

AJI – Agile Journey Index

Product Owner Extensions

Author: Bob Galen
Version: 1.1
Last Updated: June 17, 2013



Introduction

[Bill Krebs](#) is an experienced Agile Coach and practitioner. He's worked at IBM, as an independent coach, and is now coaching at AllScripts, in Raleigh, NC. He is the sole proprietor of [Agile Dimensions](#), an agile training and coaching firm.

Bill has introduced the notion of an *Agile Maturity Index*. It's an agile maturity evaluation framework that can be used to assess and communicate team performance. It established a baseline of performance that coaches and teams can leverage to focus their continuous improvement efforts on.

This is a PDF Overview of the History of the AJI and the implementation:

<http://www.agiledimensions.com/blog/wp-content/uploads/2011/10/KrebsAgileJourneyIndex.pdf>

In my writing of Scrum Product Ownership AND in my coaching, I've found it useful to be able to "quantify" the maturity level of the Product Organization. It helps in communicating gaps and planning for improvement. It also helps in determining whether an organization should even be "going" agile in the first place—assessing if they are ready?

Bill is working on a AJI book as of February 2013. I believe his target for publication is early Q2 2013. I've developed this adjunct to the AJI because I like his framework and I felt the "program side" needed a bit more definition.

I hope you find it useful...

Contents

Introduction	2
Contents	3
Product Owner	4
User Stories	5
Product Backlog	6
Estimation	7
Valuation	8
Envisioning	9
Goal Setting.....	10
Sprint Reviews.....	11
Communication.....	12
Listening	13
Product Mentoring & COE	14
Release Planning & Road-mapping.....	15
Product Leadership	16
Supporting ATDD (Optional)	17
Product ORG Supporting Scrum @Scale (Optional).....	18
Product Leadership – Blank Evaluation Sheet	19

Product Owner

Product Owner	
1	There isn't a Product Owner OR someone is assuming the PO role amongst other roles they hold. A functional manager or Project Manager is holding the role.
3	There is a Product Owner, but they are essentially under trained and overloaded with more than one team. They don't have the time to invest in their role & team.
5	Clear Product Owner with the domain experience, organizational trust, and time to "feed their team(s)" well.
7	CSPO level training. Fully understands the stakeholder communication and "expectations management" aspects of the role. Balances well across the 4 Quadrants of the role and internal vs. external focus.
9	Product Owner writes acceptance tests; supports Readiness Criteria; helps the team to define Reference Stories. Fully engages the Project Management aspects of the role with the Chief Product Owner.

The essential entry price here is having a qualified, trained, business domain aware, decisive, well connected, trusted Product Owner.

They must also have the "time" to do all aspects of their job well OR have supplemental support to do that.

One great "test" of Product Owner effectiveness resides on how well the Sprint Reviews are conducted:

1. Do stakeholders attend? Are they engaged and excited? Does the PO manage expectations?
2. Is there an agenda, clear flow, and was the team prepared?
3. Does the work align with the Sprint Goal; does the Goal synergize with attendees?
4. Do they "connect the Dots" to past & future deliverables?
5. Does the Product Owner show ALL WORK (features, bugs, refactoring, infrastructure, etc.)

(See Sprint Review criteria)

Another critical aspect to the role is external communication and representation. This is with customers and stakeholders. It's also multi-directional. Ultimately defending the team and ensuring they're creating powerful solutions. (See Communication criteria)

- CSPO – Scrum Alliance, Certified Scrum Product Owner certification. A 2-day workshop focused on the role.
- Acceptance Tests – The acceptance or criteria part of the User Story. They help define and specify the story.
- Reference Stories – Are user stories that have been selected to serve as "references" for grooming. Typically they fit a specific type of work and the estimate (size) is clear. Usually, they've already been implemented.
- Readiness Criteria – the acceptance criteria or tests that are defined "on the back of" a user story.

User Stories

User Stories	
1	Solely waterfall requirements OR no agile requirements beyond simple Product Backlog Items (PBI's).
3	Leveraging User Stories; 3-C's, INVEST; typically over (+100%) or under (> 50%) written. Few to no acceptance criteria—struggle with Story consistency and effectively “looking ahead”.
5	Stories, on average, enter sprint execution at about 70% clarity; Groomed 3-4 times before execution. Spawned appropriate story Research Spikes along the way.
7	Strong investment in acceptance criteria. Considering ATDD & executable requirements. Broad distribution of story sizes for “Sprint packing” and theme support.
9	Reference stories are established. As estimation struggles; teams take ownership and recalibrate. Whole team view towards Story writing and maturation.

Requirement transformation is one of the most difficult transitions from Traditional (Waterfall & RUP) to Agile Methods. Here are a few of the challenges:

1. Writing less; and writing iteratively (Grooming)
 2. Parsing traditional requirement documents into “chunks”
 3. Distributing out the stories to:
 - a. Product Owner – features
 - b. QA – bugs, testing, and infrastructure; Acceptance Tests
 - c. Developers: refactoring, infrastructure, and Acceptance Tests
 - d. Architects & UX teams – technical look-ahead
 4. To a variety (Village concept) of team members
 5. Only entering the Sprint w/approx. 70% clarity; leaving the sprint with 100% clarity
 6. Dealing with ambiguity
 7. Realizing that you don't have to get EVERYTHING right the first time...it's iterative!
- PBI – Product Backlog Item; often a User Story, but can be “anything” related to the work to support a product or project.
 - 3 C's – Card + Confirmation + Conversation is an acronym and a reminder of the 3-parts of the each story. The Conversation is clearly the most important part.
 - INVEST – is an acronym reminder on important characteristics for every story: Independent, Negotiable, Valuable, Estimatable, Sized well, Testable.
 - Research Spike – a User Story focused on research, knowledge acquisition, learning, prototyping, and ultimately a set of well-crafted stories.

Product Backlog

Product Backlog	
1	We have none or a solely Product Owner driven list. Priority is solely envisions from a business perspective—none others matter.
3	Product Owner “owned”, prioritized, infrequently groomed—stories/PBI’s only visited once. Team isn’t aware of “future work”.
5	No overloaded priority, Estimated by the team, some Acceptance Tests – but lightweight and inconsistent, some spike’s; 5% grooming time.
7	Excellent Acceptance Tests, Examined 3-4x as PBI’s approach execution, appropriate levels of spikes (+10%), 10% grooming time.
9	Real-time collaboration on the Backlog; Everyone (Village) takes ownership—the team actively grooms daily.

Is a Product Backlog a simple, prioritized list or something else? The more mature view is towards something else. Mature teams connect their backlogs to higher level visions of the work.

I’d liken them to capturing or supporting:

- An agile WBS
- Workflow; dependencies and x-team exchanges
- Information Radiators – they are on walls
- Story Mapping – bringing the User into the flow of user stories
- Personas are aligned with the Backlog; see story-mapping
- Roman Pitchler’s notion of a Product Canvas is an example of effective visioning
- Testing, documentation, marketing, release dynamics represented
- Regulatory and other process-centric work represented

In my SPO book I talk about Future-Casting as being an important part of the Backlog. What I’m really implying is an effective implementation of Boulder – Rock – Pebble sizing when articulating Backlogs.

It’s also effectively sized from a release perspective:

- Current Release: Rocks & Pebbles; mostly pebbles
 - Next Release: Rocks; with clearly identified Spikes
 - Next Release + 1: Boulders & Rocks; mostly Boulders with some Spikes
- PBI – Product Backlog Item; often a User Story, but can be “anything” related to the work to support a product or project.

Estimation

Estimation	
1	None or Management-driven estimates. Scrum Masters hounding team on estimates and making commitments for the team.
3	Team-based estimation; individual skills focused; a lot of concern about 'actuals'. Or SME-driven estimates.
5	Planning poker with Fibonacci-based values; lots of debate on the #'s; wide variance. Less focus on estimates as guarantees or commitments.
7	Planning poker with high level abstractions (T-shirt sizes); less focus on the #'s, more focus on work throughput / velocity; moderate variance. Achieved truly "whole team" estimates.
9	Will ask for a Research Spike if they "don't know". Estimation focuses more on discussion, thematic packing of sprints; narrow variance. Well on the road to discovering that Predictable Flow is more important than Estimation.

A general flow here is for the Product Organization and the Product Owner to "stay out of" the sizing and estimation processes within the teams. It's a truly "slippery slope" for the Product Owner as a leader and team member. They can have direct and quite subtle / indirect influence.

Ask questions and seek to understand the estimates? Yes!
 Over challenge or influence them to be reduced in any way? No!

The Product Owner 'owns' the WHAT. The Team 'owns' the HOW and the HOW LONG.

Be accepting of User Story – Research Spike's as necessary and helpful; ensure they are surfaced in grooming and are deployed in an earlier sprint than implementation need.
 If the Product Owner doesn't "like" an estimate; their recourse is to reprioritize or re-scope the User Story.

I'm finding in my coaching that teams simplify their estimation as they mature. For example, moving from Fibonacci style numbers, or even more finely grained estimates, to much simpler systems. This help in transitioning from planning, to collaboration, swarming, and optimizing throughput.

Valuation

Valuation	
1	Simple prioritization of the Product Backlog; usually only considering software functionality.
3	A more nuanced implementation of the Product Backlog; considering technical debt, bugs, research, risk, etc. Team included in prioritization and valuation.
5	Leveraging Value Poker and other collaborative techniques to create a broader view to value. In the Enterprise or at-scale, Portfolio & Program level valuation occurs across product backlogs.
7	Release criteria surround ROI measures. Portfolio value measures help with team investments. Clear value-based decision making communication back to stakeholders.
9	Raw measurement of ROI and investment decisions in the true spirit of Lean Startup. Product organizational effectiveness is measured by real usage and ROI. Product pivots easily occur.

The focus here needs to be VALUE.

Initially value in Scrum is a simple model; driving it by prioritizing the Product Backlog. But over time, it should become a more nuanced endeavor.

First, it needs to move beyond simple “feature value”. It’s a distributed valuation of: features, bugs, infrastructure investment, refactoring, and documentation. Having the conviction of not going “to Zero” in any one category for sustained periods of time.

Being clear in measuring teams output in Velocity, but also value delivered.

Teams should be working in priority order...always. If a sprint “shatters”, the highest valued items should be delivered with high quality.

In the most mature teams, measuring ROI or Usage actuals and comparing them against Product Backlog priority assumptions (business cases).

In the at scale instances, where there are multiple stakeholder views to value, the Product Organization needs to do a great job of aggregating these views, coming out with a value model, and then clearly communicating the decision(s) and the WHY behind them back to stakeholders.

Envisioning

Envisioning	
1	We don't need to 'envision' our projects; simply "Diving In and Sprinting" is good enough.
3	Well-understood Mission & Vision for the project. A Meta-Backlog is constructed when appropriate, before driving multi-team projects. It connects to higher-level business roadmaps.
5	An Agile Project Charter is completed for non-trivial projects and multi-team projects. System metaphor-like architectural diagrams/maps are available, which guide the technical direction.
7	An MMP-like construct drives the release planning and forecasting. It's also used for Date vs. Scope negotiations. Story-Mapping is completed as part of overall release planning activity.
9	Re-chartering occurs to "Reset" expectations and to re-establish a shared vision for where the project is going. Backlog Grooming focuses on short, mid, and LONG term via Future-cast oriented grooming and investigation.

A big part of the envisioning process is leveraging Information Radiators of various sorts. Another key is leveraging a Chartering Process for starting off each of your project and releases well—connecting the dots for the team as to what is important and where is the value.

The less mature teams have a tendency to "direct" these artifacts towards the teams. As they mature though, there is the intention of creating shared goals and direction. This doesn't mean the business-side isn't leading, simply not dictating.

An Agile Charter generally focuses on:

1. Project Mission and Vision
 2. Minimal Marketable Product – "Just Enough" for success
 3. Budget; ROI goals, key measures
 4. Team structures: skills, alignment, roles, needs, org structures, etc.
 5. High level Backlog and Release plan
 6. Criteria, what does success "look like"
 7. Risks, constraints, dependencies
- Information Radiators – a reference from the Crystal Methodology. Placing meaningful charts, diagrams, or pictures up on walls that represent information important for the teams progress. Literally anything can be an effective radiator. Ultimately, the measure of effectiveness is: does the team pay attention to the information; does it influence their behaviors.
 - Story Maps are a reference to the technique created by Jeff Patton. The idea is to layout stories in a functional flow from the customers' usage perspective (the Backbone). Then to connect supporting functional stories to the backbone. Then to connect supporting infrastructural stories as well. The left to right flow is priority-driven and usage driven.

Goal Setting

Goal Setting	
1	Simple, singular level of Done-Ness or Definition of Done (DoD). Rarely adhere to all aspects of it—for example: “continuation work”. 4 Levels of Done: functional work done-ness.
3	Done-ness is actively included in team-based estimation. 4 Levels of Done: Acceptance Criteria, story sign-off.
5	Team leverages the Sprint Goal in making adjustment decisions (micro-adjustments) in the sprint. 4 Levels of Done: Sprint Goals
7	4 Levels of Done: adding Release Criteria; no compromise of goals to meet sprint commitments (goals over time). Holding the line on Quality. Develop MMP / MMF level goals that drive Release Planning & Criteria
9	Transparent goal attainment measures; perhaps the notion of goal escapes. Goals part of team and organizational DNA—hold to them w/o “enforcement”. Organization doesn’t try to “Force” teams to ignore their goals.

The key movement here is from more singular goals, usually focusing around some sporadic work done-ness and acceptance criteria, to a holistic focus across all four of the below levels.

4 Levels of Agile Goals

1. Functional Work Level: Developer Done-Ness; Tester Done-Ness; etc.
2. Story Level: Acceptance Criteria, Conditions of Acceptance, and mini UAT
3. Sprint Level: Active sprint goals that guide trade-off decisions AND align with Sprint Reviews
4. Release Level: MMP, Project Charter goals, Release Criteria established (PRE) release commitment.

Goals and Criteria should be clearly posted on a team-by-team basis AND across the organization. Talk about them within the teams, retrospectives, and sprint reviews. It is quite common to have two types: Team-Based Goals and Organizational Goals. Having both is quite healthy, as it gives each team some “say” in their goals.

Remember that goals have a cost AND the Product Organization (and the customer for that matter) needs to be willing to pay those costs. So there needs to be cost / benefit alignment across the goals.

- MMP – Minimal Marketable Product. You also see MMF (Feature) or Minimum Viable Product often referenced.

Sprint Reviews

Sprint Reviews	
1	Sprint Reviews sometimes happen at the end of a sprint. We try to show “everything” that was completed. Mostly information (PowerPoint) and very little “Working Software”.
3	“Working Software” dominates the review. The PO pulls together a focus for the review. Spotty attendance by “Key Stakeholders” w/appropriate ‘excuses’. Introduce the team, appreciations, and retrospective/feedback at the end.
5	Demonstrating more than just “Software”. Test automation; bug trending, refactoring, architecture & plans, challenges, adjustments, etc. Great avenue for transparency.
7	Compelling agendas tied to Sprint Goals. Pass versus Failure. Connect the dots or regular coming attractions. At scale; coordinated reviews across teams—tied to Release Plan.
9	Great feedback; engaged audience; they keep “Coming Back” for more. External teams leverage the information to their planning & execution readiness advantage. Begins to TRANSFORM the organization!

Let’s explain this progression in two scenarios:

The team minimally prepares for their demo. Essentially everyone “shows up” on Friday morning and goes through a PowerPoint presentation that details every nuance of the sprint. It’s boring and by-rrote. Oh by the way, there are no stakeholders in the room; only the team (team, Product Owner, Scrum Master) plus a few invited friends. No working software is demonstrated, but there is polite applause at the end. Then, on to the next sprint...

Another team prepares a script and does a quick dry-run. Everyone on the team has a role in the demo. The Product Owner has sent out the agenda at least a week in advance and has personally invited interested stakeholders—assuring they’re going to make it. The demo starts crisply on time and after a brief introduction, dives into showing working software of the stories support the sprint goal; the high value stuff. Feedback is received and recorded. Questions are addressed. The software works and the audience is engaged. The demo closes with a brief retrospective and attendees are invited back to the teams seating area for follow-up demos and Q&A. Oh, and there is sincere applause.

You want to stay away from the first and move towards the latter scenario...

Another realization to think about—the Sprint Review can be a part of your agile transformation strategy. You’re literally “walking the talk” as you review your efforts and practices to your ENTIRE organization. Don’t underestimate the impact of this transparency and the exposure!

Communication

Communication: Down, Laterally, Upward, and Outward	
1	The overall view is that the Team and the Scrum Master are the primary communicators; not the Product Owner(s). Another aspect of this is not understanding or trivializing the value of information radiators.
3	Starting to engage in external communication, putting up and maintaining team-based information radiators. Starting to “take charge” of external communication emanating from the team across product and project dimensions.
5	The bridge is moving from team communications to being a part of the overall Product Organization and merging team communication into organizational communication. For example—at a Release Level.
7	Takes an active view to sharing the right level of “team and backlog state” information with x-organizational stakeholders. Defends the team without being defensive. Balanced & transparent communications.
9	Active feedback gathering in Sprint Reviews. Actively communicate trade-offs, adjustments, and plan impacts in real-time during each sprint. Instead of Continuously Deliver (CD), involved in Continuous Communication (CC)—driven/inspired from the Product Organization.

The key movement here is...leveraging the Product Owner(s) and the Chief Product Owners capabilities of effectively communicating in a 360 degree fashion across the organization.

This might take the form of:

1. Collaborating for team and x-team Sprint Reviews that “shock and awe” your organization (see Sprint Reviews criteria).
 2. Constructing feature release notes for an upcoming release.
 3. Pulling customers into the team whenever possible to interact, provide feedback, and generally collaborate with THEIR team.
 4. Requesting that stakeholders attend Sprint Ceremonies to gather more information—reinforcing and supporting “Total Transparency”.
 5. Partnering with the Scrum Master to do an outstanding job of “radiating” team state from a Sprint and Release level.
 6. Participating in the Scrum of Scrums, or equivalent Project/PMO level construct, to integrate and communicate x-team implications and progress.
 7. Working with downstream partners (customer support, documentation, marketing & sales, etc.) to assure they know “coming attractions” and have a chance to provide feedback.
- Information Radiators – a reference from the Crystal Methodology. Placing meaningful charts, diagrams, or pictures up on walls that represent information important for the teams progress. Literally anything can be an effective radiator. Ultimately, the measure of effectiveness is: does the team pay attention to the information; does it influence their behaviors.

Listening

Listening; Gaining Feedback	
1	Consideration of team feedback; but essentially driven by the business-side.
3	Attend retrospectives. Out an about with customers; listening/observing value-based usage.
5	Active participant in retrospectives; willingly take on product-level impediments and adjustments.
7	Team suggestions for new features, innovation, refactoring, bug fixes, etc. surface in the Product Backlog.
9	Seek first to understand before responding. Open minded to alternatives. Attend Sprint Planning and truly listen to team reactions to complexity and level of difficulty.

The primary movement here is moving towards a 360 degree listening style as a Product Owner. In this case the primary 'customers' are:

1. Downward – towards the team
2. Laterally – towards your peers and cross-function organization
3. Upward – towards your senior leadership team
4. Outward – towards your customers

Usually two and three are easy for most Product Owners. Four can be challenging from a “getting the time” perspective AND aggregating what you’re hearing. Quite often, one is the area that gets the least of your ear. I’d argue that it should be the one that gets the most.

I’d add to this “trusting” your team; going with their ideas, trusting their assessments and judgments, and allowing the iterative nature of Scrum to take care of the uncertainty and risks.
Try things!

Product Mentoring & COE

Product Mentoring & COE (Center of Excellence)	
1	Every Product Owner is essentially on their own. They may or may not be “part of” the Product Organization.
3	Common standards for PBI’s or User Stories. Product Owners are in the Product Organization; they are taking “ownership” of the role. Training roadmap exists.
5	Product Owners work together in mentoring each other and driving consistency across related teams. For example: Backlog Grooming goals and standards. A job description exists.
7	Product Owners start acquiring training; CSPO or equivalent certifications. Beyond that, they become part of the agile community—reading, speaking, and contributing.
9	Product Owners begin to champion other aspects of the Agile COE. For example, ATDD standards and tools. Or product quality and improved up-front design.

The key movement here is from individualized Product Ownership to one that is group-based, mentor-driven, continuously learning, and driving towards lightweight standards.

Consistency from one Backlog to the other is important. As are story-writing, acceptance criteria, story valuation, and sprint goal/success standards.

There’s a general trend towards investing providing example artifacts and training.

By definition, there’s a transition from the “technology team” owning all aspects of the agile adoption/transformation, to the Product Organization leading their aspects.

- CSPO – Certified Scrum Product Owner, designated by the Scrum Alliance
- Agile COE – An organizational method for establishing, monitoring, and driving a set of common practices or standards across agile teams.
- PBI – Product Backlog Item, an element in the Product Backlog. Very typically composed of User Stories (mini-Use Cases). But can be virtually any independent requirement of the team.

Release Planning & Road-mapping

Release Planning	
1	Entry here is simply maintaining Product Backlogs that are disparate from team to team; so no Release Planning is occurring OR it's occurring solely within the Product organization.
3	Individual team release planning is occurring. This also implies a modicum of a Release Train model exists.
5	Cross-team (Product or Project level) Release Planning is occurring. Milestones for grooming the backlog exist so that the work is well-understood in Release -1.
7	Architecture, Business, and Design look-ahead Story activity is occurring as part of the pre-Release grooming.
9	Release planning is a “whole team” exercise prior to committing to a body of work. It focuses on x-team interaction, dependencies, and integration. It also becomes the communication tool of choice (baseline) for any negotiated deviation and for Scrum of Scrums reporting.

The key movement here is from a total lack of release planning, or “sprint at a time” execution, to more look-ahead based grooming. Look-ahead in this case would include:

1. Business Epics – qualifying the business value roadmap in advance of release planning.
2. Architecture Epics – qualifying the architectural implications and strategies prior to release planning. This would include prototyping at a team level.
3. Design Epics – more so from a UX perspective, this would include analysis and investigation, as well as, User Interface and Interaction design.

These epics are groomed so that the nature of the release is reasonably well understood, at a high level, before attempting release planning. If this is a greenfield effort, the organization may very well have wrapped these efforts into a Sprint #0.

There is a strong relationship between Release Planning and Project Chartering. In this case, charter elements need to be pre-established prior to the planning. However, commit to the project (the Charter) can only occur after the team completes their planning AND has the clarity and vision to COMMIT to the project.

- Release Train – a notion from Dean Leffingwell’s SAFe model.
- EPIC’s, in this case, these are High Level User Stories that encompass the business need (WHY), the architecture, and the design for the Release.

Product Leadership

Agile Experience at a Product Leadership Level	
1	Pre-Level A agile leadership experience.
3	Level A agile leadership experience within the Product Organization.
5	Chief Product Owner has Level B agile experience.
7	Chief Product Owner has Level B agile experience.
9	Chief Product Owner has Level C agile experience. And is connected across the organization to similarly experienced leadership counterparts. Finally, agile Product steering is moving towards organizational steering with active Product engagement.

The baseline here is my view towards “experience levels” in agile teams. In this case, we’re looking for significant, Product Organization agile leadership levels.

Level A: Basic agile reading / information, but no direct experience. Consider it SHU level experience, where coaching and external guidance would be helpful

Level B: Moderate information and some direct experience. Consider this journeyman experience though. Consider this HA level experience. When the going “gets tough” B-level folks frequently regress to waterfall or historic patterns.

Level C: Deep, ‘master’ level agile experience or RI experience. Organizational leaders exist who have successfully taken at least one organization forward to high performance agility. When the going “gets tough” these folks RECOMMIT to their agile principles and tactics.

Supporting ATDD (Optional)

Supporting ATDD	
1	Loose understanding of the Agile Testing Pyramid. Leveraging ATDD or BDD tooling and templates for some value-based Acceptance Criteria automation.
3	Test Automation strategy exists and aligns with ATP.
5	Have established the notion of a Value Regression and have the processes (Definition of Done and Stop the Line) in place to reinstitute lost value.
7	ATDD or BDD support are part of your Definition of Done and a lack of adherence (in construction or coverage levels) is considered a “Sprint Escape”.
9	ATDD is part of the Agile Development landscape. Stories are naturally articulated with ATDD/BDD nomenclature (Gherkin) in mind. ATDD or BDD test automation has become an important part of System & Regression test automation – extending beyond simple acceptance tests.

The first maturation check is whether the focus is on Automation or the Conversation. If it’s solely on leveraging ATDD tools & techniques to create “test automation and run tests”, then that’s a troubling sign.

The key movement here should be towards scenario and test supported conversations in the development of each User Story.

Another maturation factor is whether the Product Owner begins to leverage crafting Acceptance Tests in the languages and tools that the team is using for ATDD. Are they not simply supporting the approach, but are they embracing it?

- Agile Testing Pyramid was first articulated by Mike Cohn. References in Lisa Crispin and Janet Gregory’s: Agile Testing book.
- ATDD – Acceptance Test Driven Development
- BDD – Behavior Driven Development (often associated with Cucumber (open source framework) and Gherkin, Cucumber’s BDD language.

Product ORG Supporting Scrum @Scale (Optional)

Product ORG Supporting - Scrum @Scale	
1	Minimal involvement in the Scrum of Scrums; more so at a “point team” level. No release level planning nor road-mapping. Product Organization essentially views responsibility as providing Product Owners for teams and continuing other processes and tactics.
3	Product organization starts to organize behind the Scrum (or other) Agile team structure. Provide Active visualization of Road Map -> Meta-Backlog -> Team Backlog mappings and delivery tracking / results.
5	Chief Product Owner is fully engaged in the Scrum of Scrums OR conversely in the Road-mapping, Portfolio, and Agile Release Train aspects of SAFe. Partnered with the Chief Scrum Master and/or Agile Coaching groups towards effective
7	Steering of Product Activities within the agile transformation from within a Product Owner COE (or similar) structure. Portfolio-based value based work is driven across multiple teams. Trade-offs are ROI & Impact decision based.
9	Architecture look-ahead, design-look ahead, and other cross-cutting concerns (regulatory, technical writing, testing, etc.) are effectively considered and planned as part of the Road Map -> Backlog instantiation process. Cross team release planning with Project Chartering, Dependencies, Milestones, and Integration.

The baseline here is initial engagement in whatever scaling model is being tried or leveraged within the organization. Many Product Organizations fail to see the transformation they need to make towards forecasting, planning, and reacting to (adjusting) execution dynamics. It’s crucial that they take the majority position in guiding these from the traditional owner—the technical organization.

In the middle levels of maturity, there is activity along the 3 Tiers of Scaled Agile, that is:

1. Portfolio
2. Project
3. Team

Typically Product Organizations start scaling at a team level—supporting individual teams, but missing out on the “Big Picture”. As time moves on, they start accommodating the upper two tiers with equal engagement and passion.

- SAFE – Scaled Agile Framework; similar views include Scrum of Scrums and DAD – Disciplined Agile Delivery

Product Leadership – Blank Evaluation Sheet

Criteria	Notes	AJI - Level
1-Product Owner		
2-User Stories		
3-Product Backlog		
4-Estimation		
5-Valuation		
6-Envisioning		
7-Goal Setting		
8-Sprint Reviews		
9-Communications		
10-Listening		
11-Product Mentoring & COE		
12-Product Leadership		
13-Release Planning & Road-mapping		
14-Supporting ATDD (Optional)		
15-Product Supporting Scrum @Scale (Optional)		
OVERALL		