Our agency recently added a PMO office and they are still geared towards waterfall and have stated Scrum is the most difficult Agile framework to implement. How do we work w/this office and change that mindset?

Luke: Scrum can be challenging for a PMO office because the time horizon for a PMO office is often quite different than the time horizon for a single Scrum team or set of Scrum teams. Scrum teams operate on a 2, 3 or 4 week cycle. PMO offices often operate on annual or quarterly cycles. I recommend exploring how to align time horizons and time cycles as a means to help PMO offices and Scrum teams work together. This newsletter provides advice on aligning on market rhythms: https://conta.cc/2NoXeC5.

What is the difference between Scrum Team and POD teams?

Luke: Not much, and quite likely, nothing is different. My understanding of a POD team comes from Dave Gray's book *The Connected Company.* Dave defines a pod as "a small, autonomous unit that is enabled and empowered to deliver the things that customers value.". In my experience that is synonymous with a Scrum team.

We are Product Support following Kanban (ScrumBan), yet pulling in small features (stories), room for scope creep.

Luke: I think there is always a room for scope creep in any Agile method. Indeed, my dev teams at Conteneo will gladly share how I often try to "slip in" one more Product Backlog Item into a Sprint :-). We find the best approach is to discuss these issues openly and put the teams in control of what they can absorb. If you're using Kanban, you might need to "bend" your WIP nd your class of service guidelines.

Our clients often ask about metrics for our teams to "prove" that we are efficient. Do you have any input on metrics in these various models? can they be "unified" in some way?

Luke: A good Agile team is going to manage a core set of metrics that help them improve over time. These metrics should include both output metrics and outcome metrics. For example, the output metric of velocity / backlog items delivered in a unit of time is one measure of efficiency. An increase in profit margins is another measure of efficiency. It helps to have a conversation about the metrics that matter the most to your organization.

We are attempting to use the Scrum framework for some of the technology teams which are more Admin/Tech teams. It has been difficult. Should we try another Agile framework such as Kanban instead?

Luke: Yes. Try Kanban and then conduct retrospectives to identify which patterns work best in your circumstances.

Many organizations are in a transition phase from waterfall to agile, and as such are forced to combine the two different process models. Do you recommended any best practices during this transition phase?

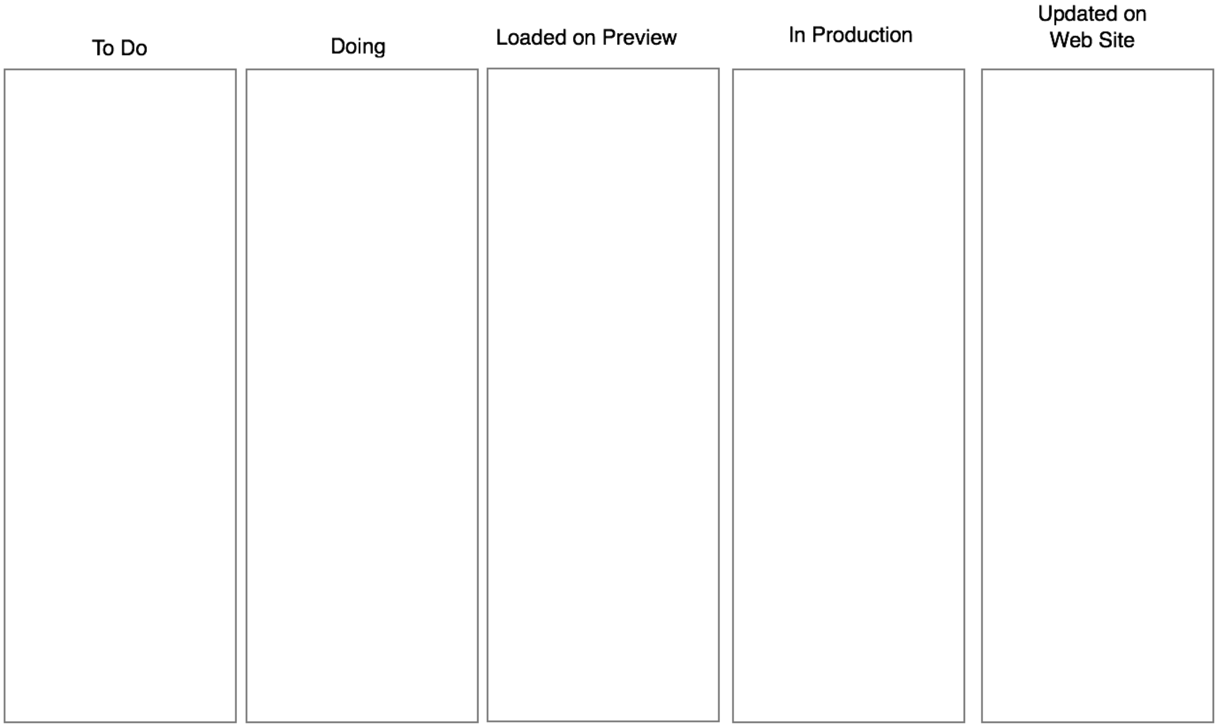
Luke: In our webinar we covered several best practices. I strongly recommend focusing on Kert's advice: find and leverage synchronization points.

Commitments in Waterfall are when requirements are signed off - 90% of the time.

Luke: I don't think this is a question. Suggest skipping it.

When is the deployment point in Kanban, Scrum vs Waterfall

Luke: My experience is that deployment in Kanban is based on the meaning of "Done". At Conteneo, we use Kanban to manage our portfolio and Scrum to manage the Sprints needed to create a portfolio-level item. We identify the deployment point in our process via a customized Kanban board in Weave (shown below). Note that we distinguish our deployment process as "Loaded on Preview" (our pre-production servers) and "In Production". The "Updated on Web Site" column refers to marketing, sales and related promotional activities.



To Do	Doing	Loaded on Preview	In Production	Updated on Web Site

Our Sprints focus on maintaining shippable quality. This means that we distinguish between quality as defined by our Definition of Done (e.g., "No stop ship bugs") and we ship (e.g., "move the code into production") when we have realized sufficient *value* for our customers.

Do other users find that Kanban is easier to use on a non-software production team like a help desk or call center where work is coming in all the time and time-boxed iterations are harder to define?

Luke: I will defer to my colleague Kert on this question.

What is the Production release cycle in Scrum, Kanban and Waterfall. Frequency

Luke: My experience is that regardless of method the frequency of production releases is a function of the size of the item being developed. If the item is small and fits within a single Sprint you can release it into production within or at the end of the Sprint. A Kanban model will base its release into production based on lead time. Once again, smaller “batch” or “items” will flow through the Kanban system more quickly. Waterfall models tend to be less frequent partly because Waterfall project items tend to be larger.

What should be a good frequency to deliver tested products?

Luke: This is not a production question but a Product Management question. Good Product Managers / Product Owners understand the market(s) they’re serving and the preferred frequency of serving these markets.

For someone who is a new grad (Masters in Digital Media [hands on agile program with industry work as a SM]) who has CSM and is studying PMI-ACP, I have little experience but tons of theory, likely when I get a job, it'll not be a pure scrum environment (like I was trained in), what approach should I have to being a Scrum Master considering EVERY place will be different, I guess you never really are truly prepared right?

Luke: It sounds like you’re as prepared as you can be. Jump in - and use your retrospectives to improve!

I work for Learning and Development team and we struggle with dependency with Agile when do we start developing training material …

Luke: This is common. We find it extremely helpful to create represent dependencies visually and make them “obvious” to all groups involved with the customer deliverable.

My understanding of "deliver to your customer" is, to show them the functional increment in order to get feedback, not necessarily to reléase to a prod environment. That's way I can say my teams deliver every sprint. Is that the general understanding?

Luke: That’s one understanding, and if it helps you and your team succeed you should keep it. I prefer to have a stricter definition that “delivery” means “putting software into a production environment for use by customers”.

Can you explain better the slide that talked about prioritization using Scrum vs Kanban?

Luke: Prioritization is a central challenge in any method because we always have more good ideas in the backlog than available resources! Accordingly, I don't think of "Scrum vs. Kanban" when it comes to prioritization. Instead, I think there are a common set of prioritization strategies that you can leverage, a topic I covered in this Collaboration at Scale webinar on "How to Prioritize a Backlog": https://info.conteneo.co/cas/how-to-prioritize-a-backlog.

Dependencies on Strategic direction/leadership action

Luke: This isn't a question, but a statement. Yes, there are times when an Agile or Kanban team is dependent on the decisions of a leadership team. To reduce the time it takes to make key decisions leadership teams should use techniques like Participatory Budgeting, Deliberative Decision-Making and related frameworks.

What tools would you use to show this when working with remote teams?

Luke: We find that photos work pretty well.

What do you think about the DevOps team helping with CI/CD?

Luke: I think this is a great idea. Try it and use your retrospectives to determine if it is working well enough to keep it.

What do you think about the spotify model?

Luke: The Spotify model is derived from the "Organizational Engineering" chapter in my book *Journey of the Software Professional*. It represents a scalable model for balancing two core demands: building the "right thing" and building the "right way". That said, any model of an organization should be created with an understanding of Conway's law and a consideration of the underlying architecture (see: https://en.wikipedia.org/wiki/Conway%27s_law).

You had mentioned a book called "Accelerate". Please repeat the author's name.

Luke: *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations Paperback* – March 27, 2018
by Nicole Forsgren PhD (Author), Jez Humble (Author), Gene Kim (Author)

Isn't flexibility the name of the game with Agile? Worked with several teams and team members have had to incorporate a Hybrid each time as much as we would love to do true Agile that is just a fleeting fallacy. The end result on each has been successful we got to the end result with compromise.

Luke: I agree that Agile teams find ways to create flexibility.

Any advice or reference on dependency management tools?

Luke: No.

In my org we find it difficult sometimes to work with our DevOps team, because it's hard to structure their time -- do you have any suggestions with how to work with a team who is sometimes unable to plan out their work entirely just due to it's nature?

Luke: My experience is that this is the kind of context in which Kanban systems do very well. I recommend asking your DevOps team to try using a Kanban system tailored to their specific work products.

What is the common release cycle on Kanban

Luke: Kanban systems base their release cycles on lead times.

For an external team that has <unknown> process (such as sales), what ceremonies would your recommend you participate in?

Luke: This depends on your role. I think that it is helpful when sales teams participate in roadmapping activities and in the selection of portfolio-level initiatives. Frameworks like *Prune the Product Tree* and *Buy a Feature* are powerful ways to integrate sales teams into these critical decision-making events.

What are your thoughts about The Journey Team Model?

Luke: I'm not deeply familiar with *The Journey Team Model* but a quick read of this article on Medium suggests that it is a compelling approach. https://medium.com/soluto-nashville/a-better-way-to-build-products-the-journey-team-model-f1843026911. If you try it and it works for you I think you've found your answer!

What's the best way to have predictability on kanban for a software development team?

Luke: I will defer to my colleague Kert on this question.

I work with a multidisciplinary product, electronics, software, mechanical etc. each of them strongly affect each other but none see eye to eye on agile/waterfall processes... what recommendations might you have to resolve this?

Luke: I find that bringing together people for key planning events is central to helping the focus their work on a shared objective while managing natural differences.

Will there be future sessions geared towards Marketing's use of Scrum & Kanban frameworks?

Luke: I will add this to the Collaboration at Scale topic backlog.

How do dependencies on other teams using different process models impact the estimation process on your agile teams.

Luke: This is one of the hardest challenges: How to create a reasonable estimate when you're dependent on another team? I find that Agile methods (Scrum/Lean Kanban) coupled with tried and true development patterns (such as API driven architectures) enable teams to create more useful estimates even when there are dependencies. These dependencies should be managed by Scrum Masters and closely monitored in Scrum of Scrum meetings.

We experienced moving to KanBan after they became a high performing team.

Luke: I don't think this is a question. Suggest skipping it.

For Kanban would you suggest a different team make up regarding skill sets?

Luke: Regardless of the individual team process model, it is important to always focus on the *work that needs to be done.* Accordingly, the Scrum or Lean Kanban team should be comprised of "T-shaped people" capable of *working as a complimentary team* delivering the necessary work. At Conteneo, we find that the minimum team size is four devs: One focused on presentation (HTML and CSS), one focused on interaction (Javascript), one focused on backend matters (databases, performance) and one full stack developer to provide glue. The team solves the problem together - as a team.

Can we add this training as a CEU (option B2?)

Luke: Scrum Alliance question.

Do you know of a tool that shows those dependencies in an electronic way vs the whiteboard.

Luke: No. You might be able to find a plugin or similar tool.

We keep our processes very close to pure scrum. Would it be beneficial to pursue SAFe?

Luke: SAFe is one of the methods designed to scale Scrum/Agile practices to larger organizations and initiatives. SAFe blends aspects of Kanban systems to manage portfolios and large investments and supports the use of Scrum or Lean Kanban at the team level. I can't speak to your specific situation, but you may wish to consider it.