What is a Product Owner?

Product owner is a role on an Agile scrum team. The product owner (PO) represents the business or user community and is responsible for working with this group to determine what features will be in the product release. Other roles in scrum include scrum master and development team.

PO responsibilities

- **Work with stakeholders** (e.g. SIGs, developers, UX designers) to **develop a vision for the product** or feature they are responsible for
- **Communicate** this vision to the development team through a **prioritized user story backlog** that maximizes value delivered
- **Be available to answer questions** in a timely manner throughout the development process
- **Test and accept user stories**
- **Test features** regularly (including for accessibility), file and prioritize bugs as-needed and be generally aware of the state of quality for area of responsibility
- **Remain on top of the current state of development**
- **Demonstrate progress** and provide status updates to stakeholders
- **Conduct user acceptance testing** (UAT)
- **Work closely with Technical Writer/Editor** to provide thorough and up to date documentation
- **Capture longer-term plans** in backlog of UXPROD features
- **Prioritize features for quarterly releases** using inputs such as business value (UXPROD rankings), technical dependencies and available capacity

Despite being supplied to the project by various vendors and academic institutions, Product Owners support the entire community. This means:

1. Listening to all perspectives and working with the SIGs to design common solutions that meet all needs
2. Capturing backlog items proposed by community members as UXPROD features so they can be ranked and prioritized for development
3. Verifying with the broader community before making changes requested by a single institution
4. Monitoring support issues and working to support all production customers with critical bug fixes
5. Working with other Product Owners and SIGs to support the needs of adjacent domains and workflows

See Getting Started for Product Owners page for more details on processes and links to resources.

"Thin thread" development

One of the key benefits of scrum development methodology is that it is **iterative and incremental**. The goal is to release testable software early and often, so that feedback can inform future development. "Thin thread" development supports this goal by putting the emphasis on development of **MINIMAL features needed to support an end-to-end workflow**, prioritizing those workflows that cross apps.

Writing user stories using "INVEST"

A great mnemonic for structuring user stories is INVEST:

<table>
<thead>
<tr>
<th>Letter</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Independent</td>
<td>The user story should be self-contained, in a way that there is no inherent dependency on another PBI.</td>
</tr>
<tr>
<td>N</td>
<td>Negotiable</td>
<td>User stories are not explicit contracts and should leave space for discussion.</td>
</tr>
<tr>
<td>V</td>
<td>Valuable</td>
<td>A user story must deliver value to the stakeholders.</td>
</tr>
<tr>
<td>E</td>
<td>Estimate-able</td>
<td>You must always be able to estimate the size of a user story.</td>
</tr>
<tr>
<td>S</td>
<td>Small</td>
<td>User stories should not be so big as to become impossible to plan/task/prioritize within a level of accuracy.</td>
</tr>
<tr>
<td>T</td>
<td>Testable</td>
<td>The user story or its related description must provide the necessary information to make test development possible.</td>
</tr>
</tbody>
</table>