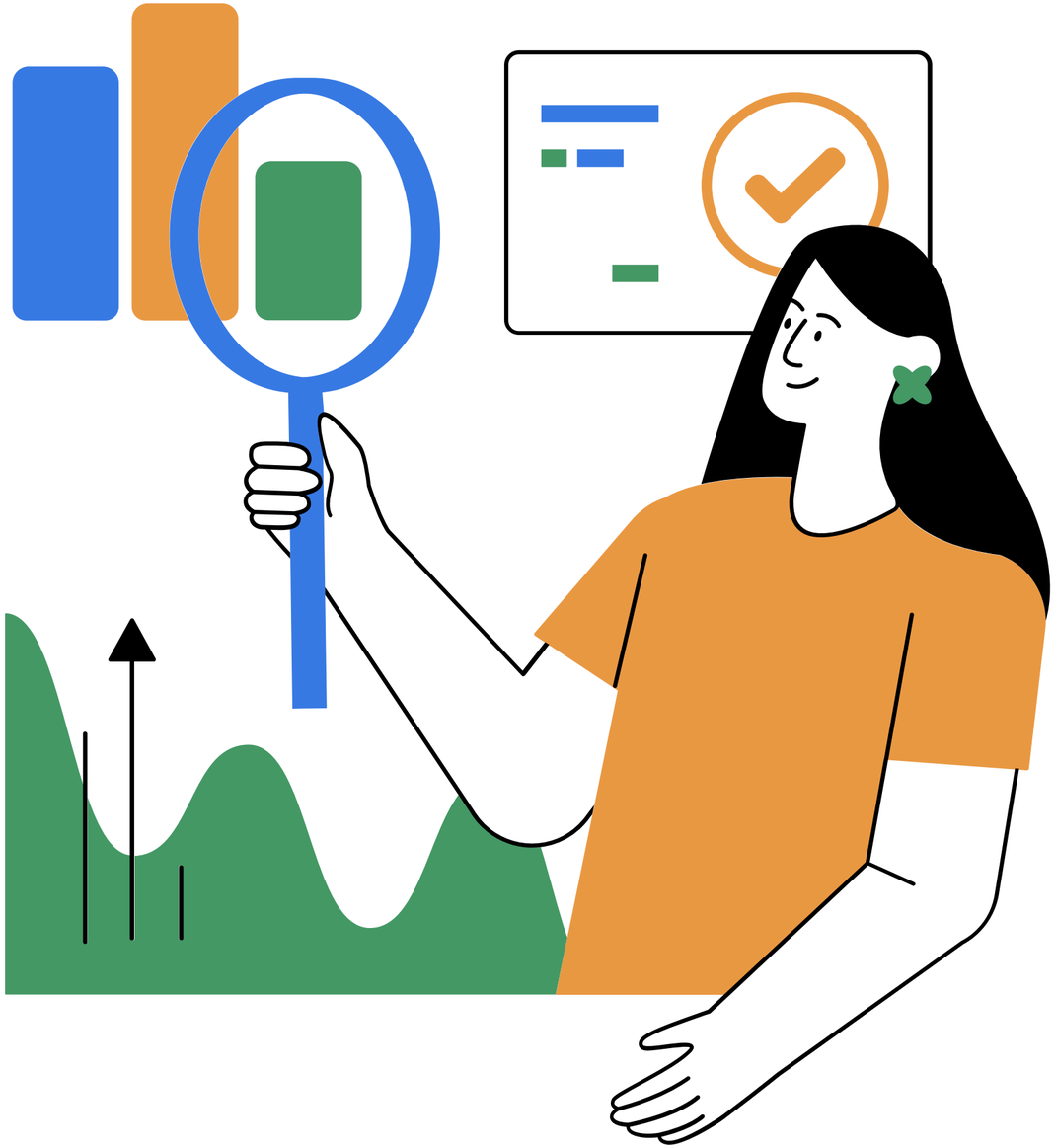
SCRUM Methodology

JSON & CSV: Data Formats Analysis



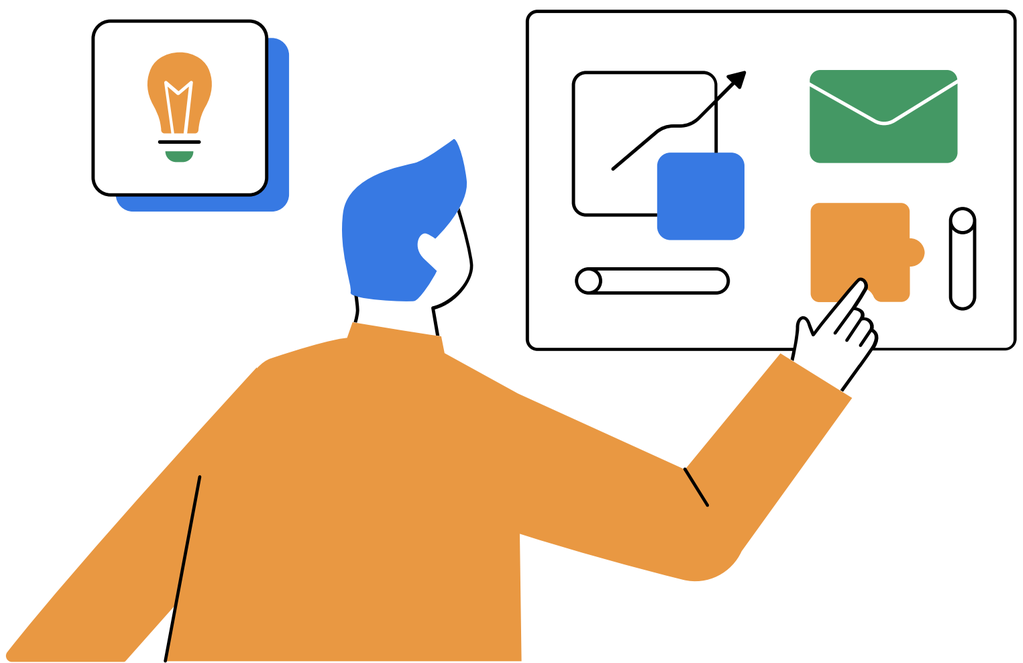
What is SCRUM?

SCRUM is an agile framework for managing complex projects, particularly software development. It emphasizes collaboration, transparency, and continuous adaptation through iterative progress.

SCRUM Pillars

- Transparency: All process aspects are visible to stakeholders
- Inspection: Continuous monitoring of progress and quality
- Adaptation: Regular adjustments based on feedback and learning

SCRUM Roles & Responsibilities



Product Owner

- Defines product vision and roadmap
- Manages Product Backlog prioritization
- Maximizes product value delivery
- Stakeholder communication interface

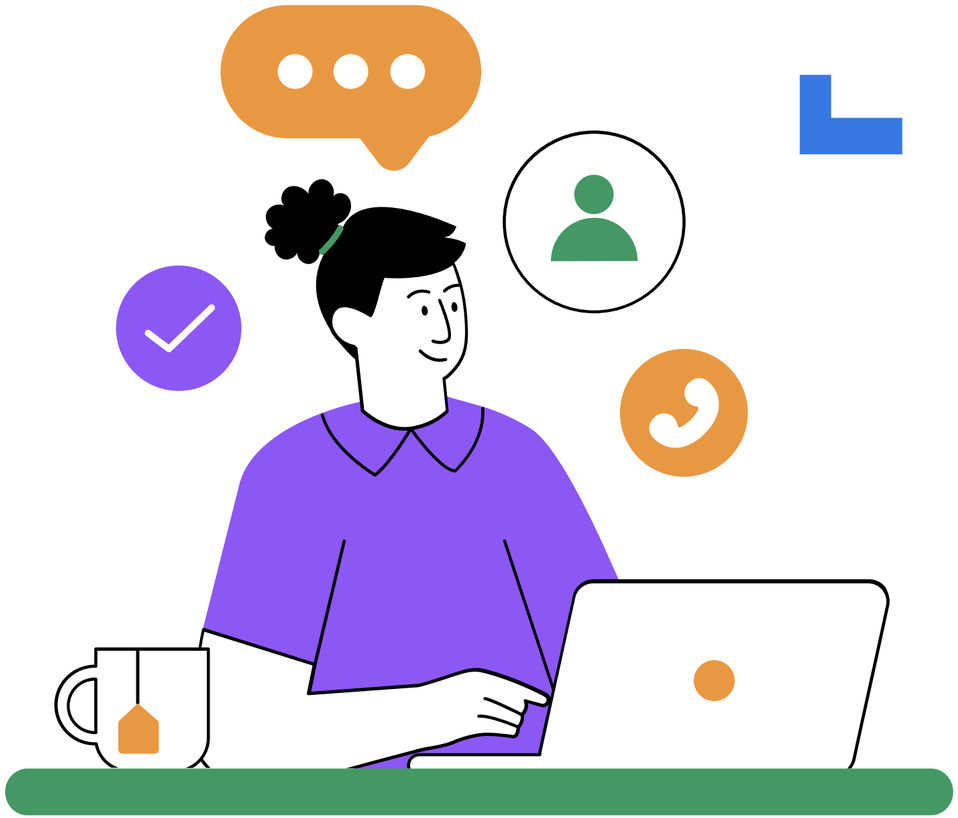
Scrum Master

- Facilitates SCRUM process implementation
- Removes organizational impediments
- Coaches team on agile practices
- Promotes continuous improvement

Development Team

- Self-organizing, cross-functional team
- 3-9 members for optimal communication
- Delivers potentially shippable increments
- Collective ownership of deliverables

SCRUM Artifacts



- **Product Backlog**: Ordered list containing all known requirements and features
- **Sprint Backlog:** Selected Product Backlog items for the current Sprint plus plan
- **Increment:** Sum of all completed Product Backlog items, ready for release

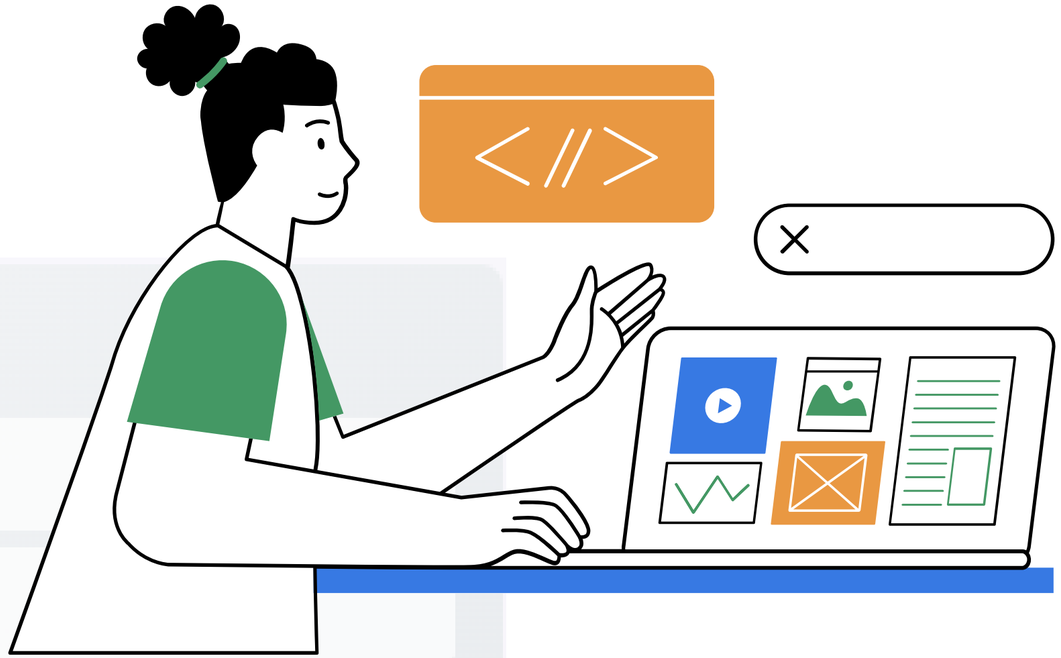
Key Metrics

Velocity: Average story points completed per Sprint

Burndown: Remaining work visualization over time

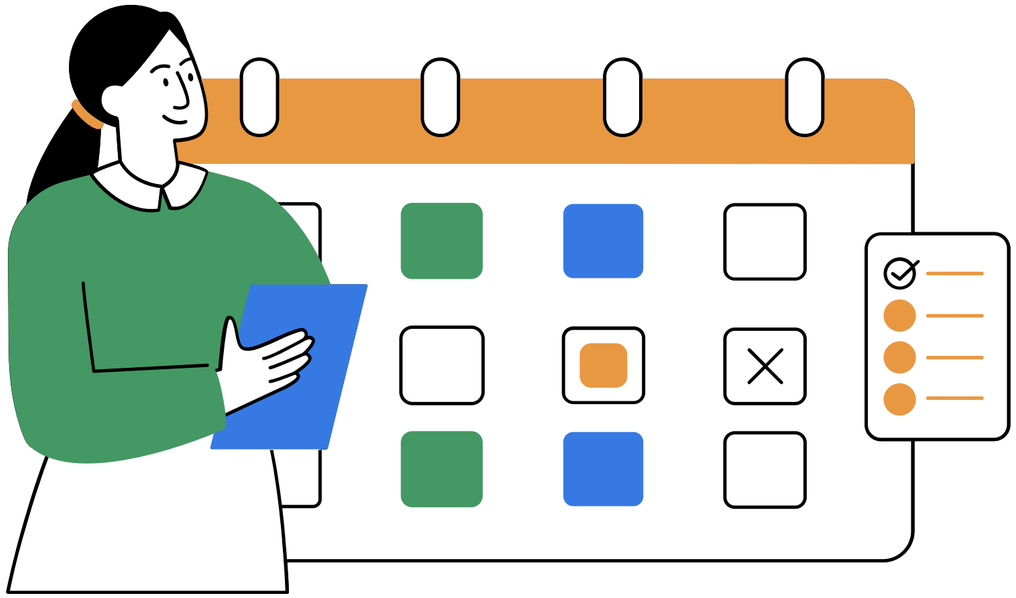
Cycle Time: Time from task start to completion

◀ Previous Next ▶



Sprint Characteristics

SCRUM Events & Ceremonies



Sprint Planning

Max 8 hours
Team plans Sprint work and creates Sprint Backlog

Daily Scrum

15 minutes daily
Team synchronizes activities and plans next 24 hours

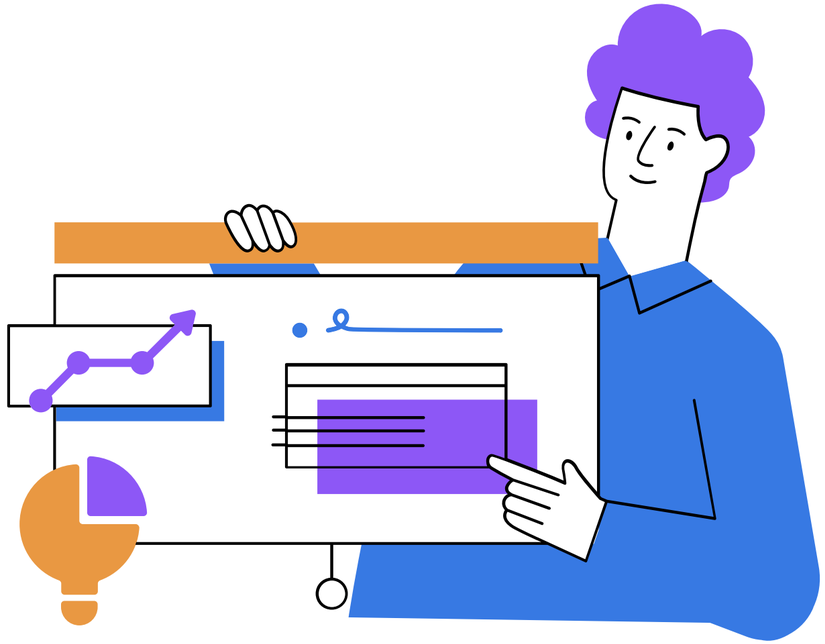
Sprint Review

Max 4 hours
Inspect increment and adapt Product Backlog

Sprint Retrospective

Max 3 hours
Team reflects on process improvement opportunities

JSON - JavaScript Object Notation



JSON Definition & Purpose

JSON (JavaScript Object Notation) is a lightweight, language-independent data interchange format. It's easy for humans to read and write, and easy for machines to parse and generate.

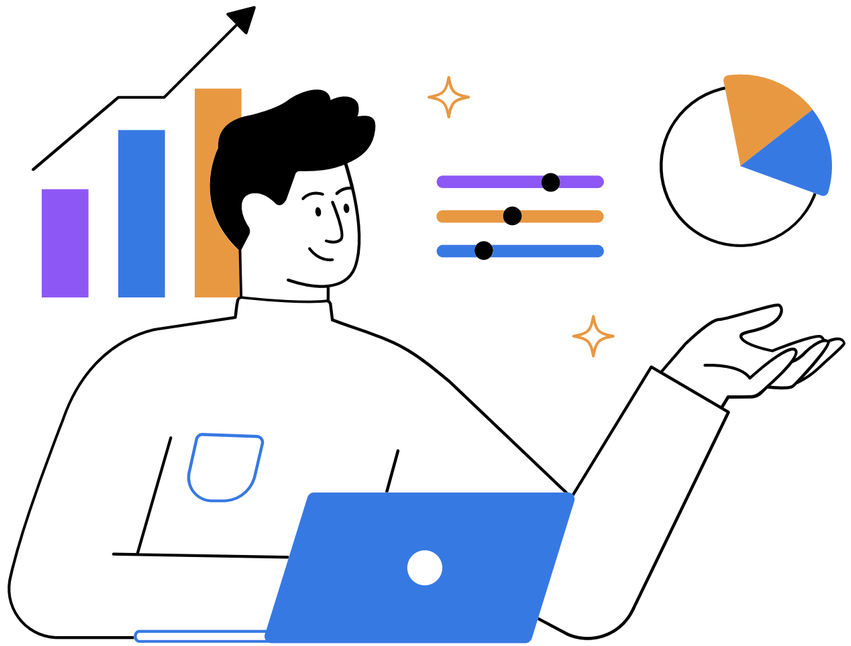
Core Characteristics

- Text-based: Human-readable plain text format
- Language-independent: Works across all programming languages
- Hierarchical: Supports nested data structures
- Typed: Supports multiple data types natively
- Standardized: IETF RFC 8259 specification

JSON Structure & Syntax

Supported Data Types

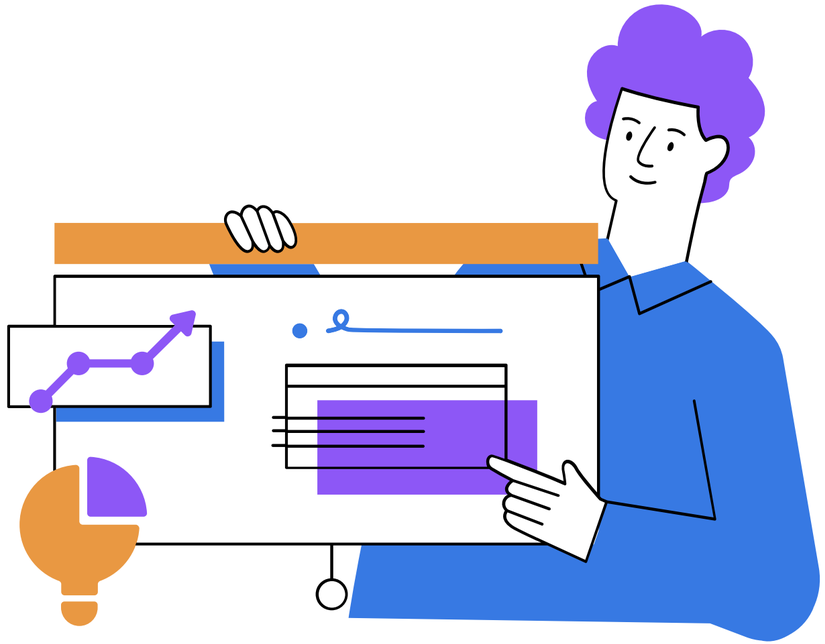
- "*string*" - Text enclosed in double quotes
- *123 or 123.45* - Numbers (integers/floats)
- *true / false* - Boolean values
- *null* - Null value representation
- *[1, "two", true]* - Arrays/Lists (ordered collections)
- *{"key": "value"}* - Objects/Maps (key-value pairs)



Syntax Rules

- Double quotes required for string literals
- Commas separate values in arrays/objects
- Colons separate keys from values in objects
- Curly braces {} for objects, square brackets [] for arrays
- Trailing commas not allowed

```
// JSON EXAMPLE: SCRUM Team Structure
{
    "team": {
        "name": "Alpha Development Team",
        "project": "SCRUM Management System",
        "createdDate": "2025-01-15",
        "members": [
            {
                "id": 1,
                "name": "Ana García",
                "role": "Product Owner",
                "email": "ana.garcia@company.com",
                "active": true,
                "storyPointsAssigned": 0
            },
            {
                "id": 2,
                "name": "Carlos López",
                "role": "Scrum Master",
                "email": "carlos.lopez@company.com",
                "active": true,
                "certifications": ["CSM", "PSM I"]
            },
            {
                "id": 3,
                "name": "María Pérez",
                "role": "Frontend Developer",
                "email": "maria.perez@company.com",
                "active": true,
                "technologies": ["React", "TypeScript", "CSS3"]
            }
        ],
        "sprintsCompleted": 5,
        "averageVelocity": 28,
        "projectProgress": 0.75,
        "nextSprintStart": "2025-03-18"
    }
}
```



CSV - Comma Separated Values

CSV Definition & Standards

CSV (Comma-Separated Values) is a plain text format for storing tabular data. Each line represents a record, and comma-separated fields represent attributes of that record.

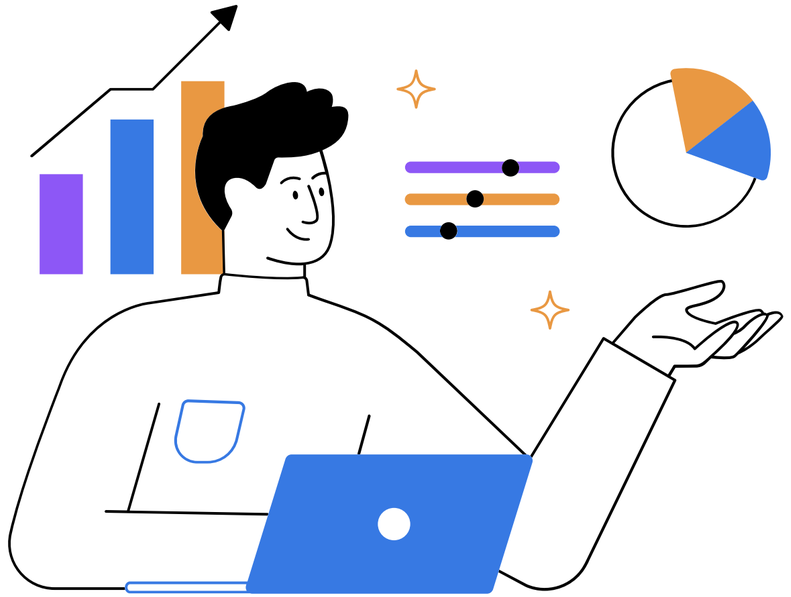
Key Characteristics

- Simple: Basic plain text tabular format
- Universal: Supported by virtually all software applications
- Compact: Minimal file size for large datasets
- Human-readable: Easily editable in text editors

CSV Structure & Rules

Format Components

- Field Separator: Comma (,) is standard, semicolon (;) as alternative
- Record Separator: Line break (CRLF or LF) between records
- Text Qualifier: Double quotes (") for fields containing delimiters
- Character Encoding: UTF-8 recommended for international support
- Header Row: First row typically contains column names



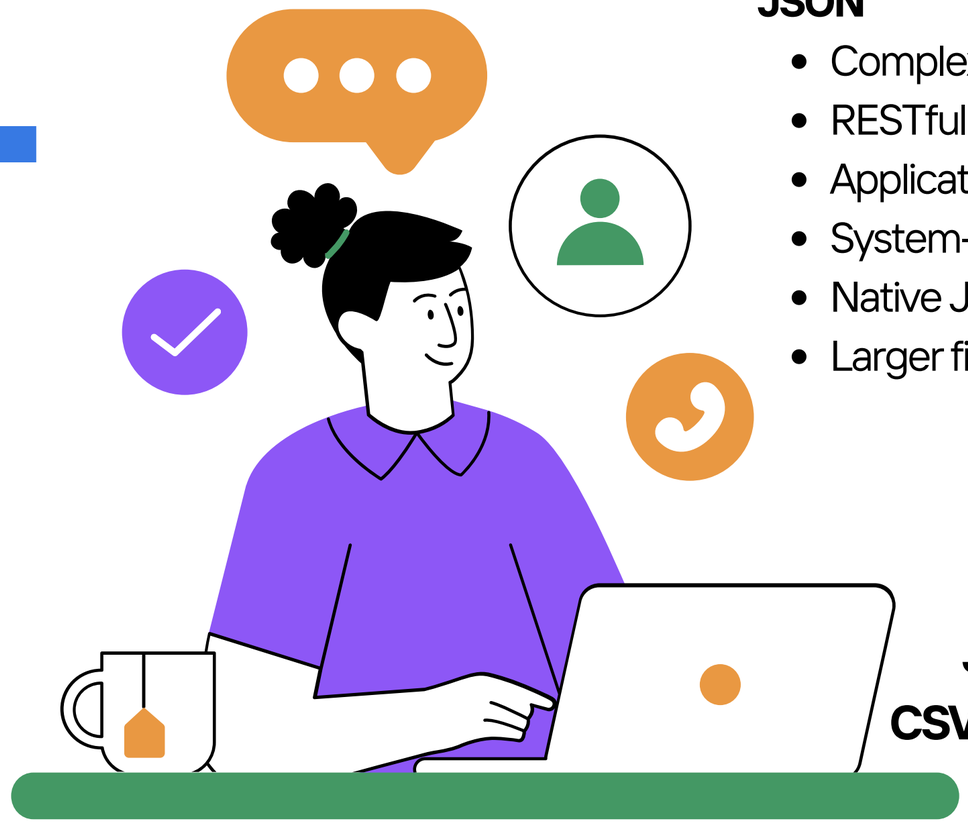
Implementation Considerations

- Numeric values stored without quotes
- Text fields containing commas must be quoted
- Embedded quotes escaped by doubling ("" becomes """")
- Leading/trailing whitespace preserved within quoted fields
- Empty fields represented by consecutive delimiters

```
// CSV EXAMPLE: Sprint Metrics Report
Sprint,StartDate,EndDate,PlannedPoints,CompletedPoints,Velocity,Status,
Sprint 1,2025-01-06,2025-01-20,25,22,22,Completed,1.38,80
Sprint 2,2025-01-21,2025-02-03,28,25,25,Completed,1.25,85
Sprint 3,2025-02-04,2025-02-17,30,28,28,Completed,1.07,88
Sprint 4,2025-02-18,2025-03-03,32,30,30,In Progress,1.07,90
Sprint 5,2025-03-04,2025-03-17,35,0,0,Planned,0,92

// IMPLEMENTATION NOTES:
// - Header row defines column structure
// - Date fields use ISO 8601 format (YYYY-MM-DD)
// - Numeric fields stored without quotes
// - Empty planned values for future sprints
// - Compatible with Excel, Google Sheets, BI tools
```

JSON vs CSV: Technical Comparison



JSON

- Complex hierarchical data structures
- RESTful APIs and web services
- Application configuration files
- System-to-system data exchange
- Native JavaScript object conversion
- Larger file sizes for simple data

CSV

- Simple tabular/flat data structures
- Data analysis and reporting tools
- Bulk data import/export operations
- Minimal storage requirements
- Universal spreadsheet compatibility
- No support for nested relationships

Performance Metrics

JSON: 2-3x larger file size but 20% faster parsing in web applications

CSV: 60% smaller files but requires additional parsing logic for complex data

Thank You