Lean Software Engineering
Essays on the Continuous Delivery of High Quality Information Systems

BUY SCRUMBAN BOOK

Scrumban
Essays on Kanban Systems
for Lean Software Development

Corey Ladas

@COREY_LADAS ON TWITTER

You can put on a Lean hat, you can take it off. You can put on a TRIZ hat, TOC hat, Deming hat. None of it is True, some of it is Useful. 2010-10-26

I am sometimes amazed at the time, effort, and passion people apply towards denying the value of change. 2010-10-25

@kelly_waters We can always describe things using ordinary descriptive language. But not knocking names, just pointing out costs. in reply to kelly_waters 2010-10-22

More updates...

Powered by Twitter Tools

Scrum-ban

As more people become interested in Lean ideas and their application to knowledge work and project management, it's helpful to find ways that make it easier to get started or learn a few basic concepts that can lead to deeper insights later. For those that are curious about kanban in an office context, it's not unusual to find people who are either currently using Scrum, or have some understanding of Scrum as representative of Agile thinking. One way or another, Scrum users are an important constituent of the Kanban audience. Since Scrum can be described as a statement in the language we use to describe kanban systems, it is also fairly easy to elaborate on that case in order to describe Scrum/Kanban hybrids. This can be useful for existing Scrum teams who are looking to improve their scale or capability. It can also be useful for more cautious new users who find comfort in an "established" method[1].

The idea of using a simple task board with index cards or sticky notes is as old as Agile itself. A simple variation of this is a task board with a simple *Pending -> In Process ->*
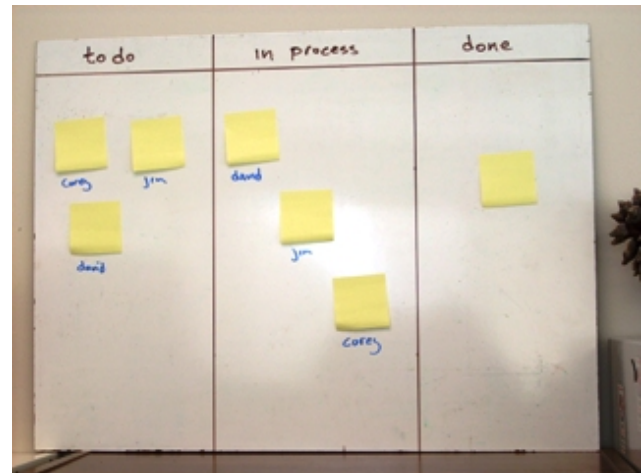
Complete workflow. The cards represent work items that are in the current scope of work. Names can be associated with the cards to indicate who's working on what. Agile teams have been using this sort of method for a long time, and a [few] [people] pointed out early on that this had some resemblance to the notion of kanban in lean systems.

Of course, a variety of electronic tools exist that perform these functions, but the simple task board represents a couple of lean principles that I find very valuable, *simple technology* and *visual control*. The utility of such a simple method of workflow management is that it is easy to manage, and more

[Find]

PAGES

A Tool for Better Reviews with Capture-Recapture
Patterns of Software Engineering Workflow
Pugh Decision Matrix
Scrum
Wideband Delphi

ARCHIVES

Select Month

importantly, *it is easy to change*. Huddling around a computer monitor, even a very large one, is in no way a substitute for the tactile and social interactivity that accompanies manipulating a large task board. Maybe someday it will. Not today. What electronic tools are good for are managing lists of things, like backlogs and bugs, and producing reports. Simple tools can be a difficult concept to explain to technology fanatics, but then, so can *value*.
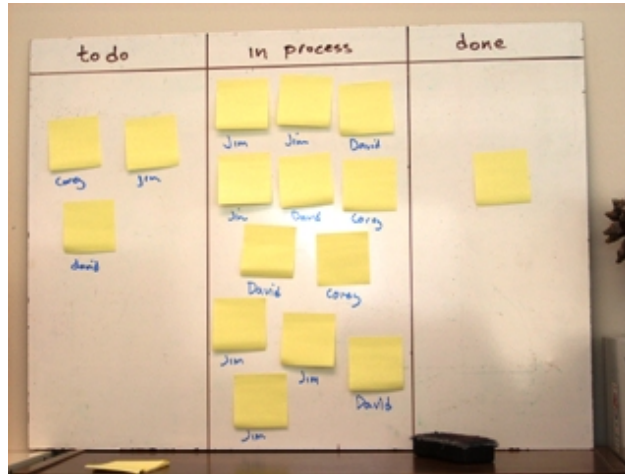
A problem with the basic index-card task board is that there is nothing to prevent you from accumulating a big pile of work in process. Time-boxing, by its nature, sets a bound on how much WIP that can be, but it can still allow much more than would be desirable.

If a kanban is a token that represents a work request, and our task board can still get out of control, then what is the problem here? The problem is that a kanban is more than just a work request on a card, and putting sticky notes on a whiteboard is not enough to implement a pull system.

A kanban is more than an index card

In a modern economy, the production and distribution of scarce goods and services are regulated by a system of money and prices. Money can be represented by currency notes, which have little intrinsic value, but that by agreement, can be exchanged for real goods and services. The existence of a neutral medium of exchange makes possible a system of economic calculation of the relative scarcity of the supply of goods in an economy. Such a system of prices

is a market. Markets communicate the value of economic
production and distribution to their participants.

If a
currency
note can be
exchanged
for an
object of
real value,
then there
must be
some way to enforce the scarcity of the notes in a way that
corresponds to the scarcity of real value in the economy. In
practice, some kind of institution must enforce this scarcity.
The health of a market economy depends greatly on the ability
of its monetary institution to coordinate the supply of money
with the supply of goods and services. In an unhealthy
economy, unstable prices make economic calculation difficult
and disrupt the communication between producers and
consumers needed for efficient production and distribution.

A kanban represents a portion of the productive capacity of
some closed internal economy. It is a medium of exchange for
the goods and services provided by the operations of a system
of productive resources. The supply of kanban in circulation is
controlled by some regulatory function that enforces its value.
That is, a kanban is a kind of private currency and the shop
floor manager is the bank that issues it, for the purpose of
economic calculation.

If you carry the currency analogy further, then you might say
that kanban is not about the cards at all. Just like money is not
about the bills. Kanban is all about the limits, the quantity in
circulation. How that is represented in a transaction is mostly
incidental.

A simple rule for understanding all of this might be:

> A task card without a limit is not a kanban in the same way
> that a photocopy of a dollar bill is not money.

If you use a durable token like a plastic card, then this is easy
to manage: control the number of cards in circulation. If all of
the available cards are already in circulation, then the next
person who comes looking for one is just going to have to wait
until one returns. *This is the very purpose of the kanban
system.* However, if you use a more disposable medium like
index cards or sticky notes, then you need another mechanism

We've developed software
or managed teams at
Microsoft, IBM, and a few
other interesting little
software companies. These
are our learnings.
Corey Ladas
Bernie Thompson

to regulate the "money supply." In our case, we simply write the quantity of kanban in circulation on the task board, and allocate new cards according to that limit.

This means that a kanban serves two functions: it is a request to do something in particular, but it is also permission to do something in general. That second notion of permission is where people who are new to lean thinking tend to struggle. But this is precisely how we can "optimize the whole" or "subordinate to the constraint."

Crunchy on the outside, chewy on the inside

Just as an unregulated index card on a cork board is not a kanban, time-boxed iteration planning is not pull. No reasonable interpretation of Lean involves building to a one-month forecast unless the cycle time for each work order is also a month. One month worth of stuff in process is certainly a much smaller batch size than 3 months or 18 months, but if your iteration backlog contains 20 work items, then that's still about 19 more than it needs to be a pull system.

Nonetheless, it is not difficult to augment Scrum with a few simple practices that move us towards a more recognizably lean workflow. The most obvious is the reduction of iteration length, although this is not without problems[2]. As we'll see, it's possible to incrementally enhance Scrum with more and more pull-like features until all that remains of the original process is vestigial scaffolding. The simple approach is to start with Scrum-like iterations and iteration planning process, and begin to add pull features to the team's internal process.

One simple technique that brings us much closer to our kanban definition is to set a multitasking limit for individuals. You might have a simple principle
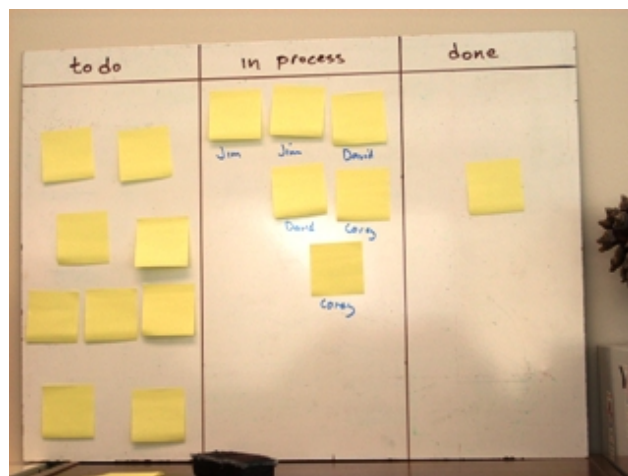
like: *prefer completing work to starting new work,* or you might express that as a rule that says: *try to work on only one item at a time, but if you are blocked, then you can work on a second*

item, but no more. In our example, that rule gives us an effective WIP limit of 6.

Another common technique is the late binding of tasks to owners. Some teams will pre-assign all of the known tasks during iteration planning. That's generally not a good idea because it artificially creates a critical path. Waiting until the "last responsible moment" to assign tasks to people maximizes knowledge and brings you closer to pull.

 Just because *anybody* can have more than one item in process doesn't mean that *everybody* should have more than one item in process. A problem with our multitasking rule is that it locally optimizes with no consideration of the whole. An implicit total WIP limit of 6 is still more WIP than we should probably tolerate for our three workers. A limit of 4 of 5 total items in process at one time still allows for some multitasking exceptions, but disallows the obviously dysfunctional behavior of everybody carrying two items. At this step, we have moved beyond a rule about individuals and have made a rule about the task cards themselves. That is, we have made our cards into kanban.

Another enhancement we can make to our previous board is to add a ready queue between the backlog and work-in-process. The ready queue contains items that are pending from the backlog, but have high priority. We still haven't bound any individual to these tasks, but as soon as somebody becomes available, they should take one of these tasks instead of picking something out of the general backlog. This enables us to decouple the process of assigning work from the process of prioritizing work, and it simplifies assignment. The ready queue also has a kanban limit, and it should be a small limit, since its only purpose is to indicate which work item should be started next.

Now we can begin to see some of the mechanics of pull and flow:
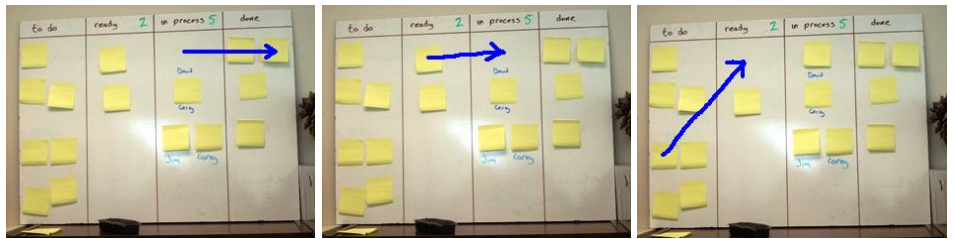
1. David completes a task and moves it into the "done" column.

2. David pulls a new kanban from the ready queue and begins working.

3. The team responds to the pull event and selects the next priority item to go into the ready queue.

At this point, we are now operating a simple kanban pull system. We still have our time-boxed iteration and planning cycle, so perhaps we might call such a thing a Scrumban system!

Now that we have a sense of capacity and pull, it's natural to think about flow. Breaking up our nebulous "in process" state into better defined states can give everybody more

visibility into the strengths, weaknesses, and overall health of the team. Even Agile workflows like Extreme Programming have relatively well-defined roles and states, and a smooth flow of work between those states is just as important as a smooth flow of work through the process overall.

Here we've broken down *in-process* into two states: *specify* and *execute*. Specify is about defining whatever criteria are necessary to determine when the work item can be considered complete. Execute is about doing the work necessary to bring that work item into a state which satisfies those criteria. We have split our previous WIP limit of 5 across these two states. Specify is considered to take less time in this case, so it is given a limit of 2. Execute consumes the remaining limit of 3. We might change this ratio as time goes on and our performance changes.

Since we are now thinking more about flow, the additional workflow detail strongly suggests using a Cumulative Flow

Diagram to track the work and measure our performance. A simple burndown tells you something about whether or not you are delivering value, but not very much about why. The CFD communicates a lot of additional information about lead times and inventories that can diagnose problems, or even prevent them.

By defining our workflow a little better, we can also account for some functional specialization. In this case, it might be a soft specialization, where some of us prefer doing one type of work more than another, even though we're capable of doing it all. It's important to understand that this kind of pull workflow system *allows* specialization but does not *enforce* specialization. The team owns the work and the workflow, and it's up to the team to figure out how to get it done efficiently.

If we let the person who's best at performing the "specify" function handle more of that work, then we may also need to coordinate handoffs between ourselves. Adding the *specify-*

complete column communicates to the team that a work item which was previously in the specify state is now ready to be pulled by anyone who wants to move it to the execute state. Work that is still in the specify state is not eligible to be pulled yet. If the owner of a ticket in the specify state wants to hand it off, he can put it in the complete buffer. If he doesn't want to hand it off, he can move it directly into the execute state as long as capacity is available. It might be that the execute state is full, and the only eligible work is to pull another ticket from the ready queue into specify.

Since we have added a new column for our handoff buffer, we are also increasing the WIP limit by a small amount. The tradeoff is that the increase in lead time due to the new inventory should be offset by the decrease in lead time due to the advantage of specialization. We also mitigate the impact of that new inventory by pooling the WIP limit across the preceding state. This has the very beneficial consequence of making the specify-complete buffer a variable throttle for the preceding station. The more work that piles up in the specify-

complete buffer, the less work can be in process in the specify state, until specify is shut down entirely. But we see it coming, it doesn't "just happen."

If we're going to allow workflow specialization and the handoffs that result, then we will also need some agreement about what results to expect at each handoff. We can do that by defining some simple work standards or standard procedures for each state. These do not have to be complicated or exhaustive. Here, they are simple bullets or checklists drawn directly on the task board. They only need to be sufficient to avoid misunderstanding between producers and consumers. These standards are themselves made and owned by the team, and they can change them as necessary according the practice of *kaizen*. Putting them in a soft medium like a whiteboard or a wiki reinforces the notion of team ownership.

Level 2 Scrumban

In the basic version of Scrumban described so far, the iteration review and planning cycle happens just as it does in ordinary Scrum. But as our production process has matured, we have also given ourselves some tools to make the planning process more efficient, more responsive, and better integrated with the business that it serves.

With the pull system in place, our flow will become smoother as our process capability improves. We can use our inter-process buffers and flow diagrams to show us our process weaknesses and opportunities for kaizen. As we get closer to level production, we will start to become less concerned with burndown and more concerned with cycle time, as one is the effect and the other is the cause. Average lead time and cycle time will become the primary focus of performance. If cycle time is under control and the team capacity is balanced against demand, then lead time will also be under control. If cycle time is under control, then burndowns are predictable and uninteresting. If burndowns are uninteresting, then goal-setting and risky heroic efforts are unnecessary. If burndowns are uninteresting, then iteration backlogs are just inventory for the purpose of planning regularity and feeding the pull system. As such, they should be the smallest inventories possible that optimize planning cost.

Since the team now pulls work into a small ready queue before pulling it into WIP, then from the team's perspective, the utility of the iteration backlog is that it always contains something that is worth doing next. Therefore, we should use the least

wasteful mechanism that will satisfy that simple condition.

A simple
mechanism that
fits the bill is a
size limit for the
iteration backlog.
Rather than go
through the
trouble of
estimating a
scope of work for
every iteration,
just make the

backlog a fixed size that will occasionally run to zero before the
planning interval ends. That's a simple calculation. It's just the
average number of things released per iteration, which in turn
is just a multiple of average cycle time. So if you have 5 things
in process, on average, and it takes 5 days to complete
something, on average, then you'll finish 1 thing per day, on
average. If your iteration interval is two work weeks, or 10 work
days, then the iteration backlog should be 10. You can add one
or two for padding if you worry about running out. This might
be a point that's been lost on the Scrum community: it's never
necessary to estimate the particular sizes of things in the
backlog. It's only necessary to estimate the average size of
things in the backlog. Most of the effort spent estimating in
Scrum is waste.

In our final incarnation of Scrumban, iteration planning still
happens at a regular interval, synchronized with review and
retrospective, but the goal of planning is to fill the slots
available, not fill all of the slots, and certainly not determine
the number of slots. This greatly reduces the overhead and
ceremony of iteration planning. Time spent batch processing for
iteration planning estimation can be replaced with a quality
control inspection at the time that work is promoted to the
ready queue. If a work item is ill-formed, then it gets bounced
and repeat offenders get a root cause analysis.

Off with the training wheels

If you have made it this far in your evolution, you will probably
realize that the original mechanisms of Scrum are no longer
doing much for you. Scrum can be a useful scaffold to hold a
team together while you erect a more optimized solution in
place. At some point you can slough off the cocoon and allow

the pull system to spread its wings and take flight.

The first step beyond Scrum is to decouple the planning and release periods. There may be a convenient interval to batch up features to release, and there may be a convenient interval to get people together to plan. If we have a leaner, more pull-driven planning method, there's really no reason why those two intervals should be the same. Your operations team might like to release once a month, and your product managers might like to establish a weekly prioritization routine. No reason not to accommodate them.

Once you've broken up the timebox, you can start to get leaner about the construction of the backlog. Agility implies an ability to respond to demand. The backlog should reflect the current understanding of business circumstances as often as possible. Which is to say, the backlog should be event-driven. Timeboxed backlog planning is just that, where the event is a timer, but once we see it that way, we can imagine other sorts of events that allow us to respond more quickly to emerging priorities. Since our system already demonstrates pull and flow, that increased responsiveness should come at no cost to our current efficiency.

The problem we are trying to solve is:

> *The ideal work planning process should always provide the development team with best thing to work on next, no more and no less.*

Further planning beyond this does not add value and is therefore waste. Scrum-style timeboxed planning usually provides a much bigger backlog than what is strictly necessary to pick the next work item, and as such, it is unnecessary inventory and therefore unnecessary waste.

The next event we might consider for scheduling planning activities is the concept of an *order point*. An order point is an inventory level that triggers a process to order new materials. As we pull items from the backlog into the process, the backlog will diminish until the number of items remaining drops below the order point. When this happens, a notice goes out to the responsible parties to organize the next planning meeting. If our current backlog is 10, our throughput is 1/day, and we set an order point at 5, then this planning will happen about once a week.

Once a week might be reasonable if people are hard to schedule or need some lead time in order to prioritize. However, if they are more available than that, then we can set the order point

lower. If the planners can respond within a day, then perhaps we can set the order point at 2. If the order point is 2, then there may be no need to keep a backlog of 10. Perhaps we can reduce the backlog to 4…and reduce our lead time by 6 days in the process.

The end state of this evolution is pull, or prioritization-on-demand. If the planners can make a good decision quickly enough, and there is no economy of scale in batching priority decisions together, then the size of the backlog only needs to be 1. At the moment the item is pulled by the development team, the planning team is signaled to begin selecting the next item. If the planning team is fast enough in their response, then the development team will never stall. If there is some variation or delay in reponse, then a backlog of 2 might be necessary to prevent stalls. But 2 is still a lot smaller and leaner than 10. Or 20. Or 50, which is something I've seen more often than I would like.

The same kind of logic can be applied to the release interval. There is an optimum batch size for releases and we should first try to find it, and then try to improve it. The result of our efforts will ultimately be features-on-demand.

Even at this level, we still have a fairly basic kanban system. From here we can add work item decomposition (swimlanes), or structural dependency flow for additional scale. Along with an enlightened division of labor, this is how we believe that Lean shows the way to scale Agile up to the enterprise.

---

1. in spite of the fact that the kanban idea is at least 40 years older
2. which I'll probably write about in another post sometime

{ 160 }
Comments

1. **Gerald Williams** | 10-Jul-08 at 6:52 am | Permalink

   An excellent article and some food for thought. Thanks.

2. **Rob Hathaway** | 10-Jul-08 at 7:35 am | Permalink

   Hi Corey,

   Your post is really interesting and I've done exactly what you've described above with one of my clients who found it massively useful…

despite being really hesitant at first. They were also doing scrum (loosely and in inverted commas) and were struggling so I got them back to Scrum proper in the fall of last year and then introduced the kanban idea's almost the way you described above over a month or so with great results. Couple of things I wanted to get your take on:

One thing that I didn't get round to changing with them was how they sized their stories – they're using Story Points but I'm wondered what your take on that would be?

Also how does this pull based work fit into a release plan? Scrum based teams have their velocity based on completed story points and can then do some basic release management work (e.g. it'll be roughly 4 iterations (or sprints) before all these stories would be completed and you can draw the burndown from this information. If you're not doing development iterations anymore what happens to velocity and how Scrum teams traditionally use it to track progress?

I mention this last question as that is one that will come up a lot with Scrum teams and may be worth amending the article to cover.

3. **Hank Roark** | 10-Jul-08 at 8:33 am | [Permalink](#)

    While I 99.44% agree with this post, I think there is still some detail that needs to be filled in. For example

    "It's only necessary to estimate the average size of things in the backlog. Most of the effort spent estimating in Scrum is waste."

    I would say this is true if there is managed variation in the size of the backlog. For example, all PBIs could be constrained to be small (e.g., everything is "3 ideal days or less") and all PBIs bigger than that broken into smaller PBIs (I think this helps support single piece flow). At some point the product owner needs to judge PBIs based on value (e.g., ROI). How can they do that if they are just basing each PBI's value based on an average?

4.   **Corey Ladas** | 10-Jul-08 at 8:54 am | <u>Permalink</u>

@Rob,

First, it's really great to hear about your results. We love hearing kanban stories from the field!

I think story point limits are a legitimate variation on kanban limits. It adds a little complexity to do it that way, but I wouldn't object if somebody felt that was best for them.

I knew somebody would call me on the release planning question. I will be writing more about that in the near future, and we'll be discussing it at the APLN conference in Seattle next week.

Throughput is continuously calculated as work items complete. We're managing throughput directly by managing work-in-process and cycle time. Kanban is all about fixing work-in-process, and that leaves us with cycle time which we manage with value stream and theory of constraints methods and such.

Release planning consumes the historical throughput metric, and can apply methods like Minimum Marketable Features, Staged Delivery, and Rolling Wave Planning. That's what we recommend: a rolling wave planning event on a regular cadence.

"Toyota naturally makes production schedules… Just because we produce just-in-time in response to market needs…does not mean we can operate without planning"

"First, the Toyota Motor Company has an annual plan. This means the rough number of cars…to be produced and sold during the current year. Next there is the monthly production schedule… Based on these plans, the daily production schedule is established in detail and includes production leveling."

–Taiichi Ohno

5.   **Corey Ladas** | 10-Jul-08 at 9:08 am | <u>Permalink</u>

@Hank,

Limiting the size of work items for downstream scheduling is one of the things we recommend.

A kanban workflow can extend the value stream before and after the typical boundaries of a Scrum team, so that some of the work that the Product Owner does is pulled into the workflow and managed accordingly. Size and effort estimates can be produced as a natural consequence of analysis, since that analysis has to be done anyway.

6. **Corey Ladas** | 10-Jul-08 at 11:08 am | Permalink

    @Hank,

    Heaven knows I don't expect anybody to agree with me 100%. You rightly point out the significance of work item sizing to single piece flow, and that is spot on.

    Mostly, I want to help facilitate a conversation about the implications of pull and flow for software development. That doesn't mean that pull and flow are "right" in some absolute sense, although I personally find them to be extremely compelling. But if we decide that pull and flow are the right answer, then this blog is mostly about figuring out what that really means, and I very much appreciate the feedback that I get here.

7. **HL Arledge** | 10-Jul-08 at 12:56 pm | Permalink

    Scrum works flawlessly for all of my teams. Why overly complicate things by adding kanban constraints? From my KISS perspective, Scrum is "leaner" than kanban will ever be.

8. **Corey Ladas** | 10-Jul-08 at 2:25 pm | Permalink

    @HL

    Lean processes are built for competition. If "good enough" process is good enough for you, then perhaps textbook Scrum is adequate for your purposes. Comfortable is a luxury, so I hope you enjoy it.

    Maybe you are not in a competitive situation. Maybe your competitors are inept. If, however, you are under any pressure for systematic

performance improvement, then the suggestions in this article address inefficiencies that are built into Scrum.

9. **Craig Daniel** | 11-Jul-08 at 8:54 pm | Permalink

    Thanks for your post. Through years of using Scrum and tweaking to make it more lean, my current team is using techniques very similar to the ones you describe.

    You've given me some more ideas for future kaizen meetings to further tweak the process.

    Thanks!

10. **Synesthesia » Links roundup for 2008-07-31** | 31-Jul-08 at 6:09 pm | Permalink

    [...] Scrum-ban [...]

11. **Agile 2008 - Scrum and Kanban | #2782 - Agile software development and best practices for building Microsoft .NET applications.** | 12-Aug-08 at 4:04 am | Permalink

    [...] section of this talk that was about applying Kanban to an existing Scrum process, "Scrum-ban", can be found on Corey's blog. What this does for you is allows you to evolve Scrum [...]

12. **Agile 2008 - Something is wrong but why? | #2782 - Agile software development and best practices for building Microsoft .NET applications.** | 14-Aug-08 at 5:04 am | Permalink

    [...] teams across time zones – Corey Ladas talk about "Scrum-ban" helped me figure out that one of the key issues with distributing teams across time zones is [...]

13. **Being Cellfish** | 17-Aug-08 at 10:40 pm | Permalink

    How to avoid having too much work in progress...

    Most teams I worked with have at some point had a lot of work in progress. Sometimes virtually the complete...

14. **Riaan's Blog** | 18-Aug-08 at 9:08 pm | Permalink

    Scrum-ban…

    Corey Ladas has written an interesting paper titled Scrum-ban in which he describes how a Scrum team…

15. **Blog Xebia France - Revue de Presse Xebia** | 18-Aug-08 at 11:26 pm | Permalink

    […] Corey Ladas propose aux équipes Scrum d'évoluer vers le Lean en introduisant un système de Kanban. […]

16. **Wayne Allen's Weblog** | 21-Aug-08 at 9:31 am | Permalink

    Corey Ladas explains Scrum-ban…

    Cory has a great post titled: Scrum-ban | Lean Software Engineering. In it he describes how a team can…

17. **Corey Ladas explains Scrum-ban | Software Testing Blog** | 21-Aug-08 at 1:03 pm | Permalink

    […] Ladas explains Scrum-ban Cory has a enthusiastic place titled: Scrum-ban | Lean Software Engineering. In it he describes how a aggroup crapper verify plus of kanban within a Scrum […]

18. **Eduardo Miranda** | 25-Aug-08 at 5:24 am | Permalink

    It looks like it's getting closer to a production like system. Do you believe that creating software is a production like activity?

19. **Corey Ladas** | 25-Aug-08 at 11:00 am | Permalink

    @Eduardo

    No, I believe that creating software is often a workflow-like activity.

20. **Gerry Kirk** | 26-Aug-08 at 2:03 pm | Permalink

   In a pull system, where does one schedule retrospectives? Does the team decide how often they should occur? Seems like everything else doesn't have a particular schedule.

21. **Corey Ladas** | 26-Aug-08 at 2:26 pm | Permalink

   Hi Gerry,

   Firstly, a pull system creates what I'd call an "event-rich environment," which means there is a great deal of context and opportunity for introspection and process improvement. The pull system is giving you permission to not wait until your next "official" retrospective to change something. "Pink tickets" or "Andon lights" ought to trigger a process that can lead to a root-cause analysis and process improvement.

   Secondly, there should often be some kind of planning process or rhythm above the level of individual work item scheduling. This particular article showed how you could use the Scrum framework for that purpose. You could also schedule retrospectives around lower-frequency events like release of an MMF. If your MMFs are sufficiently "M" then you might use their natural rhythm to trigger planning events. Or you could schedule a regular event that may or may not coincide with some other planning or release event. One suggestion is a 2-week integration/release cadence, with a 6-week (or semi-quarterly) rolling wave planning event.

22. **Gerry Kirk** | 28-Aug-08 at 7:56 am | Permalink

   Corey, I'm not familiar with using "Pink tickets" or "Andon lights", but I understand the purpose.

   Heh, what is an MMF? More jargon for me to learn heh.

23. **Clinton Keith** | 30-Aug-08 at 11:07 am | Permalink

   Corey,

   Great article! I'm applying this to game

development where there is a transition from exploring the game mechanics (fun) using Scrum to the production phases where we develop 8-12 hours of assets using a Lean-Kanban approach. Your description of Scrumban is perfect for transitioning the team.

The main thing that would prevent us from trying Scrumban across the whole project is the concern oflosing a major benefit of the iteration. Preproduction (Scrum) iterations are ideal for a "unifying audacious goal" for the team. We don't know all of our tasks (often only 50%), so we leave room in the schedule for exploration. Leveling development, decoupling iteration planning from review…this seems to deprecate audacious iteration goals.

Am I overlooking something?
Thanks
Clint

24. **Corey Ladas** | 30-Aug-08 at 3:22 pm | [Permalink](#)

Clint,

Firstly, thank you very much! Part of the thinking about the Scrumban approach is that it allows you to keep old practices that have value to you, add new practices that have value to you, and drop ones that don't. I told one story here about an evolution of a process, but there are other stories that could also be told. You could do a lot of the things that are in the article without giving up iterations. And the point was meant to be that you would only give them up if/when you recognized that they no longer have value for you. If that never happens, then you wouldn't give them up!

One thing I left out of this article was project or product planning, which can provide additional context and motivation. It seems like I only write about this in the comments…but I will have an article soon about the relationship between Minimum Marketable Features, Rolling Wave Planning, Real Options, and Kanban.

25. [**SA BizTalk Bloggers Aggregation**](#) | 01-Sep-08 at 3:19 am |

Permalink

Scrum-ban…

Corey Ladas has written an interesting paper titled Scrum-ban in which he describes how a Scrum team…

26. **Como montamos o quadro do Scrum « Blog da Bluesoft** | 17-Sep-08 at 8:35 am | Permalink

[…] montamos o quadro do Scrum Neste vídeo apresento nosso "Scrum-ban", comentando sobre os materiais utilizados para montá-lo e quais são as informações […]

27. **Daily Digest for 2008-09-18 | Pedro Trindade** | 19-Sep-08 at 12:40 am | Permalink

[…] Scrum-ban | Lean Software Engineering […]

28. **Derek** | 30-Sep-08 at 12:50 pm | Permalink

I still don't get specify and execute totally. Can you give a specific example.

29. **Corey Ladas** | 30-Sep-08 at 1:08 pm | Permalink

Hi Derek

Specify/execute is only an example. The boundary was meant to approximate what-to-build vs. how-to-build-it.

Specify is meant to be the "operational definition of the problem statement". That could be things like requirements specifications, test cases, and wireframes. In turn, these things could be represented by things like user stories and automated acceptance tests.

Execute could be schematics, design verification tests, source code, integration, acceptance testing and other V&V.

30. **Biju Vasudevan** | 01-Oct-08 at 1:39 pm | Permalink

Corey, I found the article written by you a lot of

insights to me. Many more concepts and flaws in the industry to be cleared to make it leaner and give value to the customer

31. **Testing for Agility « Agile Elements** | 13-Oct-08 at 9:51 am | Permalink

[...] rapid delivery, regular inspection, adaption, customer alignment, quality, etc. by other means (eg: kanban, FDD, etc.) then they pass the test. I'd perhaps change this to "Can teams effectively [...]

32. **Alexia** | 15-Oct-08 at 4:47 pm | Permalink

Thanks for this article, Corey. I am leading a Production Support/Minor Enhancement team in a development shop where all the features teams are using Scrum and I've been looking for a way to implement an Agile/Lean methodology which would tolerate the constant interruptions inherent to application support and maintenance. Your approach is the most promising I've found so far. Now if you'll excuse me, I'm off to fight our other Scrum teams for some wall-space!

33. **foofish.org » Blog Archive » KaizenConf goals** | 04-Nov-08 at 9:40 am | Permalink

[...] different direction, but I'm going to read up more on Lean/Kanban some more, (offhand, Scrum-Ban may prove helpful), and see if there are tweaks that can be borrowed from those ideas that would [...]

34. **new ThoughtStream("Derick Bailey");** | 20-Nov-08 at 3:03 pm | Permalink

Kanban – Pulling Value From The Supplier…

Before I start talking about how our team is going about our implementations of Lean and Kanban, I wanted…

35. **Derick Bailey.com** | 20-Nov-08 at 3:03 pm | Permalink

    Kanban – Pulling Value From The Supplier…

    …

36. **XP Days Germany Blog » Kanban-Referenzen** | 01-Dec-08 at 6:01 am | Permalink

    […] Scrum morphen in Kanban. […]

37. **John Baker** | 04-Dec-08 at 7:21 am | Permalink

    Corey,
    Do you have any productivity data to compare the improvements of a project using Scrum-ban vs. textbook Scrum?

38. **Corey Ladas** | 04-Dec-08 at 1:32 pm | Permalink

    Hi John,

    It's still pretty early, so much of the evidence is still anecdotal or speculative. David Anderson has real data on pull system performance in general. Clinton Keith has specific data on Kanban vs Scrum. There may be some others, possibly Dave Laribee or Karl Scotland. A good place to ask would be on the Yahoo! kanbandev group.

39. **Kirk Thoughts » Twitter Groups: Bringing communities together** | 12-Dec-08 at 12:35 pm | Permalink

    […] I like Twitter. A lot. Twitter has helped me connect with a diverse group of people, particularly in the Agile community. I consider myself fortunate to chat and learn from people like Lisa Crispin (@lisacrispin), Brian Marick (@marick), Esther Derby (@estherderby), as well as lesser knowns like myself. For instance, I learned about Lisa's upcoming book on Agile Testing (pre-ordered), Esther's love of gardening and good food and have a ring-side seat between the always colourful Ron Jeffries (@RonJeffries) and Bob Martin

(@unclebobmartin). I also discovered some great articles, including Cory Ladas' Scrumban. […]

40. **Clive Skipper's Blog** | 22-Dec-08 at 5:03 am | Permalink

    Another batch of Interesting Scrum and Agile Blog Posts:…

    Another batch of Interesting Scrum and Agile Blog Posts:…

41. **The Rise of Lean | Xebia Blog** | 05-Jan-09 at 1:46 am | Permalink

    […] (thinking) tools that can aid in tuning current Agile practices. A perfect example is Corey Ladas's Scrum-ban approach; a way to upgrade the default Scrum task board using Lean tools. Given the rising […]

42. **Toby Mildon** | 06-Jan-09 at 7:03 am | Permalink

    Thanks for this excellent article. We're running with Scrum but need to be that little bit more agile so am looking into Kanban. This article is good at helping map out a transition approach. Cheers.

43. **Scrum tools: visually creating Sprints - a mockup « teamwork's blog** | 21-Jan-09 at 3:28 am | Permalink

    […] scrum | Following Skype, Twitter and e-mail discussions with Rick Cogley, looking for example at Scrum-ban, we thought about how to improve the current Scrum module, creating a more "visual" […]

44. **Confluence: SV - Hugbúnaðarþróun** | 06-Feb-09 at 9:07 am | Permalink

    Improving Speed and Quality.…

    Vision/Framtíðarsýn Viðskiptavinir eru ánægðir með afköst og gæði verkefna deildarinnar. "In preparing for battle I have always found that plans are useless, but planning is

indispensable…….

45. **Confluence: SV - Hugbúnaðarþróun** | 06-Feb-09 at 9:09 am | Permalink

Scrum-ban!…

Interesting article about how to evolve Kanban…

46. **Confluence: Agile Center at Síminn** | 13-Feb-09 at 4:54 am | Permalink

Scrum and Kanban…

Scrum & Kanban Discussion on the 17/12 Goal: Investigate how netkerfi and aðgangkerfi would start to use Kanban system (with Scrum) Main items discussed: Kanban Flow optimization Just in time process Reduce the planning process Kanban….

47. **Post-Agile | Blogues chez Nihahi Technologies** | 26-Feb-09 at 7:34 pm | Permalink

[…] un résumé, vous pouvez lire ce post sur son […]

48. **Scrum-ban « Lean and Kanban** | 03-Apr-09 at 7:46 am | Permalink

[…] Scrumban is such an approach, read more here […]

49. **Projektmanagement: Von Scrum zu Scrum-ban « MacPM.net** | 06-Apr-09 at 7:06 am | Permalink

[…] hier ist nicht mehr ganz frisch, aber immer noch Wert gelesen zu werden: Scrum-ban von Corey Ladas. Es geht dabei um die Verbindung von Scrum und dem ursprünglich vom […]

50. **Alexis Le-Quoc** | 09-Apr-09 at 9:17 am | Permalink

We're implementing kanban for our IT Operations; still in the prototyping phase but it's helped so far to visualize our bottlenecks. I've posted a few pictures on my blog with little

explanation and hope to expound in the future.

Just a quick note on page 55 of your book, I found the passing reference to Axiomatic Design to require a bit more explanation. It's a bit coming out of nowhere and I'd love to have more details.

Great book.

51. **Corey Ladas** | 09-Apr-09 at 10:24 am | Permalink

Hi Alexis,

I agree about some of the references in the Scrumban book. I'm sure I drop a couple of random TRIZ references as well. I do have a couple of related articles here, and more will be coming.

http://leansoftwareengineering.....ng-really/
http://leansoftwareengineering.....structure/

52. **Paul Cornell** | 24-Apr-09 at 9:53 am | Permalink

Kanban: Some Kanban Resources...

Some kanban resources: Kanban vs. Scrum : http://www.crisp.se/henrik.kni.....-Scrum.pdf . For...

53. **Kanban: Some Kanban Resources | Coded Style** | 24-Apr-09 at 11:21 am | Permalink

[...] are designed to reduce multi-tasking, [maximize] throughput, and enhance teamwork." Scrum-ban: http://leansoftwareengineering.com/ksse/scrum-ban/. "...a kanban serves two functions: it is a request to do something in particular, but it is also [...]

54. **Scrumban - Corey's Original Blog | limitedwipsociety.org** | 29-May-09 at 1:46 am | Permalink

[...] anyone that hasn't read it Corey Ladas' blog post on Scrumban is truly worth the read. It describes how to evolve from Scrum into a more Kanbanesque process. [...]

55. **Paul Cornell** | 29-May-09 at 10:49 am | Permalink

    Kanban: Scrum-ban…

    Scrum-ban:
    http://leansoftwareengineering.com/ksse/scrum-ban/ . "The idea of using a simple task board…

56. **Kanban: Scrum-ban | Coded Style** | 29-May-09 at 11:30 am | Permalink

    […] Scrum-ban Scrum-ban:
    http://leansoftwareengineering.com/ksse/scrum-ban/. "The idea of using a simple task board with index cards or sticky notes is as old as Agile […]

57. **Breaking down user stories into tasks « AlexHamer.ca** | 02-Jun-09 at 11:23 am | Permalink

    […] conclusion that the best course of action would be to not break the stories down at all, as seen in Scrum-ban, but we're not ready yet, and I'm not sure we will be in this project. So in the mean […]

58. **BlogCoward** | 07-Jun-09 at 9:07 pm | Permalink

    pmk …

    pmk …

59. **Lean and Kanban Software Development Digest | Edge of Chaos. Agile Development Blog** | 09-Jun-09 at 6:28 am | Permalink

    […] Scrum-ban. Interesting attempt to mix Scrum and Kanban, taking the best from both worlds. Kanban with iterations is possible. […]

60. **The cost of iterations applied to Kanban as well… « Agilesparks Blog** | 10-Jun-09 at 8:03 am | Permalink

    […] on is Kanban. I believe Kanban is a great fit to many teams and situations. Specifically doing Scrumban is a great way to get the benefits of Agile project management together with the

Lean Flow Kanban [...]

61. **Daily Links for Tuesday, June 23th, 2009** | 23-Jun-09 at
4:40 am | Permalink

[...] Scrum-ban | Lean Software Engineering
[...]

62. **David Draper on agile & design » Blog Archive »
Managing WIP with Scrum** | 30-Jun-09 at 4:03 am |
Permalink

[...] to you in order to improve an existing
Scrum team or as a step in moving towards
Kanban (see also Scrumban by Corey [...]

63. **BISHOP-IT.RU** | 20-Jul-09 at 11:28 am | Permalink

Канбан в IT (Kanban Development)...

Я сегодня обещал написать несколько статей
про новую методологию гибкой разработки
Канбан (Kanban Development)...

64. **Scrum or Kanban? «** | 20-Jul-09 at 11:45 pm | Permalink

[...]
http://leansoftwareengineering.com/ksse/scrum-
ban/ View This Pollonline surveys [...]

65. **All about the code » Videos from Lean & Kanban 2009** |
22-Jul-09 at 1:55 am | Permalink

[...] in Miami is available at
http://www.sep.com/lk2009 with talks from
among others Corey Ladas on scrumban, Alan
Shalloway on going beyond Toyota, David
Anderson on Kanban, Karl Scotland on Kanban
flow and [...]

66. **SUGK: Scrum User Group Karlsruhe – Nächstes Treffen
am 5. August | ArmerKater.de - Scrum, Agiles
Projektmanagement, Innovation und Organisation,
Agile Nearshoring** | 31-Jul-09 at 4:50 am | Permalink

[...] sich im Vorfeld etwas informieren möchte

kann nachlesen bei Henrik Kniberg, Boris Gloger oder Corey Ladas (Autor des Buches [...]

67.   **tOM Trottier** | 02-Aug-09 at 1:02 am | Permalink

If you use the post-its with the sticky side at on the bottom edge, instead of the top edge), then it's easy to see the writing at the top of each note, even if crowded.

tOM

68.   **Kanban: Scrum-ban | Coded Style** | 03-Aug-09 at 7:16 pm | Permalink

[...] Scrum-ban Scrum-ban: http://leansoftwareengineering.com/ksse/scrum-ban/. "The idea of using a simple task board with index cards or sticky notes is as old as Agile [...]

69.   **Activity Modeling for Kanban "Pull" Systems** | 06-Aug-09 at 11:51 pm | Permalink

[...] Kanban - Feature development is streamlined by moving features through a kanban "pull" system. Kanban systems can take many forms. Most kanbans are comprised of two primary components: units (i.e., Goal) and cards (i.e., features and user stories). For more information about using kanban systems for software development, check out Scrum-ban | Lean Software Engineering. [...]

70.   **OCTO talks ! » Scrum sans itérations ?** | 07-Aug-09 at 1:52 am | Permalink

[...] http://leansoftwareengineering.com/ksse/scrum-ban/ [...]

71.   **links for 2009-08-11 « pabloidz** | 11-Aug-09 at 5:04 am | Permalink

[...] Scrum-ban Lean Software Engineering (tags: scrum kanban) [...]

72. **Brian** | 14-Aug-09 at 12:29 am | Permalink

    Cory – would like to watch your video on Scrumban at http://www.sep.com/lk2009/core.....-evolution, but noticed it never displays for me…seeing the same thing?

73. **Corey Ladas** | 14-Aug-09 at 9:55 am | Permalink

    Hi Brian. It doesn't work for me either.

74. **new ThoughtStream("Derick Bailey");** | 14-Aug-09 at 2:02 pm | Permalink

    Kanban In Time-Boxes: The Cadence of WIP and Sprints…

    A comment that was left on a previous post , and a response that I made to the comment, got me thinking…

75. **First thoughts on Kanban | Brian Hartsock's Blog** | 19-Aug-09 at 8:03 am | Permalink

    [...] started throwing out the idea of Kanban instead of Scrum. Really, they are wanting to start with Scrumban, where they ease into Kanban and as Rex likes to stay, "kick the ends out of [...]

76. **Huh? « Lean? Agile? Huh! – The novice** | 05-Sep-09 at 7:35 am | Permalink

    [...] phase ourselves into XP principles that suited us when I came across a blog by Corey Ladas called Scrum-ban and now I'm more confused and intrigued than [...]

77. **Scrum and Kanban - Like Chocolate and Peanutbutter | Scrum Head** | 15-Sep-09 at 2:02 pm | Permalink

    [...] (Kanban resources)Kanban vs. Scrum (Friendly comparison)One day in Kanban Land (Kanban Cartoon!)Scrumban (Fully [...]

78. **Kanban at Lonely Planet | limitedwipsociety.org** | 12-Oct-09 at 1:32 pm | Permalink

    [...] I also refrence this post/site: http://leansoftwareengineering.com/ksse/scrum-ban/ which was our original introduction and starting point. **Follows is original description with [...]

79. **Kanban development oversimplified: a simple explanation of how Kanban adds to the ever-growing Agile toolkit « Tech Notes** | 22-Oct-09 at 9:05 am | Permalink

    [...] Corey Ladas on Scrumban [...]

80. **PixelPoop » Decision "Gimbal Lock" and the Art of Temporary Commitment** | 02-Nov-09 at 1:04 pm | Permalink

    [...] prioritization in any process, but particularly in the most flexible Agile processes…like ScrumBan, of which I am a [...]

81. **Cumulative Flow Diagrams with Google Spreadsheets – BEKK Open** | 03-Nov-09 at 3:18 am | Permalink

    [...] to never work on more than a certain limited number of items at any given time. Kanban, CONWIP and Scrum-ban are similar techniques to achieve this. I won't go into detail about these techniques in this [...]

82. **Why we went from Scrum to Kanban « Derk de Geus** | 03-Nov-09 at 1:45 pm | Permalink

    [...] http://leansoftwareengineering.com/ksse/scrum-ban/ Filed under: Uncategorized Leave a comment Comments (0) Trackbacks (0) (subscribe to [...]

83. **Prakash** | 05-Nov-09 at 7:07 pm | Permalink

    Hi Corey,
    A wonderful article and a great way to simplify

Kanban learning. I don't get this line- "This might be a point that's been lost on the Scrum community: it's never necessary to estimate the particular sizes of things in the backlog. It's only necessary to estimate the average size of things in the backlog"

What is meant by estimating average size of things? All things are not the same. Let us say I have 5 MMFs in my in process (Specify) and my initial task is to find the lead time so that I can identify the limits for my backlog.

I find that the lead time for 5 MMFs is 20 days. This means 1 MMF takes 4 days. If my sprint is 10 business days I could complete approx 3 MMFs. This would mean my backlog limit could be 3. But these 5 MMFs have varying sizes. If I have to identify MMFs of similar sizes to fit in my backlog (3) I would have to spend considerable time planning, breaking stories etc.. In the process if I don't find MMFs that could fit the 3 backlog limit then what happens?

84. **We R hiring | crealytics Blog** | 18-Nov-09 at 10:00 am | Permalink

[…] Scrumban (Scrum um Kanban angereichert [incl. superduper Kanban-Board]) […]

85. **Kanban vs Scrum « Tales from a Trading Desk** | 14-Dec-09 at 12:14 pm | Permalink

[…] vs Scrum Interesting read on InfoQ. Henrik provides more info. Also worth reading David's […]

86. **Scrum and Kanban – Like Chocolate and Peanutbutter « Do It Yourself Agile** | 21-Dec-09 at 3:47 pm | Permalink

[…] (Kanban resources)Kanban vs. Scrum (Friendly comparison)One day in Kanban Land (Kanban Cartoon!)Scrumban (Fully […]

87. **fragile » The Switch to Kanban** | 04-Jan-10 at 1:50 pm | Permalink

[…] A guide to incrementally introduce Kanban ideas into a Scrum environment, known as Scrum-ban. […]

88. **Solving for the iron triangle with kanban and scrumban — Chris Heisel** | 10-Jan-10 at 3:00 pm | Permalink

[…] Scrumban is a great transitional step for teams and clients when trying to go from Scrum/Agile or, shudder Waterfall, to Kanban. […]

89. **Confluence: Telcordia SD Adapters** | 25-Jan-10 at 2:21 am | Permalink

Development Processes…

Process description {}This page contains information on the majority of processes that could be applied to IT project{} \\ Here are the most well known processes into the PP presentation format SDLC models.ppt SDLC models…….

90. **Managing WIP with Scrum | Valtech UK** | 29-Jan-10 at 7:58 am | Permalink

[…] to you in order to improve an existing Scrum team or as a step in moving towards Kanban (see also Scrumban by Corey Ladas). Daily […]

91. **Confluence: Telcordia SD Adapters** | 01-Feb-10 at 5:57 am | Permalink

Wireless Networks…

Wireless Networks description {}This page contains information on the majority of processes that could be applied to IT project{} \\ Here are the most well known processes into the PP presentation format SDLC models.ppt SDLC models.ppt Agile…

92. **ScrumBan? « Tales from a Trading Desk** | 22-Feb-10 at 2:08 am | Permalink

[...] Lean Software Engineering offer a good overview of how ScrumBan differs from Scrum – essentially improving the speed of time-to-market. Agile [...]

93. **Kanbanin ja Scrum- taulut, mitä eroa - Sekalaista höpinää** | 25-Feb-10 at 6:15 am | Permalink

[...] on. ja jos sitten haluaa hämmentää itseään, niin kannattaa tutustua uuteen käsitteeseen eli scrumban. Tärkeintä ei ole tekniikka vaan siitä saatva hyöty. Eli jokaisessa tapauksessa paras [...]

94. **Wolfgang Frank** | 04-Mar-10 at 4:59 pm | Permalink

Great post! I also really like the Scrumban book …

I used similar principles by myself in the role of a scrum master (after studying Theory of Constraints and Lean Software Development theories) and it greatly helped to make a good team even better!

I am glad you provided this nice introduction and motivation article!

Cheers,
Wolfgang

95. **Paul Boos** | 22-Mar-10 at 7:47 am | Permalink

This is a great mechanism for working. This is not at all saying this can not work, but one contextual item where a pull system may NOT work is if during Sprint Planning, you estimate when you need a particular user to help with refining the requirements (story). You can't just pull, because you "scheduled" that person at a particular time most likely for a reason (perhaps they weren't available). At the very least, it could become a problem as you would be giving them last minute notice.

Again, this is not to say this technique can't work, but just to provide a case where it may be more difficult as a consideration for those thinking of using it. If the user (and perhaps it

is the product owner, but often he or she may not be) can be totally dedicated to the team or at least to that Sprint, then this could be problematic.

Great post and I'll be looking for where I can apply these concepts.

Cheers!
Paul

96.   **henry** | 27-Mar-10 at 7:54 am | Permalink

ClearWorks New Version – 2.4 Released

Many our customers asking us about enhancements, and we are doing our best to provide requested features and functionality. Today's release is a big update of current Sevenuc best seller product (also known as agile lifecycle tool for hardware & software project) and at the same time composition with other software configuration management tools, and more automation test tools and build servers.

Update contains more elements for Lean R&D real-time collaboration platform
and reflects latest innovations in Lean Kanban created by Sevenuc and other platform vendors.

What's New in 2.4

* Workflow define for deferent project with Lean stage management.
* Event and status driven mechanism by Triggers.
* Email classfication for effective customer request life-cycle management.
* Complete release support for Lean agile project.
* Lean R&D behavior improvements for all type of statistic charts

read more on http://www.sevenuc.com

97.   **Kanban ganz kurz erklärt » MacPM.net** | 19-Apr-10 at 7:54 am | Permalink

[...] etwa dem guten alten Wasserfall oder Scrum kombiniert werden. Klar, dass daraus

sofort der Begriff Scrumban entstanden [...]

98. **maja's soup** | 30-Apr-10 at 11:30 pm | Permalink

"The idea of using a simple task board with index cards or sticky notes is as …"…

The idea of using a simple task board with index cards or sticky notes is as old as Agile itself. A simple variation of this is a task board with a simple Pending -> In Process -> Complete workflow…….

99. **When Is A Sprint A Failure? |** | 09-May-10 at 7:47 am | Permalink

[...] best of both systems you'll knock the rough edges off of both. (Some people use the term "Scrum-ban".) One of the best influences Kanban can have on Scrum is to put the concept of a sprint [...]

100. **Confluence: GB IT** | 17-May-10 at 1:25 am | Permalink

Kanban und Scrum – Literatur und Links…

Zitiert aus Wikipedia: Kanban in der IT…

101. **Dennis Stevens and Associates » Blog Archive » How is the Kanban board different then a normal Agile board?** | 24-May-10 at 12:34 am | Permalink

[...] Once you get work showing on your Kanban board you will see where work is piling up. Excess work in process raises a number of challenges. It increases the time a new item will take to travel through the system. It indicates the likelihood of an overburden on the current performers or the next performers. For example, in Scrum iterations when there is a lot of work in development that all moves to test at the end of the iteration – this is undesirable. You can limit WIP in a number of ways. In Scrum, we limit WIP at the iteration boundary and some teams limit WIP by limiting the number of work items that can be active at any one time. The Kanban board calls for explicit WIP limits and

also recommends buffer columns to mitigate the impact of variation to keep work flowing through the system. You can certainly explicitly limit WIP and include a buffer or buffers on a normal Agile board. Here is a essay from Lean Software Engineering showing ScrumBan. [...]

102. **Code Leaders and Beautiful Teams** | 24-May-10 at 1:42 am | Permalink

[...] http://www.autohotkey.com/ http://leansoftwareengineering.com/ksse/scrum-ban/ [...]

103. **Confluence: Project Management** | 27-May-10 at 1:33 pm | Permalink

Interesting Agile related links...

Here you will find links to various blog articles and training opportunities outside of DRW. Articles, Blogs, random musings on Agile Agile development is more culture than process...

104. **We're Self Organizing Into... Kanban? | Scrumology** | 28-May-10 at 9:13 am | Permalink

[...] Scrum ceremony I'm not entirely sure. It seems to be more of a Scrum / Kanban mix for now (Scrumban?), and I don't see them discarding the rest of the Scrum ceremony anytime [...]

105. **J.D. Meier's Blog** | 06-Jun-10 at 12:53 pm | Permalink

Artificial Critical Paths...

I was reading through Corey's post on Scrum-ban again, and I really liked his point that assigning all...

106. **Bookmarks for June 22nd from 17:52 to 17:52 | Travis' Blog** | 22-Jun-10 at 12:08 pm | Permalink

[...] Scrum-ban | Lean Software Engineering – ‹Previous Post Bookmarks for June 22nd from 12:43 to 12:51 [...]

107. **Books, Random Reading and Thoughts « Tales from a Trading Desk** | 25-Aug-10 at 8:08 pm | Permalink

[...] on the KanBan road needs to remember the importance of the Kanban Board. If you are new to KanBan there is a very good minibook available here. KanBan is probably [...]

108. **Susanne Reppin** | 26-Aug-10 at 12:48 am | Permalink

great article. We are doing Kanban for Administration teams and Maintenance Teams and Software Architecture Teams soon also. It's really successful and we like it.
XING AG, http://www.xing.com
Susanne

109. **Lessons Learned in Agile at foomonger's blog** | 17-Sep-10 at 10:50 am | Permalink

[...] useful for getting the entire team on the same page but it's painful. Now I prefer something more scrum-ban style and hold weekly 30-45 minute planning sessions. Just enough to fill out the backlog for a [...]

110. **Confluence: Offshore Development** | 27-Sep-10 at 11:48 pm | Permalink

Scrum meeting with Raj – September 28th, 2010...

Raj is fixing row total calculation issue in Cargotec PDF (OFF18) Raj can contact Mikko or Jukka for help if needed There is also new field required for the Parma Email Form layout, Tommi will create new issue for this Some discussion about Scrum,......

111. **Confluence: intern: Agile Space** | 30-Sep-10 at 10:14 am | Permalink

Kanban – Scrumban...

Quellen zu Kanban u.ä. Kanban and Scrum making the most of both (eBook als PDF)

Kanban
ScrumbanKanbanAndScrumInfoQVersionFINAL.pdf
Kanban in der IT (Wikipedia)…

112. **Quora** | 30-Sep-10 at 11:59 am | Permalink

What are the metrics by which you measure your agile development process? …

You introduced some good points, Parker. I would offer that both volatility and velocity (and your mention of story point inflation) measure story grooming, which is not always a proper measurement. It may be that minimal volatility indicates that a …

113. **Liste de liens concernant les Méthodologies Agiles** | 10-Oct-10 at 10:29 am | Permalink

[…] SCRUM-ban : http://leansoftwareengineering.com/ksse/scrum-ban/ […]

114. **Creating a 4 P development strategy « Implementing Agile – discoveries in software development** | 17-Oct-10 at 12:20 pm | Permalink

[…] new ScrumBan development […]

115. **Lean IT Operations : Opsban « IT-Manager Corner** | 21-Oct-10 at 9:49 am | Permalink

[…] agile systems and creating their continously improving Scrum system, which is also called Scrumban. It is basically a evolution of Scrum using the concepts of the Japanese lean Kanban methods. […]

116. **InVisible Blog » Daily Log, 2010-11-09** | 09-Nov-10 at 1:05 pm | Permalink

[…] Scrum-Ban […]

117. **Confluence: Carlo Space** | 10-Nov-10 at 1:43 am | Permalink

Kanban…

February, 2010 … I really like this one video: an interview with Henrik Kniberg…

118.   **Adil Wali » » Kanban and Agile Software Development — Part 2** | 22-Nov-10 at 7:23 am | Permalink

[...] be used in order to create hybrid agile-Kanban systems. One example is Scrumban, fully discussed by Corey Ladas in his 2008 paper, in which he describes Scrumban as "Incrementally enhancing Scrum with more and [...]

119.   **Kanban Development Oversimplified | Agile Product Design** | 14-Dec-10 at 12:35 am | Permalink

[...] Kind of a simple way to manage things when you think of it. Corey Ladas explains this well in his Scrumban article and [...]

120.   **TargetProcess v.2.20.13 Released With Scrum-ban Support | TargetProcess Product Blog** | 14-Dec-10 at 3:59 am | Permalink

[...] releases You can filter stories and bugs on Kanban Board by release and iterations now. It enables Scrum-ban development process and makes Kanban Board useful for iterative development activities like Release [...]

121.   **Specific DevOps Methodologies? « the agile admin** | 14-Dec-10 at 9:15 am | Permalink

[...] Others have made a blended process, often Scrum + Lean/kanban hybrids with the Lean/kanban part being brought in from the ops side and merged with more of a love for Scrum from the dev side. Though some folks say Scrum and kanban are more in opposition, others posit a combined "Scrumban." [...]

122.   **cusmith** | 27-Feb-11 at 8:40 am | Permalink

I work with a team that does both new feature

development as well as production support. The "pull" concept seems to break down for us is when a support issue needs to be addressed immediately which requires reallocation of individuals from their current WIP task to a support task. This type of scenario "pushes" a high priority item into the flow and could take us beyond our WIP limits that we are trying to adhere to. Maybe this is just an acceptable situation where an item is pushed into the flow instead of pulled by a team member that is available for work.

I would be interested to hear from anyone with similar experiences and how they may have adapted their process to create pull out of a push situation.

123.   **PIA Blog / Productivity by Design » Kanban ou scrum distribué : quel outil ?** | 07-Mar-11 at 1:55 am | Permalink

[…] pas succombé à la mode de l'open space ! Tout le monde à son Kanban ou dérivé (Scrumban) : du designer au commercial en passant par le développeur. Nous sommes donc adeptes du Post-It, […]

124.   **25 Blogs zu Scrum und agiler Softwareentwicklung « Produktmanagement und Vermarktung von Internet-Anwendungen** | 12-Apr-11 at 1:03 am | Permalink

[…] „Lean Software Engineering" von Corey Ladas und Bernie Thompson: Die beiden teilen ihre Erfahrungen aus Teams von u.a. Microsoft und IBM. Ihre Artikel bekommen hohe Aufmerksamkeit und sorgen regelmäßig für umfangreiche Diskussionen. Dir hat dieser Beitrag gefallen? Ich freue mich über einen Kommentar. Du kannst auch meinen RSS-Feed abonnieren oder mir auf Twitter folgen: @pherwarth. Foto: Von Rafa[EU] […]

125.   **Kanban Development Oversimplified | AgileProductDesign.com** | 02-May-11 at 1:16 pm | Permalink

[…] Kind of a simple way to manage things when you think of it. Corey Ladas explains this

well in his Scrumban article and [...]

126. **Extreme Enthusiasm » Blog Archive » Design by Contract vs. Test-Driven Development** | 28-May-11 at 2:54 am | Permalink

[...] I'm reading the Scrumban book by Corey Ladas. One thing Corey says is that Test-Driven Development is good, but not as good [...]

127. **Kanban Development Oversimplified « agileproductdesign** | 29-May-11 at 1:01 pm | Permalink

[...] Kind of a simple way to manage things when you think of it. Corey Ladas explains this well in his Scrumban article and [...]

128. **Visual Management – More Advanced Designs | b2bDoctor** | 28-Jun-11 at 3:31 pm | Permalink

[...] software development) and how it can be better scheduled and delivered. One particular post on scrumban is recommended as it builds a complex visual management board step by step from a simple three [...]

129. **Fabrice Aimetti** | 01-Jul-11 at 12:35 am | Permalink

Hello Corey,

Your article is a must read! and a must try! I have translated it into french : http://www.fabrice-aimetti.fr/.....1/Scrumban

Thank you, you're great!

Regards,
Fabrice

130. **Drew Kime** | 27-Jul-11 at 1:58 pm | Permalink

Corey, I have a hard time accepting your premise that: "The ideal work planning process should always provide the development team with best thing to work on next, no more and no less."

In a manufacturing environment, you don't

need to know anything other than, "How fast are we putting out widgets?" But in software development, the 20 items that you deployed to production last month may be completely unlike the 20 items you deploy this month.

If you have vendors or customers who need to integrate with your application, users who need training, salespeople who need updated presentation materials, etc. etc. etc., you need to be able to tell people what features will be available when. I'm not saying there has to be a year-long unalterable roadmap, but there are valid reasons to want to know more than, "What are the next five things we're doing?"

How do you square your statement about planning with all these competing needs for more planning?

131.   **Como montamos o quadro do Scrum | Blog da Bluesoft**
        | 26-Aug-11 at 7:14 am | Permalink

        [...] vídeo apresento nosso "Scrum-ban", comentando sobre os materiais utilizados para montá-lo e quais são as informações [...]

132.   **Eric Lee** | 01-Sep-11 at 2:22 pm | Permalink

        When Is A Sprint A Failure?...

        Update: this blog is no longer active. For new posts and RSS subscriptions, please go to http://saintgimp...

133.   **Should I start with scrum, or kanban? « Mike Pearce – blog** | 19-Sep-11 at 1:08 am | Permalink

        [...] team is mature enough to make the most of it, if you're not sure, then try transitioning via scrum-ban, it'll help you see the benefits and enable you to get better at the things which make kaban [...]

134.   **Confluence: Knowledge Base** | 05-Oct-11 at 4:28 am | Permalink

        Daily Process...

This page is an old draft Introduction This docume…

135.   **Scrum Process** | 10-Oct-11 at 9:06 pm | Permalink

In Scrum, we limit WIP at the iteration boundary and some teams limit WIP by limiting the number of work items that can be active at any one time. The Kanban board calls for explicit WIP limits and also recommends buffer columns to mitigate the impact of variation to keep work flowing through the system.

136.   **Confluence: ScrumMasters** | 24-Oct-11 at 10:27 am | Permalink

Articles…

Articles There are some articles that the whole world should read. List them here, in an appropriate topic section. If there isn't one, add one\! Try to keep topics alpha ordered…….

137.   **Scrum-ban | Lean Software Engineering | Brent Sordyl's blog** | 02-Nov-11 at 1:56 pm | Permalink

[…] Story: Scrum-ban | Lean Software Engineering) Like this:LikeBe the first to like this post. agile, scrum agile, kanban, scrum How Yahoo […]

138.   **PM Hut** | 03-Nov-11 at 1:02 am | Permalink

Hi Corey,

Although this post was published several years ago I only had the chance to read it today. A very interesting concept. I wonder what happened to the concept of Scrumban now. It doesn't seem that it actually worked since no one is using it nor speaking about it…

Any thoughts?

139.   **Jonathan** | 05-Nov-11 at 11:12 am | Permalink

@PM HUT You are incorrect. I and many others are doing versions of Cory's idea. For example, it

seems the majority of those using AgileZen
(agilezen.com) are using it – see the discussion
boards.

140. **Kanban development oversimplified: a simple
       explanation of how Kanban adds to the ever-growing
       Agile toolkit** | 10-Nov-11 at 1:33 am | Permalink

       [...] Kind of a simple way to manage things
       when you think of it. Corey Ladas explains this
       well in his Scrumban article and [...]

141. **How To Have Scrum In Sustenance or Maintenance
       Projects – Agile In Support Work | The Agile Radar** | 12-
       Dec-11 at 8:08 am | Permalink

       [...] Scrum-ban technique is being adapted by
       many. It is a combination of Scrum and Kanban
       method. It mainly categorizes support tasks in
       to "Not started", "In progress", "Done" on a
       white board. Post its with task description will
       be used to categorize the current pool of tasks.
       For more do check this nice article. [...]

142. **Kanban Development Oversimplified** | 20-Dec-11 at 4:30
       pm | Permalink

       [...] Kind of a simple way to manage things
       when you think of it. Corey Ladas explains this
       well in his Scrumban article and [...]

143. **Kanban System Literaturliste | GAMEDEVPORTAL** | 04-
       Feb-12 at 3:24 pm | Permalink

       [...] Scrum-ban [...]

144. **Confluence: KTPA PMO** | 14-Feb-12 at 4:09 am | Permalink

       ScrumBan...

       ScrumBan is a combination of Scrum and
       Kanban, which is highly efficient for teams that
       perform product support and maintenance work,
       ......

145. **Confluence: Technical Documentation** | 15-Feb-12 at 9:42 am | Permalink

Development Processes & Tools...

Introduction This document gives a general overvie...

146. **Joaquin** | 24-Feb-12 at 8:03 am | Permalink

Hi Corey,

your article is Great.

Is there a way of buying the Scrumban book in a DRM-free version for Kindle? (I assume that the one sold in Amazon is DRM)

Best regards,

–Joaquin

147. **Kanban | Ariadna's Kanban Blog** | 28-Feb-12 at 10:00 pm | Permalink

[...] Scrum-ban by Lean Software Engineering [...]

148. **Scrum + Kanban = Scrum-ban** | 02-Mar-12 at 8:09 am | Permalink

[...] looked around at what others have done. On my ScrumMaster training I was introduced to the idea of Scrum-ban (thanks to Corey Ladas), and this excellent text by Henrik Kniberg and Mattias Skarin. Both [...]

149. **Scrum im Marketing – Was bringt es wirklich? Rückblick und Bewertung « Marketing mal anders** | 06-Mar-12 at 3:05 am | Permalink

[...] uns optimalen Grundlagen aus dem Scrum mit denen des Kanban-Prinzips verbunden und dies über ein Scrum-Ban-Board [...]

150. **Kanban in IT | PM - Dive** | 08-Mar-12 at 7:32 am | Permalink

[...] the use of a backlog, ready, specify, complete, execute, done. Further reading can be

found in this Blogpost about Scrum-ban by
Corey Ladas. Possible Scrumban [...]

151.   **Harvin** | 21-Mar-12 at 11:15 pm | Permalink

I adopted Scrumban for a maintenance team
which handles new features as well as bugs. Our
Scrumban board comprises of To Do,
Development, Testing, Deploying and Done
columns. We often have bugs escalated by the
support team which needs to be handled
urgently. The escalations could either involve
development and testing together or just
testing (for verification) alone. Escalations
causes the current work to either stay idle or be
moved back to the To Do column (if the columns
have reached the WIP limit) which then
increases the lead time.
I am guessing this is something normal and
acceptable. Would like to hear from anyone with
similar experience and how they tackle this
situation.

152.   **Andreas Rades** | 22-Apr-12 at 1:01 pm | Permalink

super article about scrumban. Seems to be a
good approach for (software) product
development.

153.   **A non-agile, non-waterfall, winning approach to
software development « Business Analysis Resources** |
26-Apr-12 at 12:06 pm | Permalink

[...] the emergence of agile approaches, one of
these firms has now adopted an agile model
similar to Scrum-ban for projects that fit the
agile sweet spot. For other of types projects,
NANW continues to be a [...]

154.   **Development Processes Linklist | Lars Giere** | 30-Apr-12
at 1:49 am | Permalink

[...]
http://leansoftwareengineering.com/ksse/scrum-
ban/ Posted in Lists [...]

155. **Confluence: Development Process & Practices** | 01-May-
      12 at 2:20 am | Permalink

      Scrum and Kanban...

      I.   What is Scrum A light weight frame...

156. **Confluence: Pyxis Body of Knowledge** | 01-May-12 at 8:10
      am | Permalink

      1.9 Processus de travail...

      Description Ressources externes Ressource Note
      Scrum...

157. **Kanban IT Literaturliste | GAMEDEVELOPERS
      COMMUNITY** | 03-Jun-12 at 9:07 am | Permalink

      [...] Scrum-ban [...]

158. **Evan** | 07-Jun-12 at 5:44 am | Permalink

      Works great when you only have 10 features
      that can be explained on a post-it note.

159. **Kanban – The Hard Way! | James Philipps** | 25-Jul-12 at
      4:03 am | Permalink

      [...] being more work-focused than people-
      focused (indeed, a merging of the two
      techniques exists called Scrum-ban , focused on
      getting the best of both [...]

160. **Revue de Presse Xebia | Blog Xebia France** | 14-May-13 at
      1:00 am | Permalink

      [...] traduction de Fabrice Aimetti, de l'article de
      Corey Lada "Scrum-ban", décrivant toutes les
      étapes du passage d'un processus Scrum à un
      flux tiré kanban, [...]

Post a Comment

Your email is *never* published nor shared. Required fields are
marked *

Name [   ] *

Email   [   ] *

Website [   ]

Message

[Post]

six + 5 = [ ]