



No Estimates at Scale in the US Federal Government

ADAM PARKER, *Excella*

My “No Estimates” report tells the story of an empowered Agile product group of three Scrum teams who stopped using story points to estimate their work in the fast-moving days following a troubled production release. In adapting to the emerging demands, the group also made the decision to switch to Kanban and focused their attention on cycle time and throughput to manage their work. This story shows how empowered teams with a true culture of experimentation can navigate real-world challenges.

1. INTRODUCTION

What if I told you that my most agile experience happened in the US federal government? In this report, I will explain how three agile teams working as contractors for the U.S. federal government decided to embark on a NoEstimates journey and deal with scale. I feel this story is about more than NoEstimates or transitioning from Scrum to Kanban (and back) or even scaling. It’s a story about how this three-team product group managed to adapt to change while living the Agile Principles as best we could. While not perfect, this product group is the truest expression of Agile that I have ever experienced. I feel compelled to share this story as a celebration.

2. BACKGROUND

2.1 My Involvement

I first heard about NoEstimates [Zuill] in the fall of 2017 on Twitter while researching new ideas and thought leaders. Go ahead and search the Internet or social media for #NoEstimates. You will likely find many passionate, if misinformed notions of what NoEstimates advocates are trying to achieve. The premise, in short, is that teams feel that estimating work is a low value activity. On EVE-MOD, this was exactly the case, though in E-Verify Modernization’s (EVE-MOD) situation was specifically triggered regarding story points. Like many agile teams, Product Backlog Refinement (Refinement) had become more of a pointing exercise and less of a meaningful discourse among the team representatives and the Product Owner.

2.2 The Product Group

The U.S. Citizenship and Immigration Services (USCIS) is an agency within the U.S. Department of Homeland Security (DHS). E-Verify, initially launched in 1996, is a free, web-based USCIS system through which employers electronically confirm the employment eligibility of their employees. In 2016, USCIS kicked off a modernization program to escape the constraints, dependencies, and fixed costs of legacy platforms and embrace the delivery speed, scalability, and stability of a cloud-based architecture. This experience report begins two years into the EVE-MOD journey.

EVE-MOD operated in a Large-Scale Scrum (LeSS) styled framework [LeSS1]. As feature teams [LeSS2], each EVE-MOD team had a stable team composition—cross-functional and fully dedicated to the product and capable of implementing an entire customer-centric feature. At a minimum, each team included a dedicated Scrum Master, business analyst, Cloud/DevOps engineer, and several developers. Notably, these teams did not include testers or quality engineers, but instead developers wrote and maintained automated test suites which were executed in our Continuous Integration/Continuous Deployment (CI/CD) pipelines. EVE-MOD invested the time to build and maintain a stable CI/CD pipelines that allowed the teams to deploy code from commit to production in 20-30 minutes over ten times per day-consistent with high-performing teams as outlined by the DevOps Research and Assessment reports [DORA].

Explaining agility at scale can be complex. In this report, “at scale” simply refers to three feature teams acting as a product group by working off a single product backlog to achieve a common purpose. Starting with a single Scrum team in March 2016, EVE-MOD waited nearly one year before scaling up to two teams in May

2017 and, finally, scaling up to three teams in fall 2018. Whether scaling up, as EVE-MOD did, or scaling out, as seen in many Agile transformations, a critical success criterion is achieving goals with a single, healthy team before adding additional teams [MSA]. Until that condition is met, do not scale. From my perspective, running NoEstimates for 55 weeks across three teams is a success at scale.

Our journey hit some milestones that are only obvious in retrospect, as we managed late-breaking requirements, integration pitfalls, visualizing dark work, failing at Sprint commitments, and unproductive Refinements. This may seem like a litany of issues and failures, but each step was a successful opportunity to adjust and improve in a changing landscape. This mindset was echoed by much of the team, as we did not have an expectation of best outcome, but instead incrementally inspected our situation and attempted to adapt to it in the best way we could. This led to a series of steps that took us from standard Scrum to NoEstimates Kanban.

3. A YEAR WITHOUT ESTIMATES

3.1 January 2018—A Large Release Approaches

The plan was to integrate our modernized case processing application with the legacy platform, upstream systems providing data for decision-making, and the downstream systems to manually resolve difficult cases. Each of these systems—five in total—was maintained by another product group. While EVE-MOD's initial work appeared on track, late-breaking requirements to also allow for web services and the early integration testing revealed a slew of Application Programming Interface (API) communication issues between the legacy, EVE-MOD, and our partner applications. Our developers were asked to take point troubleshooting with the legacy and partner systems integrations.

Taking the lead grew to such an extent that one of my fellow Scrum Masters created a simple Kanban-style board to track this “dark” work—meaning it was not visualized on our sprint work or product backlog. The integration board had a swim lane for each partner system. One or more EVE-MOD members were aligned as points-of-contact for each partner system and had regular meetings with them to understand newly discovered challenges with the integration work. We held a standup in front of that board each morning—making it the third meeting of the morning after team level standups and then a Scrum-of-Scrums.

The second board and third standup for integration work were obvious anti-patterns to me; being the newest addition to the team, I held my initial observations. I talked privately with team members over the next several days to learn how and why they had gone this direction. I had already learned in my coaching career that I should ask more questions and make fewer judgements, or worse flippant remarks, before arriving at a conclusion. They too knew the situation was not right but believed that visualizing dark work was less bad than missing commitments and attending a multitude of meetings.

Our client delayed the launch date by a month. Then delayed another month. Each time, we knew the new date was optimistic. EVE-MOD was ready but certain partner teams continued to struggle. We hosted them in our offices; we worked together. Progress was being made but the next Go-Live date was approaching, and one or two integrating systems were not ready and so more and more of our group's time was taken up in support of our partners. By mid-March 2018, our sprint work was being overtaken by this “other” work. With the needs of our partner systems arriving continuously, conducting Sprint Planning, as had been done for two years, no longer made sense; we had to make a change. Fortunately, experimentation was one of EVE-MOD's five values—Empathy, Empowerment, Experimentation, Enthusiasm, and Execution.

3.2 April 5th, 2018—The Kanban Workshop

The proposed experiment was to move from Scrum to Kanban. While not everyone was convinced switching to Kanban was the right decision, we agreed that some decisions needed to be made to stabilize our work process. We proposed to our client a plan to stop work for a single day so that we could take the whole product group (30 people) offsite to conduct a one-day workshop. At 11 a.m. on Thursday April 5th, I kicked off the session and set the expectations for the day's workshop: a) a decision on whether we were formally switch from Scrum to Kanban, b) design (Kanban) or refactor (Scrum) our product-level operating model going-forward.

We spent the first hour of the workshop educating the team on the Kanban method. One of our developers volunteered to help with the Kanban training. Dane Weber was an experienced Scrum Master who had recently completed KMP training and was six months into a two-year foray into a developer role, which he recounted in his 2019 experience report “Undercover Scrum Master” [Weber]. Dane distilled sixteen hours of Kanban Management Professional training into 60 minutes. Using a simple Trello board, he moved each topic's card across the board as we moved through the topics including work-in-process (WIP) limits, policies, flow metrics, and more. The EVE-MOD team members then engaged in a healthy dialogue on the topic, as usual with

this group. In the second portion, we mixed the 30 team members into three breakout groups to design a Kanban board for EVE-MOD. While we fully planned to continue operating as three feature teams organized to a single product backlog, we wanted each team member to have an equal voice in the decision. The groups were given a fixed amount of time to develop their board design and policies before presenting and answering questions on their design. After discussing the team boards, the larger group worked to collectively design the EVE-MOD Product Board.

At the conclusion of the workshop, several items remained unresolved, but the group felt that we had a foundation to build upon. In addition to a physical board, we developed a plan to guide our work starting the next day and going forward. We knew taking a day for the workshop was an important investment, but it was a luxury, but the work could not wait for us to design a perfect system. We took the time to write down five rules to guide team members' actions that next day:

1. If you are working on a task, make sure it's on the Product Board. Ensure BAs are aware of it.
2. Getting things in the correct place in JIRA and on the Product Board.
3. Prioritization meeting planned for Friday 3 p.m. to make sure the correct work is entering our system.
4. Communicate with BAs if you need work before the Prioritization meeting.
5. Get a rough idea of the meetings for next week. We will adjust as we learn.

Immediately after the workshop, several members returned to our office across the street from our firm's headquarters to build the physical Kanban board with blue painters' tape and stickies. The integration board was taken down and the associated standup was removed from our calendar of events. The daily Scrum-of-Scrum was suspended until mid-May 2018. EVE-MOD hit the ground running with a standup the next morning.

3.3 Making Kanban work

The following week, we briefed EVE-MOD's Product Owner and her deputies on the decisions made in the workshop and the remaining open action items. They and our IT Project Manager had reasonable questions about what would change for them as EVE-MOD switched to Kanban. They too understood that operating in traditional Scrum sprints was not serving us, so they were amenable to changes that would stabilize the flow of work and reign in the "dark" work. We all accepted that some questions, like work estimation, would come after the workshop as we the group gained experience in our Kanban implementation.

Eleven unresolved topics remained post-workshop including Work-in-Progress (WIP) limits, Definition of Done, Meetings and Cadence, Chore Rotations, Workflow Policies, and more. EVE-MOD members then volunteered to work on these topics; we called these working groups "committees." In the days after the workshop, each committee was to refine or develop proposals to address the topic and present the proposal for discussion and approval by the EVE-MOD membership. Most committees presented their proposals and got EVE-MOD approval within the five days after the workshop.

In many respects, the EVE-MOD "day-to-day" did not change much after this transition. During the workshop, we made the decision to make the minimum necessary changes to our ceremony schedule—consistent with our strategy of making impactful changes with minimal disruptions to our otherwise healthy system. We maintained our two-week cadence both within EVE-MOD and externally with our client and partners. We continued to speak in sprint terms when describing delivered work and forecasting future efforts to our PO and stakeholders. We continued to hold retrospectives—both at the team and product levels. We held sprint reviews including product demos on the same sprint cadence. We made a concerted effort to make the changes nearly invisible to those outside EVE-MOD—few sprint review attendees, who were not part of the Product Owner's team knew any changes had occurred except that work started flowing again.

Some ceremonies did need to change and sprint planning was the most obvious victim of the transition. As post-launch activities settled, a second refinement was introduced during the two-week cadence so that we could appropriately refine the product backlog and have refined work ready for the teams. And, as I will outline later, we ended up making many changes to Refinement to accommodate NoEstimates and encourage the ceremony to focus on meaningful conversations instead box checking against our Definition of Done.

Despite reservations by technical leads, the Product Owner felt pressured to move ahead with the case processing release on April 23, 2018. Having delayed the release multiple times, USCIS leadership was itching to launch and had been convinced that the remaining issues could be addressed after go-live. EVE-MOD marched into that weekend launch knowing there was going to be some damage control. The launch did experience issues resulting in an avalanche of customer issues. EVE-MOD led the efforts to stabilize E-Verify which lasted several weeks. While the teams worked to stabilize the system and downstream operations, our

Product Owner team and project manager were exceptionally busy fielding questions and concerns from E-Verify users, and USCIS leaders.

3.4 Experimenting with Estimating and Refinement

One of the unanswered topics in our Kanban workshop was how to handle the pointing work. EVE-MOD had been dissatisfied with Refinement ceremony since I joined in January. Things were moving so fast in late March and April that the group did not stop to estimate our tickets with story points. As issues and work requests continued to steam into our backlog, we resolved to track velocity to the best of our ability. In the early days, unrefined tickets were still getting started, often without estimates, so we made assessments while in-flight or even after the work was complete. While it was not our ideal, it was our reality for several weeks as the Pointing committee shepherded ideas into experiments.

As we lived in Kanban, EVE-MOD learned that keeping the On Deck column populated with Ready work was a challenge. EVE-MOD's Scrum Masters investigated how quickly Product Backlog Items (PBI) were leaving On Deck. In doing that research, one Scrum Master identified that EVE-MOD's historical average story points per PBI for the first fifteen months was slightly above 3 story points per PBI (3.13 story points to be precise). It was at this time that I started to realize that a NoEstimates approach might work given this data and the fact that all three EVE-MOD feature teams already were tracking average cycle time.

I conferred with EVE-MOD's account leadership to determine if story points were required per our contract. While our contract did not specify story points, our client indicated their preference to have a velocity measure that could be used for historical and planning purposes.

The Pointing committee, led by developer Joe Hunt, presented their first proposal on May 3rd. In this first experiment, the group agreed to use t-shirt sizing: small (S), medium (M), large (L), and extra-large (XL) instead of our modified Fibonacci sequence (1, 3, 5, 8, 13). Prior to my joining, EVE-MOD had decided that no work was really 0 story points and had eliminated the 2-point option to force a decision between 1 or 3 story points. Staying consistent with EVE-MOD story point practices, XL tickets should be split into smaller, vertically sliced tickets in the same manner that we had handled 13-point tickets. While t-shirt sizing limited the options, we agreed to keep using it until we had a better solution.

We still needed a way to collect those story point estimates from the participants of our in-person Refinement sessions. We also wanted to collect t-shirt sizes from Refinement participants (two representatives per feature team) in a way that had low-friction (which is why we didn't use a technology solution like a mobile app). Since no one had a good idea, I presented an idea about voting t-shirt sizes with hand signals: one fist for small, two fists for medium, one hand wrapped around the opposite fist for large, and two hands on either side of the head (a la stop the madness). I hoped the group would dislike the hand signals and be inspired to find a better solution. After a single refinement session with t-shirt sizing with hand signals, EVE-MOD and the Pointing committee were inspired to find a better framework for refinement. From my perspective, their rejection of the hand signals was a success and led away from t-shirt sizing toward a bolder experiment.

3.5 June 20, 2018—The No Pointing experiment

The Pointing committee presented its second proposal on June 6th. While the group did not approve the plan, the Pointing committee took the feedback, talked with more people, and returned a week later with adjustments. Joe Hunt posted the new proposal in EVE-MOD's main Slack channel on Wednesday, June 13th.

The "Pointing Experiment" sought to address several issues with the process. T-shirt sizing was not helpful to us or our clients in determining EVE-MOD's velocity and did not allow us to forecast with confidence. The plan called for removing story points completely and to conduct our planning and forecasting based on average throughput and cycle time per two-week cadence. The belief was that given our average size and robust internal metrics, we could successfully anticipate throughput to an appropriate level of accuracy via PBI rather than via points.

Important to reporting and forecasting, the proposal addressed velocity by leveraging EVE-MOD's 15-month historical average of 3.13 story points. We would take the number of PBIs that EVE-MOD completed during our two-week cadence, effectively our throughput, and multiply by 3. If 10 PBIs were completed in those two weeks, we would report a velocity of 30 (10 x 3) story points. While this approach was not fine grained, it held to our focus on system flow. Key to this experiment was the introduction of a new ticket discussion framework, which focused on replacing of t-shirt sizing or story points with improved discussion on the goals, risks, and other key attributes of the tickets. These refinement sessions would occur once every week. This approach would, ideally, refocus our conversations on the shared understanding of the work and being "right-sized" in general rather than the specific size of any individual piece of work.

Discussion cards were intended to provide a guide for navigating refinement conversations for both the participants as well as the Scrum Master facilitating the ceremony. Refinement participants could “play” any discussion card at any time during a ticket’s discussion. Multiple discussion cards could be played together; however, the “Ready for On Deck” card must be played alone. Mark Koenig, one of our EVE-MOD Scrum Masters, purchased several sets of colored foam cards at a discount store in the days prior to the first Refinement in this experiment.

| Discussion Card | Description | Card color |
|-------------------|--|------------|
| Too Big | The ticket can be broken down into two or more smaller tickets that can be developed, tested, and reviewed, independently. (Though one might be dependent on the other). | Red |
| Needs Comment | This card is used to indicate that the ticket needs a comment. The process of discussing the ticket includes recording a high-level list of subtasks required to complete a story. For example, 1) Create a button, 2) Call API, 3) Get response and update page state. Comments could include subtasks or risks or notes to contact an integration partner. | Blue |
| Don't Understand | This card indicates that the Acceptance Criteria or a subtask is unclear. Use this card when you don't know how to do the story. | Orange |
| Ready for On Deck | This card means that the story is appropriately sized with all the necessary comments, and that the Acceptance Criteria is clear and complete. A ticket is ready to be put On Deck when all developers play this card. | Green |

Figure 1. Original definitions of discussion cards posted to Slack on June 13th, 2018.

EVE-MOD approved this plan quickly and agreed that it would start with the next sprint that began on June 20, 2018. During the time when we would have held sprint planning, we held the first Refinement without estimating work in either story points or t-shirt sizes. In Figure 3, you can see my handwriting on the whiteboard. On the left I outlined how the group should use each of the colored foam discussion cards. We later made a poster that we brought into the conference room for Refinements. On the right side of the whiteboard, I have listed out the six team representatives, two per team, who were each given a set of discussion cards to play. The column of four-digit numbers represented the six JIRA tickets that were being reviewed in this session. As I facilitated the first Refinement of our experiment, I used different colored markers to add notes beside each JIRA ticket number. As seen in Figure 3, the four tickets with green boxes were approved by all six team representatives. Two tickets were deemed not ready and were sent back for additional refinement.



Figure 2. One set of the foam discussion cards

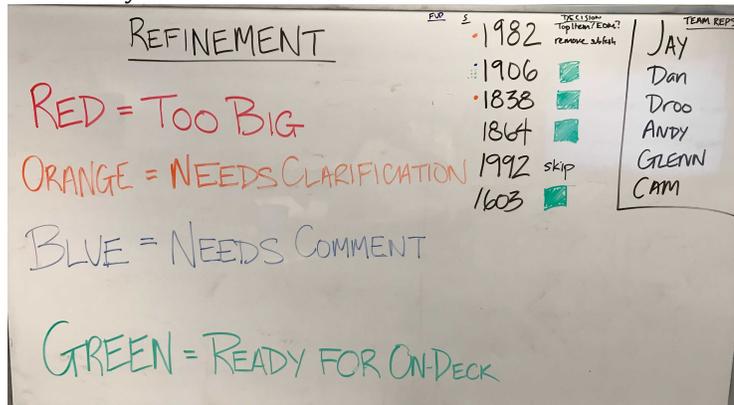


Figure 3. Whiteboard during first Refinement without estimating (6/20/2018)

3.6 NoEstimates Works Great

I have detailed the factors that led us to NoEstimates and later I will cover what prompted us to stop. Now, let us delve into why this a successful scaled adoption. Initially, we did not know what our goal throughput would be during our two-week long “sprints”. EVE-MODs teams had long tracked their trailing 3- and 6-week average cycle time (in units of days) and relied upon these measures to self-manage their flow of work during our NoEstimates year. Historically, EVE-MOD teams targeted a cycle times below 5 days and specifically less than 3 days in the Development column.

In Figure 4, an image of a physical information radiator for the Big Sillies in October 2018. The 6-week trailing average was 4.1 days from exiting the On Deck column, where teams picked up ready and prioritized PBIs, until completing a Product Owner review. On the right, you can see the PBI goals for that two-week timebox and tally marks indicating PBIs completed towards them.

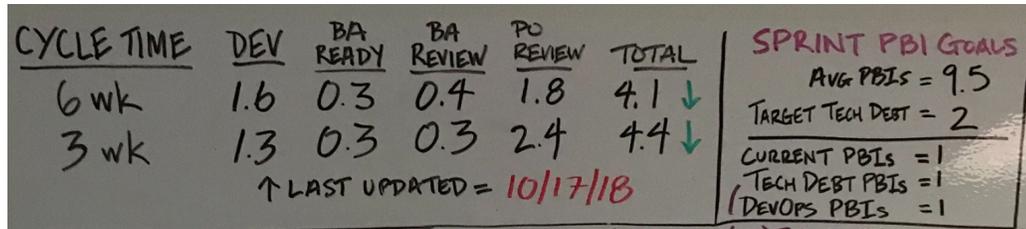


Figure 4. Example of EVE-MOD cycle time visualization at the team level

After those first, hectic few weeks, we saw our cycle times stabilize to levels that were normal before the dark work from the case processing release crept into our system. As I mention briefly in Section 3.3, our NoEstimates journey effectively began before the official experiment started on June 20, 2018. As there was, to my chagrin, little time to refine work as we wanted.

Prior to the experiment, JIRA data shows 240 completed PBIs between April 5 – June 19 and only two had story points. From April 5, 2018, until July 19, 2019, EVE-MOD completed 1,344 PBIs including 148 PBIs (11%) that had estimates. Twenty features were delivered to customers during NoEstimates.

As we stabilized our workflow, trends emerged as we built sprint review slides every two weeks. It is worth noting that we did not share team-level cycle times and throughput with stakeholders. Those measures were used for internal purposes. After returning to Scrum with NoEstimates, sprint planning started by visualizing the product group’s 3-sprint rolling average of completed PBIs. In both Kanban and Scrum, the three teams eventually normalized to deliver a combined 27 PBIs per sprint with a normal variation of +/- 3 PBIs which we used in conjunction with holidays, vacations, and training dates to establish the group’s capacity for the upcoming sprint. With the capacity set, we made a sprint commitment as a product group and only then progressed to slot PBIs with the teams. Using these methods, EVE-MOD reliably made our sprint and release commitments during NoEstimates.

3.7 Winter 2018-2019—Returning to Scrum

The switch to Kanban had served us well during that busy spring of 2018. The flow of work coming into our backlog had become more predictable in early summer. We continued to improve on our Kanban model; mostly, we made it simpler than what we created in the workshop (Figure 1). By Fall 2018, EVE-MOD was completing the integration with a new upstream partner. The group was already feeling the results of a hectic summer recovering from the previous launch. Feedback from the EVE-MOD teams told us that they found Kanban to feel like a never-ending sprint. While we had designed celebrations into our Kanban system, the teams often chose to defer small celebrations, like fancy donuts, for larger celebrations, like dinners out or trips to an escape room. The Scrum Masters were seeing and hearing signs of weariness among the teams.

Shifting from a product to team level perspective, it’s helpful to recall that I had started in January 2018 as the Scrum Master of the Big Sillies, one of the three EVE-MOD feature teams. By the fall, I had taken on the additional responsibility of serving Agile Coach for all three EVE-MOD teams. Andrew Jarding, the previous EVE-MOD Scrum Master turned Agile Coach who played a key role in building EVE-MOD from one to three teams, had shifted over to focus his time on SAVE-MOD, our sister product team.

The Big Sillies, like each EVE-MOD team, held their own team-level retrospective each sprint. While those retrospectives predominately focused on team-level improvements and reflections, they often served as testing grounds for ideas to form and gain traction before being raised at the Overall Retrospective, again following the LeSS framework. In their Sprint 68 retrospective (November 2018), the Big Sillies decide to experiment with Scrum mindset starting in the next sprint. The experiment consisted of three prongs: create a sprint backlog, size work via story points, and run a daily sprint confidence poll after team standup. By doing these three things, the team hypothesized that they would feel a greater sense of structure and less stress about the seemingly never-ending flow of work to On Deck.

We simulated a team sprint backlog in our EVE-MOD Kanban by allowing the Big Sillies to tag themselves to certain PBIs that were imminent in the prioritized Product Backlog, which was shared by all three teams,

instead of only selecting from the PBIs waiting in the On Deck column. The team assigned story points to the PBIs selected for the sprint backlog. Velocity would be monitored, but it would not be used outside of the team. The daily sprint confidence poll was also a return to an old team practice. At the end of standup each day, we verbally assessed the team’s ability to deliver on the sprint commitment and visualized the team’s agreed upon score on a chart on our team space. We continued to track the team-level cycle time and manage with team-level WIP limits, just as we had done prior to Kanban and NoEstimates.

The Big Sillies, the team I supported as Scrum Master, preferred working this way and became advocates for making the switch back to Scrum for the entire product team. The teams did not need much convincing as evidenced by making the decision convert back to Scrum starting in Sprint 72, only four sprints after the Big Sillies started their team-level experiment. After eight months of Kanban, the return to Scrum reintroduced Sprint Planning, though this did not change our use of NoEstimates. The teams agreed the return to Scrum would provide welcome structure as we advanced towards the version 30 API release planned for April 2019.

3.8 Summer 2019—Leaving NoEstimates

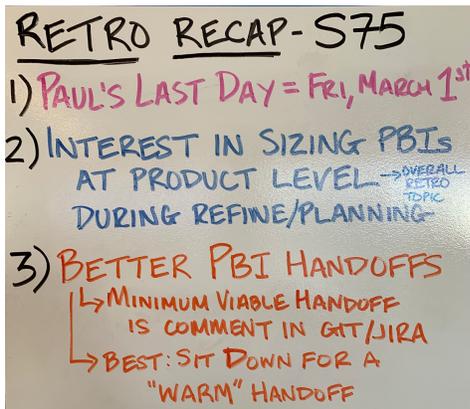


Figure 5. Big Sillies retrospective notes #2

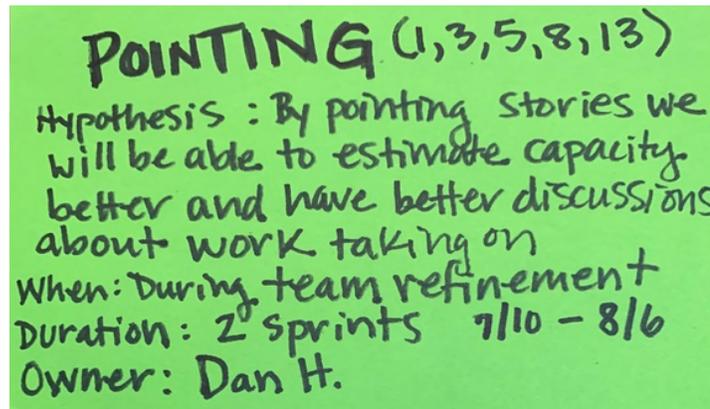


Figure 6. Sticky note describing the Pointing experiment

Leaving NoEstimates proved harder than switching back to Scrum. In Sprint 72 (Dec 2018), the teams chose to not adopt sizing PBIs with story points when they agreed to return to Scrum. The Big Sillies, without any influence from me, put sizing PBIs back on the table in their Sprint 75 (Feb 2019) retrospective with an action to raise the idea of sizing PBIs at a future Overall Retrospective topic (Figure 5, item two in blue marker).

The Big Sillies knew that making a system change as the product group was marching toward this next release was not likely to happen; however, they and other advocates wanted to keep the idea fresh in EVE-MOD minds. Several sprints followed as the API work was completed and improvements were made. The group consensus was that the change should occur shortly after the v30 release in late April 2019.

By early summer, the EVE-MOD teams were getting asked to assist another group. With an enthusiastic and positive attitude indicative of our values and culture, the group quickly jumped into help test the data loads for a partner team’s data migration to a new cloud platform. This testing work was new for the group, and they did not have prior knowledge of the business rules or the cloud platform. We observed cycle times climb week by week in our team metrics; sprint commitments became less certain. Eventually, the problem was too obvious to ignore as average cycle times climbed above 7 days and individual PBIs spiking to above 10 days. The teams acknowledged the new work and platform made refining difficult, as they did not have experience with either.

Unlike previous Overall Retrospectives, sizing PBIs was the marquee topic that emerged in the Overall Retrospective. Dan Herndon, a developer turned Cloud/DevOps engineer, did not much care whether we started pointing or not. He was tired of the debate and, as one of six team representatives in in Sprint 85’s Overall Retrospective, volunteered to serve as the experiment’s owner (Figure 6). One detail of the experiment not written on the experiment’s green sticky note was the intention to attempt to slice PBIs larger than 5 story points—although 8- and 13-pointers were not forbidden. The team knew the answer to their problems was not that allowing larger PBIs into the workflow; they should learn how to correctly understand the work and split the PBIs accordingly. We continued to use the foam discussion cards during Refinement after the Pointing returned. To no one’s surprise, the experiment became, once again, a practice after the two-sprint experiment concluded; our work started to flow again, and EVE-MOD continued forward to its next challenge.

4. WHAT WE LEARNED

This experience report is a celebration of a true expression of Agile—of inspecting and adapting, experimenting, learning, and more than I can share in a single report. It is not intended to diminish the practice of estimation or story points, but to explore situations in which alternative approaches may be useful. I enjoyed exploring with the group the value of estimation and the role it plays in refinement, planning and forecasting of work. Too often, the teams that I have worked with overvalue precise estimates at the lowest level in the belief that the increased scrutiny and detail will remove all uncertainty or risk in a desperate effort to control or predict delivery. I believe the better approach is to acknowledge the larger frame. Stepping back to see the whole system—where is it flowing, where it is blocked, where there is joy, where there is overburden.

NoEstimates can work in many situations. It worked for us because of three primary factors: stable team/group composition, a strong culture of learning and experimentation, and knowledge of our domain and technology space. The experience validated my understanding of NoEstimates and further confirmed to my satisfaction that estimates are a means to an end. Personally, I gained more hands-on experience in guiding self-managed teams through the process of maintaining a healthy, learning culture. Some of our experiments and solutions were basic and minimally viable versions and there is nothing wrong with them—often that is all the time teams have time or authority to do. I wish we had been able to explore more elegant approaches to our challenges like Monte Carlo forecasting or designing a better solution than our foam discussion cards. We did the best we could with the time and information we had available. I am proud to share what EVE-MOD accomplished.

How did members of EVE-MOD like their NoEstimates experience? While it helped them through challenging times, their feedback suggested positive yet moderate support. Most enjoyed the opportunity to explore a different approach than the standard Scrum taught and practiced. I commonly find agile teams are somewhat less interested in Agile theory than the removal of waste or low-value activities to make space for the technical, business, social, or organizational problems they are trying to solve in their daily work. To that end, they felt NoEstimates was a break or vacation from the tedium of story pointing.

The group offered two closing takeaways. First, they would attempt NoEstimates again under similar conditions: high performing team, high trust culture, and deep understanding of the product and business environment. Second, the group strongly favored making work as small as possible whether in Scrum or Kanban. Finally, I share this story because if EVE-MOD could experiment with NoEstimates and Kanban in the federal government, you can (and should) experiment everywhere. So, what is stopping you and your teams?

5. ACKNOWLEDGEMENTS

EVE-MOD product group: First, I acknowledge my fellow EVE-MOD members who shared the NoEstimates experience: Andrew Eason, Andrew Whitley, Brian Palladino, Cara Peters, Cameron Ivey, Chris Arsenault, Chris Pietro, Colyn Irvine, Dan Herndon, Dane Weber, David Chang, Drew Nickerson, Glenn Espinosa, Hugh Gardiner, Iksu Oh, Jay Danielian, Jeff Suppes, Jen Pengelly, Jay Danielian, Joe Hunt, Josh Cohen, Josh Dorothy, Khoi Le, Matt Acors, Mayowa Aladejebi, Nick Bristow, Starr Chen, Paul Schneider, and Siva Sivakumar. Special thanks to my fellow EVE-MOD Scrum Masters: Mark Koenig, Zack Ayers, and Angela Lu.

Excella: During my tenure, EVE-MOD benefited from the account leadership of Jenn Starkey, Dan Davis, Andrew Jarding, Joan Nelson, and Hank Ritchie. I'm also so grateful for Excella, Tony Solomita and all my colleagues. I have learned so much being around a wealth of talented agilists and agile-minded professionals.

USCIS: Many thanks to the USCIS employees who supported us—especially Rebecca Green, E-Verify's Product Owner, and her team, including Mariah Doolittle, in giving us the freedom to experiment and challenge the status quo. Additional thanks to Ann Nguyen and Pablo Juarez, E-Verify's IT project managers, who helped us navigate the internal demands and processes of DHS/USCIS.

REFERENCES

- [Zuill] Woody, Z., "No Estimate Programming Series intro," December 12, 2012, website: <http://zuill.us/WoodyZuill/2012/12/10/no-estimate-programming-series-intro-post/>
- [LeSS1] Larman, C., Vodde, B., Large-Scale Scrum "LeSS framework." website: <https://less.works/less/framework/index>
- [LeSS2] Larman, C., Vodde, B., Large-Scale Scrum "Feature Teams." website: <https://less.works/less/structure/feature-teams>
- [MSA] Jarding, A., Weber, D., Dekkinga, J., Spence-Goon, N., Boos, P. Hone, T. Manifesto for Scaling Agility. May 16, 2019. Retrieved July 11, 2021. website: <https://scalingmanifesto.org/>
- [DORA] DevOps Research & Assessment metrics, <https://www.devops-research.com/research.html#reports>
- [Weber] Weber, D., "Undercover Scrum Master," Agile 2019 Conference Experience Report. <https://www.agilealliance.org/resources/experience-reports/undercover-scrum-master/>