

## A Tester's Guide to Collaborating with Product Owners: Webinar Highlights

This webinar explores the 9 key patterns for Product Owner and Tester Collaboration. Bob Galen and Shaun Bradshaw share new ideas and techniques for Testers to help Product Owners deliver better received and higher value products.

### *Areas where Testers can make a difference in the life of the PO*

#### Who Owns the Backlog?

- Technically the Product Owner owns the backlog (that is what it says in the Scrum guide)
- Product Owner is not sole contributor to the backlog
- Testers need to ensure that what Product Owner asks for is what Development builds
- Adding Acceptance criteria to stories is a good opportunity for testers
- Asking questions about how users will use feature is another way that tester can contribute
- Driving conversations around testability is also a way that tester can contribute
- Good Product Owner creates space where everyone can add to the backlog
- Product Owner has final say about backlog

#### 4 Quadrants of Product Ownership

- Product ownership involves skills associated with Product Manager, Project Manager, Leader and Business Analyst
- Testers should look at each of these areas/skillsets and see where they can help. Testers have customer skill set like Product Owner's. They have customer perspective
- Testers are comfortable diving into questions such as: What does a customer want? What is the long term vision?

*Now let's get to the patterns!*

#### 1. Bridge stories from Team to the Product Owner

- Plan a story so that 1/3 of way thru coding, you run your test. This way, Product Owner gets sense of progress
- As a tester remind team of 3 Amigos based interactions



- Every story that comes to a team-three amigos should be actively collaborating
- **Nail the Demo**-Testers tend to want to validate everything in as much depth as they possibly can which can lead to areas that may not be of great importance in this sprint. Ask Product Owner what is it that he/she absolutely must see. This will help you prioritize what you are testing. (**Risk Based Testing**)
- Have mini meetings all the time
- Everyone on the team has a say

## 2. Help Write Solid Acceptance Tests

- Product Owner should have good idea as to minimal amount of functionality that each of the functions or features should have. Tester can work with the Product Owner to find that out
- Don't think of these as test cases but rather as charters
- Draw out of the Product Owner what they really need. Product Owner may not be thinking about the non-functional aspects. He/She may not be thinking of negative or edge cases. This will provide more clarity for Developer too
- Value cases/priority cases
- Developer should be looking at the Acceptance Criteria to also make sure that area is robust

## 3. Hold everyone "accountable" to Definition of Done

- Discuss how to get to Definition of Done?
- Make sure that Testing has input and that team is accountable
- Developers drive this often. Testers don't typically drive. Testers should be driving it though
- Calling out what levels of testing make it Done. Include automation
- Need reasonable Definition of Done. Robust Definition of Done
- Collaboration criteria can be a Definition of Done. (For example, criteria could say that team needs to review Test Plans by story)

## 4. Represent the Customer

- Stories are sometimes taken into a sprint when they aren't ready. This usually is not a good thing
- Need readiness criteria



- Great testers think about what the customer's problem is and how to solve it
- We can easily see what areas are probably going to be hit the most. Testing can align with that
- Does the end to end solution match the problem that we are trying to solve?
- Pareto concept/80/20 Rule: Handle the 20% of requirements that are used 80% of the time
- Need that testing voice
- Make sure we are on priority and on value
- Help Product Owner make sure that team is delivering the highest value features
- Developing Persona-Reinforce the UX team to develop Personas that can be used
- Working with the Product Owner, we need to take that sense of priority and make sure that is where we focus
- Don't go down the rabbit hole if it isn't important for this sprint

## 5. Ask Questions

- Testers ask a lot of questions. That is a good thing! Have you thought about this? Agile can be perceived as development centric. Bring that out. Start asking questions. Get in the game as testers and ask as many questions as you can... of yourself, the developer and the Product Owner
- Get to the root cause. What is the value? Why are we doing it? From a Risk Based Testing perspective, why do we have to test this?

## 6. What about the Cost of Quality?

- Quality is a team responsibility
- Testers looked to as overall champion of quality
- How mature is your definition of done?
- Is there some degree of pairing? What is the cost to that? It is producing one outcome from 2 people but the quality will go up tremendously
- Automation Investments: There is a cost to build but there will also be a return on investment
- Convince the Product Owner that if there is something extra that needs to be done, we need to be able to speak to the value of it



- Another place to look at the investment in quality is the backlog. Should be able to see value increasing over time. Quality should be increasing. Need to look at how we are handling risks
- Backlog should be more than a list of things to do. Look broadly. Are we covering regression properly?

## 7. What about the Cost of Testing?

- Quality Assurance is the prevention of a problem or defect
- Testing is detection of defects
- There is a synergy between Quality Assurance and Testing
- **Risk-based** - Some level of prioritization. More important in Agile because of limited amount of time. Have to be focused. What is highest risk?
- Don't focus much of effort to track coverage or time
- Think about how to do testing and then do the testing
- Include slack time for thinking & creativity. Outline what you want to do. Having time to imagine what system should do is important
- Testing needs to be balanced across quadrants
  - **Quadrant 1** - Automated: Unit Tests, Component Tests
  - **Quadrant 2** - Automated & Manual: Functional Tests (Story, Prototypes, Simulations)
  - **Quadrant 3** - Manual: Exploratory Tests, Usability Tests, User Acceptance Tests, Alpha/Beta Tests
  - **Quadrant 4** - Tools: Performance & Load Testing, Security Testing, "ility" Testing
- Get the right support from the teams
- Ask how much automation is being done
- Look at this as a strategy guide to have balanced testing across your application
- Tester role is to come in with frameworks
- Communicate to your team and do the right thing

## 8. The Backlog is a "plan" help focus it towards Release!

- You should see your backlog as a plan
- Good backlogs have opportunities for hardening, regression tests, milestones and key deliverables
- Agile needs regression testing
- Release train-Planning gives testers some hope that we can dig in on certain areas



## 9. Get to know your Product Owner

- Agile emphasizes relationships and trust. Get to know your Product Owner
- Talk about prioritization and goals
- Challenge them
- Become their partner
- As a tester, you have to get to know your Product Owner

## 10. Wrapping up....

- Testing has a role in collaborating with Product Owners
- Try some of the techniques that have been mentioned
- Collaborate toward delivering these results

