SCRUM

Module 3: The Product Owner Role

Overview

- This module is concerned with the functions of the Product Owner role
- For this session, we also want to deal with your issues that relate to this role
- And document issues that we can move do a deep dive in future sessions
  - Your input is essential.
  - As mentioned in the first session, we will adapt the material to your requirements
  - But I do need you to provide the feedback on your requirements

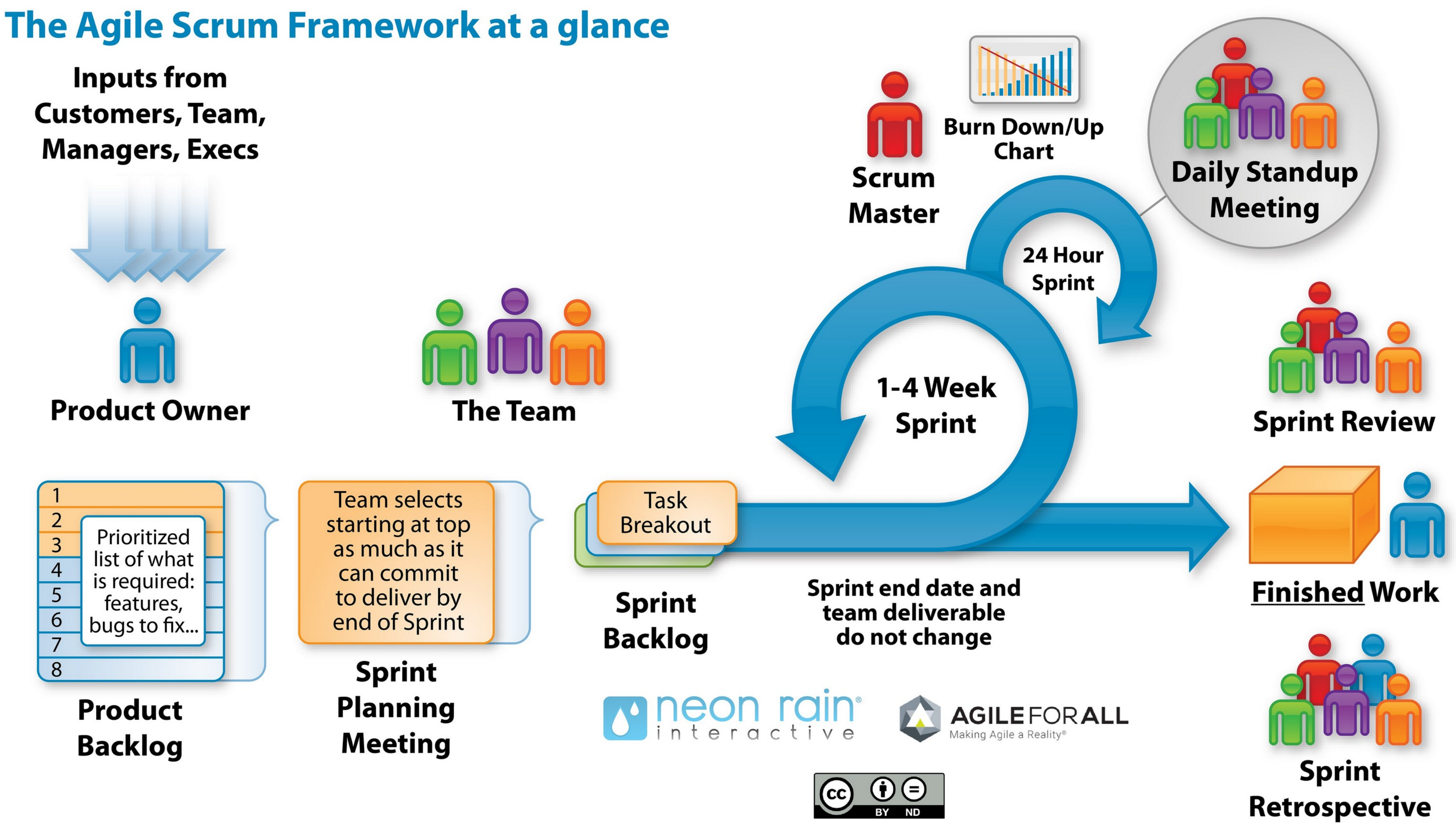
Class Discussions (repeat)

- From last week
  - I want you to start providing me with inputs to narrow the class cone of uncertainty
  - The rest of the the course should focus on high value content and activities
  - Over the next several weeks, we will be doing a deep dive into Scrum
  - But we also want to focus on how you are using Scrum, make it relate to your projects
- To do this, I need to know something about:
  - Specific issues you may be having with Scrum
  - Any places we need to review how Scrum is being applied.
  - Where we need to focus our efforts in class

Issues Identified Previously

- Managing change requests
  - On of the most common problems encountered by Product Owners are dealing with the flood of change requests received from stakeholders
  - Requires a process for prioritizing, managing and making decisions on the disposition of requests
- Managing requirements
  - Requirements management is one of the most difficult aspects of working on a project with multiple stakeholders
  - Scrum emphasizes the importance of creating value in the product backlog but doesn’t explain the steps to actually do this work
- These are two areas Scrum assumes your methodology will handle
  - In terms of processes, techniques, documentations and management

The Scrum Framework (repeat)



The Product Owner vs BA

- Product Owner
  - Owns and manages the product backlog.
  - Represents customer and business value in decision-making.
  - Focus on continuous prioritization and value delivery
- Business Analyst (BA)
  - Gathers and documents requirements.
  - Acts as liaison between business stakeholders and development team.
  - Focus on analysis and documentation, often upfront in the project lifecycle.
  - Can be thought of as the Product Owners “field agents”
  - The BA may often be a specific domain or subject area experty

Requirements vs Features

- A very important distinction to be made is between features and requirements
- Requirements
  - Represent a need the stakeholder has
    - *I need to be able to find all transactions in the last 30 days sorted by customer*
    - *No one should be able to create new customer records unless they are a manager*
  - They do not describe what the system does
  - You can add the phrase “.. and I don’t care how you do it.” without changing the meaning
- Features
  - What the system does or presents to the user
    - *The system has transaction search feature with a drop down list of all the search parameters*
    - *The system has a create option for functions but is restricted to management level user accounts*

Requirements vs Features

- Why is this important?
- A very common problem is that stakeholders may tell they want a specific feature
  - They have “solved the problem for us”
  - Their proposed feature may not actually satisfy the motivating requirement
  - The end result is that we build the wrong thing
- Developers cannot build to requirements
  - A requirement has to be converted into a feature specification before construction can be done
  - Requirements are not measurable, feature specifications are
  - We can’t plan to requirements, we can only plan to feature specifications

The Product Owner

- Product Backlog
  - Product owner is responsible for ensuring that requirements become features
  - This is one of the collaborative functions between the Dev Team, Product owners and BAs and stakeholders
- We will return to this topic in “grooming the backlog”

Product Owner Deep Dive

- PO responsible for maximizing value through backlog management including:
- Determining Acceptance Criteria:
  - Translating requirements into testable conditions for knowing when a feature is “done.”
  - “Done” is often defined as the product meeting a measurable set of acceptance criteria
  - Often includes set of acceptance tests developed with the stakeholders
    - “What should happen if someone tries to log in with the wrong password more than three time?”
  - Forces the business to articulate what they expect the “correct” response to be.
- Resolving Conflicting Requirements:
  - Negotiating trade-offs between different business units or customers.
  - Failing to do this will result in ambiguous design goals or contradictory feature specifications
- Prioritizing Requirements:
  - Developing and refining a set of criteria for prioritizing features
  - This includes guiding the business in prioritizing their requirements that motivate the features
- Supporting Go/No-Go Decisions:
  - Helping business decide whether features should be released.
- Market & Competitor Analysis:
  - Understanding external context that informs backlog decisions.

Product Owner Functions

- Vision Evangelist:
  - Constantly communicating the product vision to keep team and stakeholders aligned.
  - Develops and updates the product vision as necessary
  - Helps keep the scope of the project under control
  - Provides criteria for feature selection
- Change Agent:
  - Things will change: business context, technologies, capabilities etc, that will impact what can be done
  - Oversees and communicates changes to be business and dev team to them on the same page
  - Manages and triage change requests
- Release Management:
  - Deciding what increments go into a release consulting with dev team and stakeholders
- Risk Manager:
  - Identifying product-level risks (market changes, compliance deadlines)
- Metrics Owner:
  - Tracking success using product metrics (NPS, usage analytics, defect rates).
- Internal Trainer:
  - Educating stakeholders on how to work with the Scrum process

Change Request Management

- Not the same as change management
  - Scrum doesn’t include a change request management process
- Example change request process for a product owner
  - The Product Owner is responsible for keeping a working history of change requests
- Filter
  - Does this request even relate to our project? If not, reject it and record
  - Is this a duplicate request that we have dealt with before? If so, mark as a duplicate and record
  - Is this an actionable request
    - *There is enough content to make a decision, if not, mark as returned to submitter for clarification*
    - *Is it actually something that can be done, if not, mark it as rejected and record*
  - If not rejected, returned or marked as a duplicate, move it to intake.

Change Request Management

- Intake
  - Ensure there is enough data to make a decision
  - Request information capture, for example:
    - *An id that uniquely identifies the request*
    - *Title & description*
    - *Business justification*
    - *Expected value / impact*
    - *Urgency / deadline (if any)*
    - *Stakeholder contact and/or origin*
  - Determine the type of change request
    - *The categories are defined by the project*
  - Common types
    - *Enhancement (new feature or improvement)*
    - *Defect / Bug (fix to existing feature)*
    - *Technical / Compliance requirement*

Change Request Management

- Decision
  - Impact & Value Analysis
  - PO works with stakeholders and team to evaluate (for example):
    - *Business Value (customer benefit, revenue, risk reduction, compliance).*
    - *Effort / Complexity (initial team estimate).*
    - *Dependencies or risks (impact on roadmap, architecture, other features).*
  - The team has to have a disposition strategy
  - Often looks like:
    - *Out of scope: Reject the request (this is the second place we can do this)*
    - *Business Reject: The request has no actual business or technical value*
    - *Technical Reject: Cannot be done within the capabilities of the project*
    - *Defer: Hold off on deciding until more information is available*
    - *Accept: Implement the change request*
    - *Product or portfolio backlog: For example, should be done in the next project*

Change Request Management

- Prioritization
  - PO places the request in the backlog and orders it relative to existing items.
    - *This is a collaborative effort with the development team and Scum Master*
    - *May include stakeholders*
  - Common frameworks:
    - *WSJF (Weighted Shortest Job First)*
    - *MoSCoW (Must, Should, Could, Won’t)*
    - *ROI / Cost-benefit scoring*
    - *IT Risk Framework*
  - PO balances business urgency vs. technical feasibility.
    - *Record keeping is essential*
    - *Have to be able to refer to past requests, decisions and dispositions*

Change Request Management

- Refinement with the team
  - In backlog refinement sessions, PO and team:
    - *Clarify acceptance criteria.*
    - *Estimate effort more accurately.*
    - *Break down large changes into manageable tasks*
- If prioritized for a sprint:
  - Processed request becomes part of the Sprint Planning.
- If deferred
  - Remains in backlog, with clear communication to the stakeholder about why and when it might be addressed.
- If rejected:
  - Stakeholder receives a documented explanation, tying back to product vision or constraints.

Change Request Management

- Tracking and feedback
  - PO ensures the change request is visible on the backlog/roadmap.
  - Stakeholders get updates when status changes (accepted, in progress, released).
  - After delivery, collect feedback to measure if it achieved the intended value.
- The request management process
  - Must be defined by the team
  - Scrum does not specify one, but assumes you have one appropriate for your project

Product Backlog Grooming

- Backlog grooming (refinement)
  - Ongoing process of reviewing, clarifying, and adjusting items in the product backlog
  - Ensures they are well-prepared for upcoming sprints.
  - Ensures items are prioritized, detailed, and understood before sprint planning.
  - The change request process uses many of these same ideas
- Grooming Goals
  - Modularization:
    - *Large features/requirements are broken down into small, testable, and understandable items.*
  - Priority:
    - *Items are ranked so the team knows what to work on next.*
  - Readiness:
    - *High-priority items meet the Definition of Ready.*
  - Adaptation
    - *Reflects the most current product vision, market conditions, and stakeholder needs*

Product Backlog Grooming

- Key Activities
  - Prioritization: Reordering items based on business value, urgency, or risk.
  - Estimation: The development team estimates effort (e.g., story points, man days)
  - Splitting / Breaking Down: Large tasks are decomposed into smaller, actionable items.
  - Clarification: PO explains requirements, acceptance criteria, and business context.
  - Pruning: Removing outdated or irrelevant items.
  - Adding New Items: Incorporating feedback from stakeholders, customers, or team insights.
  - Test Development: Identify the test scenarios and test cases needed to test the item to a sufficient level of thoroughness
  - New Issues: Identify issues that rose during development
    - *Eg, Proposed architecture causes the system performance to slow down to unacceptable levels*
    - *Eg. A security vulnerability has been identified*

Product Backlog Grooming

- When It Happens
  - Continuous process, not a single meeting.
  - Common practice: 1–2 refinement sessions per sprint, usually 5–10% of team’s capacity.
  - High-priority items should always be 2–3 sprints ahead of readiness.
  - Might be triggered by changes in the environment, industry or regulatory requirements

Product Owner Deliverables

- Product Vision Document
  - Defines the why behind the product.
  - Sets a clear direction and inspires the team and stakeholders.
  - Aligns all work to long-term business goals.
  - Contents: Vision statement, target market, value proposition, success criteria.
- Product Scope / High-Level Requirements
  - Outlines the what (major features, constraints, compliance needs).
  - Provides stakeholders with clarity on boundaries (what is in-scope and out-of-scope).
  - Serves as an initial reference for prioritization.
- Product Roadmap
  - Communicates when major features or themes are expected.
  - Provides a time-based or outcome-based view of planned product evolution.
  - Helps align stakeholders, sales, marketing, and delivery teams.
  - Note: Roadmaps are directional, not commitments.

Product Owner Deliverables

- Product Backlog
  - The core Scrum deliverable — a living, ordered list of all work items.
  - Contains epics, user stories, enablers, and defects and all artifacts required for the project
  - Represents the single source of truth for what the team will work on.
- Release Plan
  - Provides a short- to medium-term plan of what increments will be delivered to customers.
  - Links backlog items to delivery dates or releases.
  - Helps manage stakeholder expectations.
- User Stories (Scenarios) & Acceptance Criteria
  - Translate business needs into testable, actionable requirements.
  - Ensure shared understanding between PO, developers, and testers.
  - Clarify definition of done and quality standards.

Product Owner Deliverables

- Prioritization Framework / Scoring Model
  - Transparent decision-making tool for why items are ranked in the backlog.
  - Balances business value, customer impact, technical risk, and cost.
  - Ensures fairness across stakeholders.
- Product Metrics & Reporting
  - Show whether the product is meeting its vision and goals.
  - Common metrics: adoption rates, customer satisfaction (NPS), usage analytics, ROI.
  - Guide future prioritization and strategy.
- Stakeholder Communication Artifacts
  - Examples: Stakeholder maps, release notes, demo decks.
  - Keep stakeholders informed and engaged.
  - Translate technical progress into business impact.
  - Build trust and transparency.

Questions