

Quotable KANBAN



Janice Linden-Reed, editor
Foreword by David J. Anderson

Quotable KANBAN



Useful and Inspiring Words

By Practitioners of

Kanban for Knowledge Work

From Around the World

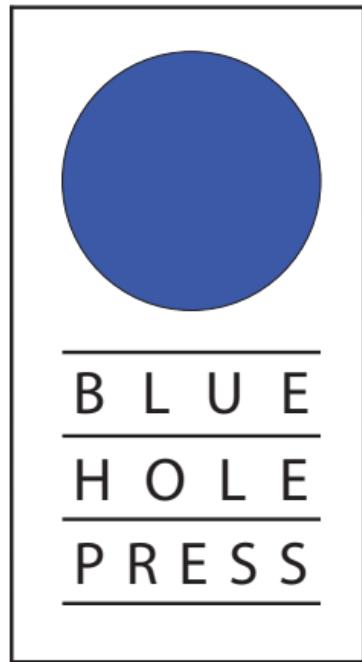


On the occasion of the 2011 Kanban
Leadership Retreat in Reykjavik, Iceland

David J. Anderson and Associates, Inc.



The opinions expressed here are the authors' and contributors' and do not express a position on the subject from David J. Anderson and Associates, Inc.



Sequim, WA

Contents

This commentary on...

Foundations

Transitioning

Visibility

Limited WIP

Measurement

Flow

Policies

Quality

Improvement

is presented to you by...



**TEAM
PROSOURCE**

and by...



Vega Information System Services, Inc.

Lean-agile, Kanban, XP
Team Lead/Coach & Software Architect

Hands-on mentoring and embedded coaching to development teams in applying lean-agile principles and practices to deliver quality software faster.

Emphasizing kanban software development w/ XP practices, domain-driven design, pattern oriented architectures, and OO design.

Contributed Articles

Why Lean-Kanban?

No Board Without a Verb!

Kanban and the Focus on Fundamentals

Estimates Are Overrated

Kanban and Lean Development with GreenHopper's Rapid Board

Products and Services

SwiftKanban by Digite

GreenHopper by Atlassian

LeanKit Kanban by Bandit

Team ProSource training

**Lean/Kanban materials
by Blue Hole Press**

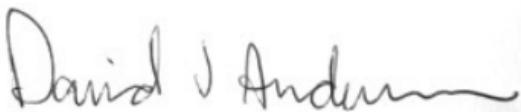
Foreword

The Kanban Leadership Retreat happened in response to requests for a gathering of the most highly trained Kanban leaders. Managers, developers, and change agents who had been through Kanban Leadership training were invited from around the world. 42 kanban leaders from 16 countries came to Iceland and explored cutting edge topics in lean and kanban when they were not exploring the beauty that Iceland offers.

Not all the great people in the global kanban community could make it. This ebook shows off some wisdom offered by those who did. This is a collection of comments from Twitter, blogs, discussion group, books, and interviews before,

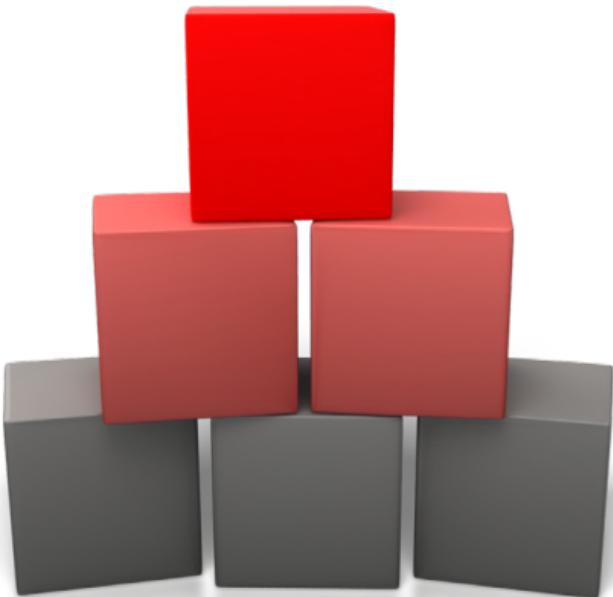
during, and after the Kanban Leadership Retreat. I hope you find it inspiring and useful as you take your own journey of continuous improvement through kanban.

I look forward to seeing you in a Kanban Leadership Workshop or even at the next Kanban Leadership Retreat.

A handwritten signature in black ink that reads "David J. Anderson". The signature is fluid and cursive, with "David" on top and "J. Anderson" below it, all written in a single continuous line.

David J. Anderson

Foundations



Foundations

Kanban's 5 core properties (Visualize, WIP Limits, Policies, Measure, Improve) include basic, intermediate, and advanced practices.

— Rick Simmons
(United States)

Kanban establishes a new balance on the relationship between people and processes, establishing a way to empower people to design their own processes. Hence, Kanban is a collaborative exercise for process design and offers thinking and action tools to empower people to evolve processes by themselves.

— Alisson Vale
(Brazil)

Foundations

People ask me, “What is the difference between lean and kanban?” Answer: lean is a destination; kanban is a means to get there.

— David J. Anderson
(United States)

Kanban is the design of a delivery system through collaboratively studying, envisioning, limiting, sensing, and learning.

— Karl Scotland
(United Kingdom)

Foundations

Kanban's value is providing insight. Use it and think of other visualizations and metrics frameworks to provide different insight.

— Larry Maccherone
(United States)

Is the hardest thing in kanban to understand and accept the concept of capability?

— Jesper Boeg
(Denmark)

Foundations

Card walls are not inherently kanban systems. They are merely visual control mechanisms. They allow teams to visually observe work-in-progress and to self-organize, assign their own tasks, and move work from a backlog to complete without direction from a project or line manager. However, if there is no explicit limit to work-in-progress and no signaling to pull new work through the system, it is not a kanban system.

— David J. Anderson
(United States)

Kanban fails when people don't want to face the truth.

— Hillel Glazer
(United States)

Foundations

Demand analysis is the backbone of the Kanban Method and lies at the basis of decision making. You'll see that organizations not studying their demand are also not implementing Classes of Service which means they are not able to allocate against shifting variability and more likely to end up with a static Kanban Board.

— Maarten Volders
(Belgium)

Kanban is a gateway drug to lean.

— Chris Hefley
(United States)

Foundations

Our efforts with point estimating and task breakdown/hours sizing were never as accurate or useful to us as were our efforts with focusing on cycle times, delays, limiting WIP, and swarming.

— Frank Vega
(United States)



Working with your distributed team like they were right next to you?
It's gotta be Swift-Kanban!!

Connect. Communicate. Collaborate. Wherever and whenever.  **Swift** from digité

Swift-Kanban - a simple yet powerful web-based Kanban tool, which facilitates distributed or collocated teams manage their work based on key Kanban principles!

Use Smart Lanes to visualize your value-stream, define WIP limits, manage your backlog and releases, collaborate with your team, and measure and improve your process and deliver increasingly better value to your customer.

Available in both **SaaS** and **on-premise** delivery models!

Free offer!



Signup for Trial and get an **Additional** month of the Team License for **FREE!**
<http://www.swift-kanban.com/signup>

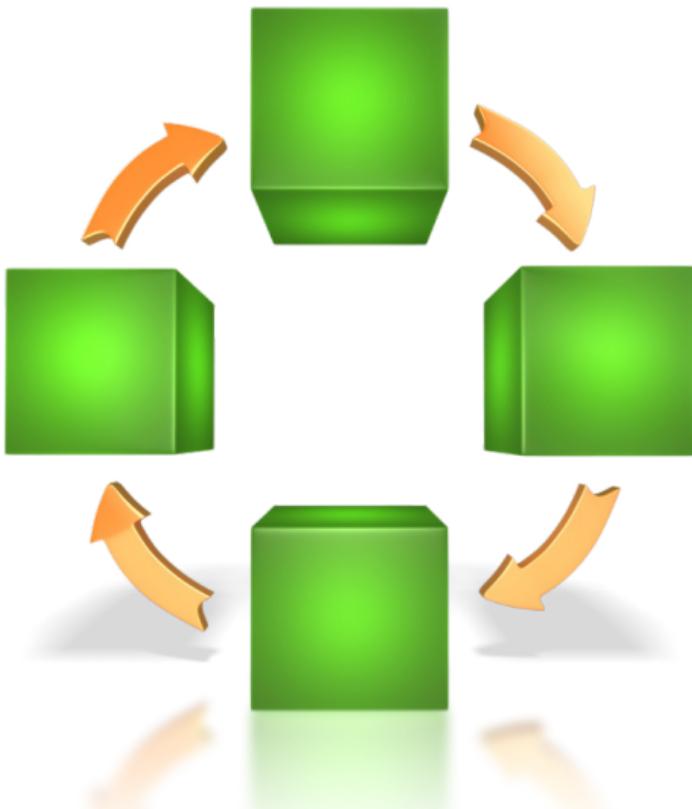
*Offer valid till 31st Oct, 2011. Use Promotional Code: **ICE2011**

digité
How work-really gets done

Compete. Differentiate. Win.

To learn more, visit www.swift-kanban.com

Transitioning



Transitioning

If so few people really “get” Lean and Agile, then we should be looking at those doing the explaining. That would be us.

— Marius de Beer
(South Africa)

Having huge respect on the impact change make on people, we were very careful with making only a few changes. And we took great care in explaining why we thought applying these changes were a good idea. Ultimately, you want the team to pull a change into the process instead of pushing a change on the team.

— Ketil Jensen
(Norway)

Transitioning

Context is relevant, so Kanban allows you to design processes that fit to the context, instead of manipulating the context to fit a specific process

— Allison Vale
(Brazil)

The Kanban Method can be viewed as a complex adaptive system that is intended to catalyze a lean outcome within an organization. It was seed conditions, adaptive simple rules, several feedback mechanisms, catalytic probes and attractors, to drive (somewhat predictable) emergent behavior.

— David J. Anderson
(United States)

Transitioning

IMO the decisions made at different levels in the org differ in detail, but the same heuristics apply ... if principles are valid.

— John Clifford
(United States)

I've provided value to teams (happened to be non-software teams) with just one or two organizing meetings around Kanban. Even if they do nothing more than visualize their work, it delivers value.

— Rick Simmons
(United States)

Transitioning

If you want to sell kanban to senior management, learn about their pressures and needs.

— Simon Marcus
(United States)

A well designed Classes of Service package with good knowledge of Cost of Delay and a better strategy for managing market risk can be a good start to open a dialog within an organization. Only when you have all the Kanban pieces in place, a normative experience from your CFO can create a higher emotion, which pushes the dialog towards action.

— Maarten Volders
(Belgium)

Transitioning

Your situation is unique and you deserve to develop a unique process definition tailored and optimized to your domain, your workflow, the risks that you manage, the skills of your team, and the demands of your customers.

— David J. Anderson
(United States)

Turn “thou shalt ...” into “how do we...”

— Hillel Glazer
(United States)

Transitioning

So you can start with doing a lean transformation and that's always a good way to go especially if you've got top-down buy-in from all the executives, but oftentimes that's not the case. Even a small team, even one that doesn't have a whole lot of power in your organization can start by visualizing their work with a kanban system and use that to drive change and to drive improvement, to drive the beginnings of a lean journey.

— Chris Hefley
(United States)



LeanKit

Kanban



Easy to Use

LeanKit is as simple as sticky notes on a whiteboard with all the power of Kanban to instantly communicate status.



Global Availability

We're cloud-hosted, supporting your teams wherever they may be - PCs, touch screens, and iPhones & iPads via our new iOS app.



Flexible Design

Our board editor allows you to quickly and easily map even the most complex workflow with an unlimited number of vertical and horizontal lanes and sub-lanes.



Powerful Metrics

Every card move is tracked to generate metrics on cycle time, variability, efficiency, workload distribution, and what-if forecasting.



Visibility



Visibility

One of Kanban rules is visualization.
This is one I like the most since I
just can't sit at ease next to the clear
whiteboard. I have that urge to grab
a marker, a bunch of sticky notes and
make the board a little less white.

— Paweł Brodzinski
(Poland)

Making things visible on high level is
already a disruptive change!

— Markus Andrezak
(Germany)

Visibility

By calling something an impediment, we are making a judgment, which could create conflict. Instead, visualise the data.

— Karl Scotland
(United Kingdom)

When I describe Kanban I emphasize it is about visualizing (indiscriminately) instead of “Visualize process workflow”. I think teams shouldn’t stop at the process workflow; it’s only part of the visualization. Any information important to the team and stakeholders needs to be presented and visualized.

— Olav Maassen
(Netherlands)

Visibility

My approach now, to give us better predictions of lead/cycle times, is to even further visualize the value stream and apply improvements to the actual uncertainty in our process. We have surely a lot of possible bottlenecks not visualized or identified.

— Mats Yforgs
(Sweden)

Make the need for change visible, obvious, and undeniable.

— Joshua Bloom
(United States)

TEAM PROSOURCE

Kanban for projects

www.TeamProsource.eu

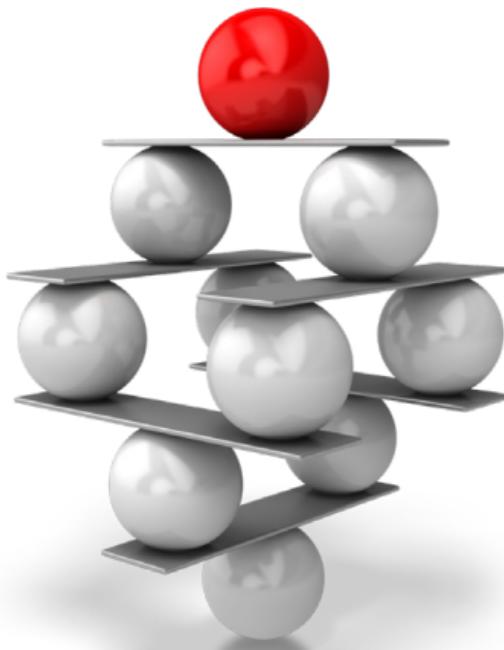
TeamProsourceEU



Follow us on LinkedIn

FOLLOW US ON twitter

Limited WIP



Limited WIP

A Kanban pull system cannot be overloaded if capacity has been set appropriately.

— Dominica DeGrandis
(United States)

I think one of the main reasons behind the project success was the teams' ability to do small positive changes helped by the incredibly powerful technique of limiting work in progress.

— Ketil Jensen
(Norway)

Limited WIP

Little's Law means that WIP is a leading indicator for cycle time, whereas things like velocity or throughput are trailing indicators (metrics).

— Daniel Vacanti
(United States)

That we recommend cadence as an improvement to many contexts doesn't mean we are doing sprints. Limited WIP over sprint commitment.

— Yuval Yeret
(Israel)

Yes! WIP limits, properly used, compel us to learn.

— Daði Ingólfsson
(Iceland)

Limited WIP

Once you balance demand against throughput and limit the work-in-progress within your workflow, magic will happen. Only the bottleneck resources will remain fully loaded.

— David J. Anderson
(United States)

Measurement



Measurement

Metrics can pull or be pulled. Metrics should be backed with a theory or be used as canaries.

— Joshua Bloom
(United States)

Data brings existing conflicts between expectation and reality to the surface. You generally want to resolve these conflicts by changing behavior to bring reality into alignment with expectations or altering the expectations. Not looking at the data is the equivalent of ignoring the conflict.

— Larry Maccherone
(United States)

Measurement

Define measures that create knowledge about whether the current Kanban system design is meeting its purpose.

— Hermanni Hyytiälä
(Finland)

Statistical Process Control charts show where the process is unpredictable.
Driving out variability in your system results in greater predictability.

— Dominica DeGrandis
(United States)

Kanban with GreenHopper for JIRA

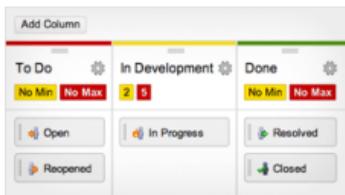
[GreenHopper](#) adds agile project management to any JIRA project. Whether you're a seasoned expert or just getting started with Lean and Kanban concepts, GreenHopper is perfect for visualizing your existing process and stimulating incremental improvement.

Principles of Kanban

Kanban is a catalyst for change through small, incremental improvements to your existing process – be it scrum or otherwise. Rooted in Lean manufacturing, Kanban is a signaling system that can be effectively applied to software development, DevOps, IT Operations, and many other processes.

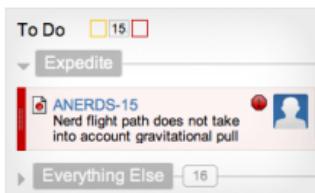
Visualize Your Workflow

First, you'll want to map the columns of the [Rapid Board](#) to the steps of your existing workflow. Quickly populate the Rapid Board with any unit of work – like user stories or bugs – using a JQL Filter. You can also use JQL to create Quick Filters for viewing select data on your board.



Limit Work In Progress

Set minimum and maximum limits on each column to identify spare capacity (yellow) or bottlenecks (red), respectively. This is your Kanban – or signal. Limiting work in progress creates a "pull system" filling the "in-process" columns from right to left across your board to optimize flow.



Horizontal Swimlanes

Separate work items on your Rapid Board using Swimlanes to help map out your value stream by team, class of service, or any other categorization. Swimlanes are creating using JQL similar to Quick Filters.

Manage The Flow

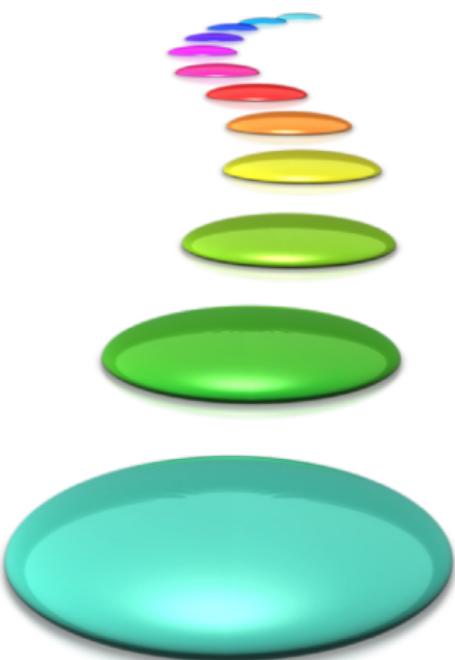
View Control Charts to measure average cycle time – the time issues spent "in process" – and identify specific outliers. Cumulative Flow Diagrams also show historical work-in-progress providing indication of the average lead time, major bottlenecks, and variabilities in your process flow.



Try GreenHopper»

Free 30-day trial

Flow



Flow

Finding it increasingly hard to avoid using Kanban on Agile projects, since Kanban so obviously exposes and relieves the real bottlenecks

— Jesper Boeg
(Denmark)

Today I've seen a glimpse of the PMO of the future: project differentiation, classes of service.

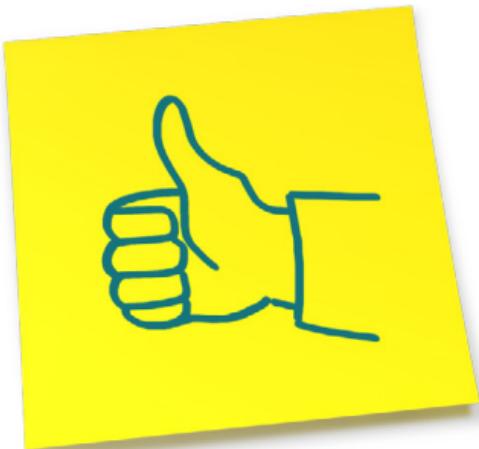
— Patrick Steyaert
(Belgium)

Flow

A lot of small tasks are failure demand, i.e., they are tasks created because we have failed to deliver the actual value in some way. If we can collect data on this and analyze it we might be able to get rid of the root cause of the failure demand and remove it, freeing up resources that instead can be focused on delivering value.

— Joakim Sundén
(Sweden)

Policies



Policies

Thinking of a process as a set of policies is a key element of the Kanban Method.

— David J. Anderson
(United States)

I observe that some Kanban teams write policies like the 10 Commandments.
I prefer policies that tell me what 2 do instead of what not 2 do.

— Klaus Leopold
(Austria)

A WIP Limit is a policy. All of the core kanban properties can be represented in policies. The challenge, and crucial element, is that the policies are understood and willingly agreed to.

— Janice Linden-Reed
(United States)

Policies

Customer satisfaction is king. Policies on cadence-based continuous flow delivery are a golden gift to the king.

— Masa Maeda
(United States)

Policies

The kanban system relies on the policies being explicit, and the board should certainly reflect those policies, but I have never seen a board that by itself completely defines the policies. I have seen many come close by posting the working agreements for each phase's ready and acceptance criteria on (or next to) the board, and by hanging each role's "pick order" near the board, but I can't think of any teams that use only the board to make the policies explicit. Remember that the board is a visualization and a model, it is not the system.

— Eric Willeke
(United States)

Policies

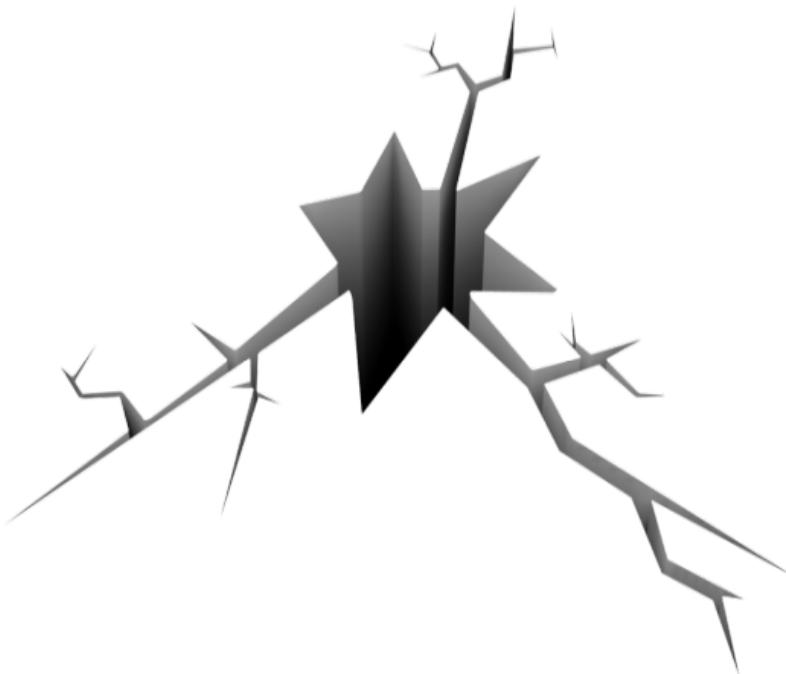
Today we squeezed 24 priorities that nobody understood into 3 classes of service that everybody understood within a minute! Yes we Kanban :-)

— Klaus Leopold (Austria)

The mindset of striving to minimise all work outside the “Build business value” category seems at best simplistic, at worst blinkered both to risk and to the bigger economic picture. Rather than minimise or ignore these other aspects, it seems much more useful to make choices and policies explicit and invest carefully across classes. A portfolio-based approach if you like.

— Mike Burrows
(United Kingdom)

Quality



Quality

Defect estimation is waste. Either you have to fix them, so it doesn't matter, or you don't ... so it doesn't matter.

— John Clifford
(United States)

The cost of failure is not just the cost of correction but also the cost of building bugs!

— Hillel Glazer
(United States)

Building the right thing and have high defect rate is much better position than building the wrong thing flawlessly.

— Paweł Brodzinski
(Poland)

Quality

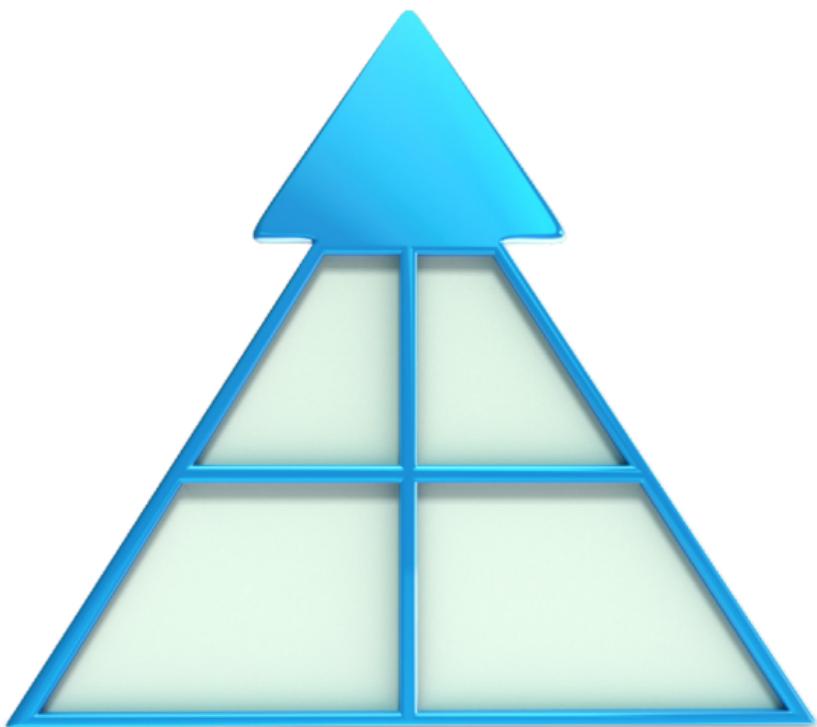
Probably Deming would have said: “It’s nonsense to blame testers for the quality of the software. Quality starts at the board of directors.”

— Klaus Leopold
(Austria)

Reducing work-in-progress, or shortening the length of an iteration, will have a significant impact on initial quality. It appears that the relationship between quantity of work-in-progress and initial quality is non-linear; that is, defects will rise disproportionately to increases in quantity of WIP.

— David J. Anderson
(United States)

Improvement



Improvement

Initially it is called continuous irritation,
eventually continuous improvement.

— Katherine Kirk
(United Kingdom)

Waste is the same way ... you can be actively spending a ton of time arguing about it, focus a lot of people on trying to find it or identify it ... instead, help grow the culture to where waste doesn't take hold ... it just sheds itself out of the organization naturally.

— Eric Willeke
(United States)

Improvement

I have seen teams that adopt Kanban and then stop innovating. They go through the mechanics but miss the drive to challenge their own process any further. What results is only minor steps.

— Olav Maassen
(Netherlands)

You can't add things to your process to fix a process. You have to remove the dysfunction.

— Håkan Forss
(Sweden)

Improvement

The slack capacity created by the act of limiting work-in-progress and pulling new work only as capacity is available will enable improvement no one thought was possible.

— David J. Anderson
(United States)

What is all this continuous improvement for? Context should drive answer, but question should be asked. Otherwise, what is success?

— Simon Marcus
(United States)

Improvement

Kanban allows leadership to flourish in every mind. It is permaculture applied to knowledge workers. Kanban helps management to switch to distributed leadership.

— Alexis Nicolas
(France)

Kanban is a catalyst for process improvements, not a development process. Just a reminder.

— Håkan Forss
(Sweden)

XOXO

Waiting for bus to airport. Goodbye
Reykjavik, thanks for all the fish! Had a
great time at #klris. Met amazing people.
Love this community!

— Joakim Sundén
(Sweden)

Why Lean-Kanban?

By Alan Shalloway and Jim Trott,
Net Objectives

Software development across an enterprise involves more than efficiently managing a collection of teams. It requires managing a vision across the organization, prioritizing work to realize that vision, and creating contexts and methods that let teams thrive and deliver value. This article describes Net Objectives' own journey and why we feel Lean and Kanban, taken together, make for effective software development at the enterprise.

Net Objectives has been in the business of helping transform software development organizations for over 12 years. We started with technical practices: design patterns, refactoring, test-driven development, and so on. From there, we expanded to process and

management practices: starting with Scrum, then Lean, and most recently adding Kanban. Along the way we broadened our offerings to include practices that span technical and process, such as acceptance test-driven development and emergent design. We expanded to support more of the value stream such as product portfolio management and business consulting. Why do we keep expanding our offerings? Because we have found it is not enough to fix individual pieces of the development process. Like “whack-a-mole,” fixing one piece at a time merely raises up another problem somewhere else. And the common approach – starting out by introducing Agile practices to a team – yields even more uneven results. You may truly help the team to improve; but if the team is not what was impeding software development, you have not significantly helped. No, it requires a holistic approach in order to create effective IT and

Why Lean-Kanban?

By Alan Shalloway and Jim Trott, Net Objectives , © 2011

product development. Technical, process, management, value stream, portfolio, ...

This is not a simple task. It has been over a decade of learning for us.

As development organizations get larger – 50 or more people – there is a desire to find an overall method that will work for every team. Consistency, it is hoped, will help teams collaborate more effectively. It rarely works that way. The truth is that teams work in different contexts. They face different challenges technically and with customers. They are comprised of different people. They have had different experiences. Uniformity is just not possible: there will always be some differences. At the same time, “every team gets to choose for themselves” is also not desirable. Collaboration would certainly not happen as effectively as required.

Scrum offered a middle path between the two. And it has indeed proven to be extremely successful in many organizations for many teams. And yet, as good as it was at the team level, it has rarely scaled throughout the organization. Where it has scaled well, it required an external force such as a CEO mandate.

The heart of the problem lies in resolving the dilemma of integrating a customer focus at the team level with the business/management focus required for large-scale adoption of agile methods. Focusing only on a team and giving them power to delight customers is good but may result in a loss of business focus. A management focus on only business value, however, may leave teams disempowered, weakening their ability to innovate. Somehow we need to create a context within which empowered teams can thrive in their own context while also working on those aspects of IT or product development that provide the most value

Why Lean-Kanban?

By Alan Shalloway and Jim Trott, Net Objectives , © 2011

to the business. As good as it is, Scrum and other team-based approaches are insufficient. Agility at the business level involves different approaches than agility at the team level.

Here is what we have learned. Lean thinking provides the necessary mindset to “see globally but act locally.” It provides a global perspective that helps the organization take local actions that optimize the whole. By creating a global perspective, it helps provide clarity on which projects that add the biggest value to the business. Kanban then equips teams to self-organize in order to manifest that value. And this is why we added Kanban to the mix.

But Kanban is more than a tool at the team level. A cornerstone of Lean is “respecting people.” This goes beyond the common view to “let people figure out how they must work.” In Lean, respecting people means to respect how they react to change, how they work in

groups, how groups of people interact, and how people learn best. Kanban was specifically designed to include all of these aspects of respecting people. Kanban is an integration of how people behave with how the work is done. Using Lean principles, it lets organizations start where they are, lets teams adapt and improve through fact-based incremental steps, gives tools to even out the flow of work, and gives the business the means to prioritize work based on value.

The combination of Lean, Kanban, and agile and technical practices, offers a cohesive set of perspectives that are consistent with each other and work at all levels of the organization. It gives the organization enough practices and principles to guide their work while avoiding a prescriptive approach that may or may not work (on the one hand) or having to figure it all out for themselves (on the other).

Why Lean-Kanban?

By Alan Shalloway and Jim Trott, Net Objectives , © 2011

It has worked successfully at all sizes of organizations. And putting them together has been a journey of discovery.

Why Lean-Kanban?

By Alan Shalloway and Jim Trott, Net Objectives , © 2011

No Board Without a Verb!

By Arne Roock, IT Agile

You might be familiar with the phrase “No sentence without a verb!” from your school days. Using a nominal writing style is not only considered bad habit in language lessons. The same also holds for Kanban: “No board without a verb!”

What does that mean? When people design a new Kanban system and they’re coming up with the relevant process steps, they often end up with something like “Offer required” or “In Quality Gate.” First I thought it was just me as an external coach who didn’t really understand what this should mean. Then it became clear that even the team didn’t know what this meant. Does it mean the client requested an offer and now the task is waiting to be done? Or is there anybody working on this task

right now? And if so: What exactly is going on here?

Things become much clearer if we insist on using verbs for all column headings in case there's an activity going on. (As a side note: I don't consider "Waiting for" to be a verb in this context!) If we apply the rule then we get columns like "Create an offer," "Check with CTO," or "Perform Acceptance Tests." These are telling us very clearly what is going on in our system.

But sometimes teams struggle hard to come up with or agree upon a suitable verb. This means we triggered a very fruitful discussion about our processes.



The column headings should communicate clearly what is going on in the different process steps.

Of course it's not sensible to use verbs in each and every column! This holds for queues (“Testing Done”) and buffers (“Ready for Engineering”).

If we divide action steps from queues and buffers, things become simpler:

- We can have a discussion about who is in charge of performing which activity (and if our current assignments are useful).

- Often it's a real surprise to realize how often tasks are supposed to wait within our value stream.

This insight becomes even more striking with a Cumulative Flow Diagram. We can see the effect of all these queues and buffers over time because they're increasing our lead times.

I can't imagine a workflow without at least one activity in it (or even none). But for some kind of boards it might be okay to visualize very few activities. The reason for this is that in some cases you perform the activities so quickly that it would be too much overhead to even move the tickets to the respective column. And if the columns are empty all the time, then it might be worth considering deleting them – unless you want to show the fact that they are empty all the time.

At it-agile we're using a Kanban system for visualizing and improving our sales processes. Here we're dealing with several activities that only take a couple of minutes ("Send confirmation mail") while waiting for client replies takes way longer. So our board consists mainly of queues and buffers. And knowing that fact is of great value!

It becomes absolutely clear that it is of no use to improve our action steps. Instead we need to think about ways to decrease all the waiting times! Almost always this seems to be impossible, but in fact it's not!

Kanban and the Focus on Fundamentals

By Mahesh Singh, Digité, Inc.

Our Kanban journey began early in 2010 when we decided that we would build a product in the Kanban space that would address some of the basic issues we saw our prospects face in adoption of Agile methods such as Scrum and XP within their organizations that were historically used to doing waterfall or iterative or some hybrid Agile method that combined more than one type of processes.

While the presence of established competitors was a strong reason to look beyond the “popular” Agile methods, we also felt a strong appeal for Kanban existed because of its focus on 3 key fundamentals:

- Evolutionary change

- Improvement of existing processes
- Engineering rather than Management processes

We have seen numerous organizations and teams take on large (revolutionary) process improvement initiatives – be it 6-Sigma or CMMI or even Agile. We have seen them become consumed with “the task of process improvement” over and over again. As a result, while there is improvement in the interim, in terms of consistency with which a process is followed in the team or the organization, the overall magnitude in terms of the time, effort and cost, of making the transition becomes huge and management begins to question the benefits! Kanban, with its promise of evolutionary change, with one stroke, takes care of this fundamental issue. It allows teams to take on only those aspects of change that they can comfortably handle.

Secondly, with an evolutionary approach, Kanban necessarily tackles current processes in an organization and helps improve them, rather than force a range of new processes on it. This is fundamental to understanding how Kanban can be applied not just to traditional software processes – but also to popular Agile processes such as Scrum and indeed, to non-software processes – be they within IT or in general business functions such as Sales or Marketing or Legal! So besides being attractive to software teams, their pull for non-software teams, presented us with a much bigger opportunity, which we are already starting to see materialize!

However, the one aspect of Kanban that excites more than others is that it forces teams to focus on the “engineering” or the “delivery” processes rather than the “management” processes. This is where, I feel, Kanban truly distinguishes itself from its “competitors,” if

one may term them as that. Through the use of statistical control charts, Kanban helps teams identify their normal performance and their deviations from the normal – the outliers. Rather than advise teams to simply do better documentation or better management or better reviews, it encourages teams to do better root-cause analysis – and attack root causes for poor performance – be that better or more specific testing, better development or design practices or better collaboration and requirements elicitation from customers. And in doing so, I believe Kanban shows itself to be far more effective than any other framework to start making incremental, lasting improvements in the final quality of the product or service being delivered.

Far too often, after initial gains have been realized through ‘traditional’ Process Improvement initiatives, teams start to question the extra overhead of “following process”

and wonder “what is in it for them.” With Kanban, these teams are much more likely to realize lasting gains through evolutionary, (incremental) improvements to basic existing delivery/ engineering processes and measuring their own performance in a much more meaningful manner.

Estimates Are Overrated

By Markus Andrezak

Wherever I go, whomever I meet, in each and every organization, estimates seem to be a problem (and mine is in no way better). There are various sources of problems with estimates. The most common symptom I see is teams taking too long to estimate. This leads to the typical complaint that estimation method X bears too much overhead, the meetings are too inefficient, boring and cumbersome, and finally the whole of method X is under question.

A typical situation in this context is a long-ish discussion: whether feature US12 has 5 or 8, maybe even 13 feature points. Gosh! You can read books on this issue. They even tell you what to do. This situation is supposed to be a clear indication of uncertainty in

requirements, which you can resolve in several ways, most of them involving even more overhead. Heavens no ... the enemy enters the stage: Variability! The PO must really be a prick! A more modest assumption may be that the team is unsure of the implementation approach, architecture, design, task breakdown, yadayadayada.

First of all: I don't blame the teams in this scenario. The environment in which they live must have signaled to them – in whatever subtle way – that their estimations matter. Hell, the estimates don't only matter, they are part of your commitment and before you say "f\$%^ no," consider that the team's commitment is a commitment to a timeline depending on no less than a tight sprint plan and release plan, which again depends on the correctness of all estimations of the other projects, on the limited variability of all the features any talented PO can think of, etc. If

you leave this blog for only 5 minutes and try to brainstorm on the dependencies of our current project you will become blind and depressed at the same time. If all those dependencies were really embedded in your ‘planning’ and thus in your estimations and your commitment you might as well stay home and never touch your keyboard again. But your rent needs to be paid, so you come back to your senses after the 5 minute break and another cup of tea, destined to go back to work tomorrow, no matter what. At least you now know that a) your estimate cannot contain all uncertainties and is only a best guess snapshot of your current knowledge, and b) your estimate does not matter that much given that context.

Another symptom is that teams cannot decide on the estimation process (planning poker based on Fibonacci, planning poker based on any natural numbers, splitting stories above

a limit of 21, estimate in days and numbers—no, stop, I'm back to just hours—, classical function point analysis!...) I'm getting dizzy over all the decisions you could feel forced to make. And people can get religious over it. They do.

What I will tell you now may let you relax a little over all those micro decisions and estimates. It may well be that what I think and say only matters for some areas, maybe 40% of software development. I personally think it matters for around 80% of software development (or all of it once one got the point). Anyway, here is what I think.

Don't care (too much) about your estimates. They don't matter too much for several reasons. One not so subtle reason is that while one might think it makes sense to get the variability out of the estimations by science and sound reasoning, I think that most of the

variability simply levels out statistically. One time you estimate too high, the other time you estimate too low. If you have a systematic error in your estimations, that will be learned by the team over time and level out this way.

The more subtle reason, which will be much debated when you bring up this topic, is that estimations were brought into the world for the wrong reason and support the wrong reason and they don't even do it well. Estimates support the wrong approach. Estimates are good for chasing timelines, but they are no good at all for judging value created, value to be created, or complexity of a certain topic. To explain that, let me change the perspective: Think of my job where I am working on a standing organization honing and fine tuning a standing Internet product for years. So, I am talking about product development right now, just to make the point more clear. If you are an Amazon, EBay, Google,

whatever revenue-generating product (even a small one) with a margin in the internet you are only talking about product development. By that I mean the constant refinement and changes to your product. And my statement is that timelines do not matter in product development in general. There may be some very few exceptions, e.g., the infamous press conference to announce your product change, when suddenly your product goes bang into another direction the same day at 10 a.m. If your Corporate Communications Dept. is clever they know how to pull the event off and make it a success without so much as generating an issue when a certain percentage of the product is not ready. (Yes, I'm saying that even in this case the exact timeline makes no sense). What matters is the right spirit, (fast) user feedback, ideally a constant flow of product changes to try out new angles and, if you're really good, an approach for handling

the white space out there to stay innovative and not getting “MySpaced” by another Facebook.

Timelines are only good for blaming people. Blaming people who have been stupid enough to commit to timelines they could not assume responsibility for and then couldn’t finish 110% of the features. The world is too complex to maintain timelines and have people getting shot at.

In general, going for the timeline is the same mistake as going for utilization. Reaching any or both targets does not prove a thing. You may keep a timeline and keep utilization high and still have a crap product. It’s the same as not committing to a Scrum sprint regarding a set of features or user stories but rather on a goal. And the goal would be something like, publish product part X – and in the sprint have a lively productive discussion on how this is feasible,

which features are necessary, which can be simplified or de-scoped, which ones need to be more elaborate etc. If you're lucky you can get closer to value generation by having users' opinions as a main influence in these discussions. And guess what? For this approach you don't need exact estimations. I go as far as to say you don't even need clear requirements. What you need is much harder to achieve but harder to describe: You need a clear goal and a deep and thorough common understanding of what it means to reach that goal.

In this scenario, no doubt, good estimations can help to come up with good options on the way of reaching the common goal. But being a few points off does not matter at all. But for sure, an organization with that level of understanding would never burn an individual or a team for not keeping a timeline. (And the teams themselves would be the first ones to

ask themselves what went wrong and punish themselves for being off too far.)

So, my point is: Set clear goals on a quite abstract level, estimate to catch a glimpse of your current understanding of the complexity of your current goals (or your next release, etc.). Reach those goals by an intelligent negotiating process with the team coming up with a set of good options and choosing the most pragmatic and realistic ones. In this setting, estimates support you on a very high level but need not be exact. As a rule of thumb, discussing an estimation of a User Story for more than one minute seems to be overhead to me. Really.

Now, why do I also think the same holds for any project rather than product development? Simply because product development is done in ... projects. So, a project is simply an organizational form and in itself means nothing. In the scenario of a software shop delivering

a project to a client, the project will simply be a part of some other product development process. And absurdly, the one project (yes, yours) cannot be so important with regard to the timeline as to treat it differently than product development should be treated. I know, I know – your client doesn't see it that way. But that doesn't mean you have to see it that way and you can still do an excellent job, delivering all you have to deliver.

To get a grip on the overall risk and variability of your current project, just listen to some examples. I once committed to building a major international job portal based on three lines of requirements in Excel with a team of six developers. I knew what I had to do. I had to meet the client each and every week to exactly find out what he needs, where we need to be strict, where we have leeway, finally to make clear to him that if I fail the problem is bigger for him than for me. What role would

exact estimations play in this context? Or: do you think that you're much better off in your current project, just because you have 157 User Stories in a DEEP product backlog? How deep is the level of common understanding of the details of those User Stories throughout the team? What if you overlooked that line that means you do not only have to integrate that SAP HR system but adapt it also? Or on a smaller level – what if your release cycle is 6 weeks and you miss the release window by a day? (Yes, you now lost six weeks based on a three day User Story). What use are your fine-grained estimations now? I guarantee you that the little variability you have in your estimations is negligible compared to the rest of the uncertainty in your project and that there is a) no way to drastically get rid of it and b) no cheap way to decrease that level of uncertainty to a very small level. Further it is just the wrong lever you're pulling.

The bad news is that what you have to do is much harder and on a different level, but much less technical with less process. You will have to be clear about what you want and you have to be good about what you're doing as a profession. This means you have to have open discussions in your team. Where estimates help you is in getting a good, shared sense of your basic speed.

If you ever get the chance to play a round of the getKanban board game, you will have a lot of counterintuitive learning experiences. One of them is that the variability of the User Stories pulled across the board plays a minor role. Lead time is king here, and the learning that waiting times dominate actual work times lets lead time dominate even more. That means the less lean a system is, the less important good estimations are. In fact, the less lean the system is, the less probable are good estimations. The other observation is that (nearly)

no one blames the variability of the dice (1200% in getKanban). And finally you will learn that simply by watching the lead time of your work together with the visualization of the workflow, the explicit rules and the daily discussion on what to do, you will never miss a beat. The teams, with all the variability in the game, very seldom miss a fixed date ticket. By playing getKanban you can experience the power of variability. You can also experience the minimal use of estimations compared to good prioritization and the creation of real opportunities by actively exploring variability.

Kanban and Lean Development with GreenHopper's Rapid Board

By Nicholas Muldoon, Atlassian

In this peek of the Rapid Board we will take a look at some key features of GreenHopper, features that are going to make Kanban development and operations teams jump for joy! Existing GreenHopper customers will be over the moon too, as the Rapid Board answers many of the most requested features, including the ability to visualise multiple JIRA projects at once, a separate field to track sprint information, permanent URLs for each page, ranking performance and auditing improvement, plus a ton of other great features.

Using the Rapid Board teams can visualise critical issues, minimise their work in

progress, and measure cycle time. Below we look at four areas:

- the perfect visualisation
- swimlanes and quick filters
- limiting work in progress
- reporting and wallboards

Build the perfect visualisation

The Rapid Board is built upon the flexibility and power of JQL, the JIRA Query Language. Using JQL brings flexibility to GreenHopper, as views can include issues from multiple JIRA projects. A **single team responsible for three products** might use the following statement for a GreenHopper view:

```
project in ("Angry Nerds", "FourWalls",  
"PairOn") AND team = Dreamteam AND is-  
suetype = Bug
```

DreamTeam		
Reporting	Only My Issues	
To Do		
GP-527	Single quotes in Confluence Page Title causes GreenPepper Page Decorator not to appear	View
GHS-402	Ranking is Incompatible with the Searchable Attachments Custom Field	View
GHS-474	Exporting Confluence page to PDF doesn't render embedded GH portlet correctly.	View
In Progress		
GHS-1357	As a GreenHopper user I would like the ability to see many jira projects in the one view	View
GP-799	PHPextension is limited to ~3 parallel executions (cf support La Poste)	View
GHS-1361	As a GreenHopper user I would like the sprint field to be separated from the fix version field so that I may manage my sprints independently of my release versions	View

Focus using swimlanes and quick filters

Leverage the power of JQL for swimlanes and quick filters as well. Preset swimlanes and quick filters are based on common defaults, but teams can change them to suit any need. Here we can see a team using an Expedite swimlane based on either priority or due date:

```
(priority = blocker OR duedate <= now())
AND resolution = Unresolved
```

Quick Filters are buttons at the top of the board to slice the issues however you see fit, adding another layer of flexibility for teams. A Quick Filter might be used as a toggle during your

daily ‘walk the wall’ during standup, to show just those issues updated in the last 24 hours, as shown in the lower figure.

The image contains two side-by-side screenshots of a Jira board interface. Both screenshots show a 'To Do' column on the left and an 'In Progress' column on the right. The top screenshot shows a 'Reporting' tab at the bottom with filters: 'Updated < 24h' (which is highlighted in blue), 'Unassigned', and 'Only My Issues'. The bottom screenshot shows the same reporting tab but with different filter options: 'Updated < 24h', 'Unassigned', and 'Only My Issues'.

To Do

- GHS-1 Chart board breaks when using context options (Filter/Assign to me)
- GP-527 Single quotes in Confluence Page Title causes GreenPepper Page Decorator not to appear
- GHS-402 Ranking is Incompatible with the Searchable Attachments Custom Field

In Progress

- GP-799 PHPExtension is limited to ~3 parallel executions (cf support La Poste)
- GHS-1357 As a GreenHopper user I would like the ability to see many jira projects in the one view
- GHS-1361 As a GreenHopper user I would like the sprint field to be separated from the fix version field so that I may manage my sprints independently of my release versions

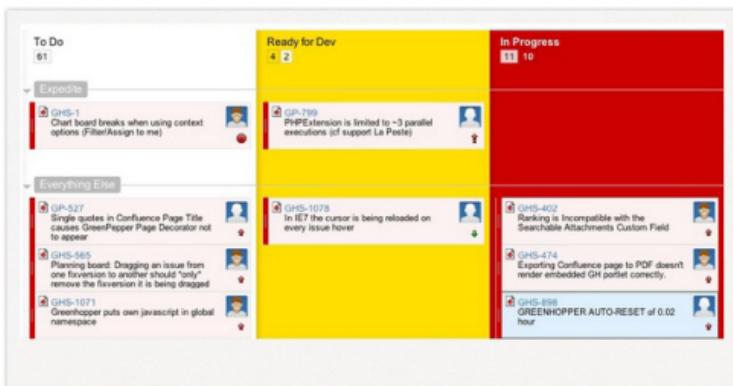
Quick Filters allow you to see specifics

Limit work in progress using column constraints

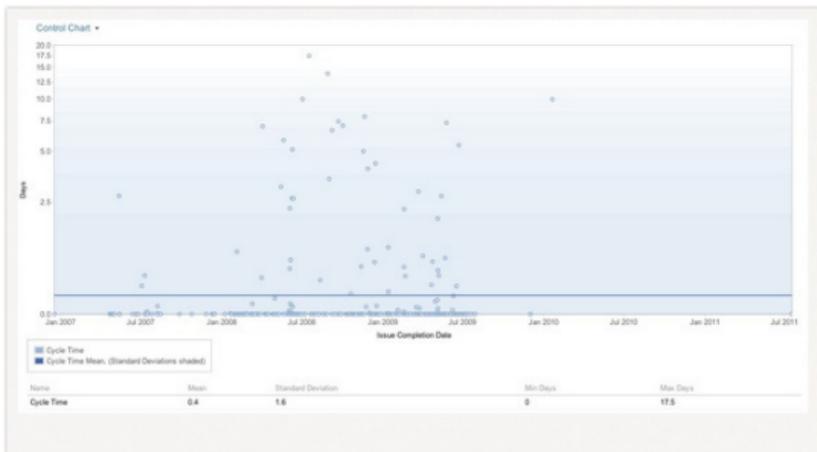
GreenHopper is flexible for however your team works. Whether your team limits work in progress (WIP) to ‘team size plus one’ or

‘team size in half’ to encourage pairing, maximum column constraints give an easy, visual indicator of WIP.

As a product owner I use a minimum column constraint on the ‘Ready for Development’ column. This prompts me to pull in more stories and replenish the queue as necessary. In the example below we can see both my need to replenish the queue (yellow) and that the team has too much work in progress (red).



Identify opportunities for improvement using reports



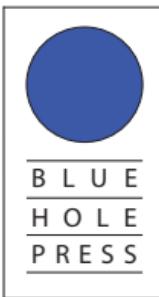
The Rapid Board includes a control chart and cumulative flow diagram. The control chart can be used to identify the lead time (time taken from when the story was raised), or the cycle time (time taken from when work began).



Of course, you can also take all of these great charts and place them on your information radiator using JIRA Wallboards. This is a great way to demonstrate what a team is working on at present, as well as sharing the progress of your team with the rest of the organisation. Build status, Twitter searches, sales and support statistics, plus loads more.

Many of the quotations by David J. Anderson in
Quotable Kanban are from his book,
*Kanban: Successful Evolutionary Change
for Your Technology Business*,
published by Blue Hole Press.

You can order *Kanban*
by David J. Anderson
as a print book or ebook at
<http://www.kanbaninaction.com>



Publishing books and materials
connecting the **GLOBAL**
Lean and Kanban Community

Books, Translations, Electronic Books, Whitepapers