alok.giri@ncit.edu.np

What is Scrum?

Scrum is lightweight, Agile framework for managing and completing complex projects. It is empirical.

Scrum Framework Roles

- **Scrum master**
    - Facilitates all Scrum events
    - Ensures team follows scrum principles
    - Protect team from outside distractions - Does not act like a boss or traditional manager

Analogy: Like a coach in a football team– not playing, but helping everyone play better.

- **Product Owner**
  - Owns the product vision
  - Maintains and prioritizes the Product Backlog - Talks to stakeholders and understands customer needs
  - Accepts or rejects work as “Done”

Analogy: Like a restaurant manager who ensures customers are happy and the chefs cook the right dish.

- **Development Team**

- 5–9 people ideally
- Have all required skills (design, code, test, deploy)
- Self-organizing: they decide how to do the work - No titles or sub-roles inside the team (all are Developers)

Analogy: Like a rock band—each member has a role, but they all own the final performance.

Artifacts

- Product Backlog
    - Owned by Product Owner

- Dynamic and constantly evolving
- Refinement: breaking down, estimating and clarifying items

Example:

.Items: "Login page", "Search filter", "Add to Cart" .
Higher-priority items are better defined

- Sprint Backlog
  - Created during Sprint Planning
  - Developers commit to these items for the sprint
  - Includes tasks and effort estimations

It answers: What can we deliver in the next 2 weeks, and how will we do it?

- Increment
  - Must meet Definition of Done (DoD): passed tests, reviewed, deployable
  - Delivered to stakeholders during Sprint Review
  - Multiple increments may exist, but each increment is additive and complete

Scrum Events

- Daily Stand-up
  - Everyone answers:
    - What did I do yesterday?
    - What will I do today?
    - Any blockers?
  - Encourages team accountability and coordination
  - Same time, same place
- Backlog Refinement

- PO+Dev team meet to:
    - Clarify user stories
    - Estimate effort (story points)
    - Break large items into smaller ones

Backlog refinement avoids surprises in Sprint Planning

- Sprint Planning
    - Sprint goal is defined
    - PO explains top backlog items
    - Dev team selects items they can commit to
    - Tasks are broken down

- Sprint Review
    - Dev team showcases what’s “Done”
    - Stakeholders give feedback
    - PO may adjust backlog accordingly

It’s not a status meeting. It’s about getting real feedback on working software.

- Sprint Retrospective
    - Held after Sprint Review, before next Sprint Planning
    - Only team members + Scrum master

- 3 key questions:
  - What went well?
  - What didn’t go well?
  - How can we improve?

Kanban

- Introduction to Kanban
  - Visual system for managing work
  - Helps teams visualize tasks, limit WIP, and improve efficiency
  - Popular in Agile and Lean environments

Kanban was originally developed at Toyota for lean manufacturing. Its goal is simple: visualize how work moves, and continuously improve it.”

- Kanban Principles
    - Visualize Workflow
    - Limit Work In Progress (WIP)
    - Focus on Flow
    - Continuous Improvement (Kaizen)
- Visualize Workflow
    - Use boards to show task stages (e.g., To Do,

Doing, Done)

- Improves visibility, accountability, and clarity

- Limit Work In Progress (WIP)
    - Set limits on how many tasks are in-progress - Encourages task completion over multitasking - Prevents overload and bottlenecks
- Focus on Flow
    - Monitor how tasks move across the board

- Improve speed and predictability
    - Address delays early

- Continuous Improvement
    - Reflect regularly (retrospectives, daily checks)
    - Tweak workflow based on insights
    - Small improvements = big gains over time

Visualizing Workflow

- Create a Kanban Board

- Columns represent stages (e.g., To Do → Design → Dev → QA → Done)
- Swimlanes divide task types (e.g., Bugs, Features)

A Kanban board represents your team’s process. Customize it to your real-life workflow. Swimlanes help manage multiple types of work.

- Kanban Board Components
    - Columns = Workflow stages
    - Cards = Tasks or user stories
    - Swimlanes = Parallel task types
    - Drag cards to indicate progress

Boards are dynamic. As work progresses, cards move right. Teams can instantly see priorities, progress, and blocks.

Managing Work in Progress

- Prevent overload
- Improve focus
- Shorter delivery cycles

If everything is in-progress, nothing is done. By limiting WIP, you free up capacity and increase quality output.

- Monitoring Flow Efficiency
    - Cycle Time: Start to Finish of a task
    - Lead Time: Request to Delivery
    - Shorter times = Better flow

Use these metrics to analyze efficiency. If your cycle time is long, work might be stuck. If your lead time is too long, prioritization might be off.

Extreme Programming (XP)

Extreme Programming (XP) is an Agile software development framework that emphasizes technical excellence, rapid feedback, and customer involvement.

- XP encourages small, frequent releases - Built for highly dynamic and uncertain projects - Focuses on engineering practices (vs. Scrum's process framework)

XP Values

- Communication – Constant feedback between team members
- Simplicity – Build only what is needed now - Feedback – Get early and frequent feedback (tests,

customers)
- Courage – Refactor, delete code, challenge assumptions
- Respect – Value each team member’s input and pace

XP Roles

- Developer- Writes and refactors code, tests -
Customer- Provides user stories, business goals -
Tester- Ensures coverage via automated/manual tests -
Tracker- Monitors progress and velocity
- Coach- Guides the team in XP practices

Practice 1- TDD

- Red: Write a failing test
- Green: Write the minimal code to pass - Refactor: Clean up code without changing behavior

Benefits

- Instant feedback
- Cleaner code
- Better test coverage
- Encourages simplicity

Practice 2- Pair Programming

TWO BRAINS, ONE KEYBOARD

- 2 developers work together:
    - Driver: Writes the code
    - Navigator: Reviews, thinks ahead, finds bugs
- Switch roles every 30 mins - 1 hour

Benefits: Fewer bugs, shared knowledge, faster problem solving, onboarding becomes easier

Continuous Integration (CI)

- Developers integrate code frequently (multiple times per day)

- Each integration is verified by automated tests

Benefits:

- Detect integration issues early
- Shortens feedback cycle
- Maintains a working system

Lean Software Development

- What is Lean
    - Origin: Lean manufacturing at Toyota
    - Focus: Maximize value, minimize waste

    - In software: Focus on flow, quality, and fast delivery

Lean is about doing more with less

Lean Principles

- Eliminate Waste
    - Waste = anything that doesn’t add value to the customer
    - Examples:
        - Extra features no one uses
        - Waiting for approvals
        - Defects and rework

        - Too many meetings
- Build Quality In
    - Prevent defects instead of fixing them later
    - Use:
        - Automated testing
        - CI/CD pipelines
        - Code reviews and pair programming
- Defer Commitment
    - Don’t make decisions too early
    - Keep options open until the last responsible moment
    - Reduces risk of poor decisions

Booking a flight just in time with enough data vs. too early and getting wrong date

- Deliver Fast
    - Speed = feedback + learning
    - Deliver small, usable chunks
    - Helps validate ideas early

Example: MVP -> deploy fast -> get real feedback

- Respect People
    - Empower teams to make decisions
    - Promote ownership and responsibility

    - Avoid micromanagement

Pathao working mechanism.

- Optimize the Whole
    - Don’t optimize parts at the cost of the whole system
    - Focus on end-to-end flow, not just development

Dev builds fast but QA gets overwhelmed = not real optimization

- Continuous Improvement (Kaizen)
    - Encourage team to inspect and adapt regularly

- Retrospectives are key
- Encourage experimentation

Eliminating Waste

- What is waste in software?
  - Anything that doesn’t directly contribute to delivering value is waste.
- Types of waste:
  - Partially done work, extra features, relearning, handoffs, task switching, delays, defects

Continuous Improvement

- What is Kaizen?
  - Japanese word for “Change for better” - Ongoing, incremental improvements - Driven by everyone, not just management

Practices for Continuous Improvement

- A and B Testing
- User Testing (Beta Testing)
- Team Reviews