

# Domain VII Continuous Improvement (Product, Process, People)

The PMI-ACP Exam consists of 120 questions which can be categorised into seven domains. The Seventh domain: Domain VII Continuous Improvement (Product, Process, People) is the knowledge about "Continuously improving the quality, effectiveness, and value of the product, the process, and the team." (source: PMI-ACP Examination Content Outline).

Domain VII Continuous Improvement (Product, Process, People) accounts for 9% of all questions in the PMI-ACP Exam (i.e. ~11 questions among 120 PMI-ACP Exam questions)

According to the PMI-ACP Exam Content Outline, Domain VII Continuous Improvement (Product, Process, People) consists of 6 tasks:

1. **Review product, processes, and practices periodically** to look for rooms for improvement and efficiency enhancement.
2. **Conduct frequent retrospectives and experiments** to continually improve team processes and effectiveness.
3. Gather **feedback from stakeholders** on product increments and demonstrations to enhance value delivery.
4. Develop a team of generalising specialists by providing **learning and practising opportunities**.
5. Perform **value stream analysis** on existing processes to **remove wastes** and improve efficiency.
6. **Disseminate knowledge** gained during carrying out the project works to the whole organization for organizational improvement.

## PMI-ACP Study Notes: Domain VII Continuous Improvement (Product, Process, People)"

Below is a collection of the key knowledge addressed in Domain VII Continuous Improvement (Product, Process, People) and the 6 tasks related to the domain:

### Integration, Testing and Experiments:

- **Continuous Integration** (as a core practice in XP)
  - to continuously integrate changes (usually in small trunks) to the codebase by merging the new codes as soon as practicable (i.e. once ready)
  - to avoid code conflicts and minimize risks of incompatibility
  - on every integration, the codebase needs to be tested (usually by unit testing with automated testing tools / regression testing tools) typical setup for continuous integration:
    - ✓ A source code repository
    - ✓ A check-out and check-in process
    - ✓ An automated build process (compiles codes, runs tests and deploys)
  - if errors are found, fixing the broken build is of top priority
- **Continuous Improvement**
  - the Deming's PDCA Cycle (plan – do – check -act)
  - make use of process improvement and self-assessment for improved product
  - e.g. code refactoring and pair programming
- **Testing (Exploratory and Usability)**
  - Exploratory testing - seeks to find out how the software actually works, and to ask questions about how it will handle difficult and easy cases by asking test subjects to try the software
  - Usability testing - a special type of exploratory testing with emphasis on the usability of the software interface (whether the test subject will be able to perform core tasks on the interface without instructions and help) help to provide insights on the design of the software:
    - ✓ Users' expectations / habits
    - ✓ Users' ability to understand / comprehend the design of the interface
    - ✓ Users' value of the functions of the software
  - both will provide valuable feedback early in the project to enhance value delivery and avoid failure later on

- **Learning Cycle**
  - Agile software development is about learning - from little known about the end product in the beginning to (hopefully) delivering the maximal value in the end
  - understanding of the requirements as well as the technology to make the product feasible increase incrementally during the project each retrospective/review is an opportunity to learn
  - it is recommended to keep learning cycles short so that new knowledge gained can be fed into the project as soon as possible

## Review and Retrospective:

- **Retrospective**
  - an Agile process for self-evaluation to be performed at the end of each iteration (somehow similar to the "postmortem" meeting or "lessons learned" meeting in traditional project management)
  - a continuous process improvement for timely implementation
  - involved the **Agile development team only with a timebox of up to 1 hour**
    - ✓ a valuable learning opportunity for the Agile team
    - ✓ analyse, adapt, and improve the entire development process improve productivity, capability, quality and capacity
  - actionable improvement tasks are to be implemented right in the next iteration for instant improvements
  - focus on **what went well, what went wrong and how the team can improve** in next iteration and beyond without finger-pointing typical agenda:
    1. set the stage – get people comfortable to speak and outline the topics for discussion
      - check-in – everyone express in 1 or 2 words about the expectation of the retrospective
      - focus on / focus off – which side to focus on (e.g. dialogue vs debate)
      - ESVP – choose 1 from among “explorers, shoppers, vacationers and prisoners” that describes their feeling anonymously
      - working agreements – work on different topics in small groups first
    2. gather data
    3. generate insights
    4. decide what to do – identify the high priority items to devise an action plan
    5. close the retrospective – express appreciation and feelings
      - Plus / Delta – what should be done more / what should be changed
      - Helped, Hindered, Hypothesis – three flip charts for participants to add ideas on
  - the improvement stories chosen in the retrospective will be treated as **non-functional backlog**
  - **Retrospective Meeting vs Review Meeting**
    - Retrospective meeting is for the development team only with the primary aim for process improvement
    - Review meeting is for demonstration / getting acceptance of deliverables with management, product owner and stakeholders
  - **Retrospective Meeting vs Lessons Learned Meeting**
    - Retrospective meeting is carried out once per iteration and identifies areas for improvement
    - Lessons Learned meeting is carried out once at the end of the project / phrase as the project closure activity and all the lessons learned are to be identified and documented (according to PMBOK Guide) for future references (not as feedback to the project itself)
  - **Intraspectives**
    - intra = inside / within, (per)spective = a particular way of viewing things / inspection: intraspective = inspecting within / seeing inwardly intraspективs
    - in Agile project management is an ad hoc discussion / meeting by the Agile team to review on the team practices or teamwork during the sprint, often called for when something went wrong
  - **Pre-mortem (rule setting, failure analysis)**

- A premortem is the hypothetical opposite of a postmortem in which team members are asked to generate plausible reasons for the project's assumed failure.
- To make it safe for team members to voice out their reservations about the plan / project direction / etc.
- Can identify possible causes of failure which are missed during risk analysis

## Value Stream Analysis and Mapping:

- The objectives of value stream analysis:
  - to provide optimum value flow to customers through value creation processes
  - by eliminating wastes in every process through analysis (e.g. value stream mapping) and enhancements
- **Value Stream Mapping**
  - **Simplicity** – the art of maximizing the amount of work not done – **is essential**
  - Value stream mapping is originally a graphical tool for analysing the flow of materials in manufacturing from its beginning through to the customer (e.g. in lean manufacturing).
  - It is later adopted for value creation of services (e.g. IT development projects).
  - usually involves the following steps:
    1. understand the current state (**current state**)
      - ✓ create a visual map of the value flow of the current state
      - ✓ distinguish between value-adding processes and non-value-adding operations (including wastes)
      - ✓ find delays, wastes and constraints
    2. analyse and modify (the ideal **future state**)
      - ✓ create a new value stream map for the desired state after optimization (e.g. removing delays, wastes and constraints)
    3. communicate and carry out the improvements
      - ✓ ensure all team members understand the values of the improvement work
      - ✓ develop a roadmap for implementing the actions
    4. verify and validate the improvements

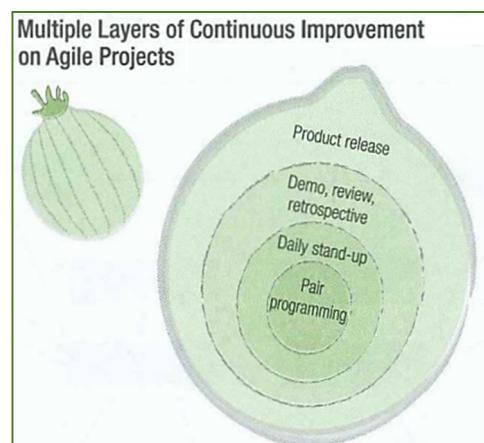
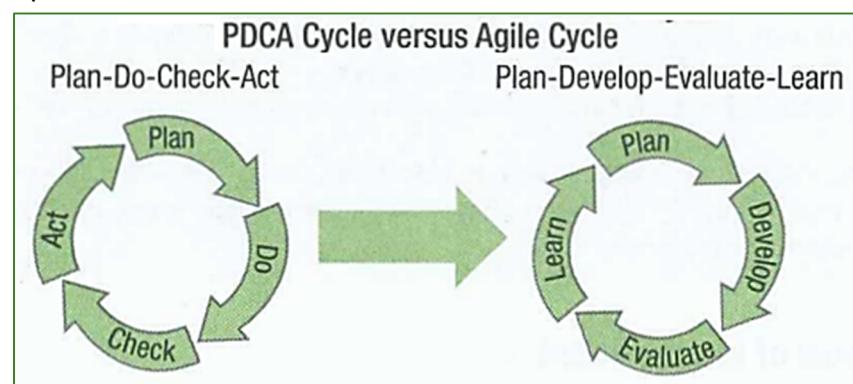
## Agile vs. Traditional:

Most traditional projects capture the majority of their lessons learned at the end of the project. The intent is to allow the organization to apply these lessons to future projects that are in a similar business or technical domain or have similar team dynamics. This approach, frankly, is too little, too late. We need to apply the benefits of learning as we go—on our current project, and as soon as possible.

So continuous improvement is more like a journey than a destination, since it is an ongoing part of the iterative life cycle that drives agile methods.

## Kaizen:

Kaizen is a process for continuous improvement, which means **"change for the better"**. It's true that agile retrospectives aren't exactly the same as kaizen, which looks for continuous daily improvements rather than learning after each iteration. It's basically an attitude or mindset that can help us understand why agile methods approach improvement as a continuous team-based, iterative process.



## Continuous Improvement—Process:

The main triggers for driving process changes on agile projects are the iteration retrospectives,

## Process Tailoring:

As a general recommendation, teams that are new to agile should use their methodology “out-of-the-box” for a few projects before attempting to change it. That’s because the problems a new team encounters with a standard technique or practice may be due to their lack of skill or experience using that technique, rather than issues with the technique itself. Scrum are less keen on tailoring.

## Hybrid Models:

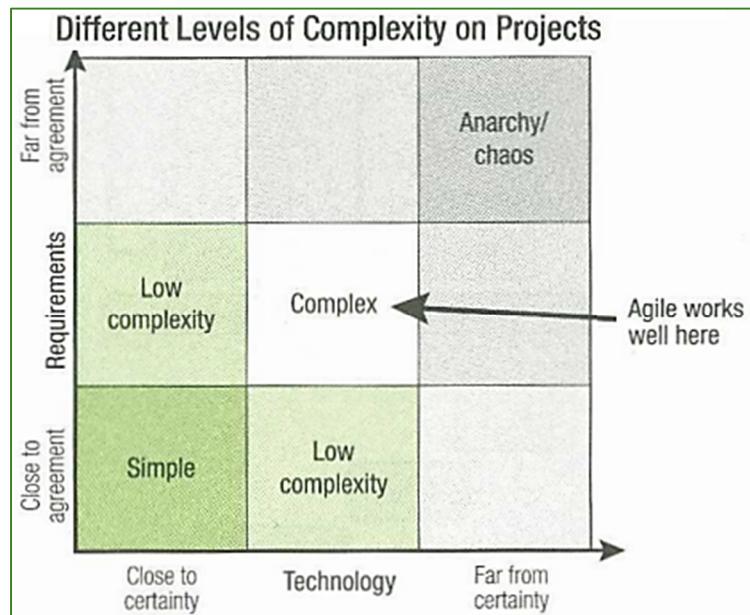
A hybrid is simply a combination of two different types of things.

## Agile-Agile Hybrid: Scrum-XP:

- XP provides great technical guidance but not much in the way of project governance guidance.
- Scrum provides a project governance model but not much in the way of how to do the work.

## Agile-Traditional Hybrids:

Even if the plan for the execution of the project is linear rather than iterative, we think a team can always benefit from using certain components of agile—such as daily stand-ups, retrospectives, and local decision making. And on an agile project, it might be best to use a traditional approach for certain portions, such as a procurement effort that has to follow a strict purchasing workflow—we simply plan, specify, and order the equipment in good time, and then wait for it to arrive.



## Systems Thinking:

When a team is considering changing their process, it can be helpful to understand the systems-level environment for the project. This type of analysis is called systems thinking.

## Process Analysis:

Process analysis involves reviewing and diagnosing issues with a team's agile methods.

Anti-Patterns	Success Criteria	Success Patterns
One size for all projects	The project got shipped.	Face-to-face communications
Intolerant	The leadership remained intact.	Excess methodology weight is costly
Heavy	The team would work in the same way again.	Larger teams need heavier methodologies
Embellished		More ceremony for more critical projects
Untried		Increasing feedback and communications reduces the need for intermediate deliverables
Used once		Discipline, skills, and understanding over process, formality, and documentation
		Efficiency is expendable in nonbottleneck activities

## Value Stream Mapping:

The value stream mapping process involves the following steps:

1. Identify the product or service to be analysed.
2. Create a value stream map of the current process, identifying steps, queues, delays, and information flows.
3. Review the map to find delays, waste, and constraints.
4. Create a new value stream map of the desired future state of the process, optimized to remove or reduce delays, waste, and constraints.
5. Develop a roadmap for creating the optimized state.
6. Plan to revisit the process in the future to continually refine and optimize it.

$$\text{Total cycle time} = \text{Value-added time} + \text{Nonvalue-added time}$$

$$\text{Process cycle efficiency} = \text{Value-added time} / \text{Total cycle time}$$

## Project Pre-Mortems:

A pre-mortem exercise typically includes four working steps:

1. imagine the failure
2. Generate the reasons for the failure
3. Consolidate the list
4. Revisit the plan.

## Continuous Improvement—Product:

Just as we continuously refine our processes, so do we continuously improve the evolving product.

## Reviews:

The scientific method is a process for investigating things and learning new knowledge or correcting previous knowledge. This process involves making observations, thinking of a hypothesis to explain our observations, conducting experiments to test our hypothesis, and then confirming, adapting, or rejecting our hypothesis based on the data we have gathered.

Agile reviews follow some basic ground rules or guidelines:

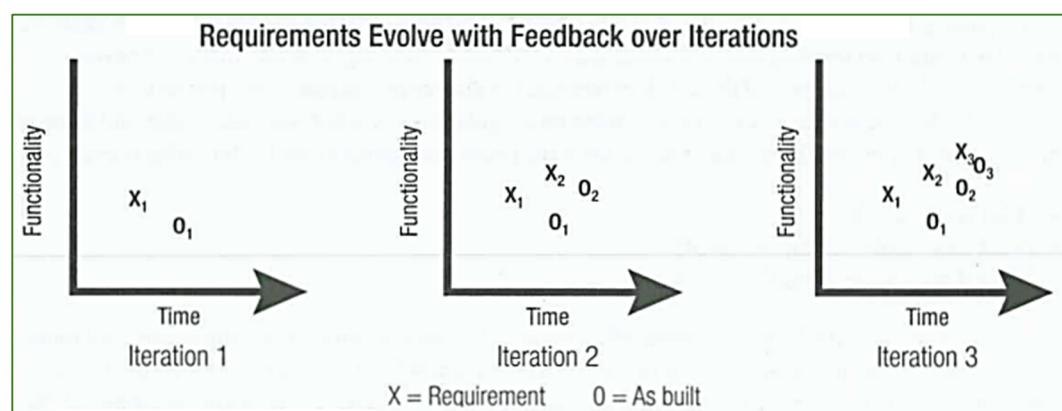
- **Let the data speak for itself.** We don't try to prejudge the results of our experiments or filter out unlikely suggestions. Instead, we run the experiments that we agree are the most promising, and then examine the results.
- **Respect individuals.** We value everyone's view equally; we don't judge suggestions based on the person's role or seniority.
- **Diverge then converge.** We encourage diverse suggestions to increase the likelihood of generating valuable insights and ideas. To generate these ideas, we diverge—we generate our ideas individually so that they aren't influenced or inhibited by the views of our colleagues. Then we converge—we gather together to review the ideas, agree on the most likely root causes of issues, or identify the best suggestions for further experiments.

## Feedback Methods:

So, when we demonstrate functionality to the product owner, it serves two purposes. First, we learn about any differences between what was asked for and what we interpreted and built (the gulf of evaluation).

And second, we learn

about any new or adjusted functionality that is required (**IKIWISI**) ("I'll know It When I See It!").



## Continuous Improvement—People:

During the retrospective, we struggle with the following questions:

- What is going well?
- What areas could use improvement?
- What should we be doing differently?

As problems are identified, we brainstorm solutions, and then we commit to trying the selected solutions for one or two iterations before meeting again to discuss (in another retrospective) whether the situation has improved. If the changes helped—great! We adopt them as part of our processes on the project. If they did not help, we consider the effort a learning opportunity and decide whether to try something else or revert to our earlier process.

## The Benefits of Retrospectives:

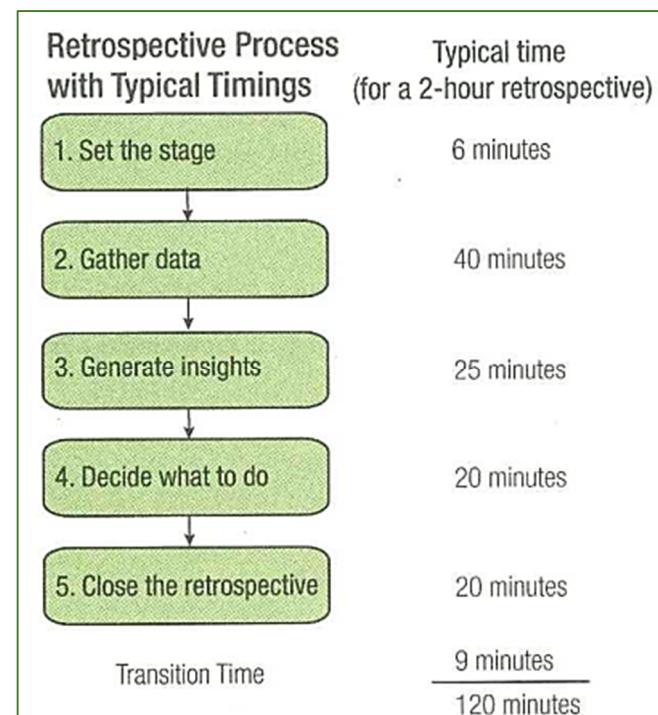
Retrospectives offer a number of benefits for teams, including the following types of improvements:

- **Improved productivity:** By applying lessons learned and reducing rework, the team can get more productive work done.
- **Improved capability:** Retrospectives provide a venue for spreading scarce knowledge, and as the number of people who have the scarce knowledge increases, so does the number of people who can perform tasks associated with the knowledge.
- **Improved quality:** We can improve quality on our projects by finding the circumstances that have led to defects and removing the causes.
- **Improved capacity:** Retrospectives focus on finding process efficiency improvements, which can improve the team's capacity to do work.

## The Retrospective Process:

### Step 1: Set the Stage:

- **Check-In:** Participants answer a series of check-in questions with one or two words or a short phrase.
- **Focus On/Focus Off:** The team members discuss productive and unproductive ways of participating and agree to stay in the “Focus On” column.
- **ESVP:** Participants anonymously identify their attitude toward the retrospective as Explorer, Shopper, Vacationer, or Prisoner.
- **Working Agreements:** The team brainstorms and then defines the working agreements they would like to put in place for the retrospective.



✓ **Explorers:** Explorers are eager to discover new ideas and insights, and they want to learn everything they can.

**Shoppers:** Shoppers will look over all available information and will happily go home with one useful new idea.

**Vacationers:** Vacationers aren't interested in the work of the retrospective, but they are happy to be away from their regular job.

**Prisoners:** People who classify themselves as prisoners feel like they are being forced to attend the retrospective and would rather be doing something else.

In setting the stage, we aim to:

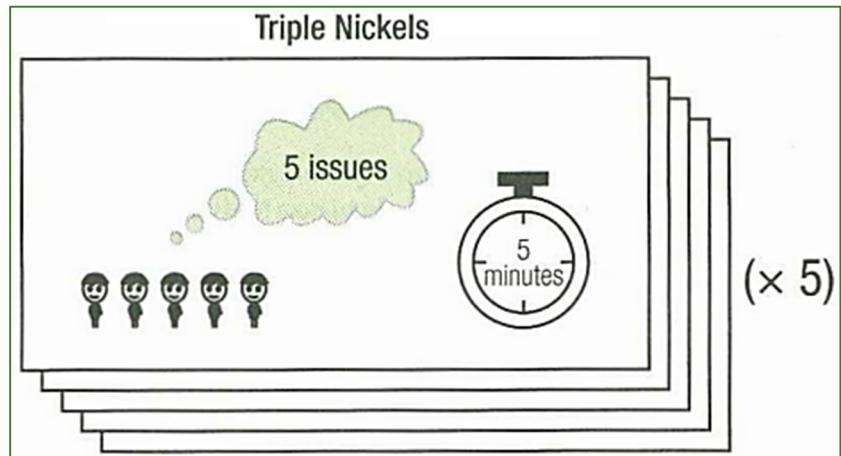
- ✓ Explain why we are doing the retrospective.
- ✓ Get people talking so they are comfortable contributing throughout the retrospective.
- ✓ Outline the approach and topics of the retrospective.
- ✓ Establish ground rules.
- ✓ Determine if people feel comfortable enough to contribute to the retrospective

## Step 2: Gather Data

- **Timeline:** The team members create a timeline to track the progression of the iteration.
- **Triple Nickels:** The team is divided into five groups that spend five minutes gathering or building on five ideas, five times.
- **Colour Code Dots:** The team members identify where their energy was high and low during the course of the iteration.
- **Mad, Sad, Glad:** The participants identify their emotional reactions throughout the iteration.
- **Locate Strengths:** The team identifies what went well during the project.
- **Satisfaction Histogram:** The team members create a graph that shows how satisfied they feel about a particular area or issue.
- **Team Radar:** The team assesses how they have performed against their previous process improvement goals.
- **Like to Like:** The team compares their reactions to events that occurred over the course of the iteration.

### Triple Nickels:

Triple Nickels is a data-gathering exercise in which participants spend five minutes gathering data on at least five ideas related to a specified topic. (This technique gets its name from a shooting competition that engages five targets from five yards in five seconds.) The team is divided into groups of five (or if the entire group is smaller than seven people, the exercise is done as one group), and the groups conduct five rounds of expansion.



## Step 3: Generate Insights

- **Brainstorming:** The team focuses on generating a high volume of ideas that will be filtered afterward. Common approaches include Free-for-All, Round-Robin, and Quiet Writing.
- **Five Whys:** The participants analyse the underlying cause of a problem by asking "Why?" five times to move beyond their automatic answers and identify the actual root cause of the issue.
- **Fishbone Analysis:** The team uses this diagramming tool—often along with the Five Whys exercise—to display their root cause analysis of problem.
- **Prioritize with Dots:** To determine their priorities, the team members use the dot voting technique.
- **Identify Themes:** The participants identify recurring patterns in the strengths they identified in Step 1.

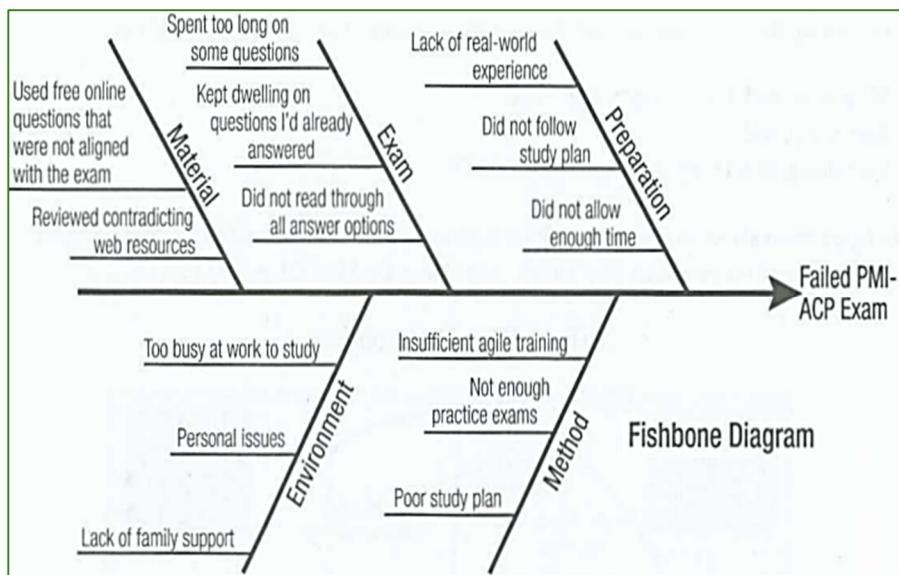
### Brainstorming

Brainstorming exercises aim to generate a large number of ideas that are then filtered into a select list of ideas that will move forward in the process. The high volume of ideas generated during such exercises helps counter the common phenomenon that a team's best ideas rarely emerge first. There are many different approaches to brainstorming, including the three methods: Quiet Writing, Round-Robin, and Free-for-All.

### Five Whys:

- ✓ **Question 1:** Why did we get that system crash in the iteration demo?
- ✓ **Answer 1:** We tried to access sales data for a store with no sales.
  
- ✓ **Question 2:** Why does accessing a store with no sales cause a problem?
- ✓ **Answer 2:** The fetch routine returns a null value that is not handled by the system.
  
- ✓ **Question 3:** Why don't we catch null values and display a more meaningful error message?

- ✓ **Answer 3:** We do catch them where we know about them, but this was the first time we had seen it for sales.
- ✓ **Question 4:** Why aren't all query returns coded to handle nulls?
- ✓ **Answer 4:** I don't know; it has never been a priority.
- ✓ **Question 5:** Why is it not a priority, as it seems like it's really a weak link in the system?
- ✓ **Answer 5:** Agreed. We should add it to the module walk-through checklist.



### Fishbone Analysis:

#### Step 4: Decide What to Do

- **Short Subjects:** The team uses an action wheel with categories such as “Keep, Drop, Add” or “Start Doing, Stop Doing, Do More Of, Do Less Of” to identify problem-resolution actions.
- **SMART Goals:** The participants transform their list of action items into goals that are SMART: Specific, Measurable, Attainable, Relevant, and Timely.
- **Circle of Questions:** Each participant asks a question about how to improve one of the issues that has been identified, to be addressed by the next person in the circle.
- **Retrospective Planning Game:** In this exercise, the team plans the tasks required to reach the process improvement goals they have identified for the next iteration.

#### Step 5: Close the Retrospective

- **Plus/Delta:** The team records what they want to do more of (“plus”) and what they want to change (“delta”) in two columns.
- **Helped, Hindered, Hypothesis:** The team members provide feedback about the retrospective itself—what helped, what hindered, and any ideas they came up with (“hypotheses”) for improving future retrospectives.
- **Return on Time Invested (ROTI):** Participants discuss the benefits of retrospectives, and then grade the meeting on a five-point scale to show whether their time was well spent.
- **Appreciations:** Team members have an opportunity to express their appreciation to each other for specific efforts during the iteration.

