

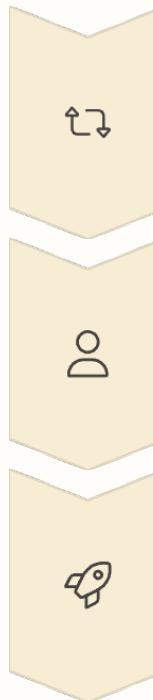


Agile Project Management and Scaling Agile

Dr.Vani Vasudevan



Introduction to Agile Project Management



Iterative & Incremental

Agile delivers work in small, consumable increments. Value is delivered frequently rather than all at once.

Collaborative

Teams work closely with customers. Feedback drives continuous improvement.

Adaptive

Embraces change even late in development. Prioritizes people over processes and tools.

Agile Values and Principles

4

Core Values

The Agile Manifesto outlines four key values that guide agile practices.

12

Principles

These principles expand on the values. They provide practical guidance.

17

Years

Agile has been transforming projects for nearly two decades. It continues to evolve.



Agile Manifesto Values

Individuals and interactions

People drive success through collaboration.
Prioritize team dynamics over rigid processes.

Working software

Delivering functional products trumps excessive documentation. Value is in what works.

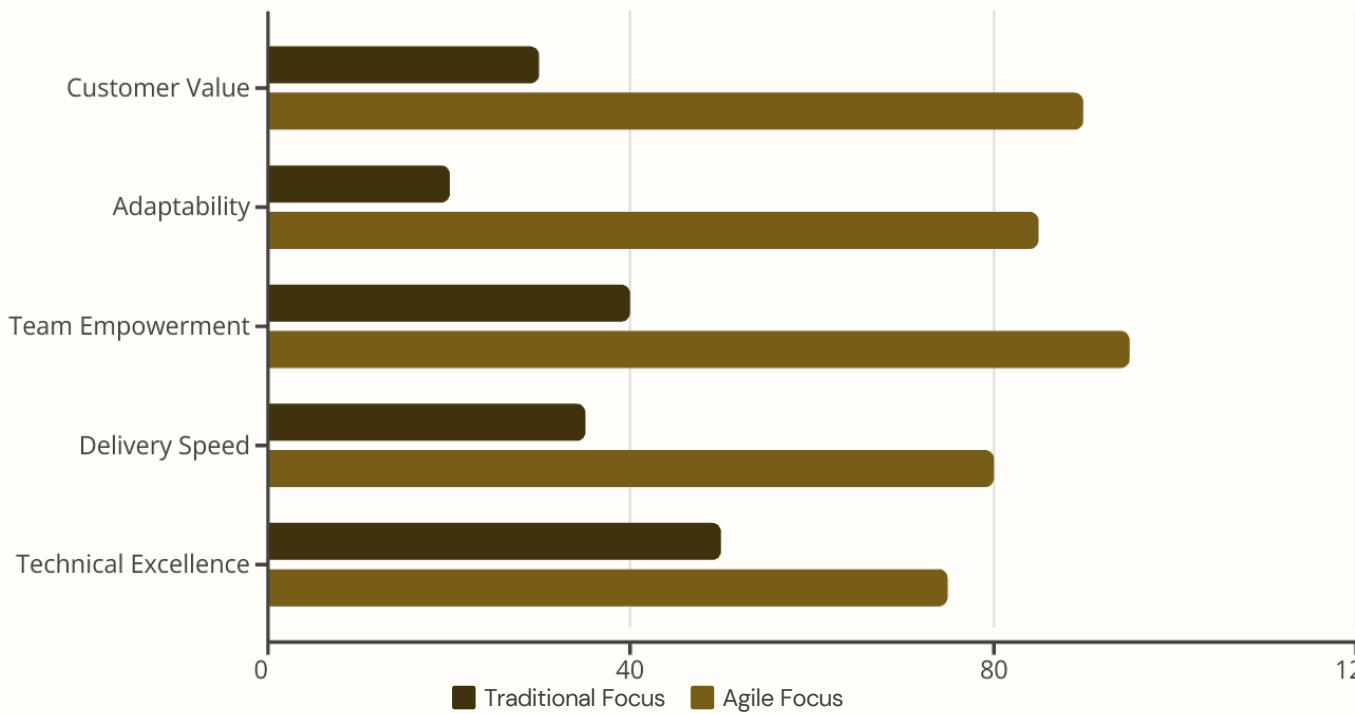
Customer collaboration

Work with customers throughout development.
Partnership beats contract negotiation.

Responding to change

Adapt to evolving needs.
Flexibility is more valuable than rigid plans.

Core Agile Principles





Benefits of Agile Approach

Faster Time to Market

- Incremental delivery of features
- Early and continuous value
- Reduced time-to-market by 50%

Higher Quality

- Built-in testing practices
- Continuous feedback loops
- Reduced defect rates

Greater Adaptability

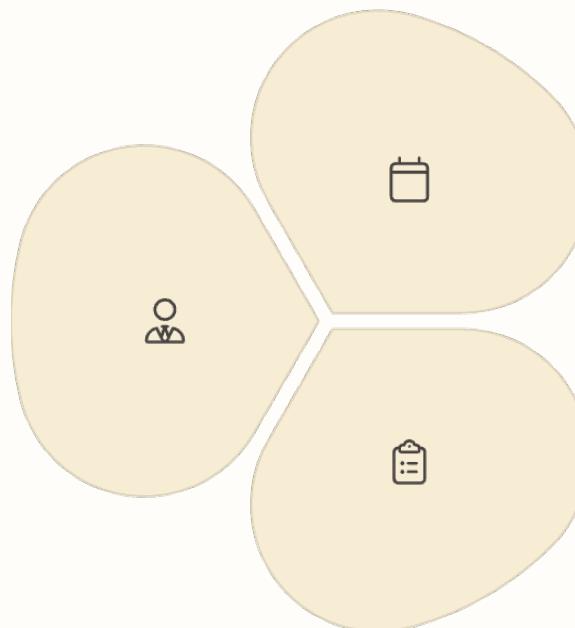
- Responds to market changes
- Adjusts priorities continually
- Embraces new requirements

Improved Satisfaction

- Customer input throughout
- Higher team engagement
- Better aligned solutions

Scrum Framework

- Roles
- Product Owner
 - Scrum Master
 - Development Team

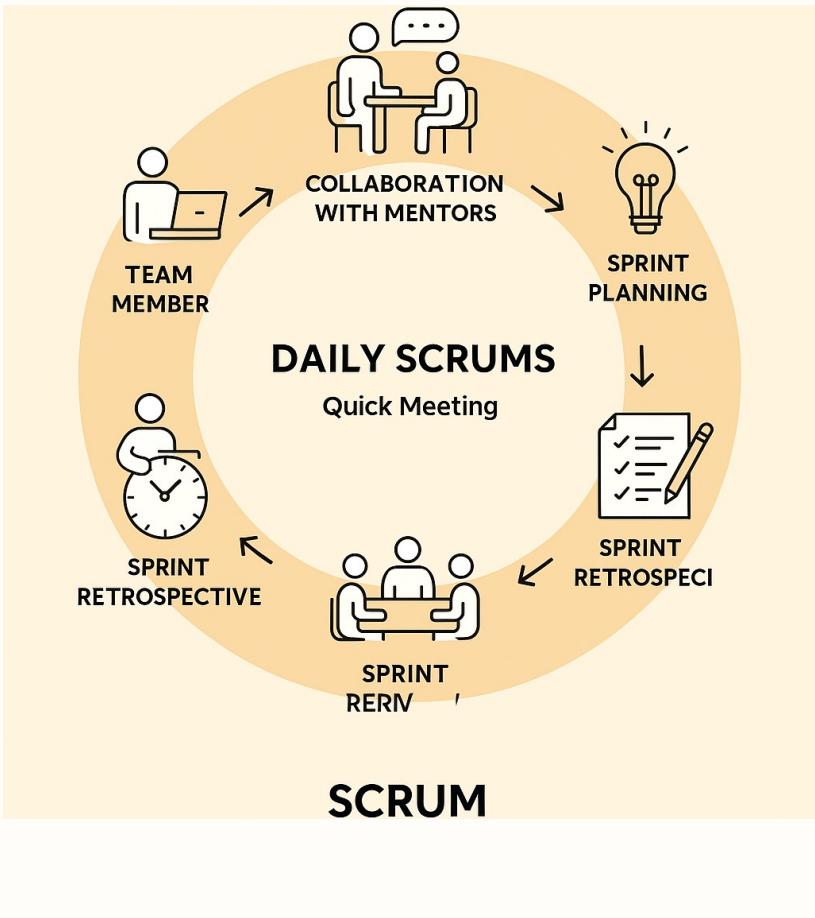


Ceremonies

- Sprint Planning
- Daily Standup
- Sprint Review
- Retrospective

Artifacts

- Product Backlog
- Sprint Backlog
- Increment



Scrum Deep Dive



Sprint Planning

Team decides what to build. Stories pulled from backlog. Estimation happens here.



Daily Scrums

15-minute synchronization. Three questions: done, doing, blockers.



Sprint Review

Demo completed work. Gather stakeholder feedback. Adjust product backlog.



Sprint Retrospective

Inspect and adapt process. What worked? What didn't? What's next?



Scrum Roles and Responsibilities



Product Owner

Maximizes product value. Manages backlog. Sets priorities. Represents stakeholder interests.



Scrum Master

Facilitates scrum process. Removes impediments. Coaches team. Protects from interference.



Development Team

Self-organizing group. Cross-functional skills. Collectively responsible for delivery.

Effective Product Backlog Management



Prioritization

Order items by business value and risk



Refinement

Clarify details and acceptance criteria



Sizing

Estimate effort using story points



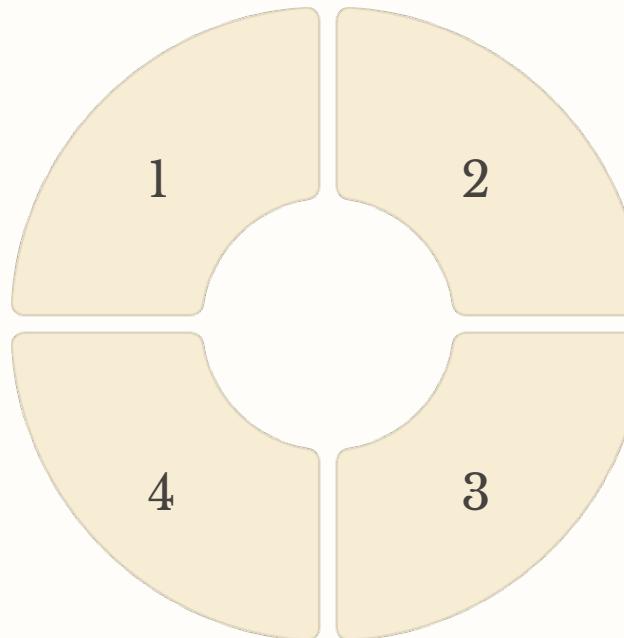
Slicing

Break large items into manageable pieces

User Stories in Agile

Format
As a [user], I want [feature] so that [benefit].

Sizing
Story points reflect relative effort.
Fibonacci sequence often used.



INVEST Criteria

- Independent
- Negotiable
- Valuable
- Estimable
- Small
- Testable

Acceptance Criteria

Define when a story is complete.
Specify expected behavior.

Effective Sprint Retrospectives



What went well?

Celebrate successes and positive patterns



What could improve?

Identify pain points and challenges



Action items

Create specific, measurable improvements



Kanban Framework



Visualize Work

Make work visible on a Kanban board. Visualize workflow stages from start to finish.



Limit Work-in-Progress

Restrict items in progress. Avoid multitasking to optimize flow and reduce cycle time.



Manage Flow

Focus on continuous delivery. Optimize for smooth, predictable workflow without fixed iterations.



Implementing Kanban Effectively

Visualize Current Workflow

Map existing process states.
Create columns for each step.
Include policies.

Apply WIP Limits

Set maximum items per column. Focus on finishing not starting. Prevent bottlenecks.

Manage Flow

Measure lead time and cycle time. Remove blockers.
Optimize for smooth delivery.

Implement Feedback Loops

Hold cadence meetings.
Review metrics regularly. Make process policies explicit.

Kanban vs. Scrum Comparison

Kanban

- Continuous flow
- No required roles
- Changes anytime
- Optional timeboxes
- WIP limits
- Optional estimation

Scrum

- Timeboxed sprints
- Defined roles
- No changes mid-sprint
- Required ceremonies
- Sprint commitment
- Required estimation

Scrumban: Hybrid Approach

Scrum Elements

- Roles and responsibilities
- Regular planning
- Retrospectives

Kanban Elements

- Visualized workflow
- WIP limits
- Pull system

Best For

- Support and maintenance
- Scrum transition
- Unpredictable work



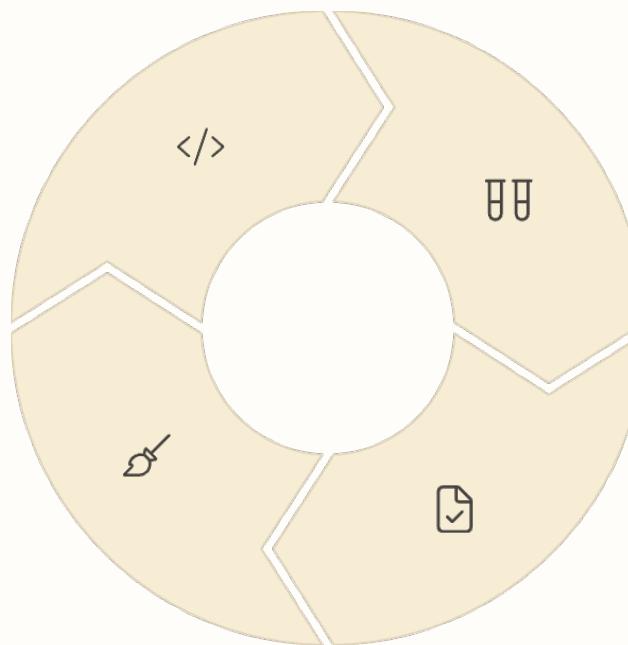
Extreme Programming (XP)

Pair Programming

Two developers work together at one workstation

Refactoring

Improve code structure regularly



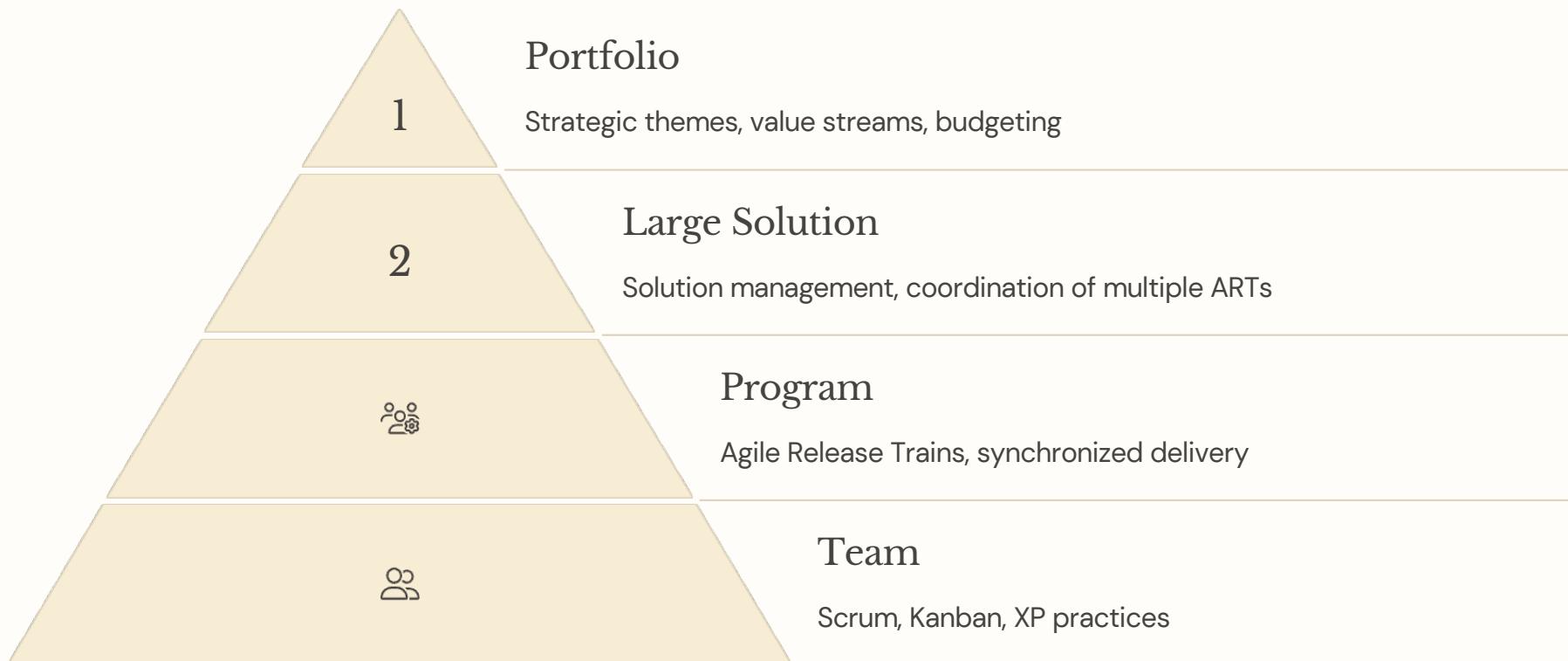
Test-Driven Development

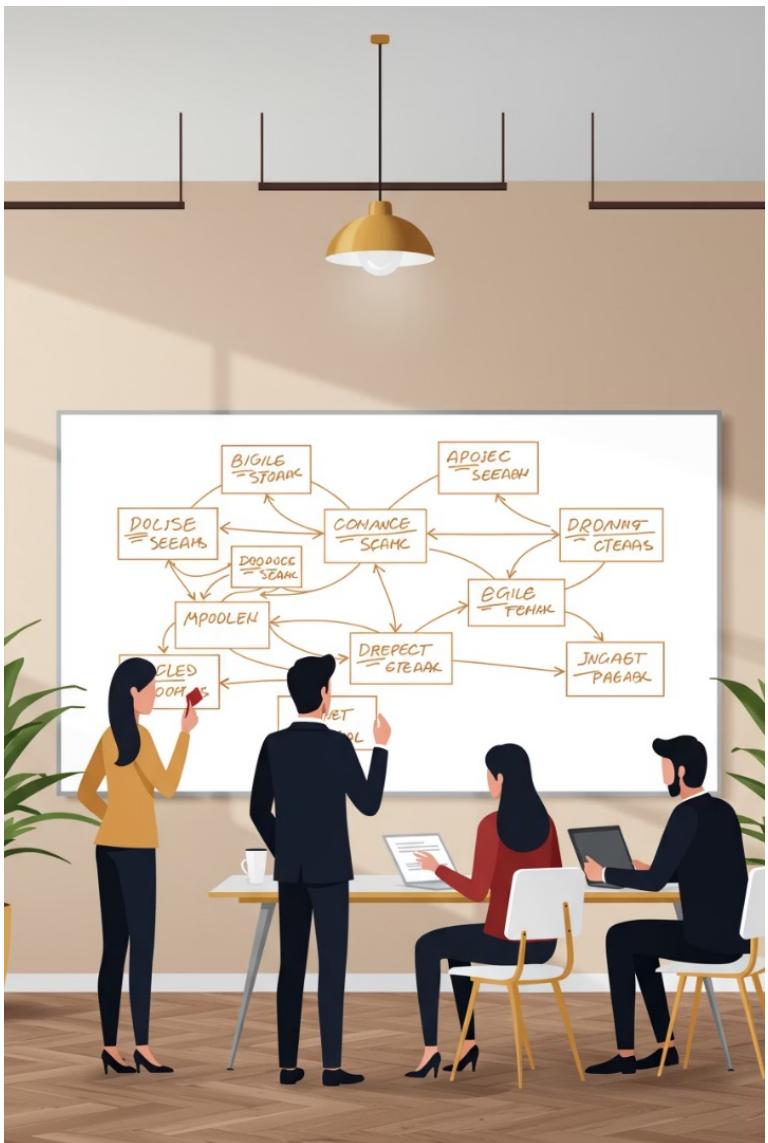
Write tests before code

Continuous Integration

Integrate and test changes frequently

SAFe (Scaled Agile Framework)





Scaling Agile: Challenges

Communication Barriers

Information silos form between teams. Dependencies become harder to manage. Coordination requires more effort.

Vision Dilution

Product vision gets lost across teams. Technical standards may vary. Integration issues increase.

Process Overhead

Governance can become bureaucratic. Agility decreases with scale. Alignment becomes more difficult.

SAFe Implementation Roadmap



Reach the Tipping Point

Build leadership case for change.

Train Change Agents

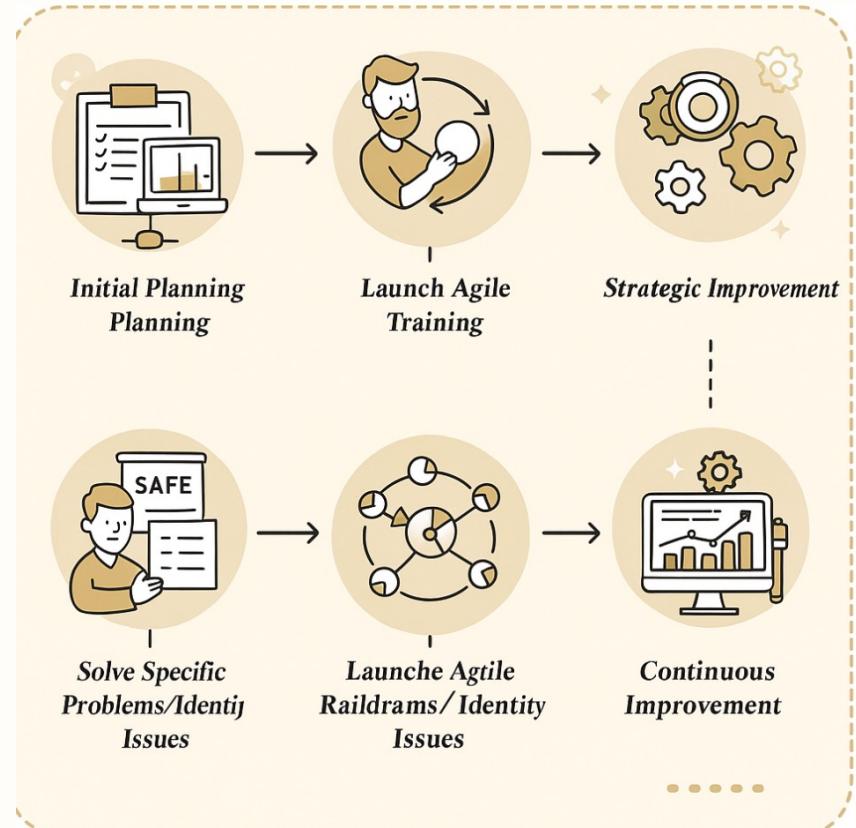
Prepare SAFe Program Consultants.

Create Implementation Plan

Design Agile Release Trains.

Launch ARTs

Begin delivery with first trains.



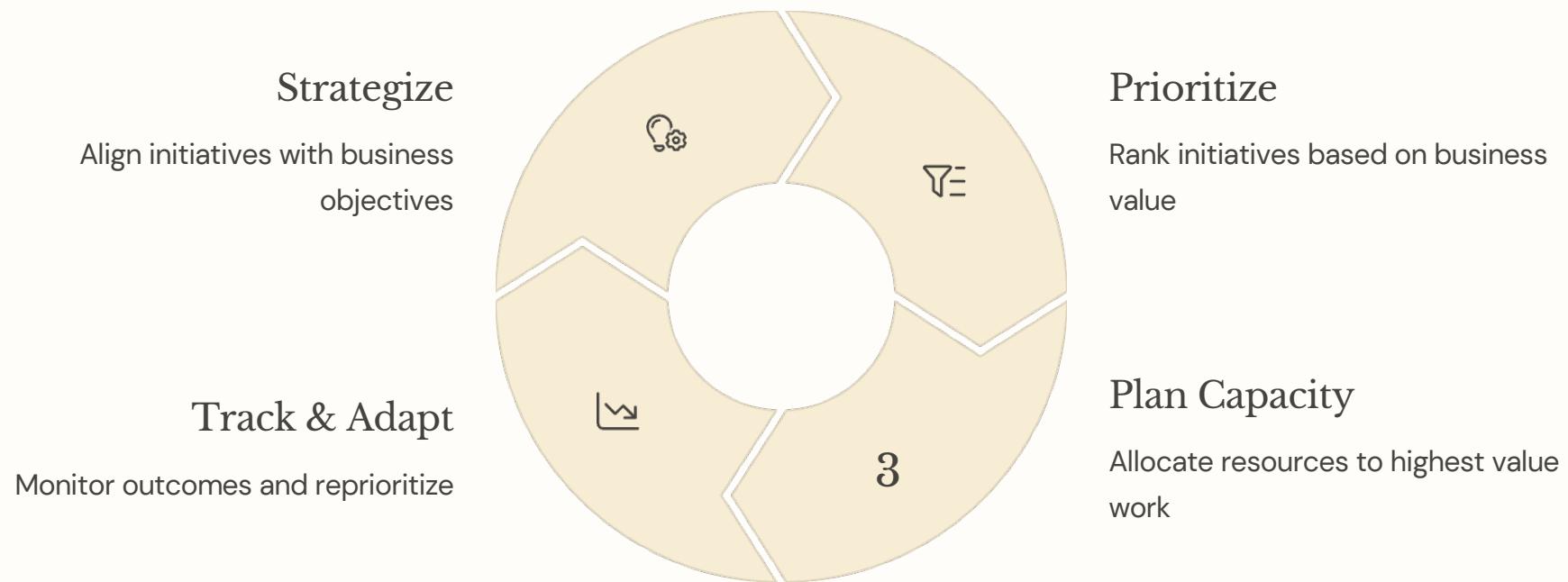
Enterprise Transformation

SAFe Core Competencies



SAFe defines seven core competencies essential for business agility. Each competency contains multiple dimensions.

Agile Portfolio Management



Lean Portfolio Management



Strategy & Investment Funding

Align investments with strategic goals



Agile Portfolio Operations

Coordinate value streams across organization

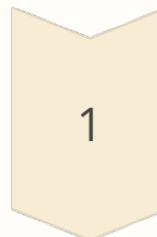
3

Lean Governance

Ensure compliance with minimal overhead



Value Stream Mapping



Customer Request

Initial input triggers the process. Value stream begins here.



Analysis & Design

Value-add: 2 days. Wait time: 5 days.



Development

Value-add: 5 days. Wait time: 3 days.



Testing & Delivery

Value-add: 3 days. Wait time: 7 days.



PI Planning in SAFe

8-12

50-125

Week Planning Horizon

Program Increment typically spans
8-12 weeks of coordinated work.

Participants

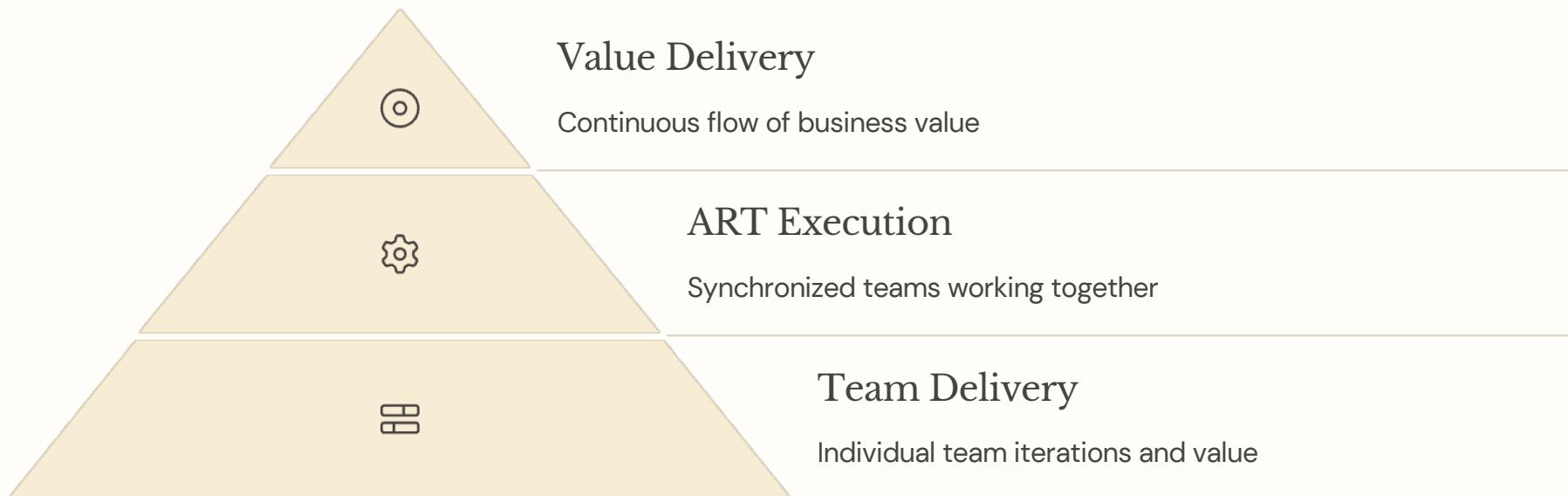
All teams in an Agile Release Train
attend the PI planning event.

2

Days

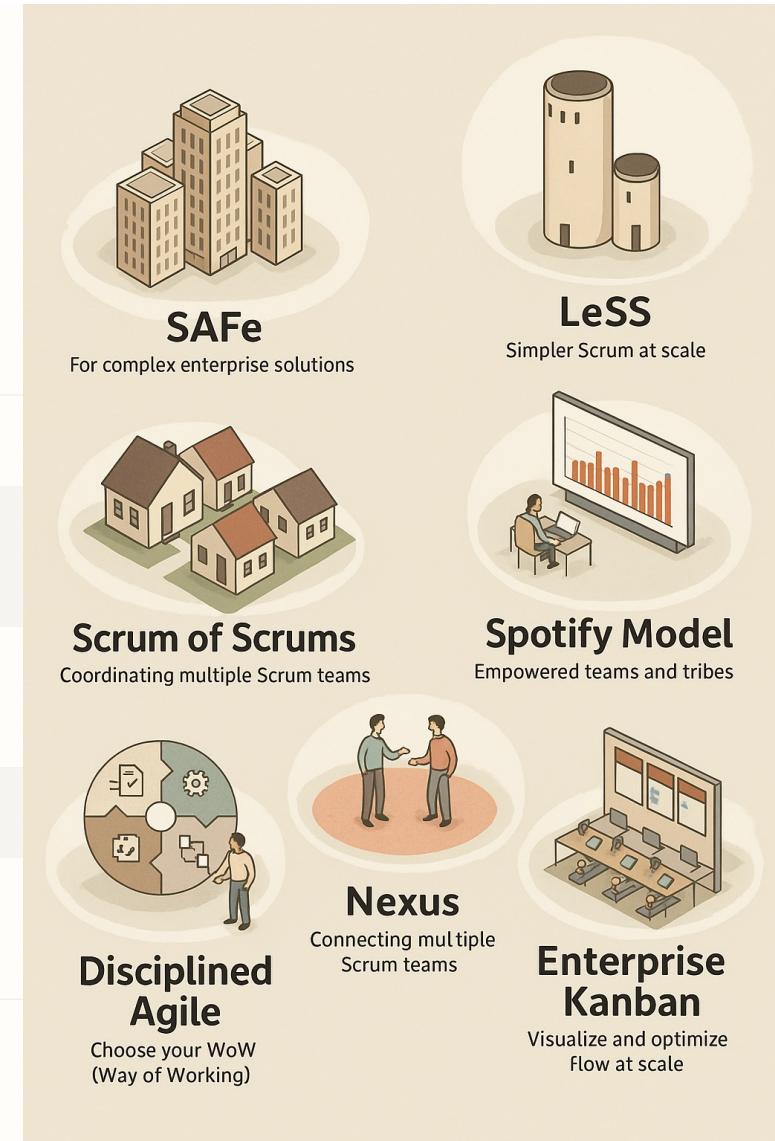
Standard PI planning duration.
Teams align on vision and
dependencies.

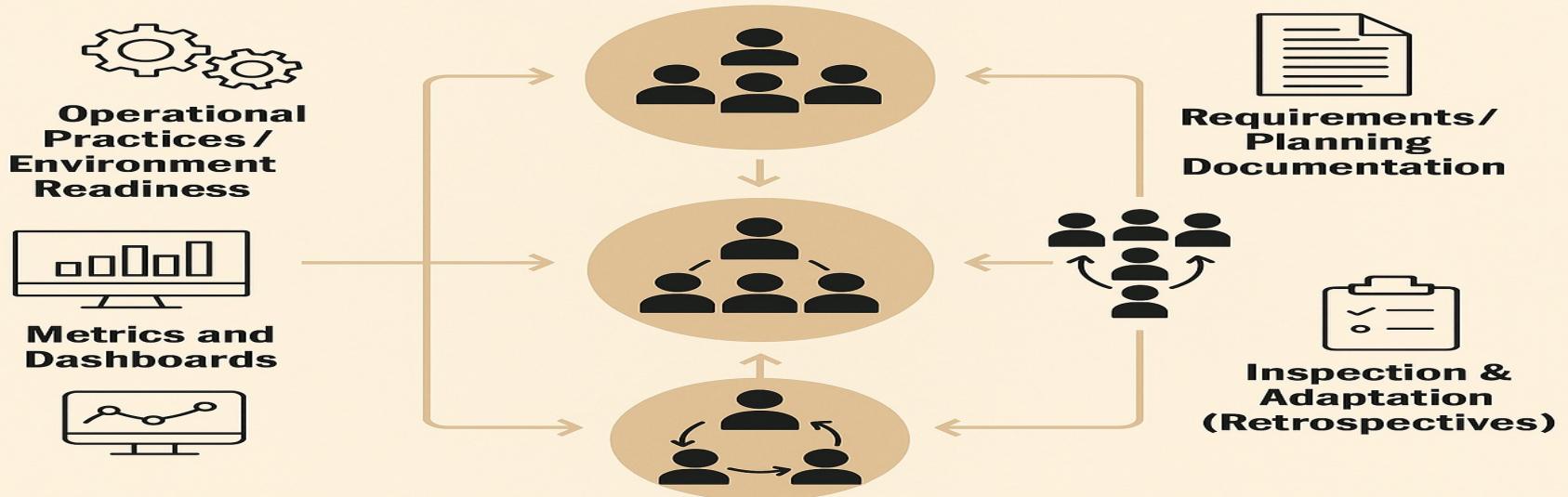
SAFe Program Level



Frameworks for Scaling Agile

Framework	Best For	Key Features
SAFe	Enterprises	Structured, comprehensive
LeSS	Minimalists	Few roles, lightweight
Nexus	Scrum users	Integration team
Spotify	Autonomy focus	Squads, tribes, chapters





LeSS (Large-Scale Scrum)



Minimalist Scaling

Apply Scrum principles at scale.
Avoid adding roles and processes.



One Product Backlog

All teams work from single prioritized backlog. One Product Owner manages it.



Synchronized Sprints

All teams work in same sprint cadence. Coordinate through Scrum of Scrums.

Nexus Framework



Nexus Daily Scrum

Representatives meet daily to discuss cross-team dependencies and integration issues.



Integration Team

Dedicated team focuses on cross-team dependencies and ensuring integration.



Nexus Sprint Planning

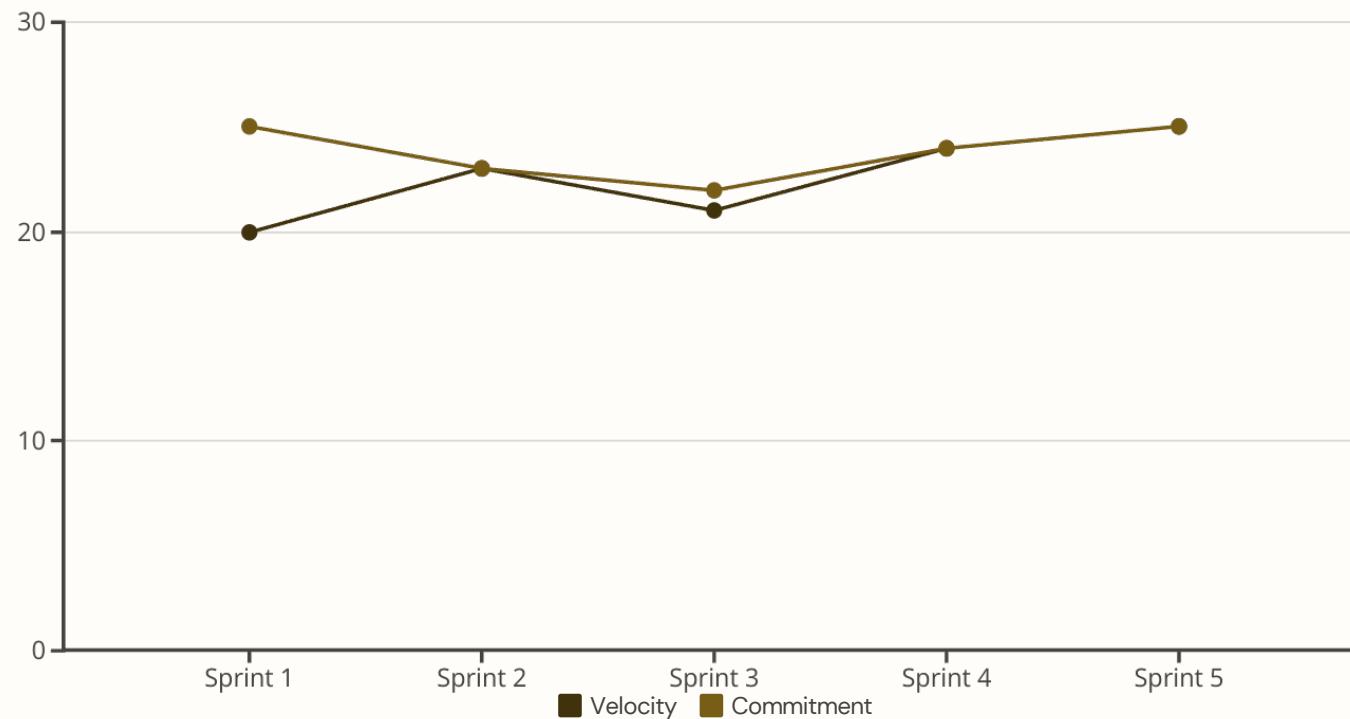
Identifies dependencies and sequences work across teams for optimal flow.

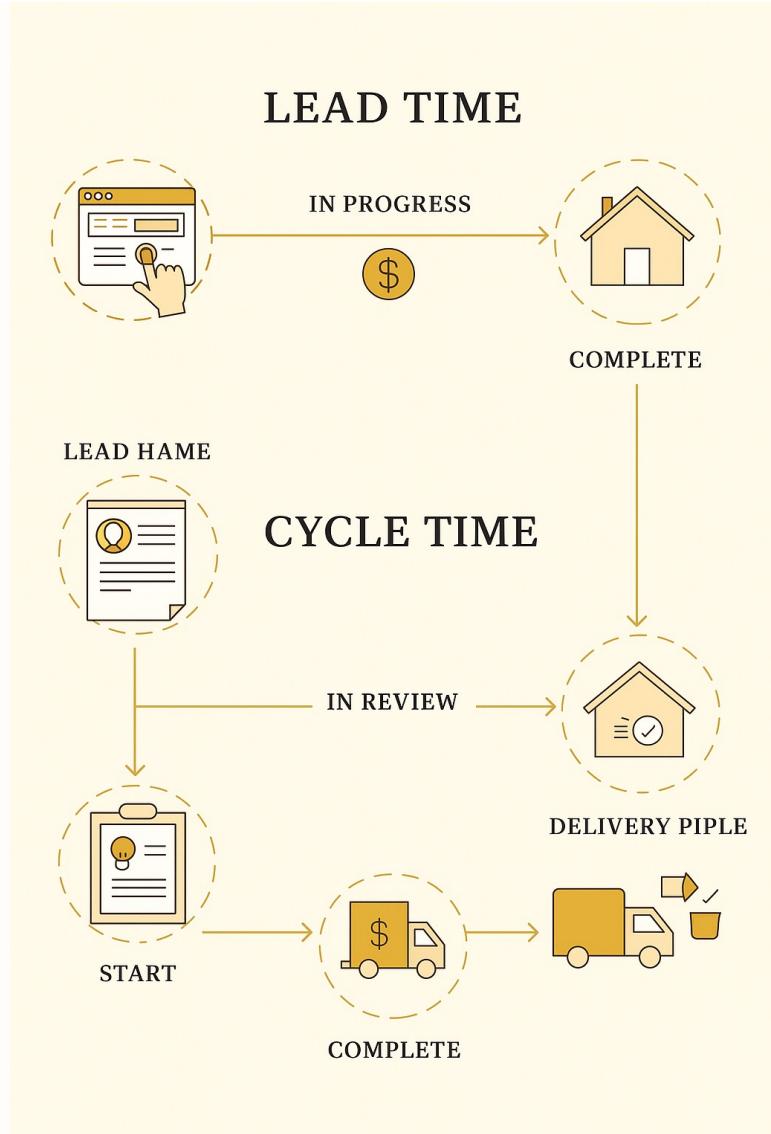


Spotify Model

Structure	Purpose	Leadership
Squad	Feature team	Product Owner
Tribe	Collection of squads	Tribe Lead
Chapter	Functional expertise	Chapter Lead
Guild	Community of interest	Coordinator

Velocity and Capacity Planning

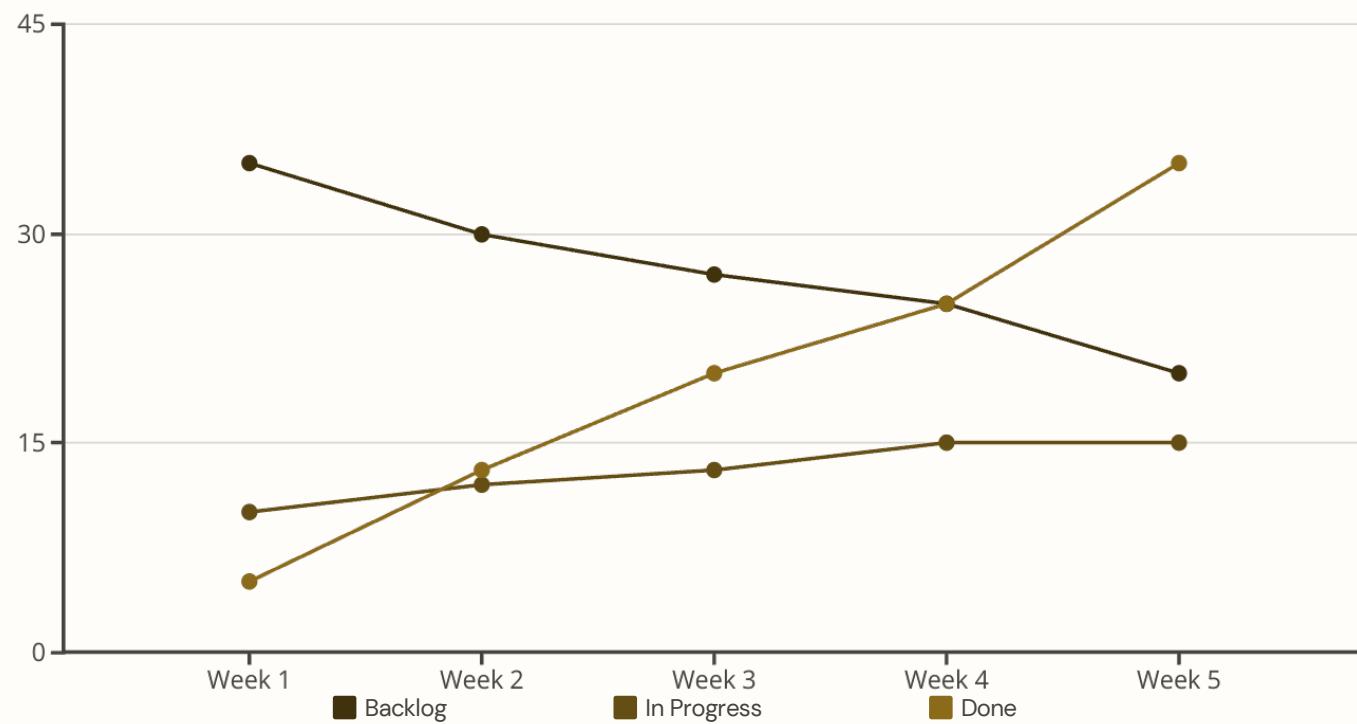




Lead Time vs. Cycle Time

- 1 Request Received
Customer submits feature request. Lead time clock starts.
- 2 Work Starts
Team begins development. Cycle time clock starts.
- 3 Work Complete
Feature finished. Cycle time ends. Lead time continues.
- 4 Delivery
Customer receives feature. Lead time ends.

Cumulative Flow Diagram



Burn-down vs. Burn-up Charts

Burn-down Chart

Shows remaining work over time. Starts high and decreases.
Good for sprint tracking.

- Focuses on work left
- Simple to understand
- Shows if behind schedule

Burn-up Chart

Shows completed work over time. Starts low and increases.
Good for release tracking.

- Shows scope changes
- Tracks total scope
- Better for longer timeframes

Agile Estimation Techniques

Planning Poker

Teams simultaneously reveal point estimates. Uses Fibonacci sequence. Drives discussion on differences.

T-Shirt Sizing

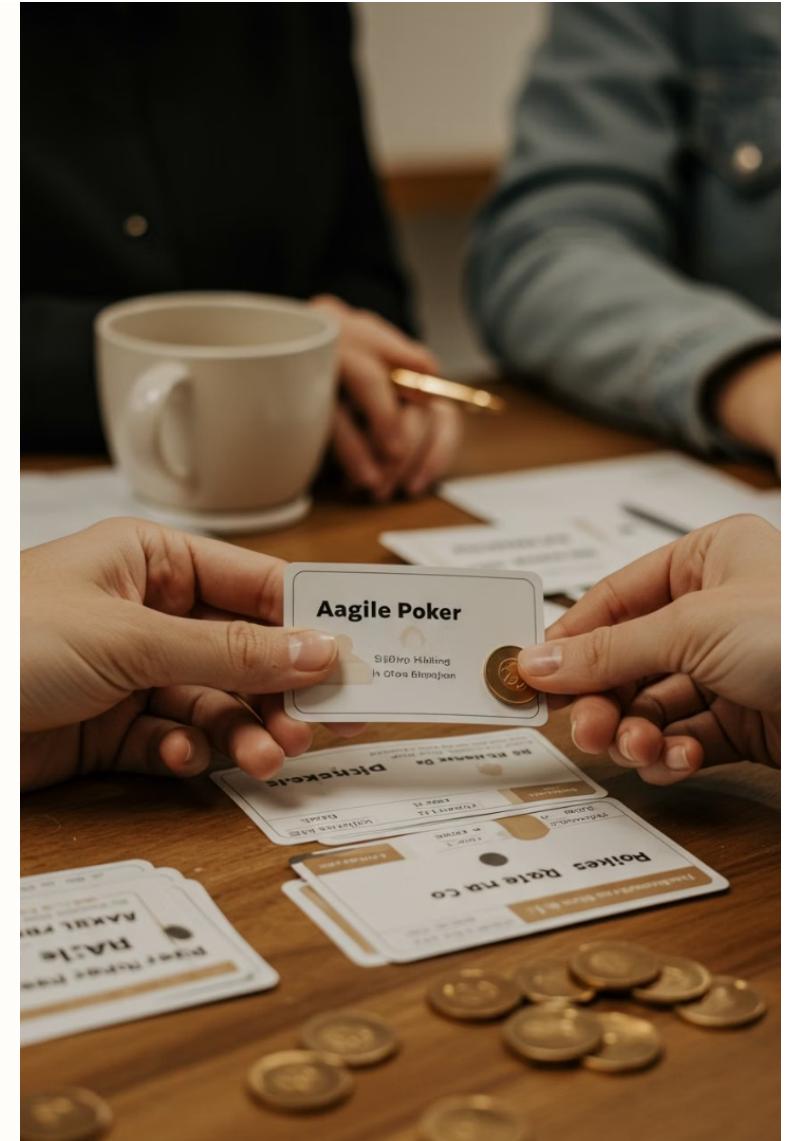
Simple XS to XL scale. Quick relative sizing. Good for initial rough estimates.

Dot Voting

Team members place dots on items. Visual prioritization technique. Shows consensus quickly.

Affinity Estimation

Group similar sized items together. Works well for large backlogs. Highly visual method.



Definition of Done (DoD)

Code Complete
Follows standards and patterns

Deployable
Ready for production

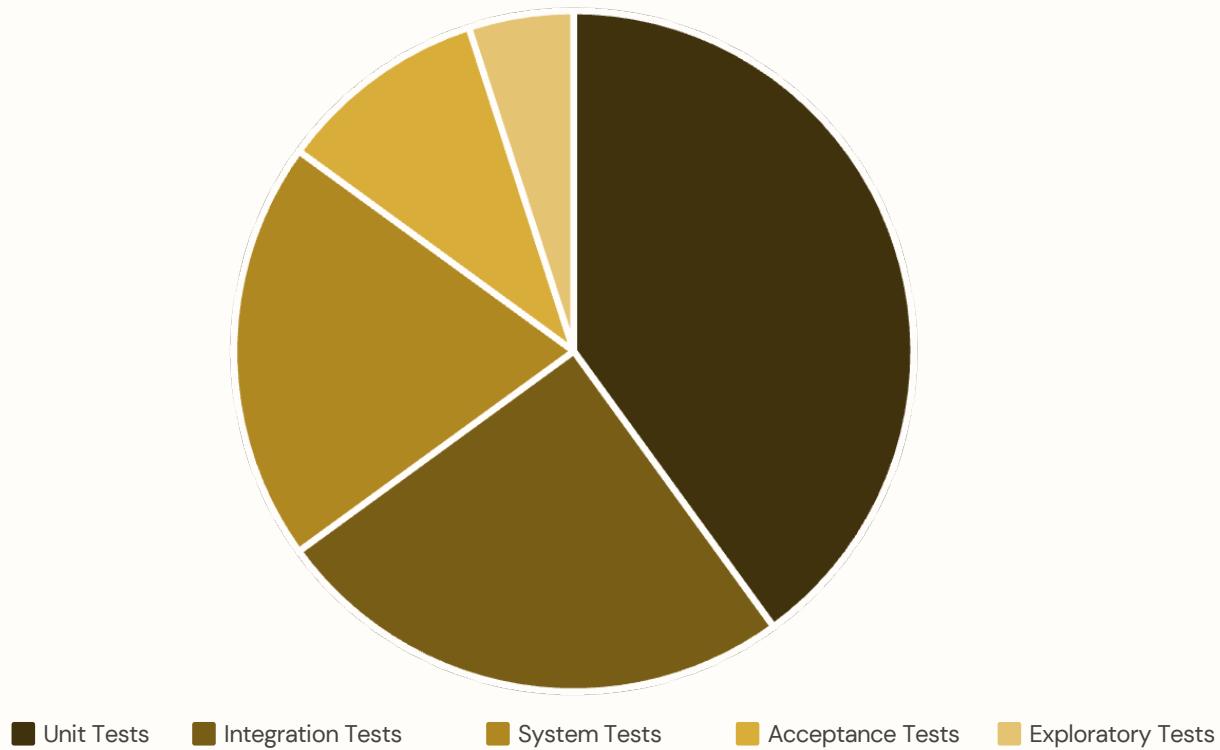


Tested
Unit, integration, and acceptance

Reviewed
Code review and QA complete

Documented
Technical and user documentation

Agile Testing Quadrants



Agile Metrics

⌚ Velocity

Measures story points completed per sprint. Shows team capacity and delivery rate.

〽 Burn Charts

Tracks remaining work (burn-down) or completed work (burn-up) over time.

⌚ Cycle Time

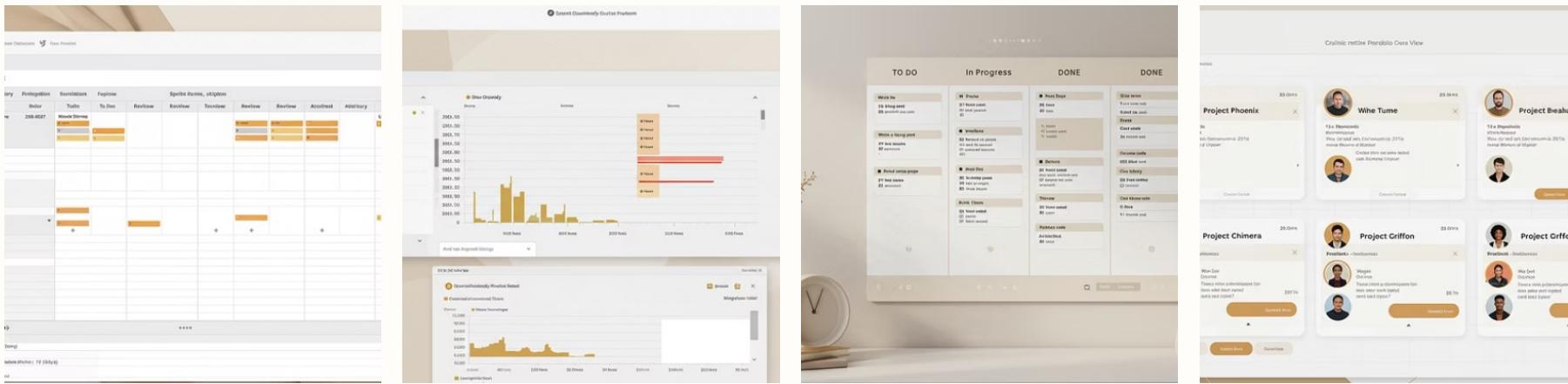
Measures time from work start to completion. Indicates process efficiency.

% Defect Rate

Tracks quality through bugs and issues. Lower is better.



Tracking Progress with Agile Tools



Modern agile tools provide real-time visibility into project status. They enable remote collaboration and automated reporting.

Agile Governance



Risk Management

Proactive identification and mitigation



Quality Assurance

Built-in testing and Definition of Done



Compliance Integration

Regulatory requirements embedded in backlog



Portfolio Management

Lean funding models and value stream alignment

Stakeholder Management in Agile

Identify
Map all stakeholders and their interests

Adapt
Respond to changing stakeholder needs



Engage
Involve early and continuously

Collaborate
Co-create solutions through feedback

Agile Culture and Mindset Shift

From

- Command and control leadership
- Functional silos
- Detailed upfront planning
- Process-focused metrics
- Risk avoidance

To

- Servant leadership
- Cross-functional teams
- Adaptive planning
- Value-focused metrics
- Calculated risk-taking

Organizational Transformation to Agile

Leadership Commitment

Secure executive sponsorship. Establish transformation vision. Create sense of urgency.

Pilot Implementation

Start with small teams. Generate quick wins. Demonstrate value.

Scaling Strategy

Train teams and leaders. Adapt structure gradually. Address organizational impediments.

Continuous Improvement

Gather feedback regularly. Make data-driven adjustments. Celebrate successes.



Conclusion & Key Takeaways



Adaptability is Key

Agile PM embraces change. It delivers value faster through iterative development.



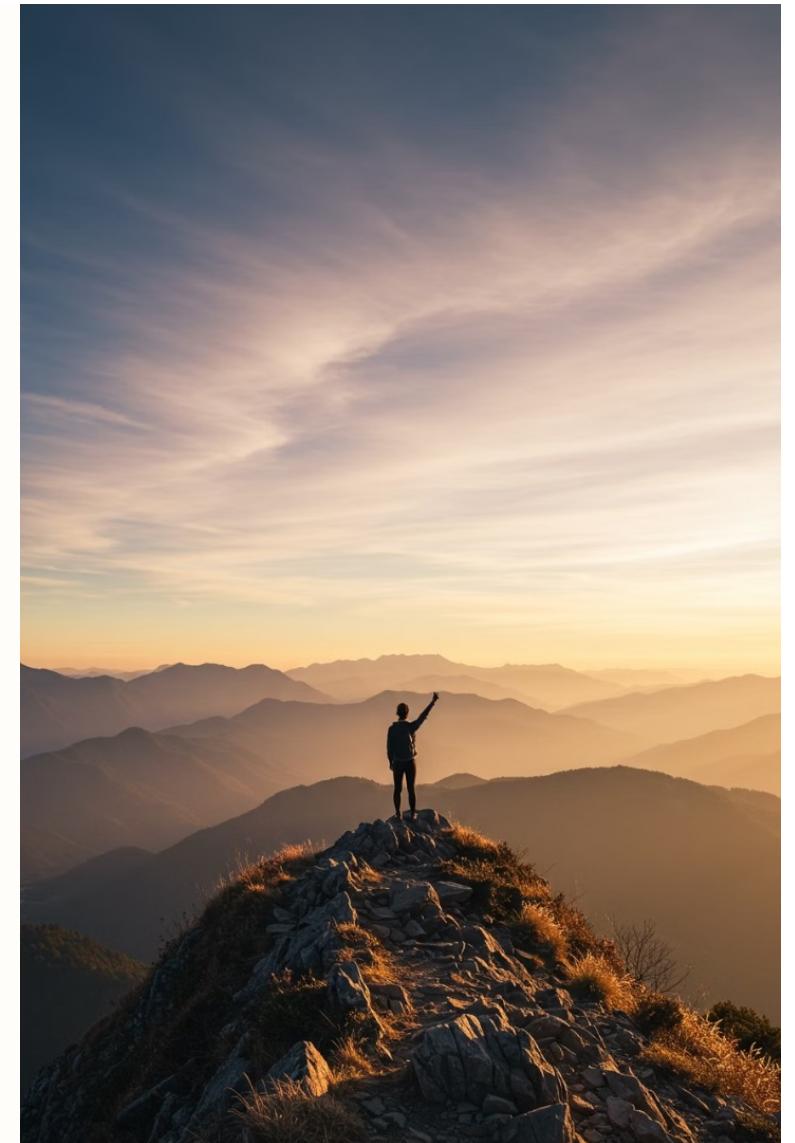
Framework Selection Matters

Scrum, Kanban, and SAFe offer unique benefits. Choose based on your context.

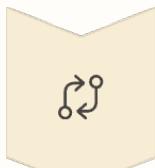


Scaling Requires Alignment

Culture, metrics, and governance must support agility. Leadership commitment is essential.



DevOps and Agile Integration



Continuous Integration

Automated build and test with every code change.



Continuous Testing

Automated testing throughout the pipeline.



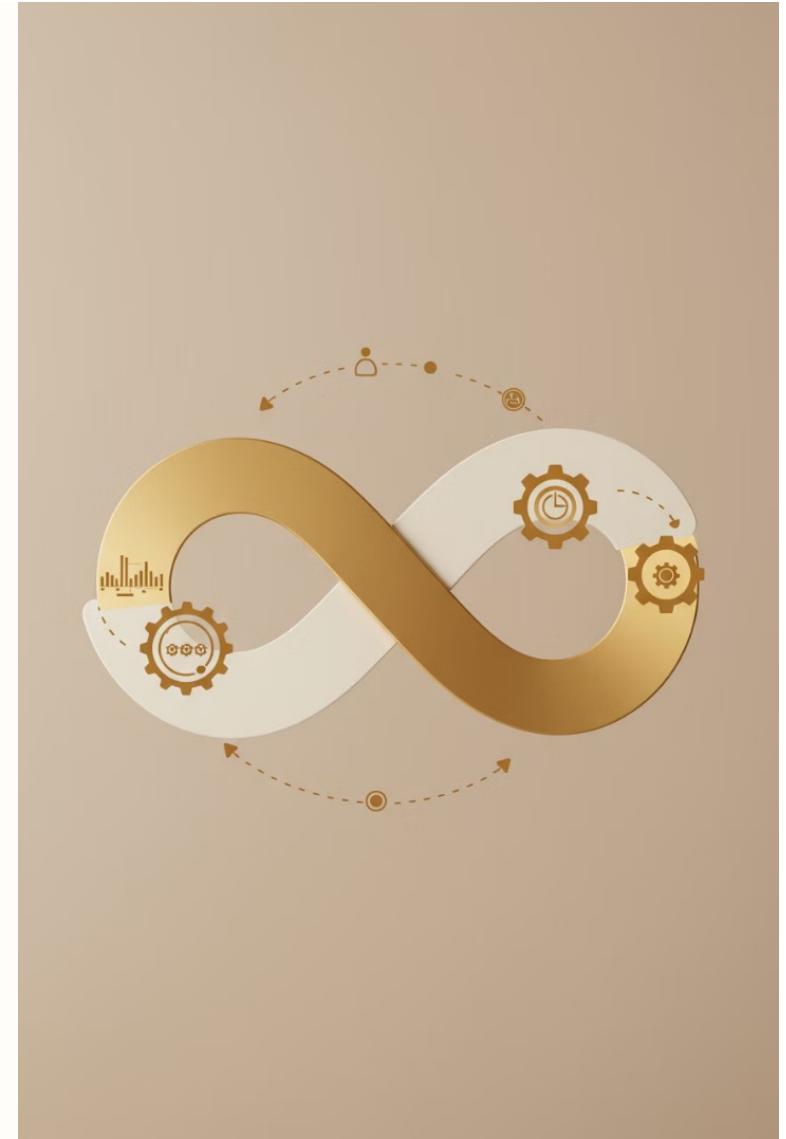
Continuous Delivery

Automated deployment to staging environments.

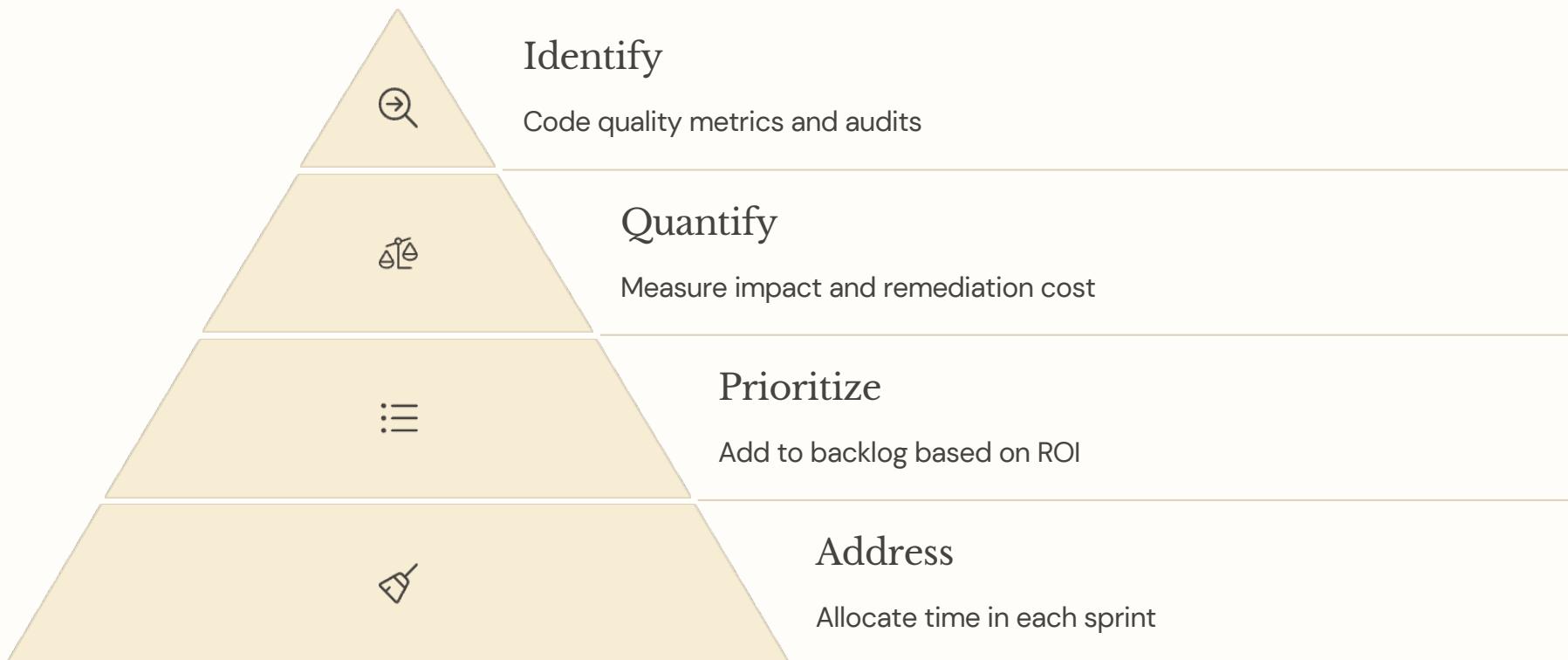


Continuous Deployment

Automated release to production.



Technical Debt Management





Distributed Agile Teams

Communication Tools

- Video conferencing
- Digital whiteboards
- Chat platforms
- Virtual task boards

Coordination Practices

- Overlap work hours
- Clear documentation
- Recorded meetings
- Follow-the-sun handoffs

Team Building

- Virtual social events
- Regular face-to-face
- Cultural awareness
- Team agreements

Agile Contracting Models

Fixed Price Per Sprint

Client pays for team capacity. Scope remains flexible. Suitable for uncertain requirements.

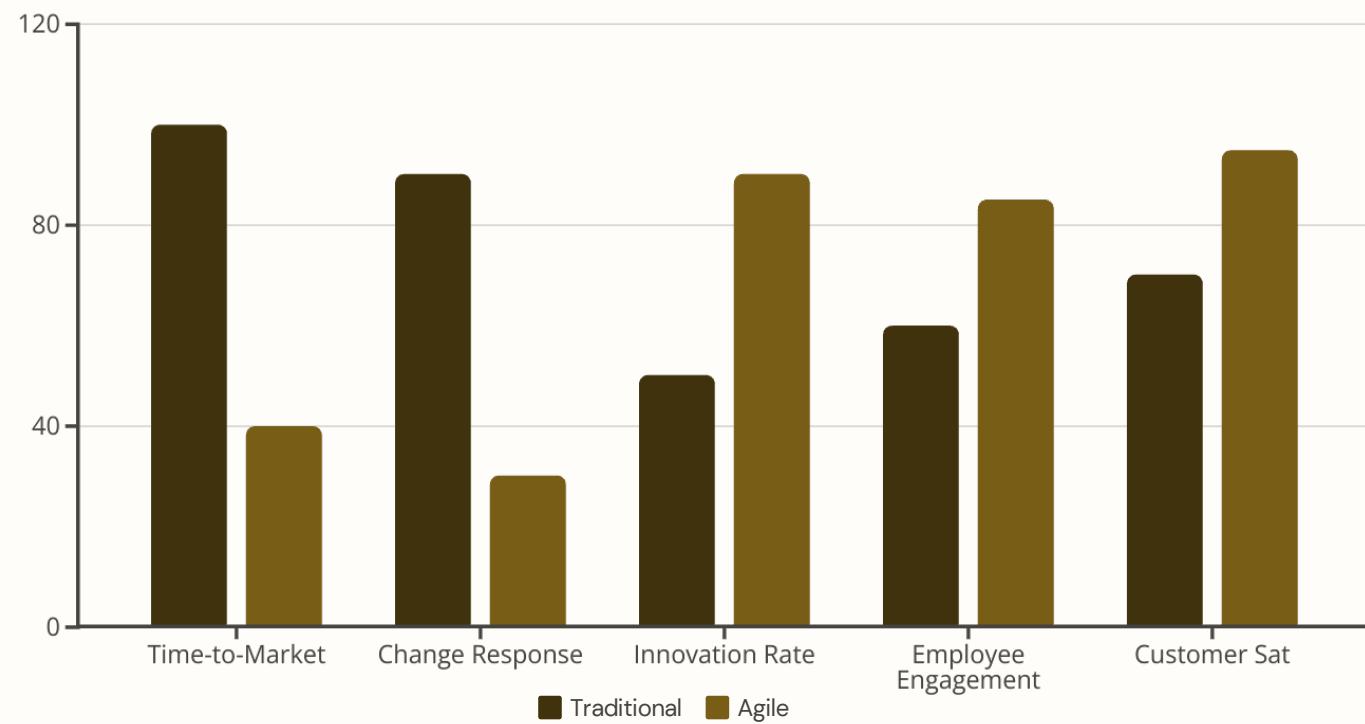
Time & Materials with Cap

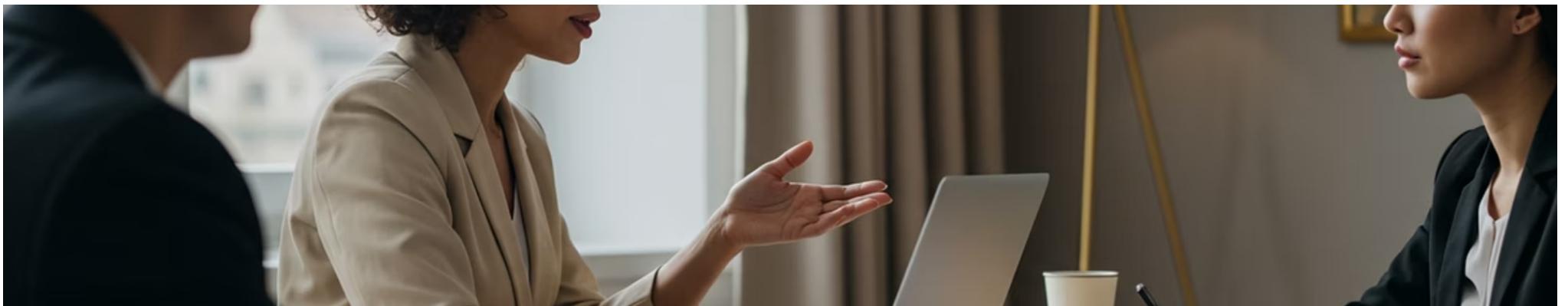
Pay for actual work with upper limit. Provides budget predictability with agility.

Target Cost

Shared risk/reward based on targets. Aligns vendor and client interests.

Measuring Business Agility





Agile Leadership Behaviors



Servant Leadership

Leaders remove obstacles. They enable teams to succeed. They coach rather than direct.



Creating Safety

Encourage experimentation. Welcome failure as learning. Build psychological safety.



Fostering Innovation

Promote creative thinking. Allocate time for exploration. Reward new ideas.



Agile HR Practices

Talent Acquisition

- Team-based hiring
- Cultural fit emphasis
- Cross-functional skills

Performance Management

- Frequent feedback
- Team-based goals
- 360° reviews

Learning & Development

- Self-directed learning
- T-shaped skills
- Continuous improvement

Shift from Projects to Products

Project Mindset

- Temporary teams
- Fixed scope
- Success = on time, on budget
- Hand-offs between groups
- Deadline-driven

Product Mindset

- Stable teams
- Evolving scope
- Success = customer value
- End-to-end ownership
- Value-driven

Building Agile Teams



Common Agile Adoption Mistakes



Process Over People

Following ceremonies but missing the mindset. Creates "cargo cult" agile.



Ignoring Technical Practices

Adopting ceremonies without engineering excellence. Creates mounting technical debt.



Imposing Standardization

Forcing identical practices on all teams. Prevents context-specific adaptation.

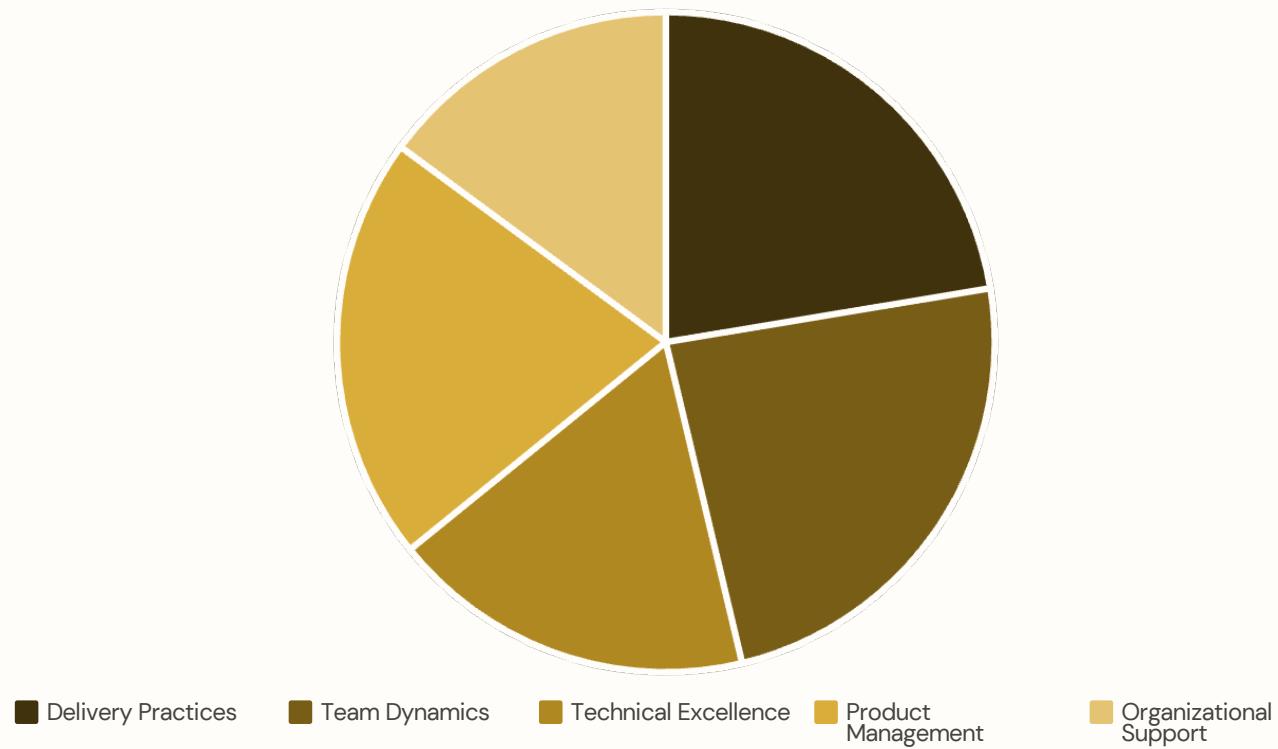


Waterfall Thinking

Sequential phases within sprints. Undermines iterative and incremental benefits.



Agile Maturity Assessment





Overcoming Resistance to Agile

Understand Root Causes

Identify specific fears and concerns. Listen to understand, not to respond. Empathize.

Demonstrate Value

Show early wins. Use metrics that matter. Share success stories from similar organizations.

Provide Support

Offer training and coaching. Create safe spaces. Allow experimentation and learning.

Involve Resistors

Make them part of the solution. Leverage their expertise. Address legitimate concerns.

Agile in Regulated Environments



Traceability

Link requirements to tests and code.
Maintain automated audit trails. Ensure documentation without burdening teams.



Risk Management

Apply risk-based approach. Front-load compliance for high-risk areas.
Integrate into Definition of Done.



Automated Compliance

Build compliance into CI/CD pipeline.
Automate documentation generation.
Implement continuous validation.

Agile Budgeting Approaches



Fund Teams, Not Projects

Allocate stable funding to persistent teams. Measure outcomes, not outputs.



Dynamic Portfolio Management

Review allocations quarterly. Adjust based on market feedback. Kill underperforming initiatives.

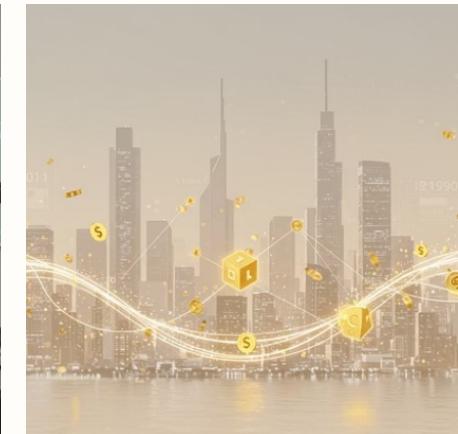
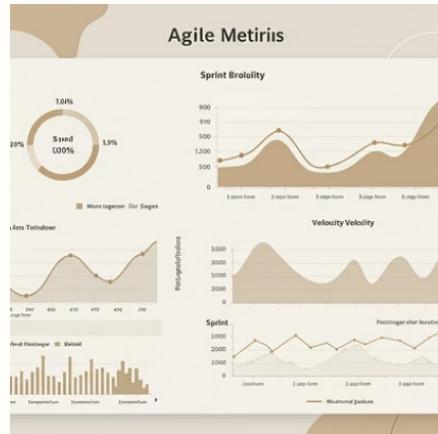


Beyond Cost to Value

Assess strategic alignment. Measure customer impact. Consider total cost of ownership.



Future of Agile



Agile continues evolving with technology advancements. AI will enhance estimation accuracy. Automation will streamline ceremonies.



Your Agile Journey: Next Steps

Assess Current State

Conduct maturity assessment.
Identify strengths and gaps.
Survey team satisfaction.

Develop Transformation Strategy

Choose appropriate frameworks. Set measurable goals. Secure leadership support.

Implement and Learn

Start with pilot teams. Gather feedback continuously. Adjust approach as needed.

Scale and Evolve

Expand successful practices.
Build internal coaching capability. Maintain continuous improvement.

Recommended Agile Textbooks

Martin, R. C, Agile Software Development, Principles, Patterns, and Practices.

Chapters 1–4: Agile values and principles

Chapters 7–9: Technical practices and code quality

Chapters 20–21: Design patterns in agile environments

Cohn, M. Agile Estimating and Planning.

Chapters 3–5: User stories and estimation techniques

Chapters 7–8: Velocity and capacity planning

Chapters 17–19: Portfolio management and budgeting

Schwaber, K., & Beedle, M. Agile Software Development with Scrum.

Chapters 2–4: Scrum roles and ceremonies

Chapters 5–6: Product backlog management

Chapters 9–10: Scaling Scrum for larger organizations

Resources for Agile Learning

1. **Essential Books:** "Agile Practice Guide" (PMI), "Scrum Guide" (Schwaber & Sutherland), "SAFe 5.0 Distilled" (Leffingwell)
2. **Online Learning:** Scrum.org, Scaled Agile Framework (scaledagileframework.com), Mountain Goat Software, Atlassian Agile Coach
3. **Communities:** Agile Alliance (agilealliance.org), Scrum Alliance, LinkedIn Agile groups
4. **Tools Documentation:** Jira, Azure DevOps, VersionOne, and Monday.com knowledge bases
5. **Advanced Research:** IEEE conferences, Journal of Systems and Software, Harvard Business Review case studies