DMAIC: Breaking the Cycle of Recurring Project Problems

A structured approach for Agile project managers to permanently solve delivery challenges

by Kimberly Wiethoff, MBA, PMP, PMI-ACP

Managing Projects The Agile Way

#ManagingProjectsTheAgileWay #DMAIC #LeanSixSigma #ProjectManagement #ProcessImprovement #AgileDelivery

The Problem: Project Groundhog Day

Every project manager has experienced it: the same issues surfacing sprint after sprint, despite your best efforts to address them.

- Testing delays that push deadlines
- Approval bottlenecks holding up progress
- Stories consistently rolling over to next sprint
- Scope creep that never truly gets contained

Introducing DMAIC: Beyond Agile Problem-Solving

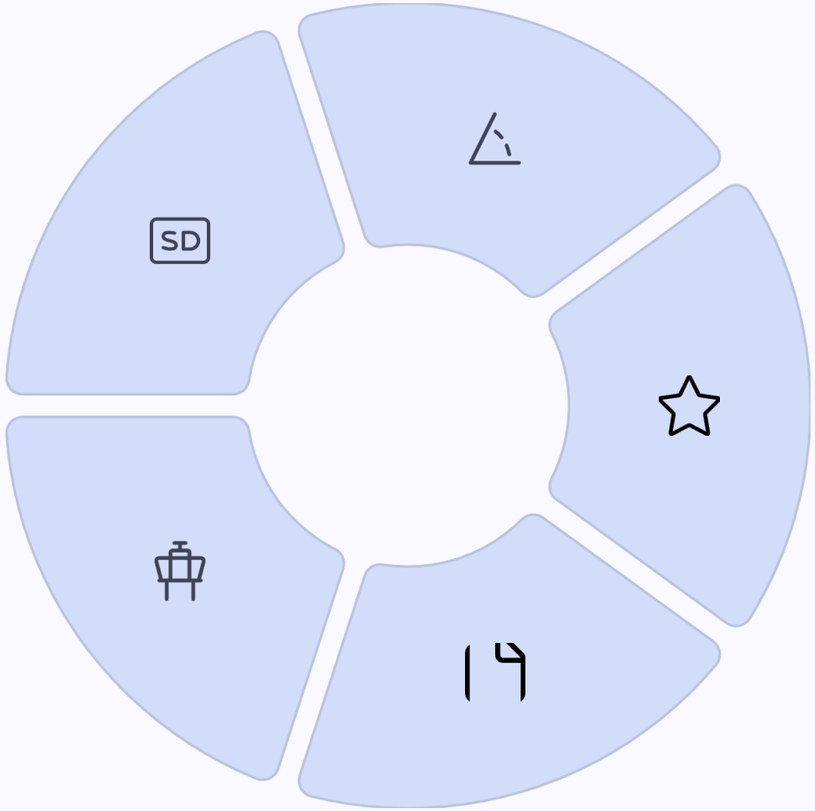
While Agile gives us iterative improvement, it doesn't always provide the structure to solve persistent problems at their root.

Define

Clearly state the problem and desired outcome

Measure

Gather data to understand the current state



Analyze

Identify root causes of the issue

Improve

Design and implement solutions

Control

Sustain the gains and prevent regression

When to Use DMAIC in Agile Environments

DMAIC is your go-to approach when you hear these recurring statements:

> "We never finish all the stories we commit to."
>
> "Testing is always the bottleneck."
>
> "Stakeholder feedback comes too late."
>
> "We're constantly reworking the same feature."

These aren't one-off issues. They're **patterns** that need systematic resolution.

- Recurring blockers
- Quality issues
- Communication breakdowns
- Hand-off failures
- Resource constraints

Step 1: Define

Key Activities:

- Clearly articulate the recurring issue
- Identify all relevant stakeholders
- Establish project boundaries
- Define what success looks like

Example: "Our team consistently carries over 40% of stories into the next sprint, impacting delivery timelines and team morale."

Clarity is key—vague problems yield vague solutions.

Step 2: Measure

Gather Quantitative Data

- Sprint velocity trends
- Defect counts
- Cycle time metrics
- Story point completion rates

Collect Qualitative Data

- Team surveys
- Retrospective themes
- Stakeholder interviews
- Process observations

Establish Baseline

- Document current performance
- Identify patterns and trends
- Create visualization of data
- Set measurable improvement targets

Data removes assumptions and shows whether the issue is systemic or situational.

Step 3: Analyze

Use proven analytical techniques to identify true root causes, not just symptoms:

- **5 Whys** - Ask "why" repeatedly to drill down to fundamental causes
- **Fishbone (Ishikawa) diagrams** - Map potential causes across categories
- **Pareto charts** - Identify the vital few causes creating the majority of issues
- **Value Stream Mapping** - Visualize waste in the process flow

Example: "Why are stories rolling over?" → "They're too big." → "Why?" → "Unclear requirements." → "Why?" → "Insufficient backlog refinement."

Step 4: Improve

1 Brainstorm Solutions

Generate multiple potential improvements based on root cause analysis

2 Prioritize Interventions

Select high-impact, low-effort solutions first

3 Pilot Changes

Test improvements in a controlled environment

4 Evaluate Results

Measure effectiveness against baseline data

5 Refine Approach

Adjust solutions based on feedback and results

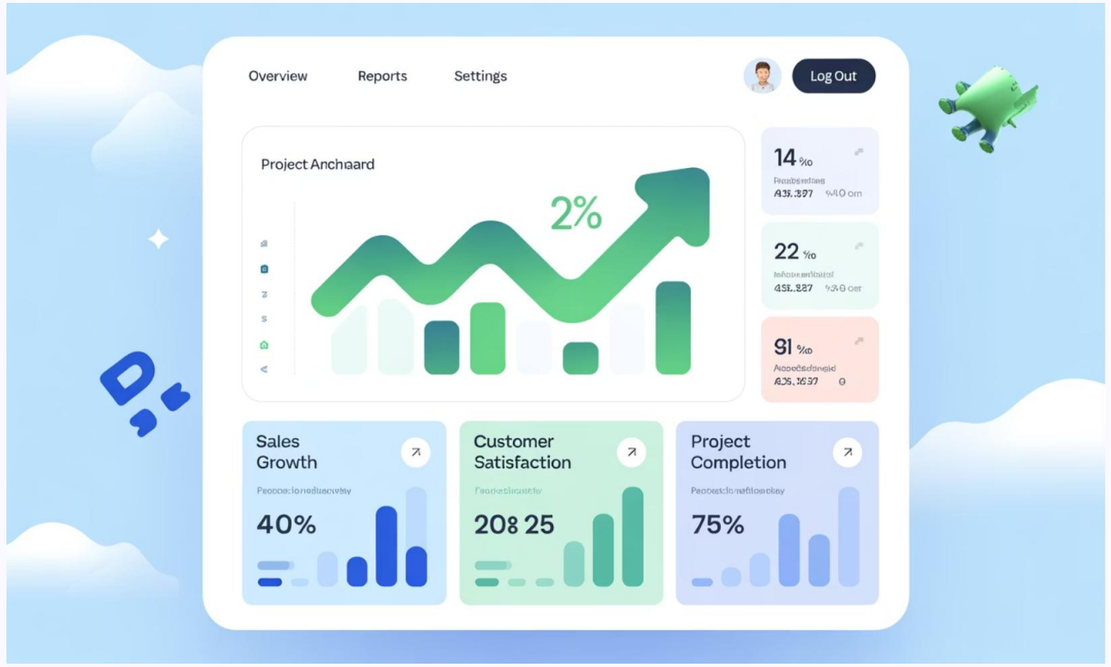
Start small and iterate. Improvement doesn't need to be complex–it just needs to stick.

Step 5: Control

Sustain your improvements by:

- Creating new process documentation
- Adding checklist items to ceremonies
- Establishing ongoing monitoring metrics
- Updating team training materials
- Building verification into retrospectives

Control is about **building stability** so the issue doesn't resurface when pressure mounts.



Monitor key performance indicators over time to ensure improvements stick.

Real-World Example: Story Rollover Crisis

Define

40% of stories carried over every sprint, causing missed deadlines and stakeholder frustration

1

Measure

Analyzed 3 months of Jira data, story completion rates, and team survey responses

2

Analyze

Root causes: poor backlog refinement and lack of clarity in acceptance criteria

3

Improve

Introduced weekly refinement meetings and implemented a Definition of Ready checklist

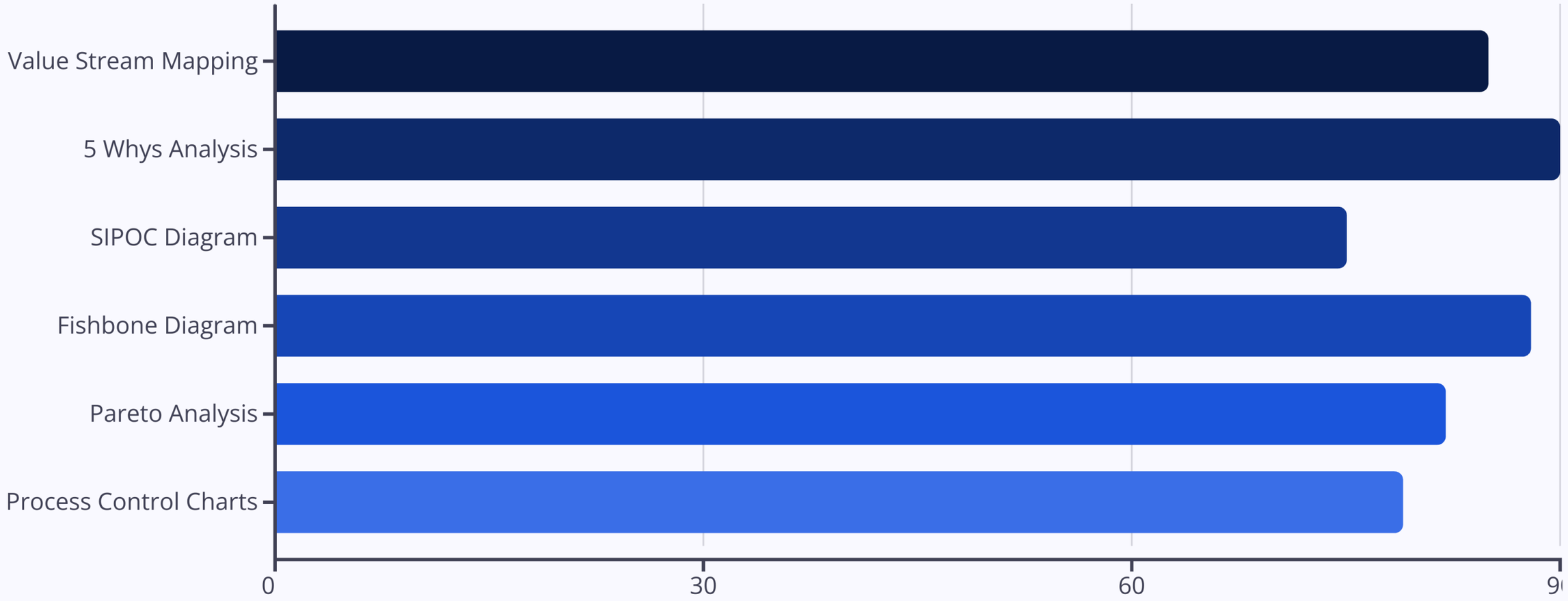
4

Control

Monitored carryover rates in retrospectives; improved from 40% to under 15% in two sprints

5

DMAIC Tools for Agile Project Managers



Based on survey of 200 Agile project managers who implemented DMAIC for process improvement

Integrating DMAIC with Agile Ceremonies

Sprint Planning

Use Define and Measure insights to adjust capacity planning and story prioritization

Daily Standups

Track improvement initiatives and identify new data points for the Measure phase

Sprint Review

Demonstrate progress on DMAIC metrics to stakeholders alongside feature demos

Retrospective

Use Analyze techniques to dig deeper into issues and evaluate Control effectiveness

DMAIC doesn't replace Agile—it enhances it by providing structure to solve persistent problems.

Common Pitfalls and How to Avoid Them

Pitfalls:

- Rushing through the Define phase
- Collecting insufficient data
- Fixing symptoms instead of root causes
- Implementing too many changes at once
- Neglecting the Control phase

Best Practices:

- Take time to get stakeholder alignment
- Set clear metrics before starting
- Use multiple analysis techniques
- Test one solution at a time
- Build monitoring into existing ceremonies

Key Takeaways

Move from reactive to proactive problem-solving

DMAIC provides the structure to permanently resolve recurring issues, not just patch them temporarily.

Leverage data to drive decisions

Replace assumptions with measurements to identify true causes and validate improvements.

Integrate with existing Agile practices

DMAIC complements Agile frameworks by adding depth to continuous improvement efforts.

Build sustainable solutions

The Control phase ensures improvements stick, even when teams are under pressure.

In Agile environments, we move fast—but fast doesn't always mean smart. DMAIC transforms teams from problem managers to problem solvers.