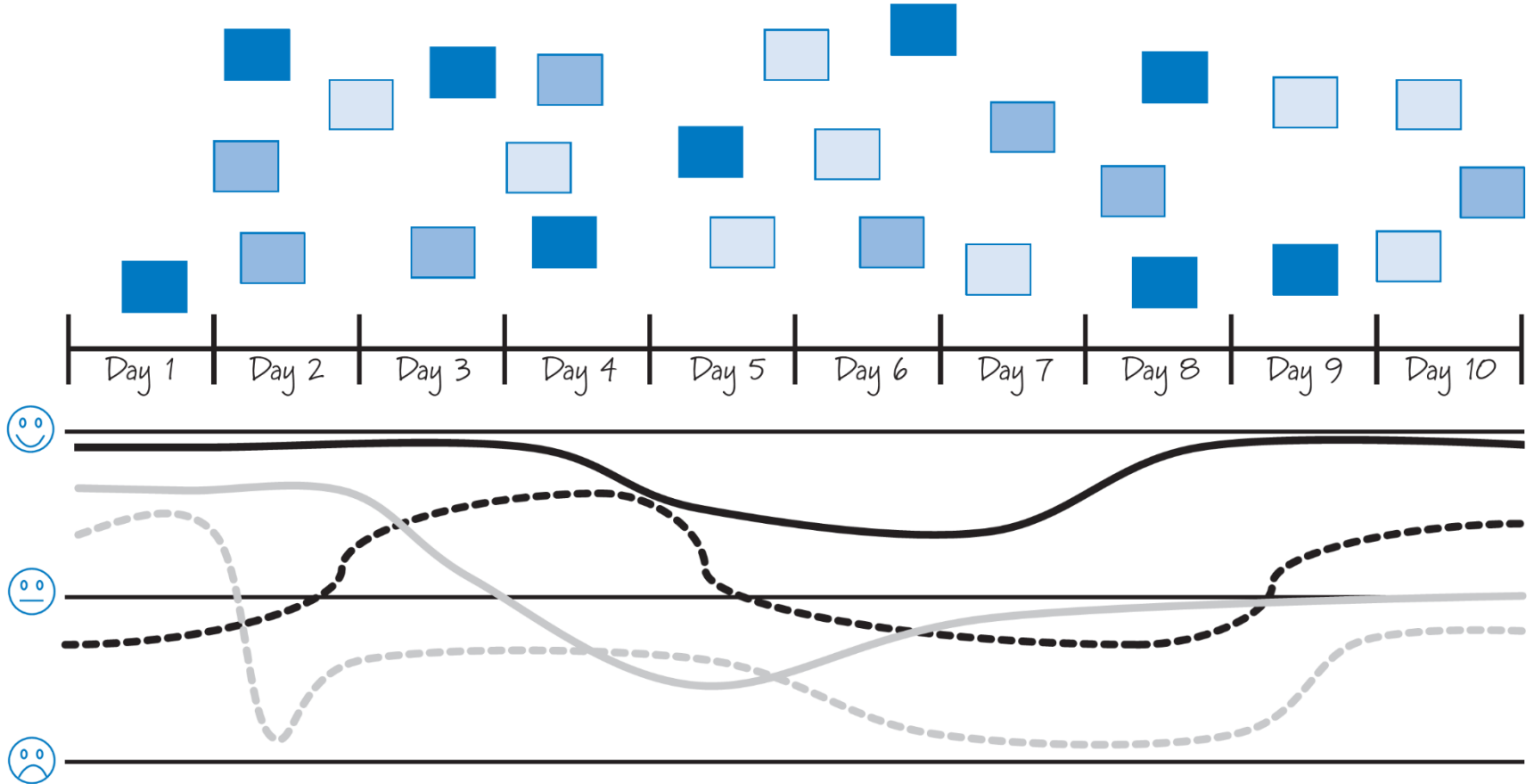
Share Context: Event Timeline – example



Share Context: Emotions Seismograph

- The emotions seismograph is a **graphical** representation of the **emotional ups** and **downs** of the participants over the course of the sprint.
 - Usually created as a **complement** to the event timeline.
 - Helps expand the shared context beyond the objective data (what happened) to include some **subjective data** (how the team felt about it).
- **Each participant** is invited to draw a curve showing how they felt or what their energy level was like over the course of the sprint.
 - It is frequently convenient to draw the seismograph **directly under the event timeline** so that the two sets of data can be visually correlated.

Share Context: Event Timeline and Emotions Seismograph



[Rubin 2012]

Activities: Identify Insights

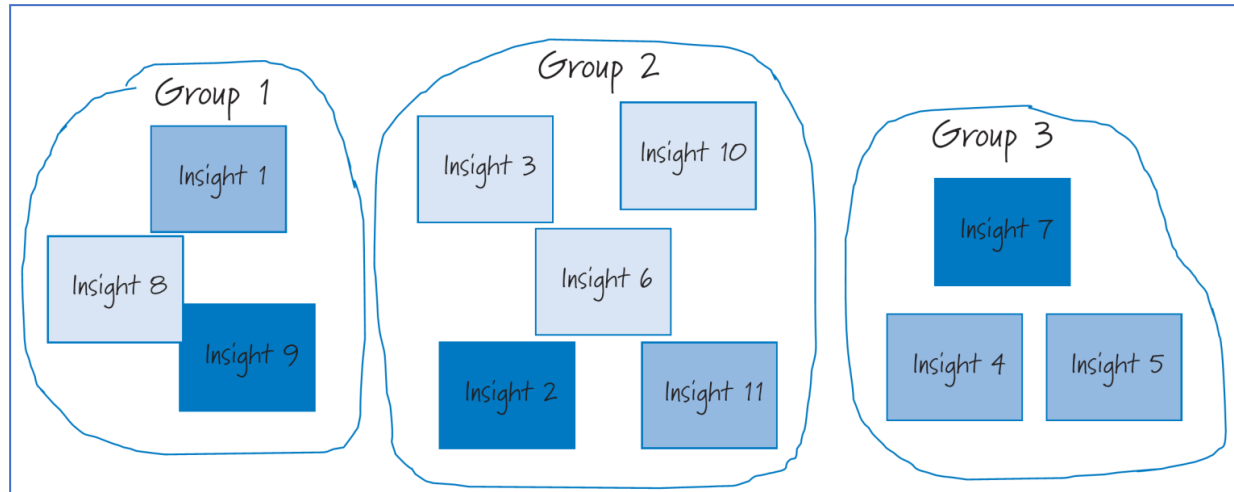
- Once a shared context has been established, the participants can examine, understand, and interpret the data to identify **process improvement insights**:
 1. Participants should start by **mining** the shared context data; they can look at the timeline and seismograph and ask the following questions:
 - What worked well?
 - What did not work well?
 - Where are some opportunities to do things differently?
 2. Participants **brainstorm** insights, capture them on cards and place them on a wall or any other surface so that everyone can see them.
 3. Another source for insights might be the team's **insight backlog**, a prioritized list of **previously generated insights** that have not yet been acted upon.
 - Participants mine this backlog to see which insights should be included in the current retrospective, represent them by cards and place them on the wall.
 4. Participants **cluster** the insights into meaningful groupings to indicate similar or duplicate cards.

Identify Insights: Grouping Insight Cards

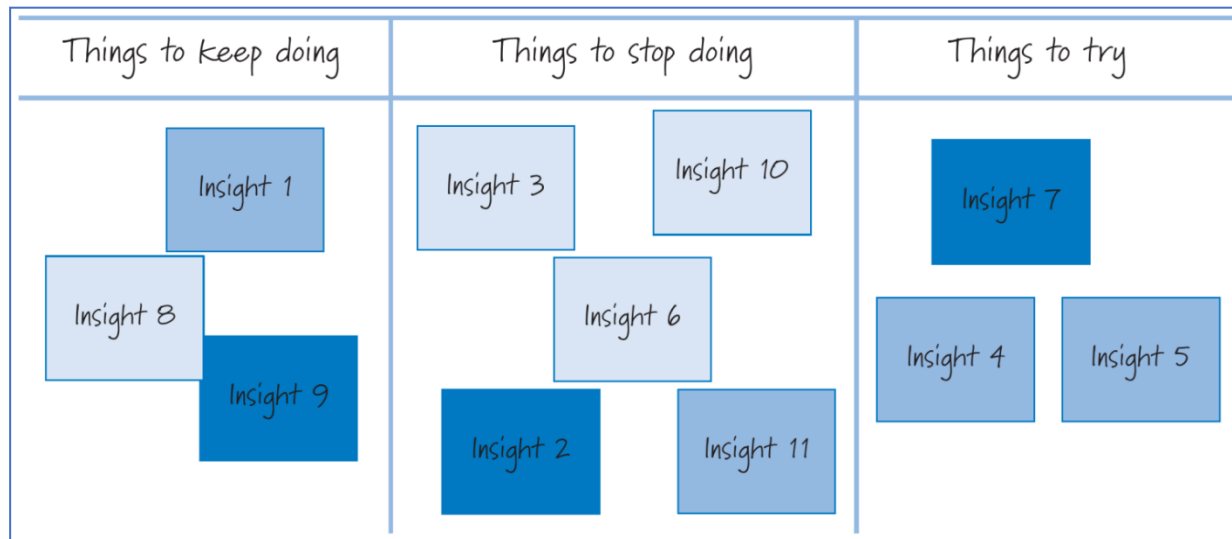
- **Silent grouping:** A time efficient and effective grouping technique during which people collaboratively create the **groupings without verbal** discussion.
 - The clusters can be **emergently determined** during the grouping activity; they are in fact similarity groups.
 - Alternatively, you can divide the wall into **predetermined category** areas (such as things to keep doing, things to stop doing, things to try) before the retrospective.
- Some of the insights might lead to discussions among the participants to better understand underlying causes, or important patterns or relationships.

Identify Insights: Grouping Insight Cards

Similarity
Groups



Predetermined
Groups



[Rubin 2012]

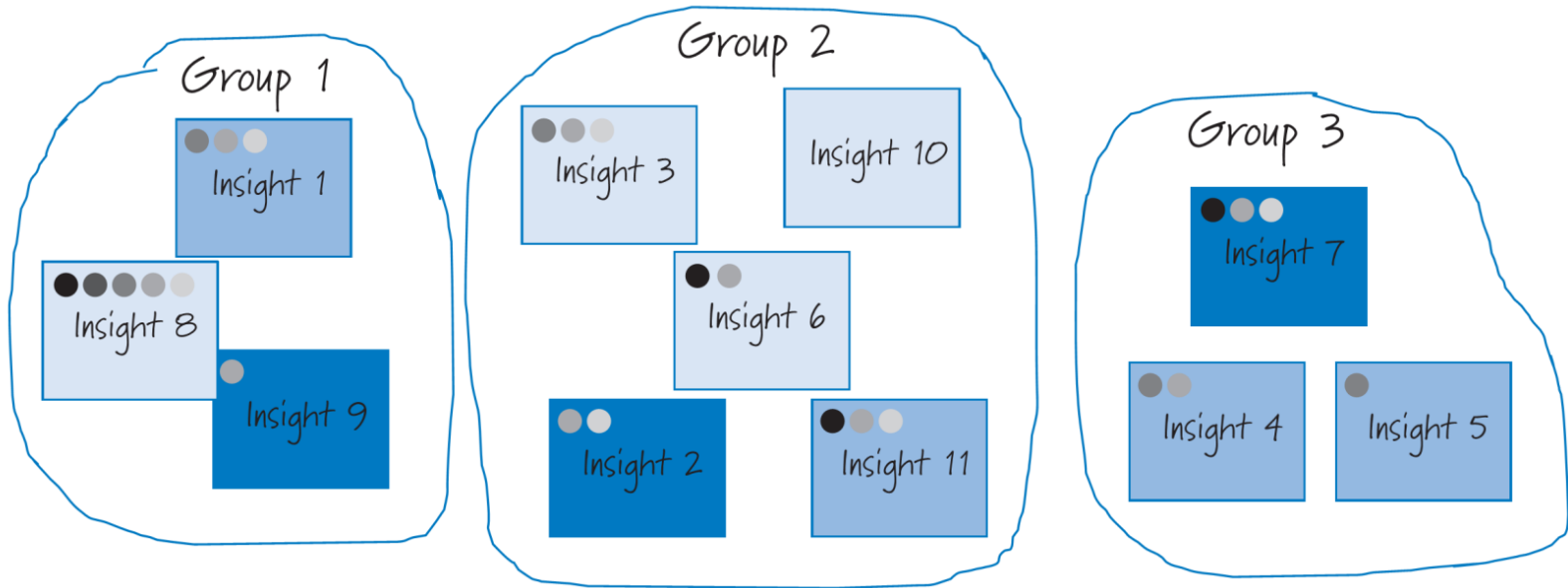
Activities: Determine Actions

- **Insights** are our **ideas** or perceptions of things that can be **improved**.
- To extract long-term value from these insights we need to move from discussing them to taking **demonstrable Actions** to leverage them.
 - For example:
 - Insight: “We’re wasting too much time because the code management system keeps failing.”
 - Action: “Have Talya apply the vendor patches to the code management system to make it more stable.” Talya can take this action in the next sprint.
- The participants should also take time to review **what happened** to the improvement actions from the **last retrospective**.
 - If those actions have not been completed (or even started), the participants **need to know why** before they start addressing new insights.
 - They might choose to **carry forward previous actions** or prioritize them against the new insights they have just identified.

Determine Actions: Select Insights

- Participants first need to determine **which improvement** insights to act on immediately and which to defer.
 - Retrospectives frequently identify many more improvement insights than the Scrum team and organization can digest and act on in a short period of time.
- **Dot Voting:** A practice for **prioritizing** insights:
 1. Each participant is given a small number (perhaps three to five) colored dots.
 2. The participants then simultaneously place their dots on the improvement insight cards that they feel are the highest priorities to address.
 - A person can put all the dots on one card or spread them over several cards.
 3. Once everyone has voted, the cards with the greatest number of votes should be considered first.
- The **number of selected** insights depends on how much **capacity** the participants are able to dedicate to the insights over the next sprint.

Select Insights: Dot Voting



[Rubin 2012]

Determine Actions: Decide on Actions

- Most **actions** should be **broken down** into specific **tasks** that one or more Scrum team members will perform during the upcoming sprint.
- Some actions do not require specific task-level work; for example, “People should make the effort to show up at the daily scrum on time.”
- Sometimes the actions represent impediments that the **Scrum Master** must resolve, perhaps through collaborating with the organization.
- Sometimes it is not possible to immediately address an insight; we might need to **explore the insight** before we can actually make an improvement.
 - In such cases, the proper action might be to investigate and collect data during the next sprint so that we can better understand the problem.

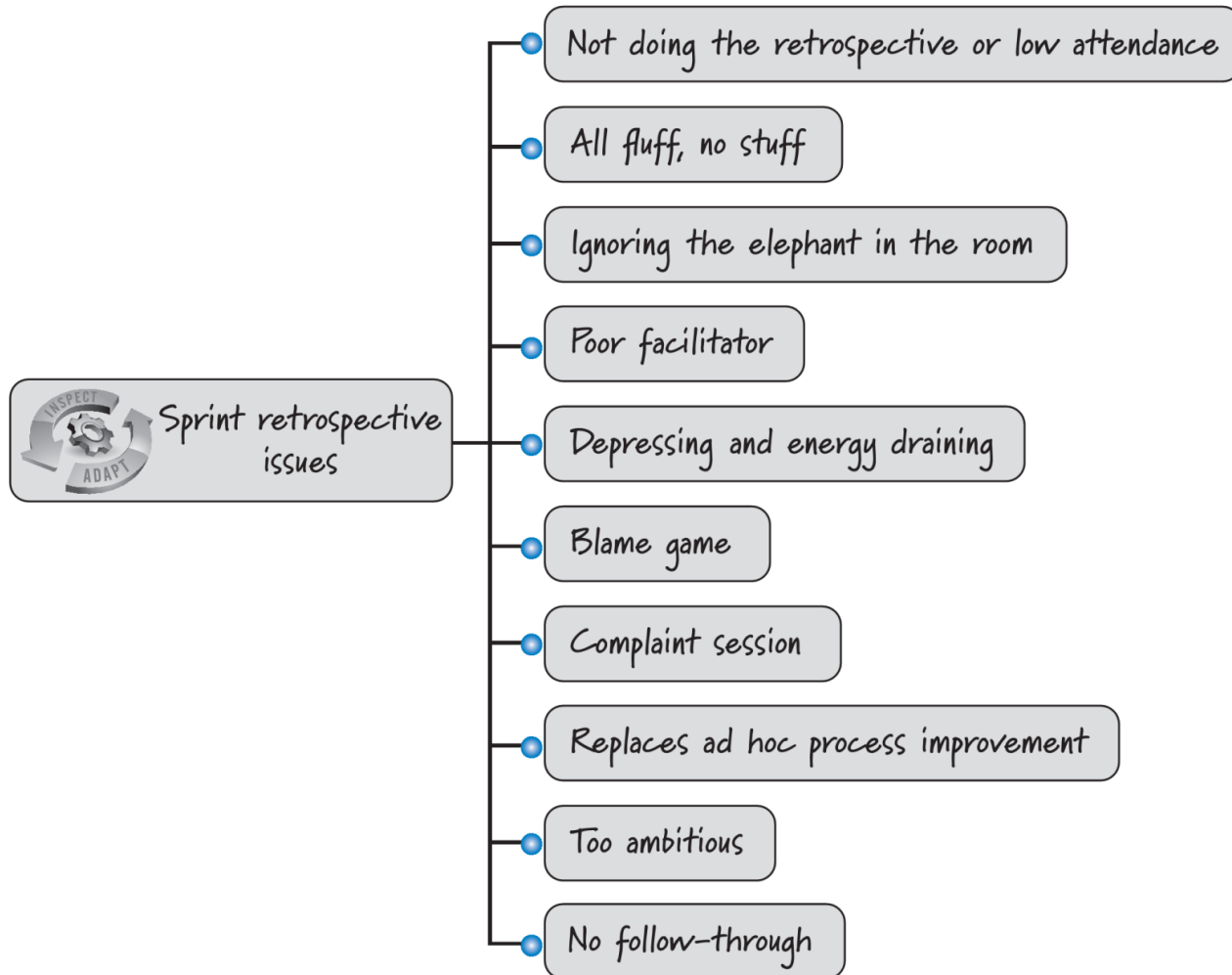
Activities: Close the Retrospective

- Close by recapping what **actions** the team has decided to take based on what the participants learned.
 - This might be as simple as describing each committed action item and who is going to work on it.
- Closing is also a good time to **appreciate people** and their participation.
 - Each participant should say a few kind words of appreciation regarding the contributions made by others.
 - Be sure to also recognize any non-Scrum team members who took time out of their busy schedules to participate in the retrospective.
- It's a good idea to spend a few minutes asking the team for suggestions to **improve** the team's approach to performing a **retrospective**.

Sprint Retrospective: Follow Through

- To ensure that what happens in the sprint retrospective does not just stay in the sprint retrospective, the participants should **follow up on the actions**.
 - Some actions (such as that everyone shows up on time for daily meetings) need only to be reinforced by the team members and the Scrum Master.
 - Others will need to be **addressed during the forthcoming** sprint-planning activity.
- The easiest way to handle the improvement actions is to **populate the sprint backlog** with tasks corresponding to each action before adding new features.
 - The team's available **capacity** to work on new features would then be **adjusted** downward by the estimated time these improvement tasks will take.
- Warning: One approach that does **not** work is to have an “**improvement plan**” for the team that is separate from the work it will do each sprint.
 - This approach will almost always lead to the improvement plan being subordinate to the typical feature-driven sprint plan.

Sprint Retrospective Issues



[Rubin 2012]