USER RESEARCH METHODOLOGY

1

Heatmap Analysis

See what grabs attention—and what doesn’t.This helps you optimize layouts, calls-to-action, and content placement based on real engagement patterns.

2

Session Recording

Watch real visitor journeys in action.Understand how users navigate, where they get stuck.These recordings uncover friction points analytics alone can’t catch.

3

Funnel & Analytics Review (GA4)

Find where and why users drop off. Using GA4, I analyze your funnel performance step by step, pinpointing high-impact drop-offs

On-Site Surveys & Polls

4

Hear directly from your visitors.Capture in-the-moment feedback about user intentions, objections, and experiences..

User Testing

5

See how real people interact with your site.We observe real users completing tasks and narrating their thoughts. .

Insights Playbook

6

Your research, prioritised. All findings are distilled into a clear, actionable report with prioritized recommendations.,

Agile vs Traditional Development

Key differences between agile and waterfall development approaches

WATERFALL

Sequential Phases

Complete each development stage before moving to next phase

Detailed Planning

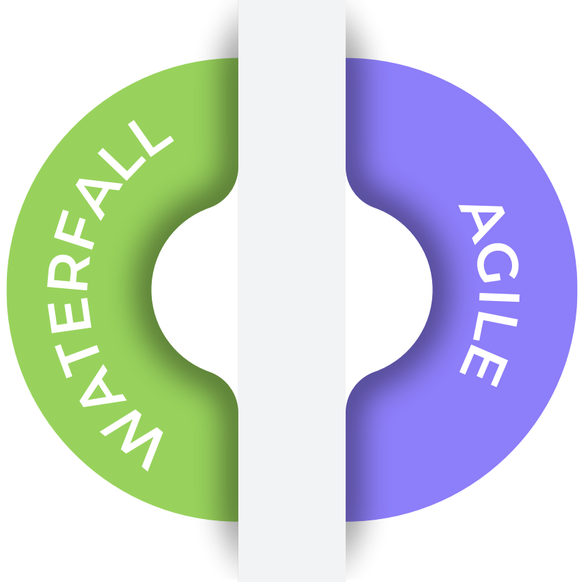
Extensive upfront planning with fixed requirements and scope

Limited Flexibility

Difficult to accommodate changes once development begins

Client Review at End

Customer sees product only after complete development



AGILE

Iterative Cycles

Parallel development activities within short sprint cycles

Adaptive Planning

Continuous planning adjustments based on feedback and learning

High Flexibility

Easy accommodation of changing requirements throughout development

Regular Client Feedback

Frequent customer reviews and input during development process

Scrum Framework Overview

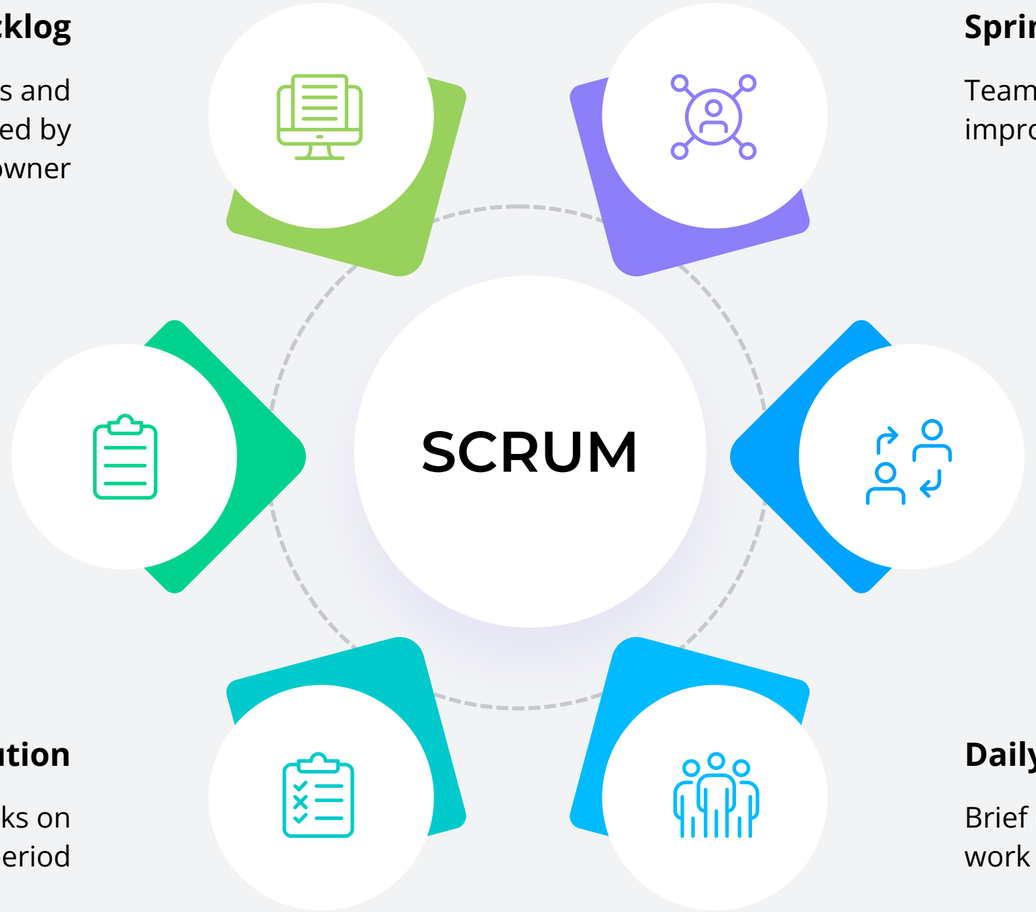
Complete scrum process from planning to delivery and retrospective

Product Backlog

Prioritized list of features and requirements maintained by product owner

Sprint Retrospective

Team reflection on process improvements for next sprint



Sprint Planning

Team selects backlog items and commits to sprint deliverables

Sprint Review

Demonstration of completed work to stakeholders for feedback

Sprint Execution

Development team works on selected items during sprint period

Daily Standups

Brief daily meetings to coordinate work and identify blockers

Sprint Planning Process

Detailed workflow for organizing and preparing successful sprint cycles

Backlog Refinement

Review and prioritize product backlog items before planning meeting

Capacity Planning

Assess team availability and determine realistic sprint workload

Story Selection

Choose highest priority items that fit within sprint capacity

Task Breakdown

Divide selected stories into specific development tasks and estimates

Commitment Setting

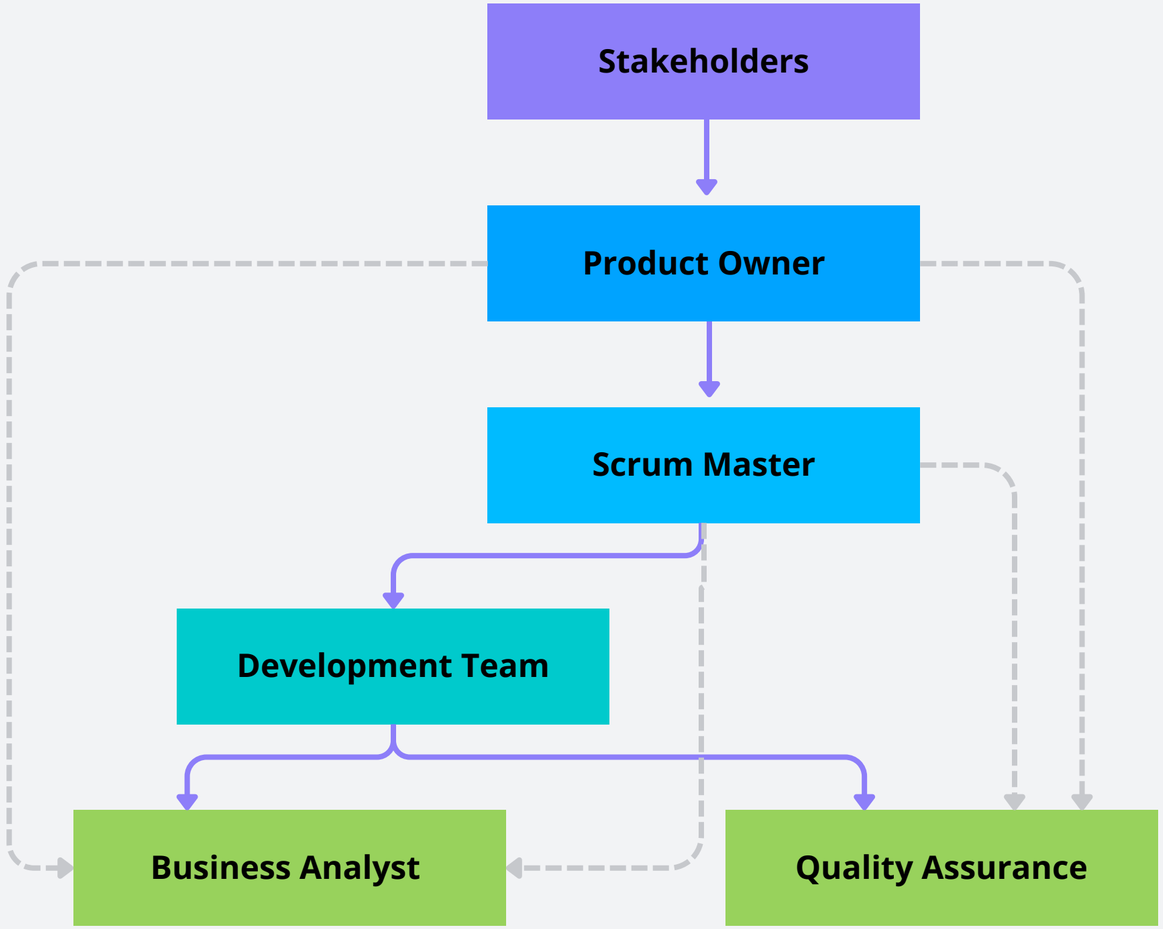
Team agrees on sprint goal and deliverable commitments

Sprint Kickoff

Begin development work with clear objectives and shared understanding

Agile Team Roles

Key responsibilities and interactions within agile development teams



Business users and customers who provide feedback and requirements

Defines requirements, prioritizes backlog and represents customer interests effectively

Facilitates meetings, removes blockers and ensures adherence to processes

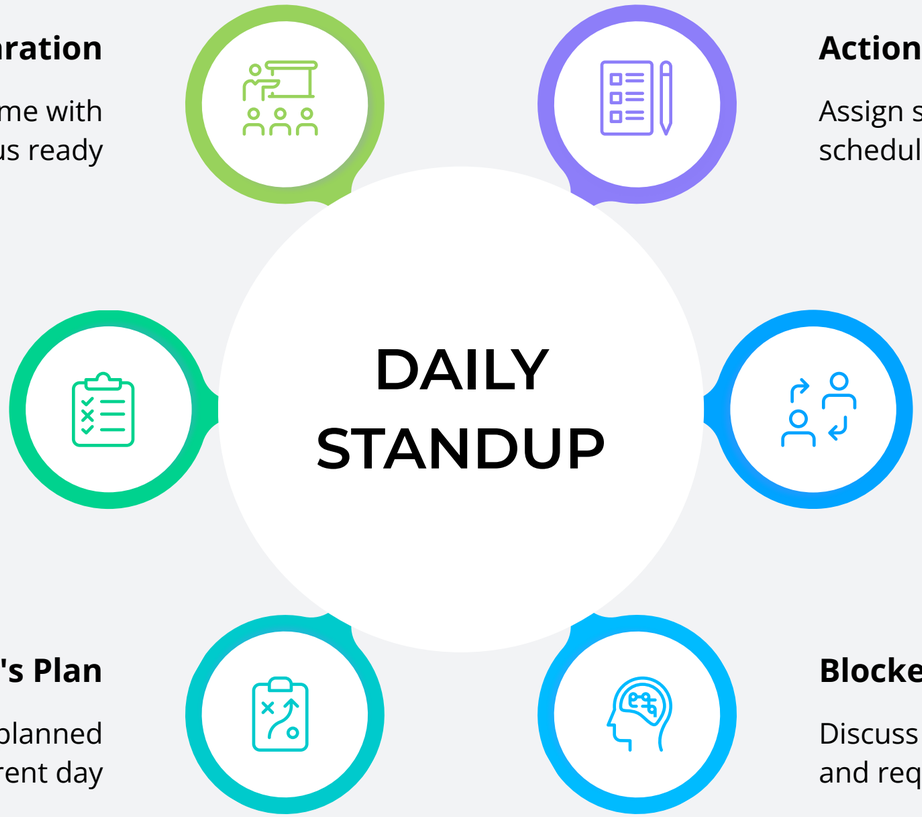
Cross functional group responsible for designing, coding and testing software

Bridges communication between technical team and business requirements

Ensures software meets quality standards through testing and validation

Daily Standup Structure

Efficient format for conducting productive daily team coordination meetings



Meeting Preparation

Team gathers at designated time with current work status ready

Action Items

Assign specific follow up tasks and schedule additional meetings

Yesterday's Progress

Each member shares completed work and accomplishments from previous day

Coordination Discussion

Address dependencies and collaboration needs between team members

Today's Plan

Team members communicate planned activities and goals for current day

Blocker Identification

Discuss obstacles preventing progress and request assistance needed

User Story Development

Structured approach to creating clear and actionable development requirements

Story Writing

Create user stories following standard format with clear acceptance criteria

Story Splitting

Break large stories into smaller manageable development tasks

Story Estimation

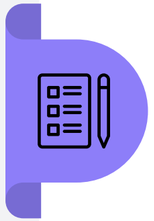
Assign complexity points using planning poker or similar techniques

Acceptance Criteria

Define specific conditions that must be met for completion

Story Prioritization

Rank stories based on business value and technical dependencies

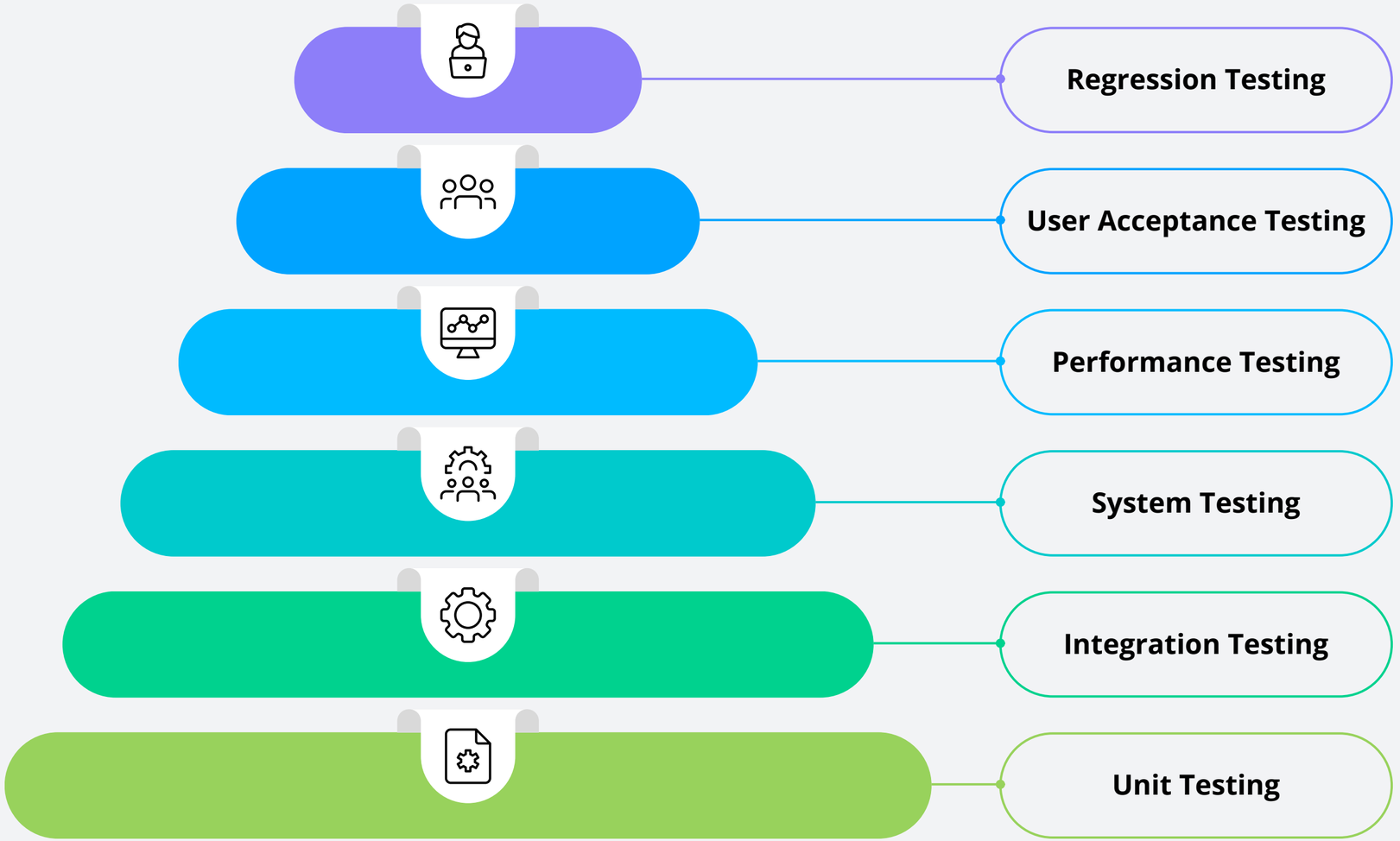


Story Validation

Confirm stories meet requirements before adding to sprint backlog

Testing and Quality Assurance

Comprehensive testing strategy ensuring high quality software delivery



Regression Testing

Ensure new changes do not break existing functionality

User Acceptance Testing

Validate software meets business requirements and user expectations

Performance Testing

Assess system speed, responsiveness and stability under various loads

System Testing

Complete application testing to ensure all features function properly

Integration Testing

Verify interactions between different system components work correctly

Unit Testing

Automated tests for individual code components and functions

Sprint Review and Demo

Structured approach to showcasing completed work and gathering stakeholder feedback

Demo Preparation

Set up working environment and prepare demonstration scenarios

Feature Showcase

Present completed user stories and new functionality to stakeholders

Feedback Collection

Gather input on demonstrated features and overall product direction

Requirement Updates

Document new requirements and changes based on stakeholder input

Next Sprint Planning

Discuss priorities and objectives for upcoming development cycle

Action Items

Record follow up tasks and schedule additional meetings needed

CONTINUOUS IMPROVEMENT PROCESS

Performance Metrics

Track velocity, quality indicators and team satisfaction regularly

Retrospective Meetings

Conduct regular team discussions about process effectiveness

Improvement Identification

Analyze data and feedback to identify enhancement opportunities

Results Monitoring

Measure impact of changes and adjust approach accordingly

Training and Development

Provide skill building opportunities for team members

Process Adjustments

Implement changes to workflows, tools and team practices