

Stacia Heimgartner Viscardi

In this issue of *Exhibition*, editor Ellen Snyder-Grenier talks with Stacia Viscardi to learn how her work on organizational thinking in the world of software development might help us think about developing exhibitions in new ways.

Q Ellen You work in the technology world as an Agile coach, Scrum master, and organizational transformation expert. For those who may not yet be familiar with Scrum/Agile project management, can you briefly describe it?

A Stacia Agile was created in 2001 as a result of meeting of 17 leaders in the software industry. The outcome of this meeting was something called the “Manifesto for Agile Software Development” (www.agilemanifesto.org), which captured their modern thinking and means by which they had been helping their own respective clients deliver software at least a decade prior:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and Interactions

over processes and tools

Working Software

over comprehensive documentation

Customer Collaboration

over contract negotiation

Responding To Change

over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

The Agile Manifesto has 12 principles that enable agility, from working in iterations, to providing the right environment, to maximizing value. It’s important for anyone new to Agile to remember that it’s not



Stacia Viscardi is a Business Agility coach, Certified Scrum Trainer, and organizational transformation guide, devoted to creating energized and excited teams that delight their customers. She founded AgileEvolution in 2006 and has helped companies like Cisco Systems, Martha Stewart Living, Primavera, DoubleClick, Google, the *Washington Post*, and many others find their way to agility. Coauthor of *The Software Project Manager’s Bridge to Agility*, and author of the recent *Professional ScrumMaster’s Handbook*, Stacia has taught and coached the agile and lean mindset in 22 countries.



a methodology. There are, however, various Agile methods that help teams and organizations achieve agility, such as Scrum, XP, DSDM, Crystal, and Feature-Driven Development.

Scrum, created by Ken Schwaber and Jeff Sutherland (who were both at the Utah meeting in 2001), is an agile project-management framework that utilizes empiricism as a means to an end. Scrum teams are small, cross-functional, self-organizing, and work directly with the customer and/or end users to deliver the desired product. Since the future cannot be predicted, and since customers and users often don't know what they want until they see it, Scrum teams work in sprints – one- to four-week timeboxes within which the team delivers a working/tangible increment of the overall solution so that the customer/end users may give feedback. The desired product emerges after a number of sprints based on tight feedback cycles between the team and the customer/end user. Finally, teams meet at the end of every sprint in what's called a “retrospective,” wherein they discuss process and team improvements to carry forward into future sprints. In this way, impediments are resolved and progress doesn't stagnate.

You can read more about Scrum at the ScrumAlliance (www.ScrumAlliance.org).

Q Ellen How did Agile project development come to be? What were the challenges that gave birth to it?

A Stacia Agile's roots are in the technology world. Until very recently, a majority of technology projects were delivered by the “waterfall process.” Waterfall projects were so named because one silo of experts would hand off a big batch of work to another silo of experts; when this was visually represented in a project plan – called a Gantt chart – it looked like a waterfall (see graphic here: <https://blog.codinghorror.com/microsoft-project-and-the-gantt-waterfall/>).

The perils of waterfall were/are many, but here are three for your consideration:

1. It is impossible to know the future. Projects were traditionally budgeted up front, which pressured people to think of everything up front, which meant that projects entered into long analysis and design phases and went through many gates by which to pass approval into execution. When a project passed approval gates, it was accompanied by a project plan that included promised scope, timeframes, and costs, detailed hourly tasks, and resources for each of the tasks, among other things. Common wisdom is that we know more about a project as more time passes, yet waterfall forced most of the decisions to be made up front, when the least was known. Up-front decision making like this is laborious, expensive, and flawed, in part due to natural emergence over time. Perhaps worst of all, project teams would be forced into “death marches” in order to deliver these unrealistic, flawed plans, leading to human burnout and buggy systems. Agile methods like Scrum encourage teams to break down initiatives into iterations, and to use these iterations to inspect and adapt progress to a goal alongside the customer. (Check out Cynefin framework for more about decision making in ordered and unordered contexts: www.infoq.com/articles/cynefin-introduction.)

Additionally, granular estimates are given by the silos in silo – they don't talk to each other. Business analysts, for example, make a bunch of assumptions about a work package that varies from the developers' assumptions, which varies from the testers'. When the project is underway, the developers hand code off to the testers, and it varies (sometimes greatly) from the specifications written by the analysts. The testers had written their test cases, in silo, based on the analysts' specs. This results in bug-laden code that enters a vicious find/fix/validate cycle, one silo bouncing code

to another like an endless game of ping-pong. Finger-pointing ensues.

2. Waterfall, or phase-based project plans, did not have value at their core; rather, project managers drove silos of people to complete phases. Step into any traditional project status meeting for five minutes and you'd hear, "Phase 1 is 25% complete. Phase 2 not begun." In a technology project this sounds like, "Analysis phase 100% complete. Code 50% complete. Test cases underway." While, generally speaking, phases are a great way to break down projects into smaller manageable deliverables, Agile has us think about what constitutes a phase. Instead of silo phases, let's take a system and start building it out with the customer/end user in mind, *feature phases* if you will, or *product increments*, as Scrum calls them. For example, instead of building out an entire database up front for features we cannot even imagine yet, let's build out a small increment of the entire system feature-by-feature and have the customer/end user look at them every sprint and give us feedback. So, in an Agile project review, the language is more likely to be user/customer-centric: "Check out the login feature we built for you. Does it meet your needs? We learned that social media login is really popular with users; should we add that to our backlog?" Agile teams also embed testing within the team as a way to ensure what's built is built properly. Since a defect is up to 100 times more expensive to fix after the fact, let's find and fix things as we go instead.

3. People. I've talked to many of the Agile Manifesto signatories and a main undercurrent of why they created such an important movement was simply – and most importantly – people. I remember Ken Schwaber telling me once that Scrum was meant to make software development a profession. While I was not involved with the creation of the Agile Manifesto, I've personally witnessed people in traditionally

managed projects being told what their estimates are, disrespected, treated as "resources" – as cogs in a wheel – burned out, no work/life balance, working weekends, not seeing their children or spouses enough, threatened with a bad performance review or perhaps being fired – all in order to meet someone else's ultimately unrealistic commitment in a project plan. This was hugely unfair, and demoralizing. While people suffered in this system, so did their respective companies! If people are treated as cogs, and stop voicing ideas for improvement for the sake of meeting deadlines, the innovation of the company stagnates. People are just too busy and not motivated to think creatively about improvement. The people doing the work usually have the best ideas about how to improve the work. This is not a new concept – it's even called "Genchi Genbutsu," or "Go and see for yourself" in the Toyota Production System (check out more about this concept at <https://blog.toyota.co.uk/genchi-genbutsu>).

With technology moving faster than ever, with the ever-increasing mobility and autonomy of today's knowledge worker, with customers who expect unrivaled experiences, Agile couldn't have come at a better time. Regardless of the other pitfalls, waterfall or phase-based thinking is simply too slow for today's needs.

Q Ellen You bring agility to companies and their processes. What are the benefits of being agile?

A Stacia Agile gets everyone on the same page about value. For example, in a siloed company, sales and marketing understand what's valuable to a customer/end user, but the development "resources" do not. They're often many steps removed. Agile gets developers and their cross-functional teams *in front of* the customer/end users. Now that's a real dialogue.

When leaders engage their people in an invitational, open way, Agile can be a great motivator. In such

an environment, Agile teams get to set their own commitments during a sprint or iteration. They get to talk to customers/end users. They get to reflect and make their own improvements. They're more in charge of their direction than ever. Free lunches and quiet rooms are wonderful job perks, but autonomy, trust and the freedom to be creative are the real incentives to stay. The modern knowledge worker needs this context in which to be their very best. Agile, then, becomes a way to motivate, retain, and attract top talent.

Agile makes the delivery of value crystal clear. Every iteration ends with a review of what was delivered; all key stakeholders, customers/end users are invited to give feedback on the functionality completed for that timebox.

When customers and end users become part of the ongoing iterative dialogue, the team is able to rapidly fold feedback into the next iteration, or whenever it is appropriate to do so.

Q Ellen You write that you “believe in the power of an empirical process and a self-organizing, cross-functional team.” Generally speaking, what is involved in the process to transform a team into this new model?

A Stacia It's important to remember that transitioning to agile ways is a move from efficiency-based to value-driven work. A shift from manager-led workers to self-organizing individuals and teams. A movement away from projects and into outcomes. These changes are not trivial and often take years to even partially realize, due in part to the requisite growth in the complexity of the mind of the collective organization. A transition of this magnitude is more likely to succeed when leadership approaches it with an open, invitational approach instead of a forced, imposed approach. Read more about Open Space Agility at <http://openspaceagility.com>.

I believe that leadership's role is first to invite their people to consider change, and then set the value direction and subsequent stage for teams to deliver at their very best. Delivering at best does not mean hitting every deadline perfectly; rather, it means delivering with quality, negotiating scope/time/cost, responding to change as it emerges, growing and learning as individuals, working together to create the best solutions, and removing impediments to delivery. For a team to do its best, it needs a value goal, the right environment/safety, support, autonomy, focus, and nurture. I find that leadership usually feels they are doing this already, but when we dig in they begin to realize otherwise. I readily recognize that people are doing their best. Yet it's my job to teach another way.

My process is straightforward: understand the overarching transformation goal; work with leadership and teams to engage in open dialogue to shape the evolution together, in an agile way; educate and coach teams; create sticky knowledge; help the organization create a master impediment backlog and drop the fear of doing so; inspect and adapt. Here are some of the questions I use to get started.

Goal: Do teams regularly meet with their customers and/or end users to get first-hand feedback on what they've built? Do product managers keep roadmaps up to date and regularly communicate them, and do they see roadmaps as directional instead of predictive? How proficient is the organization at understanding the impact of the last release, and how to pivot as a result? *If we zoom out, is there a transformation goal? In other words, can leadership clearly state why transforming to Agile thinking and principles is a desirable or important thing to do?*

Environment: Do teams have a workplace conducive to collaboration? Are teams collocated, or are they in remote locations of the world? Can they regularly

engage at a whiteboard to exchange ideas? Do they have the technical infrastructure and tools to release quality product increments at the appropriate cadence? Are they safe to explore technical solutions, experiment a bit, to learn the best way to do things; in other words, is the environment safe for failure and the resulting learning? *Have teams been invited to shape what the transformation looks like?*

Autonomy: Are teams able to make most decisions for themselves? Does the team have ideas to make themselves more autonomous? Perhaps teams are based on silos, vs. cross-functional or feature team. Are there ideas for improvement in the supply chain to reduce teams' wait time? Do we have the right contract models so that our vendors may also work in autonomous, agile ways? *Do teams maintain an impediment backlog that they routinely review with leaders as they seek to improve?*

Q Ellen Your company provides training globally to, among others, new product development professionals. If we think of a museum exhibition as a product and visitors as customers, how might using a Scrum framework help exhibition practitioners create better products that are better suited for their customers?

A Stacia Silos are a carryover of Frederick Taylor's *Principles of Scientific Management*, which he published in 1911. Taylor's practice is efficiency-based, which was important in its time, and I do not intend to take away from that. However, in a creative, rapidly-moving space, while we do want to remain efficient at what we do, we need to prioritize outcomes above attaining 100% efficiency. Consider a cross-functional Scrum team for a new exhibition, made up of curators, educators, developers, display/staging staff, and visitors, among others, and that team stays together from beginning to end. Their goal is to deliver an exhibition using sprints, let's say two-week timeboxes, and each sprint must result in something the visitor

can see, experience, and give feedback about. I highly suspect that they would rapidly learn from each other and make quick adaptations as they deliver the exhibition in this value-based way. I have made this sound easy in this paragraph, but I suspect it would not be. There would be particular challenges around the existing budgeting and fundraising model. It's also challenging for new teams to step outside their existing processes to think differently about delivery. Perhaps, for example, an early exhibition team sprint could end with three different themes to present to a sample visitor group with the purpose of getting their feedback on how best to proceed. Always try to bring the visitor front and center.

Exhibition teams could also use iterations to test what they've done. Perhaps the display cases and lighting might be greatly influenced by the artifacts on display. Lighting that works well for pottery may damage paintings. Until we see glass cases under the building lighting, we may not realize they produce a glare that makes it difficult for the visitor to clearly make out the details of the artifact. If spell-checking the label copy is done late, and hurriedly, perhaps misspellings make it out into the world. It's quite possible that the actual artifacts curated don't match what was in the early fundraising materials. Working in an integrated, cross-functional team with a visitor/customer and testing frame of mind helps us find and fix these issues early.

Another aspect or benefit of agile and lean that could benefit exhibition teams is cross-learning. When we bring together a cross-functional team whose goal is to deliver value above sub-optimization, they find novel ways to learn from each other. I've seen database developers learn user-interface design. Testers learn to code. And vice versa. Sure, this is not efficient. It costs to do so. Yet, consider the bottlenecks in your exhibition delivery system - the chokepoints wherein one group gets backlogged and cannot keep up with

the rest of the system. Bottlenecks are expensive as they create upstream and downstream wait time, among other issues. They also provide key areas to encourage cross-learning. Over time, people are more motivated if they learn more and the team moves better together as a unit in the long run. Would you rather spend money to work within and around existing bottlenecks, or build knowledge within your group to create better flow? Which has the better return on investment?

Agile is a people-oriented, value-driven, adaptive system of thinking that allows for emergence of ideas based on learning through delivery. As a museum holds closely our history and precious artifacts – each the result of the emergence of *individuals and their interactions* throughout time – I cannot imagine a better way than Agile to bring these objects and stories to light.



originstudios



EXHIBIT DESIGN

ORIGINSTUDIOS.COM

GRAPHIC DESIGN | CONTENT DEVELOPMENT | INTERPRETIVE PLANNING



DESIGNED & BUILT ON PROVEN
PROCESS, HELPING YOU FOCUS

217.893.4874

info@taylorstudios.com

www.taylorstudios.com