



Mastering Agile & Scrum

From Student to Professional Engineer

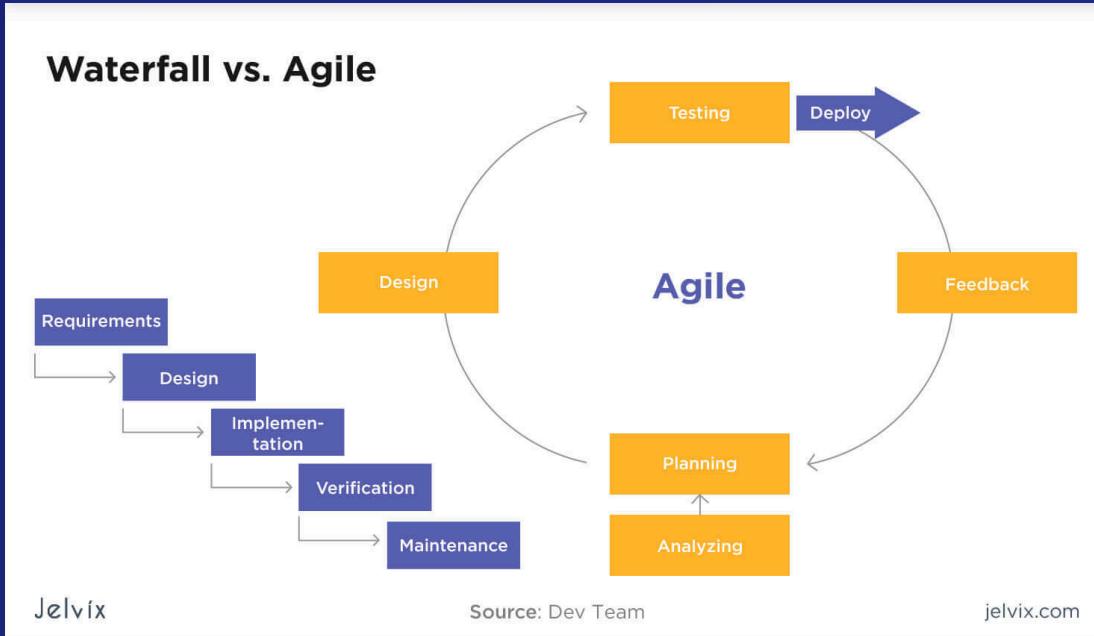
IIT Gandhinagar AI/ML Training Program - Day 5

PM Session

Friday, February 27, 2026



The Waterfall Problem



Limited Flexibility

Cannot adapt to changing requirements
Locked into initial design decisions



Late Feedback

Problems discovered only at the end
Stakeholders see product after months/years



High Risk

Testing delayed until completion
Expensive rework when issues emerge



Projects took months/years, often failed to meet expectations

The Solution: Agile Manifesto

A Paradigm Shift in Software Development (2001)



Individuals & Interactions

OVER Processes & Tools



Working Software

OVER Comprehensive Documentation



Customer Collaboration

OVER Contract Negotiation



Responding to Change

OVER Following a Plan

💡 These values guide every decision in Agile development

12 Agile Principles

1 Satisfy customer through early delivery

5 Build around motivated individuals

9 Technical excellence enhances agility

2 Welcome changing requirements

6 Face-to-face conversation

10 Simplicity is essential

3 Deliver working software frequently

7 Working software measures progress

11 Self-organizing teams

4 Business & developers work daily

8 Maintain sustainable pace

12 Regular reflection & adaptation

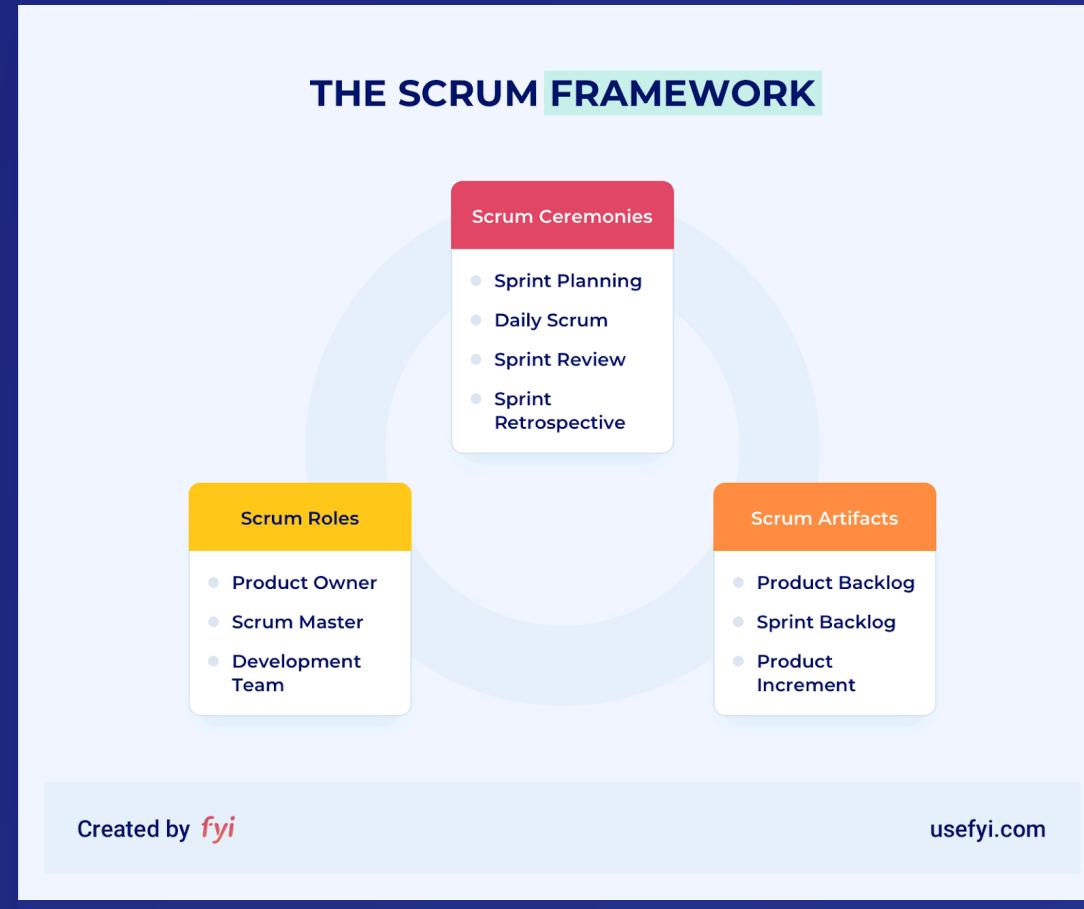
Scrum Framework Overview



Most Popular
Agile Framework



Structure with
Flexibility



Empiricism



Iterative



Self-Organizing



Time-Boxed

Scrum Team Roles



Product Owner

- 👤 Voice of the customer
- 📝 Prioritizes backlog
- .DEFINE Defines features
- ☑️ Accepts/rejects work



Scrum Master

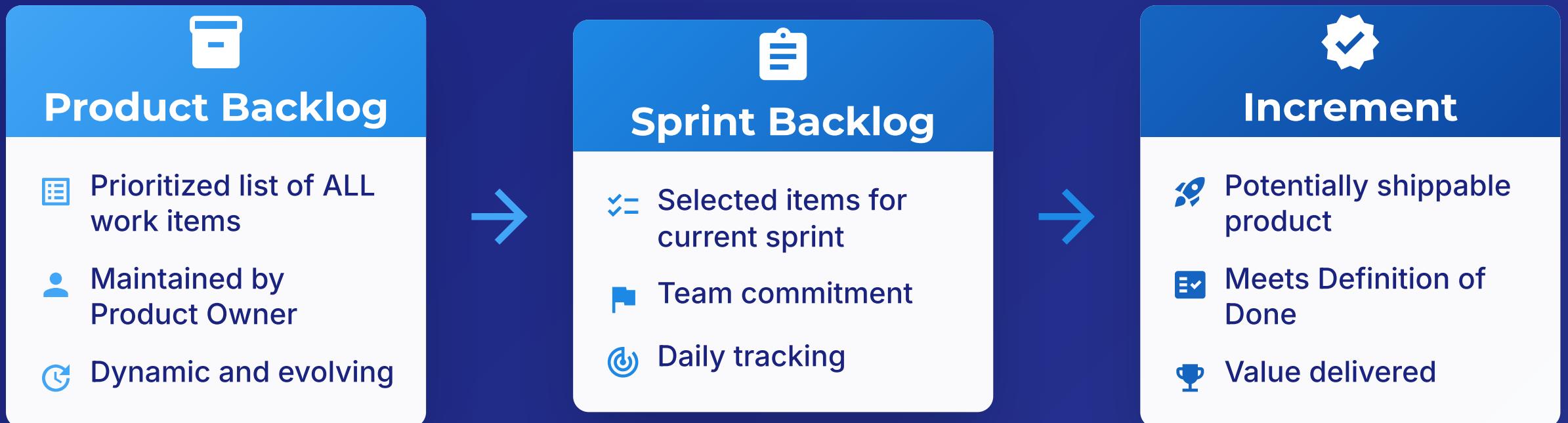
- 👥 Facilitator
- 🚫 Removes impediments
- 🛡️ Shields team
- 🔀 Enforces Scrum practices



Development Team

- 👤 Cross-functional
- 🏃 Self-organizing
- 👤 5-7 members typically
- FLAG Commits to sprint goals

Scrum Artifacts: The Work Products



Scrum Ceremonies: The Rhythm of Work

Sprint Planning

When: Sprint Start

Plan sprint work • Select backlog items • Set sprint goal

Sprint Review

When: Sprint End

Demo completed work • Stakeholder feedback • Update backlog

Scrum Methodology

Template



Daily Standup

When: Daily

15 minutes • 3 questions • Synchronize team

Sprint Retrospective

When: Sprint End

Reflect on process • Identify improvements • Action items

User Stories: Expressing Requirements

Story ID: Story title:

User story

As a: <role>
I want: <some goal>
So that: <some reason>

Importance:

Estimate:

Acceptance Criteria

And I know I am done when:

Type:

- Search
- Workflow
- Manage Data
- Payment
- Report / View

” Story Format

As a [WHO], I want [WHAT], so that [WHY]

Example: As a data scientist, I want to visualize model performance, so that I can identify optimization opportunities

INVEST Criteria

Independent

Negotiable

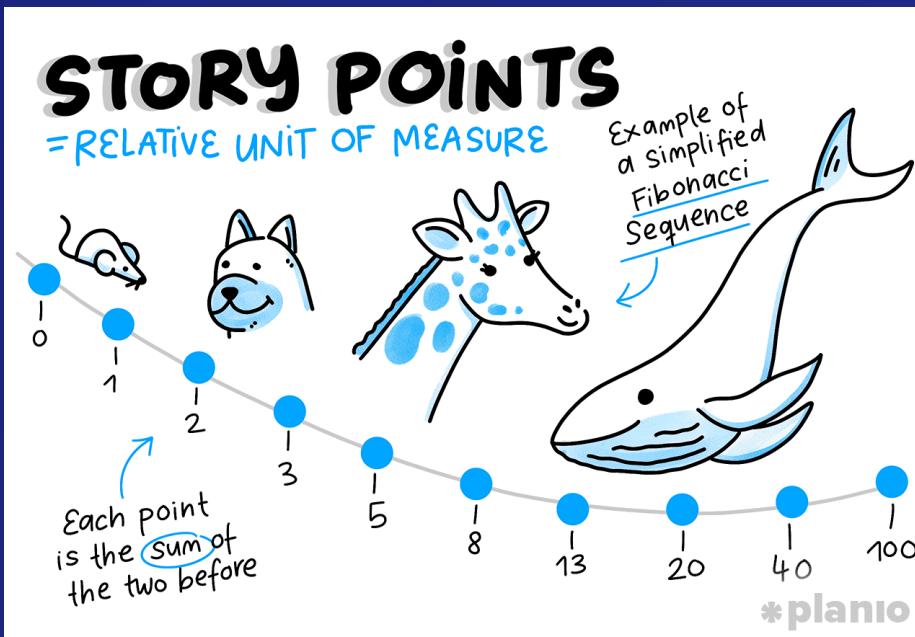
Valuable

Estimable

Small

Testable

Story Points: Estimating Work



What Story Points Measure

Story Points measure **Complexity, Uncertainty, and Effort (CUE)**

Fibonacci Scale

1 2 3 5 8 13 21 34 55

↘ Smaller numbers = easier to estimate accurately

↗ Larger gaps = handle uncertainty

i Key Insight

Not about time - about **relative complexity** compared to reference story

Daily Stand-up: Your Daily Checkpoint



Format & Timing 15 min max Same time daily

The 3 Questions

- 1 What did I **accomplish** yesterday?
- 2 What will I **work on** today?
- 3 Are there any **blockers**?



NOT for problem-solving

Status updates only

Raise issues to discuss later

Stand-up Preparation Template

Think Before You Speak

Empty Template

Yesterday's Accomplishments	Today's Plan	Blockers/Impediments
[Fill in what you completed]	[Fill in your plan]	[List any blockers]

💡 Best Practices

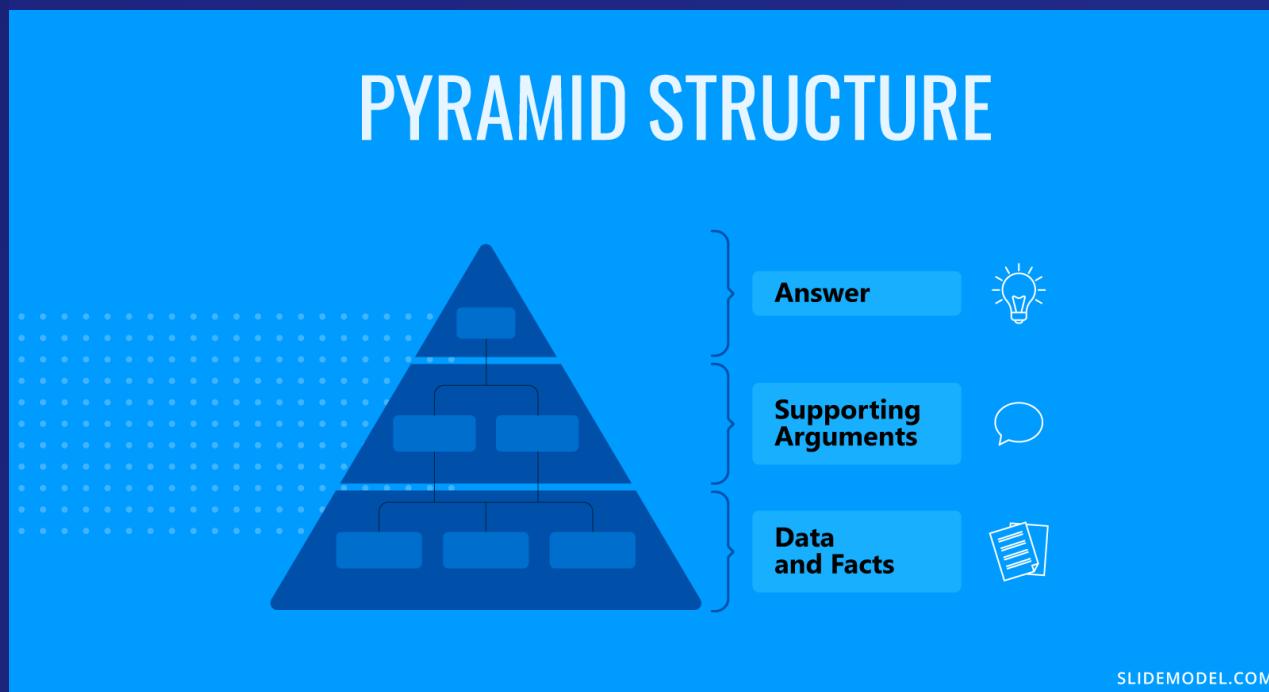
- Fill this BEFORE stand-up
- Be specific and concise
- Lead with results, not activities

Filled Example

Yesterday's Accomplishments	Today's Plan	Blockers/Impediments
<ul style="list-style-type: none">✓ Completed data preprocessing for ML model✓ Validated data quality metrics	<ul style="list-style-type: none">○ Start feature engineering (2-3 features)○ Begin model training pipeline setup	<p>❗ Waiting for API access credentials</p> <p>Impact: Can't access production data</p>

McKinsey Pyramid Principle

Communicate Like a Pro



✗ Bad Communication

"I worked on preprocessing, then feature selection, then... we should use XGBoost"

✓ Good Communication

"I **recommend XGBoost** for our model. Tested 3 algorithms, XGBoost achieved **95% accuracy**, 15% better than baseline. Here's the data..."

💡 Key Takeaway

Lead with the **answer**, support with **key points**, back with **evidence**

Stand-up Communication

Do's & Don'ts

DO's ✓

✓ Be concise and specific

✓ Lead with results/outcomes

✓ Mention blockers clearly

✓ Use Pyramid Principle

✓ Keep it under 2 minutes

DON'Ts ✗

✗ Don't ramble or tell stories

✗ Don't solve problems in stand-up

✗ Don't make excuses

✗ Don't discuss technical details

✗ Don't skip your update

■ Your goal: Sound like a professional with 4-5 years experience

Practice Scenario: AI/ML Project Stand-up



Scenario: Building recommendation system | Sprint 3 of 6 | Day 10

X Novice Response

Yesterday:

"Yesterday I spent a lot of time working on the model. I tried different things like XGBoost and random forest. I also looked at some data."

Today:

"Today I'll continue working on it."

Blockers:

"I'm facing some issues with the code."

✓ Professional Response

Yesterday:

"Trained 3 models, XGBoost performed best (87% accuracy)"

Today:

"Will tune hyperparameters and integrate with API"

Blockers:

"Need production dataset access from DevOps team - critical for sprint goal"



Vague, no metrics, no clear plan, blocker unclear



Specific metrics, clear plan, blocker with impact

Agile for AI/ML Projects

Special Considerations

Unique Challenges in AI/ML Projects

Data Uncertainty

Quality & availability varies

Model Experimentation

Multiple iterations needed

Rapid Iteration

Quick feedback cycles

Stakeholder Education

AI capabilities & limits

Case Study: Recommendation System Project

6-Month Deadline

Tight timeline delivery

2-Week Sprints

Iterative delivery cycle

Daily Standups + Reviews

Continuous alignment

Test-Driven ML

Automated model testing

Key Lessons Learned

Communication is critical

Flexibility essential

Automated testing crucial

Balance tech vs business

Applying Agile in Your Training Journey

🕒 Daily Standups

- 🕒 Every training day at start
- ↗ Track progress
- ⚠ Share blockers
- ⚙ Practice professional communication

↔ GitHub Integration

- ⬇ Micro-commits with messages
- 📁 Use repos for projects
- ➡ GitHub Discussions for Q&A
- 💼 Build your portfolio

⌚ Sprint-Based Learning

- 📅 Weekly sprints
- ⌚ Track velocity
- 🚩 Sprint goals & backlog
- ⟳ Continuous improvement

⌚ Weekend Project

- ⇐ Apply all Agile concepts
- 📝 Create user stories
- 🔢 Estimate story points
- 🚀 Deliver increment



Agile is a way of working, not just a methodology

Sprint Planning: Your Weekend Project

Steps to Plan

1 Review requirements & objectives

2 Break down into user stories

3 Estimate story points (Fibonacci)

4 Prioritize by value

5 Commit to what you can deliver

Example Project Breakdown

As a user, I want to visualize weather data
Display charts and graphs for weather metrics

As a user, I want to compare 3 cities
Side-by-side comparison of weather forecasts

As a user, I want to see forecasts
Predictive weather analysis for next month

 Total: 16 story points for 2-day sprint

Putting It All Together: Real Project Scenario

Building an AI-powered code review assistant - Sprint 1

Sprint Backlog

As a developer, I want to upload code

2

As a developer, I want AI analysis

8

As a developer, I want suggestions

5

As a developer, I want to export report

3

Total: 18 story points

Sprint Execution Timeline

1-2

Setup & upload feature

Project setup, code upload functionality

3-4

AI integration

Connect AI model, implement analysis logic

5-6

Suggestions & export

Implement suggestions, add export functionality

7

Testing & demo

Testing, bug fixes, sprint review demo



Expected: Functional prototype with core features

Key Takeaways

Agile Values

-  Individuals over processes
-  Working software
-  Customer collaboration
-  Respond to change

Scrum Components

-  3 Roles
-  3 Artifacts
-  4 Ceremonies
-  Sprints

Stand-up Best Practices

-  Lead with results
-  Use Pyramid Principle
-  Mention blockers
-  Be concise

Your Journey

-  Practice daily
-  Build portfolio
-  Continuous improvement
-  Think like a professional



Remember: Agile is a mindset, not just a process

Mock Stand-up: Let's Practice!

Instructions

- 1** Fill your stand-up template (5 minutes)

- 2** Review your past 2 days of work

- 3** Identify what you'll work on tomorrow

- 4** Note any blockers you're facing

- 5** Practice Pyramid Principle - start with key result

Group Formation

 **Option A: Entire cohort together**
First time practice - everyone participates

 **Option B: Split into groups**
Groups of 5-6 for more practice time

What to Remember

- ✓ Be concise (under 2 min)

- ✓ Lead with outcomes

- ✓ Use professional language

- ✓ Don't apologize or make excuses

Continue Your Agile Journey

Resources

 Scrum Guide (scrum.org)

 Agile Manifesto

 McKinsey Pyramid Principle book

 Jira/Trello for practice

 Daily stand-up templates

Practice This Week

 Daily stand-ups with template

 Create user stories for weekend project

 Estimate story points

 Track your progress in GitHub

Apply everything you learned!

What's Next

 Weekend project implementation

 Sprint reviews & retrospectives

 Advanced Agile patterns

 Real-world case studies

Your Agile career starts now!



Remember: Practice makes perfect. Every day is an opportunity to improve.