Open Banana Booky da Kanban – For Smart Worky Stuff

Johnny Banana Coleman

2025-07-02T09:00:00Z

Before-Blabla

Dis doccy wanna giv *open and bendy* guidey for Kanban *and Zoom-Zoom, pulling togethery ideas from lotsa different tribes*. *It wanna be one togethery reference for dose tribes, on top of dere own stuff.* Depending where yu standy, lotsa other ways can holdy-hand with Kanban, so it can hug a big rainbow of Banana-giving and big-boss headache.

Da slanty-letters is dere for da Creative Commons adaptation notice on da front page; slanty-letters is NOT for shouty-emphasis. A Big Letter at da start of a word mean it is a termy-term rule listed in da appendix of dis doccy, e.g., Banana is a maybe-benefit or a got-it-already benefit for a Stakey-holder, includy meeting da needs of da customer, da end-user, da decidey-peep, da company, and da big blue planet.

What Is Kanban, Den — for Brainy Worky Stuff

Kanban is a planny for making da *Zoom-Zoom* of *Banana* betta through a *system*. It is a beep-beep system for calling in Work or stuff-on-da-shelf. It got da following tree do-dos, all holding hands:

- Makey and *Looky* a worky-flow.
- Bossing da *Worky Bits* in a worky-flow, all da time.
- Making da *Zoom-Zoom* betta.

When peeps actually do dem, all dese Kanban do-dos togethery is called a Kanban system. Da peeps who join in da Banana-giving of a Kanban system is called Kanban system peeps.

Why Use Kanban?

Right in da middle of *understandy* Kanban sit da idea of *Zoom-Zoom*. *In a Kanban system, Zoom-Zoom* is da moving of Banana through dat Kanban system. Because most *Kanban* worky-flows exist to make Banana big, da planny of Kanban is to make Banana big by making da Zoom-Zoom good, which mean hunting for da right balance of giv-da-right-stuff, no-wastey, and no-surprisey:

- A giv-da-right-stuff worky-flow is one dat giv stakey-holders what dey *wanty,* when dey *wanty* it.
- A no-wastey worky-flow spendy da available *room-to-work* da best way it can, to giv Banana.
- A no-surprisey worky-flow mean yu can *sensibly* guessy da Banana-giving inside a livable amount of not-knowing.

Da planny of *a* Kanban *system* is to *let* Kanban system peeps ask da right questions sooner-sooner, as part of a fix-fix-forever effort chasing dese goals. Kanban system peeps should aim for a balance dat last long-long time. *Kanban is also a way to squish down over-piling (too-much-work) and to boss da demand so dat Work get given out best-best for da room-to-work yu got. It is no perfect, but it should push constant fix-fix and a good Zoom-Zoom of Banana.*

Sidey-benefits is happier Kanban system peeps, betta quality, and being able to bendy to da demand. A good Kanban system boss itself, i.e., da Kanban system beep-beeps and tweaks itself for troubles with nobody poking it.

Because Kanban *can Looky* almost any worky-flow, it belong to no one job-kind only. Brainy Worky peeps in monies, in lighty-and-watery, in doctor-place, and in computery (to name just few), all got Banana out of Kanban do-dos. Kanban in most places where Banana get given.

Kanban Brainy-Think

Da Kanban system borrow from *lotsa ways and understandy* includy, but no only, systems thinky-think *(5)*, lean rules *(4)*, queue brainy-think (how big da batchy *(6-7)* and how long da liney *(1,13-14)*), wibbly-wobbly *(2,11)*, and quality checky-check *(2,8,10)*. Keeping a Kanban system getting betta-betta on top of dese ways and understandy is one way companies can try to make da giving of Banana good. Lotsa other *Banana*-loving ways share da *ideas* dat Kanban standy on. Because dey so samey-same, Kanban can be used to make dose giving-ways stronger.

Da Kanban Do-Dos

See Da Work: Makey and Looky Da Worky-Flow

Making da *Zoom-Zoom* good need yu first to say what *Zoom-Zoom of Banana* mean round here, *da (best-case) smooth moving and giving of maybe-benefits or (best-case) got-it-already benefits for Stakey-holders*. Da out-loud shared understandy of Zoom-Zoom among Kanban system peeps in dere own place is called a Definitiony of Worky-Flow. *Definitiony of Worky-Flow* is a big-big idea of Kanban. Ev'rything else in dis booky lean heavy on how da worky-flow got defined.

To tell yu how to run da worky-flow best and to help da fix-fix, at da teeny-minimum, Kanban system peeps must-must make dere *Definitiony of Worky-Flow* using ALL of dese bits:

1. A say-so for da single bits of Banana moving through da worky-flow, called *Worky Bits (or Bits)*.
2. *Depending on da Worky Bit, for at least one togethery 'started' and 'finished' point pair:*
   - ☐ One or more say-so'd states dat da *Worky Bits Zoom* through from 'started' to 'finished.'
   - ☐ *Worky Bits* between da 'started' and 'finished' points, even if dey sitting in a Queue or a Buffer, count as:
     - *'Started but Not Finished Work' (SNFW) or*
     - *Work in Progresso*/*Process* (WIP).
   - ☐ A say-so for how da WIP get bossed from 'started' to 'finished.'
   - ☐ *A bundle of* out-loud policies about how *Worky Bits* can *Zoom* through each state from 'started' to 'finished' *with no boo-boos*. *For example, Kanban system peeps might got a policy dat is out-loud about fixing any known boo-boo in a Bit before moving it to da next state, so dat no known boo-boo get passed to da next process.*
   - ☐ A *Time Expecty* (SLE): A guessy of how long a *Worky Bit* should take to *Zoom* from 'started' to 'finished.' *Noticey dat dere is no pinky-promise dat what happened before will happen again.*
   - ☐ A *Looky* of da *Time Expecty* on da Kanban boardy.

Da order yu do dese bits in is no importanty, long as dey ALL get *done*. Kanban system peeps often need extra *Definitiony of Worky-Flow* bits, like what dey value, what principles dey hold, and what dey promisey each other, depending on da *situationy of da* Kanban system *peeps*. *Dere is stuff in da appendix of dis booky and elsewhere to help pick da right options*.

Kanban system peeps also often need more dan one *Definitiony of Worky-Flow*. Dose many *Definitionies of Worky-Flow* could be for many gangs of Kanban system peeps, for different floors o' da company, etc. While dis booky say no minimum and no maximum number of *Definitionies of Worky-Flow*, it do cheer yu on to make a *Definitiony of Worky-Flow* wherever da Kanban system peeps need to gluey *Zoom-Zoom* to *Banana*.

Letting da Zoom-Zoom happen is da act of growing a smooth and balanced system dat make Banana. Da Definitiony of Worky-Flow should make sure da system is balanced so da Zoom-Zoom of Banana is good. Kanban system peeps do dis by getting betta at checking dat Banana really got delivered, and by throwing away Work dat giv no Banana.

Da *Looky* of one *or more Definitionies of Worky-Flow* is *called* a Kanban boardy. Dere is no special rules for how a *Looky* should looky. Thinky about all da bits of a *Definitiony of Worky-Flow* (like *Worky Bits*, policies) plus any other local thingies dat might sway how Banana *Zooms*.

In a computery team, Kanban might Looky da feature-making from idea to put-it-out. In a shouty-selling team, it might follow a campaign from drawy-drawy to launchy.

Kanban system peeps is limited only by dere own silly imagination for how dey make *Zoom-Zoom see-able and how dey grow on-purpose talky-talks with da right peeps at da right time*. *It is a good idea to Looky every step in a worky-flow so dat wastey stuff no stay hidden.*

Manage Da Work: Bossing Da Worky Bits in a Worky-Flow

Bits in da worky-flow must-must get bossed actively. *Active bossing of Bits in a worky-flow can looky like lotsa things, includy, but no only, dese:*

- *Boss 'Started but Not Finished Work' (SNFW) or Work In Progresso/Process (WIP)*.
- *Make sure Worky Bits* no get oldy-oldy for no good reason, using da *Time Expecty* as da looky-stick.
- *Un-stucky da stucky things dat is blocking Work or blocking processes*.

A common do-do is for Kanban system peeps to luk over da live *Bits regular-regular*. Dis luk can happen all-da-time or on a regular beaty. Kanban system peeps must-must out-loud boss how many *Worky Bits* is in a worky-flow from 'started' to 'finished,' straight-on or roundabouty. Dat bossing can be showed on a Kanban boardy any way *da Kanban system* peeps thinky is good.

Using WIP limits (16) in Kanban for Brainy Worky Stuff usually mean dat demand can get bigger dan da team's room-to-work, so WIP limits (16) is used to boss and balance da Zoom-Zoom of Worky Bits and stop da over-piling.

On da other hand, a Toyota just-in-time (JIT) pully system stop demand from getting bigger dan supply, because da next request no get served until da one before is done — a self-limiting or self-bossing system built to line up making-stuff with da real customer demand and keep da stuff-on-da-shelf teeny, in a steady, no-surprisey factory place.

Making only what is needed just-in-time is da corner-stone of da Toyota Production System. Da Kanban system in da Toyota Production System pull exactly what is needed when it is needed.

For Brainy Worky Stuff, Kanban system peeps should start Work on (*pick*) a Bit only

when dere is a clear beep-beep dat dere is room to do it. When WIP droppy below da control setted in da *Definitiony of Worky-Flow*, dat can be a beep-beep to pick new work. Kanban system peeps should holdy back from picking more *Work* into a given part of da worky-flow *beyond da matching WIP control(s) or picking Work bigger dan dere room-to-work. When needed, da Work should be choppy-chopped into smaller Bits dat still got Banana in dem.*

Dere is no rule saying yu must got a pile of Worky Bits dat is no yet Work In Progresso/Process, what peeps often call a backlog. A backlog just grow by itself and can hold all sorts of Work-getting-ready stages. If yu got one, it no need to be a listy shape and it no need to be in order.

Best-best, Work should go into da Kanban system guided by policies, no shoved onto one peep. In da hunt to boss idle work, no idle peeps:

- *Da Kanban system peeps should self-organize round da Work and da Definitiony of Worky-Flow.*
- *Kanban system peeps should 'start' Work when dey is ready to work on it, pulling in new Work by how it is being ordered.*
- *Kanban system peeps — and peeps outside da Kanban system too — should out-loud stop Work from being shoved onto Kanban system peeps.*
- *Watchy-out for re-ordering 'Started but Not Finished Work' (SNFW) or Work In Progresso/Process (WIP), because it make dose Bits go oldy (sit doing nothing) and lead to longer or wobblier Elapsed Times from 'Started' to 'Finished.'*

Rightsizing, an optional but recommended do-do, mean checking whether Worky Bits fit da Time Expecty, or is too big for da Time Expecty and so need choppy-chopping into smaller Worky Bits dat still got Banana in dem.

Rightsizing, in a Brainy Worky place, lean on da idea dat Worky Bits need to be at or under a biggest-size (whatever da Kanban system peeps say), but dey no must all be da same size. If a Worky Bit is so HUUUGE dat it can't get done in a sensible time (like, it would breaky da Time Expecty), even after starting it, Kanban system peeps should thinky about choppy-chopping it into smaller Bits dat each got da chance to giv Banana. Same-same da other way: Worky Bits can be squished togethery.

Bossing da room-to-work often need more dan just WIP control. Bossing WIP help da *Zoom-Zoom* and often make da togethery focus, promisey-keeping and work-togethery *of da Kanban system peeps* much betta. Any okay-okay exceptions to bossing WIP should be *said out-loud* as part of da *Definitiony of Worky-Flow*.

Fix Da Work: Making Da *Zoom-Zoom* Betta

Given an out-loud Definitiony of Worky-Flow, it is da Kanban system peeps' jobby to keep making dere *Zoom-Zoom* betta *by getting* a betta balance of giv-da-right-stuff, no-wastey and no-surprisey. Studying da system all da time can pointy-point at da fixes. *Kanban system peeps often luk over* da *Definitiony of Worky-Flow* to talky-talk and *take on da changes dat is needed*.

Fixes is often right-now-when-needed. Fixes is no limited by how big or how wide dey is. Sometimes a fix is outside what da Kanban system peeps can boss or bendy. On-purpose talky-talks, growing change, and taking away Blockers at every floor is da key to fixing.

Betta still, peeps who show leadership, also called leaders, Go See, Listen, and really understand, to gather da facts dat feed da deciding. Dis is called Genchi Genbutsu. Leaders do Genchi Genbutsu so often dat da truth pop out. Knowing what to do is one thing, but on-purpose, never-stopping, round-and-round, kind-hearted doing toward fixing (includy shorter feedback loops) is another.

Kanban like growy-growy change, but it no forbid bigger, structural change, guided by evidence and a clear understandy of da system. Changes should be on-purpose and fit da place dey in.

Feed Da Zoom-Zoom Fixing With Da Right Measure-Bits or Numbers

- **Blocked Elapsed Time for Finished Items (BETFI):** Da piled-up time dat one 'finished' Worky Bit (or a bunch of 'finished' Bits) spend in a stucky condition from 'started' to 'finished,' but no in a Queue or Buffer state. [measure-bit for one Bit, number for many Bits]

  [!FOOTNOTE] Da Definitiony of Worky-Flow should got a policy saying what a Blocker is (round here) and how to beep-beep about it.

- ***Cumulative Queueing or Buffer Time (CQBT):*** *Da piled-up time dat one 'finished' Worky Bit (or a bunch of 'finished' Bits) spend in Queueing or Buffer states from 'started' to 'finished.' [measure-bit for one Worky Bit, number for many Worky Bits]*
- ***Elapsed Time from 'Started' to 'Finished' (ETSF):*** Da (usually *rounded-up) count of elapsed time chunks (often calendar days) from* when one *Worky Bit* 'started' *to* when a *Worky Bit* 'finished.' *Only 'finished' Bits get ETSFs. [measure-bit]*
- **Flow Distribution:** Da Looky and chewing-over of what kinds of Worky Bit got 'finished' or 'completed' over time, so peeps can actively boss it and keep a healthy balance of effort. [number]

  [!FOOTNOTE] Da Definitiony of Worky-Flow should clearly say what any Queue and Buffer states is.

- ***Flow Efficiency:*** Da ratio of da busy-working time to da whole time a Bit or a bunch of Bits spend in da worky-flow, includy waity times, between da 'started' and 'finished' points on a Definitiony of Worky-Flow. *It is showed as a percentage. It can trick yu, because time sitting in busy states may no be real busy time. ((ETSF-(CQBT+other non-value-adding time))/ETSF) 100. [number] Example of other no-Banana-adding time: Blocked Elapsed Time for Finished Items*

- **Number of Blockers:** Da count of stucky things, part-way or all-da-way, at one point in time (usually right-now datetime), stopping da Zoom-Zoom of Worky Bits from 'started' to 'finished.' [measure-bit]
- **Process Cycle Efficiency:** Measure da Work-efficiency of a system or bits of it. Yu work it out by dividing da Banana-adding time by Time to Market and den timesing by 100 to get a percentage. Dis mean Kanban system peeps got to measure ALL da Banana-adding and ALL da no-Banana-adding time (includy, but no only, waity time). ((T2M-(CQBT+other non-value-adding time))/T2M) 100. [number]
- ***Time Expecty (Service Level Expectation):*** A guessy of how long a *Worky Bit* should take to Zoom from 'started' to 'finished.' Da *Time Expecty* itself got two bits: a chunk o' elapsed time and a how-sure number stuck to dat chunk (like, '85% of *Worky Bits* will be 'finished' in eight days or less'). *It is built from a pick of Elapsed Time from 'Started' to 'Finished' out of all history, a slice of history, or, if dere is no data or no enough data, a clever guessy. [number]*
- **'Started but Not Finished Work' (SNFW)** or **Work In Progresso/Process (WIP)** *or* ***Flow Load***: *Da* count of *Worky Bits* 'started' but no 'finished'. *[measure-bit]*
- **Throughputty:** Da count of *Worky Bits* 'finished' per chunk o' time. Da measuring of throughputty is da exact county of *Worky Bits*, *no monies. [number]*
- **Time to Market, also called Customer Lead Time:** Da (usually rounded-up) count of elapsed time chunks (often calendar days/weeks) from when a Stakey-holder's order for one Worky Bit come in, to when da Worky Bit got given to da Stakey-holder. It is one example of an ETSF. [measure-bit for one Worky Bit, number for a product or service]
- **Total Worky Bit Oldy-ness (TWIA)** or **Total Elapsed Time for 'Started' but Not 'Finished' Items (TETSNFI) :** Da whole elapsed time from when all da in-progresso ('started' but no 'finished') Worky Bits 'started' up to a said datetime, usually right-now. [number]
- **Worky Bit Oldy-ness (WIA)** or ***Elapsed Time for 'Started' but Not 'Finished' Items(ETSNFI)*** : Da (usually *rounded-up) count of elapsed time chunks (often calendar days) from da datetime dat one 'not finished' Worky Bit* 'started' *to a said datetime, usually right-now. By poking da oldy-oldy Bits first, feedback loops get shorter and da Zoom-Zoom get betta. [measure-bit]*

Da *Zoom-Zoom* numbers *and measure-bits* go with da right 'started' and 'finished' points dat da Kanban system peeps setted in dere *Definitiony of Worky-Flow*. *If dere is many sets of 'started' and 'finished' points, some zoom-zoom numbers and measure-bits often go on each 'started' and 'finished' pair.*

If Kanban system peeps no know where to start, dis booky suggest:

Time to Market, and for each togethery 'started' and 'finished' pair:

- *A Time Expecty (must-have for at least one 'started' and 'finished' pair),*
- *Worky Bit Oldy-ness or Elapsed Time for 'Started' but Not 'Finished' Items (ETSNFI),*

- *Elapsed Time from 'Started' to 'Finished' (ETSF), and*
- *Throughputty.*

Long as Kanban system peeps use da *Zoom-Zoom* numbers *and measure-bits* like dis booky say, *and dey fit da place dey in,* dey can cally dem any other names dey fancy. It is up to da Kanban system peeps to decide da best way to *use* dese *Zoom-Zoom* numbers *and measure-bits, like Looky-ing dem in charty-charts or checking da wibbly-wobbly. A leany-forward focus on outcomes, effecty, and Banana is a good idea.*

Outcomes, Effecty, and Banana

Kanban system peeps should regular-regular hunt for evidence of outcomes/effecty, e.g.:

- *Customer outcomes could focus on giving measurable Banana to customers, e.g., less Failure Demand, customer long-time cost squishing, or customer jobs got done (18).*
- *User outcomes could look hard at da exact changes in user behaviour dat fix problems or make experiences betta, e.g., 'completing' Worky Bits betta at da teeniest costs, or easier-to-use stuff.*
- *Product Stakey-holder outcomes could gluey dose behaviour changes to product numbers, like trends in product customer take-up, sticky-ness and convergence, plus trends in feature take-up, decidey-peep and user numbers, and product Time to Market.*
- *Business Stakey-holder Effecty, e.g., rule-following, business long-time cost squishing, business results, trends in market share, customer happy-ness across all products, etc.*
- *Outcomes for Kanban system peeps, like betta can-do, thinking about for example psychological flow (15), how often dey release, tooly stuff, skills, technical debt, user experience (UX) debt, customer experience (CX) debt, human-centered-design debt, technical domain can-do, market domain can-do, business domain can-do, and a weather/culture for net fixing.*

Any of da above ways can be useful. Also thinky about dese:

- **Failure Demand** (17)**:** Demand caused by failing to do something, or failing to do it right for da customer. It is a beep-beep for a possible fix. It show where room-to-work is getting wasted because of earlier failures, poor Work, or bad deciding. For example, a customer support team might get da same calls over and over because da billy-bill instructions is muddly. [number]
- **Time to Validated Banana, also called Time to Value or Time to Outcome:** Da *rounded-up count of elapsed time chunks (often calendar days/weeks) from when a Stakey-holder's order for a Worky Bit come in, to when da Banana got checked-for-real. It is one example of an ETSF dat look at valuable, measurable outcomes. [measure-bit]*
- **Banana Checked-For-Real:** A Worky Bit dat reach da 'finished' point and giv da Banana it meant to giv to da Stakey-holder (includy, but no only, customer or

user), meeting da out-loud policies, e.g., quality or experience standards. Often got evidence and luk-sees with it.
- **Banana No-Good:** A Worky Bit dat reach da 'finished' point, or get luk-ed at, but fail to giv da Banana it meant to, no meeting what da Definitiony of Worky-Flow expected, often needing re-work or a chuck-out, guided by evidence and luk-sees. Thinky about da place yu in.

By measuring dese kinds of outcomes, effecties, Banana numbers and Banana measure-bits, Kanban system peeps make sure dey is no just giving Work fast (outputs), but giving real Banana and real fixes (outcomes and effecties) to Stakey-holders, includy but no only customers and users.

Getting clear on Worky Bits should happen right-now-when-needed, to keep da wastey away. No stare too hard at outputs and too teeny at outcomes. *Kanban system peeps should leany-forward, on-purpose and regular-regular luk over da numbers or measure-bits and keep making dem betta.*

Last Blabla

Only da Kanban Do-Dos, da teeny-minimum Definitiony of Worky-Flow, and a pick of numbers or measure-bits is must-must; ev'rything else is up to yu. Thinky about da place yu in. Kanban system peeps should grow a kind-hearted Zoom-Zoom of Banana.

Feedback from results ('result feedback') mean da data dat come back after yu change something, whether it is county-numbers or wordy-words about outcomes, effecties, or even shifts in da market weather. Dis feedback can sway da Stakey-holder Banana outcomes, plus da inputs, effort, stuff, or costs going forward. (Noticey: Peeps is NOT 'resources.').

In real life, Kanban is a journey of always-learning and always-bending. By starting with dese core do-dos and keeping da fix-fix going, Kanban system peeps can get a betta Zoom-Zoom of Banana dat last. Kanban system peeps should start simple-simple and grow dere Kanban system as dey learn.

Oldy Story of Kanban

Da now-now state of Kanban can be tracked back to da Toyota Production System (and its grandpapas) and da work of peeps like Taiichi Ohno *(9)*. Da togethery bundle of do-dos for Brainy Worky Stuff, what peeps now mostly call *Kanban (12),* mainly got borned on one team at Corbis in 2006. Dose do-dos spreaded zoom-zoom quick to a big and mixy-mixy international tribe dat keep making da whole thing betta and betta.

Big Thank-Yous

Da peeps thanked here no must agree with what is writted in dis doccy, and dat is okay-okay. Still, Da Open Banana Booky of Kanban owe a HUUUGE pile of tank-yus to:

- All who helped grow Kanban, includy dose who preferred no to be named

- *Kanban Guide July 2020 or December 2020 version luky-overs: Jean-Paul Bayley, Jose Casal, Colleen Johnson, Todd Miller, Eric Naiburg, Steve Porter, Ryan Ripley, Dave West, Julia Wester, Yuval Yeret, and Deborah Zanke*
- *Kanban Guide May 2025 version luky-overs:* Magdalena Firlit, Tom Gilb, Colleen Johnson, Christian Neverdal, Prateek Singh, Steve Tendon, and Julia Wester
- *Open Guide to Kanban luky-overs: Jim Benson, Andy Carmichael, Jose Casal, Magdalena Firlit, Michael Forni, Martin Hinshelwood, Christian Neverdal, Nader Talai, Steve Tendon, and Nigel Thurlow*
- *Peeps who shaped da thinking: Russell L. Ackoff, Jim Benson, Andy Carmichael, Emily Coleman, John Cutler, W. Edwards Deming, Dominica DeGrandis, Tom Gilb, Joseph M. Juran, Siegfried Kaltenecker, Henrik Kniberg, Klaus Leopold, John Little, Troy Magennis, Taiichi Ohno, Donald G. Reinersten, Sam L. Savage, Walter Shewhart, Nader Talai, Steve Tendon, Nigel Thurlow, and Donald J. Wheeler.*

Extra-Bits Bag (Appendix)

Bossing Work In Progresso/Process = Bossing 'Started but not Finished Work'

Bossing *'Started but not Finished Work', also called* WIP control, can be showed on a Kanban boardy any way *da Kanban system* peeps thinky is good, includy, but no only, painter's-tape spots, Total Worky Bit Oldy-ness or Total Elapsed Time for 'Started' but Not Finished Items (TETSNFI), queue controls, WIP control numbers, or WIP control ranges.

Dere is also some optional no-Kanban options dat some tribes hold up, like CONWIP(16), Simplified DBR (16), or DBR(16):

- **CONWIP (Constant Work In Progresso/Process)** (16)**:** CONWIP is a pully system dat keep one fixed total 'Started but Not Finished Work' (SNFW) or Work In Progresso/Process (WIP) limit across da whole worky-flow, 'starting' new Work only when a 'finished' or 'completed' Bit go out, bossing da Zoom-Zoom with one system-wide squeeze. Example: A computery support team allow only 15 open tickets at a time; when a ticket get fixed, a new one can be 'started.' No ev'rybody like it.
- **DBR** (3,16)**:** A clever-clever way dat boss da Zoom-Zoom Squeeze with Buffers before da Zoom-Zoom Squeeze and at da system outputs, making Throughputty as big as possible while guarding against wibbly-wobbly in tangly systems. Example: In a product-making gang, UX luk-over (da main Zoom-Zoom Squeeze) set da beaty (drum) with a Buffer of designs in front of it, a second Buffer before legal-approval stop da over-piling, and new Work only get let out when both Buffers got room. No ev'rybody like it.
- **Zoom-Zoom Squeeze (Flow Constraint)** (16)**:** Da bottleneck with da teeniest room-to-work in da Definitiony of Worky-Flow. Dere can be many bottlenecks (all with less room dan da demand need), and da Zoom-Zoom Squeeze is da squeeziest one. It hold back da Kanban system's whole Throughputty, setting da

beaty at which Banana get given. Example: In a computery team, if da testing take da longest and hold back da releasing of features, den testing is da Zoom-Zoom Squeeze dat set da system's beaty. In Brainy Worky Stuff, bottlenecks often go all wobbly-wobbly and can hop round da worky-flow in surprisey ways. But sometimes bottlenecks is stubborn-stubborn.

- **Simplified DBR (Drum-Buffer-Rope)** (3,16)**:** A simple-simple scheduling way where da Kanban system's Throughputty set da worky-flow beaty, and Throughputty act as a re-filly beep-beep like in CONWIP. Say dere is a Kanban system using Simplified Drum-Buffer-Rope, and da Definitiony of Worky-Flow is built to hold up to 15 Bits, with 12 busy in progresso/process (drum) and a Buffer of 3 Bits ready to start, so da Work keep going smooth if any of da 12 Bits hit trouble, by pulling from da Buffer, keeping da Zoom-Zoom with, for example, 13 in progresso/process and 2 in reserve. Da rope beep-beep for a re-filly when a Bit get given, keeping da total inside da 15-Bit limit, and da system rush to rebuild da Buffer if it run dry, poking at troubles early to keep da Zoom-Zoom going. No ev'rybody like it.

If Kanban system peeps need to pick which Worky Bit to 'start'

Here is some optional no-Kanban tricks dat some but no all tribes hold up:

- **Class of Service** (21)**:** A shape-type for one or a bunch of Worky Bits, like standard, (real and so no made-up) fixed date, expedite, or intangible. Which class of service yu pick might show da felt Banana, Uh-oh, or Cost of Delay. It is more useful as an input for deciding which Bit(s) to 'start' next when dere is room, dan for re-ordering Worky Bits already In Progresso/Process (which is bad for Zoom-Zoom). Easy to over-pile da Kanban system when yu use it wrong, e.g., an 'expedite lane' might get beaten by a 'super-expedite lane,' and den things go silly-silly. Prone to wonky Zoom-Zoom even when used right.
- **Cost of Delay (per chunk o' time)** (7)**:** Da rate of Banana-loss per chunk o' time for one or more Bits, no to be muddled with Delay Cost. It is often useful as input for deciding which Bit(s) to 'start' next when dere is room, rather dan re-ordering Worky Bits In Progresso/Process (which is bad for Zoom-Zoom). Like most picking-inputs, it is often built on clever guessies. It can also be real after da fact. For example, da Cost of Delay for a Worky Bit is $10,000 per week. Kanban system peeps should steppy careful before thinking about dis way.
- **Data-informed Rightsizing** (24-25)**:** Sometimes betta dan da other options, because Kanban system peeps rarely know da effort or da Banana up-front. It leave more room for grabbing a chance.
- **Delay Cost (total)** (7)**:** Da whole piled-up loss over a stretch of time from one delay stretch, for one or more Bits. It can be real or guessed, and it matter a lot to say which one yu mean. For example, if da Cost of Delay for a Worky Bit is $10,000 per week and it is late by 3 weeks, da Delay Cost is $30,000.
- **Impact Estimation Table (IET)** (22)**:** Weigh up options against what Stakey-holders expect or will put up with.
- **Opportunity Cost:** Da Banana or benefit yu lose by choosing to work on one

or more Worky Bits instead of other maybe-valuable Worky Bits, because dere is only so much room-to-work. It show da swappy-swaps yu make when picking inside yu room-to-work in a Kanban system, where focusing on one or more Worky Bits mean giving up others dat could also have given Banana. Kanban system peeps often use numbers like Cost of Delay or Delay Cost to put a size on opportunity cost. Because Banana — and so Opportunity Cost — go from hard-to-guess all da way to can't-guess-at-all, Kanban system peeps should steppy careful before trying dis way.

- **Random:** Can be betta dan da other options, because da effort or da Banana is no known up-front.
- **Real Options** (23)**:** Waity on promises until yu got enough info, treating decisions as valuable, going-off-soon options, to keep yu bendy and boss da Uh-oh.
- **Uh-oh (Risk):** Do da uh-oh-iest Bit first. Uh-oh can include da chance dat da Banana can no be picked at all.
- **Shortest Job First** (24-25)**:** Pick da Worky Bit with da teeniest felt effort, putting rightsized Worky Bits before other Worky Bits. Dis can make feedback loops shorter and outcomes quicker. But it can also make yu 'start' late on a bigger, uh-oh-ier Worky Bit.
- **Slack** (19)**:** Slack is leaving some room-to-work unused so yu can cope with demand surges, no-planned work, or surprises popping up. In a Kanban for Brainy Worky Stuff place, it is an on-purpose set-aside or policy of spare room or time inside da Definitiony of Worky-Flow, to soak up wibbly-wobbly, handle surprise knocks, or let da fix-fix happen without squishing da Kanban system's Throughputty. Example: Kanban system peeps might keep a Slack by holding dere 'Started but Not Finished Work' (SNFW) or Work In Progresso/Process (WIP) to 80% of dere room-to-work, leaving time for urgent asks or tidying up processes without making da planned work late. Slack is a big idea in Lean.
- **Banana divided by Effort:** Guessed Banana (usually a clever guessy) divided by Guessed Effort (usually a clever guessy). Real Effort and real Banana tend to be all over da place. Kanban system peeps should steppy careful before thinking about dis way. Optional: thinky about Uh-oh too.

Termy-Term Rules in da Context of Brainy Worky Stuff

- **Buffer** (16)**:** A buffer is a WIP-limited (or 'Started but Not Finished Work' limited) area dat hold Work for a little while to smooth da Zoom-Zoom and stop da over-piling, and it also work as a WIP-bossed queue. No to be muddled with Slack. No ev'rybody like Buffer; more columns can lead to MORE 'Started but Not Finished Work' (SNFW) or Work In Progresso/Process (WIP).
- ***Definitiony of Worky-Flow:*** Da out-loud shared understandy of Zoom-Zoom among Kanban system peeps in dere own place, includy but no only, *da out-loud bundle of agreements and policies dat say how Worky Bits get picked, pushed along, and 'finished' through da worky-flow's separate stages.*
- **Out-loud policy:** An out-loud policy in a Kanban system is a clearly-said, can-see rule or guidey dat make da assumptions about da worky-flow — like

when Worky Bits 'start' or move — see-through for Kanban system peeps. Dese policies should be Looky'd on da Kanban boardy and be easy to get at, so all Kanban system peeps understand and follow da same process. By making policies out-loud, Kanban system peeps squish da muddle, line up dere doing, and hold up da good Zoom-Zoom of Banana.

- **'Finished' (or 'Completed'):** When da Elapsed Time from 'Started' to 'Finished' clocky stop for a 'started' and 'finished' pair in a Definitiony of Worky-Flow.
- **Zoom-Zoom (Flow):** Da (best-case smooth) moving and giving of Worky Bits through da Definitiony of Worky-Flow. A balanced Kanban system keep da Throughputty going. In a perfect world, Work dat went into da system (Brainy Worky Stuff) would zoom like a river, never stopping, finding da easiest path until it reach da customer. No to be muddled with da Definitiony of Worky-Flow (DoW). In Kanban, Zoom-Zoom > keeping-everybody-busy.
- ***kanban:*** *A kanban (signboard in Japanese) is a Looky cue dat nudge yu to pick, 'start,' or move a Worky Bit. Nothing should be made or moved without a kanban beep-beep.*
- **Kanban or Kanban system**: Da whole bundle of ideas in dis booky. *Kanban grow from da idea of a beep-beep system (a way to call for Work or stuff-on-da-shelf in a making-stuff system).*
  When dis booky say Kanban, assume a Kanban system.
- **Kanban Boardy:** A Looky version of one or more Definitionies of Worky-Flow.
- **Brainy Worky Stuff (Knowledge Work):** Da making, using, or bossing of info with thinky-brain processes, to fix (often) tangly problems, make decisions, or invent new things, usually needing know-how, good judgy, and work-togethery. Often, Brainy Worky Stuff & its wastey bits is invisible.
- **Round-and-Round (Iterative):** Worky Bits get worked in repeated cycles, with each cycle going back over and tidying da same Work using feedback, testing, or new light-bulb moments. Kanban is no by-nature bad for creative round-and-round work, but it may need some careful thinky or bending.
- **JIT:** Toyota Just-in-Time — making only what is needed, when it is needed, in da amount needed, to squish da wastey and make things efficient.
- **Measure-bit (Measure):** A measure-bit is a raw, unit-specific data dot standing for one quantity, like 'count of Worky Bits completed dis week' or 'time to complete a Worky Bit,' feeding in as a ground-floor input for tracking Zoom-Zoom performance. Example: Kanban system peeps write down a measure-bit of 10 Worky Bits completed to date.
- **Number (Metric):** A number is a worked-out calculation built from one or more measure-bits, to giv context for worky-flow performance, like 'average Throughputty' or 'Throughputty per week.' Example: Kanban system peeps work out a number of 4 days average Elapsed Time 'Started' to 'Finished' by dividing da total time to complete 10 Worky Bits by da count of Worky Bits.
- **Pull:** Work get picked (whether 'started' or 'not started' on da Definitiony of Worky-Flow) only when dere is room, chosen by Kanban system peeps, and it stop da over-piling, best-case beep-beeped by a customer, straight-on or roundabouty.

- **Push:** Work get shoved onto Kanban system peeps or into da Kanban system without thinking about da room-to-work or whether da Kanban system peeps is ready to ‘start’ it.
- **Queue:** A queue in Kanban is a waity area for Worky Bits, often with no strict limits, but it can act as a Buffer if Work In Progresso/Process (WIP) limits (16) or ‘Started but Not Finished Work’ (SNFW) limits is dere.
- **Uh-oh (Risk):** Da chance dat a bad thingy could happen.
- **Steady system:** Put simple-simple, a system dat can keep meeting da demand put on it. Dere is more exact descriptions (7,8,20). Brainy Worky Stuff tend to make a wider range of Worky Bit sizes dan factory work. Uneven sizes no must lead to wobblier elapsed times (because waity time is often da biggest thing, etc.) or Throughputty, but dey can (because of outside dependencies, etc.). Dis booky reckon dat ways built for factories no must be useless for Brainy Worky Stuff.
- **Stakey-holder**: A thingy, one peep, or a gang o’ peeps who is boss of, nosey about, or changed by what go into da Kanban system, what happen inside, and what come out. Includy but no only customer, decidey-peep, or user.
- **‘Started’:** When da elapsed time clockies ‘start’ for a ‘started’ and ‘finished’ pair in a Definitiony of Worky-Flow.
- **‘started’ and ‘finished’ pair:** Each of da one-or-many ‘started’ points on a Definitiony of Worky-Flow should got a matching ‘finished’ point on da same Definitiony of Worky-Flow.
- **Takt:** Da word Takt (English ‘tact’) come from da German word meaning rhythm, beaty, or cycle. Takt is about keeping time in music. Now-now, Takt is mostly used in factory places. Takt is a ground-floor measure-bit in da Toyota Product System and Lean Thinking, used to work out da room-to-work needed to meet demand in a steady system. Throughputty, unlike Takt — which set da hoped-for perfect beaty from da demand — measure da real output per chunk o’ time. Takt also help get a balanced system dat meet demand steady, by letting Kanban system peeps work out da room-to-work needed at each stage of a process. Working out Takt is hard-hard in Brainy Worky Stuff, because yu got to understand demand in very wibbly-wobbly places. No always great for Brainy Worky Stuff.
- **Work:** Mean one or more Worky Bits, ‘started’, ‘not started’, ‘finished,’ or ‘not finished.’
- **Worky Bit (Work Item):** A Worky Bit, also called a Bit, hold da chance of Banana. Lotsa words can be used for da different sizes of a Worky Bit, long as it got da chance of Banana. Worky Bits with no chance of Stakey-holder Banana is maybe-wastey, e.g., peeps focus on ‘finishing’ little sub-tasks across many Worky Bits instead of focusing on ‘finishing’ one Bit at a time. Bossing ‘Started but Not Finished Work’ (SNFW) or Work In Progresso/Process (WIP) for maybe-wastey Bits often squish da work-togethery and da focus needed to giv da Banana sooner. Thinky about da place yu in.
- **Worky Bit Type:** A sorting-box for a Worky Bit. Examples includy but is no only brands, customers, features, boo-boos, project work, user experience (UX) research, customer experience (CX) research, human-centered design, day-to-day work, problem statements, hypotheses, other research, and experiments.

Useful for making sense of things.

- **Banana Checked-For-Real (Validated Value):** Banana confirmed with evidence or luk-sees (best-case both), formal or loosey, by Stakey-holders; often after one or more rounds of result feedback (and re-work), by inside and outside Stakey-holders. No ev'rybody like it.
- **Banana (Value)**: Either a maybe-benefit or a got-it-already benefit for a Stakey-holder. Examples includy meeting da needs of da customer, da end-user, da decidey-peep, da company, and da big blue planet.
- **Looky, looky-making (Visualize, visualization):** Any way to put idea in da head so ev'rybody go "AAAAH, me see!", includy just 'splaining it clear. No must be picture.

References

References is put here to point readers at chances for more study. Dey no must hold up what is writted in dis booky:

1. *Little, J. D. C. (1961). A proof for the queuing formula: L = λW. Operations Research, 9(3), 383–387. https://doi.org/10.1287/opre.9.3.383.*
2. *Deming, W. E. (1986). Out of the crisis. MIT Press. (Peer-reviewed through its academic adoption in quality management.)*
3. *Goldratt, E. M. (1990). Theory of Constraints. North River Press. (Peer-reviewed through academic adoption in operations research.)*
4. *Womack, J. P., & Jones, D. T. (1996). Lean thinking: Banish waste and create wealth in your corporation. Simon & Schuster.*
5. *Ackoff, R. L. (1999). Ackoff's Best: His Classic Writings on Management. NY: John Wiley & Sons.*
6. *Hopp, W. J. and Spearman, M. L. (2004) 'To pull or not to pull: what is the question?', Manufacturing & Service Operations Management, 6(2), pp. 133–148. https://doi.org/10.1287/msom.1030.0028.*
7. *Reinertsen, D. G. (2009). The Principles of Product Development Flow: Second Generation Lean Product Development. Redondo Beach, CA: Celeritas Publishing*
8. *Shewhart, W. A. (1931). Economic Control of Quality of Manufactured Product. NY: D. Van Nostrand Company.*
9. *Ohno, T. (1988). Toyota Production System: Beyond Large-Scale Production. Portland, OR: Productivity Press.*
10. *Juran, J. M. (1992). Juran on Quality by Design: The New Steps for Planning Quality into Goods and Services. New York: The Free Press.*
11. *Wheeler, D. J. (1993). Understanding Variation: The Key to Managing Chaos. Knoxville, TN: SPC Press.*
12. _Wikipedia (2025) 'Kanban (development)'. Available at: https://en.wikipedia.org/wiki/Kanban_(development) (Accessed: 22 June 2025)._
13. *Kingman, J. F. C. (1961) 'The single server queue in heavy traffic', Mathematical Proceedings of the Cambridge Philosophical Society, 57(4), pp. 902–904. doi: 10.1017/S0305004100035783, and the stable URL*

is https://www.cambridge.org/core/journals/mathematical-proceedings-of-the-cambridge-philosophical-society/article/single-server-queue-in-heavy-traffic/81C55BC00A68FE6D5385638AA0B0AF37.

14. *Roser, C. (2018) 'The Kingman Formula – Variation, Utilization, and Lead Time', AllAboutLean.com, 2 March. Available at: https://www.allaboutlean.com/kingman-formula/ (Accessed: 22 June 2025)*
15. *Csíkszentmihályi, M. (1990) Flow: The Psychology of Optimal Experience. NY: Harper & Row*
16. *Tendon, S. and Müller, W. (2015). Hyper-Productive Knowledge Work Performance: The TameFlow Approach and Its Application to Scrum and Kanban. Plantation, FL: J. Ross Publishing.*
17. *Seddon, J. (2019). Failure demand | Vanguard. [online] Vanguard-method.net. Available at: https://vanguard-method.net/library/systems-principles/failure-demand/ [Accessed 22 Mar. 2019]*
18. Christensen, C.M., Hall, T., Dillon, K. and Duncan, D.S., 2016. Know your customers' 'jobs to be done'. *Harvard Business Review*, 94(9), pp.54-62.
19. DeMarco, T. (2001). *Slack: Getting Past Burnout, Busywork, and the Myth of Total Efficiency*. Broadway Books.
20. Leopold, K. (2017) Little's law and system stability – an interview with Daniel Vacanti. Leanability. Available at: https://www.leanability.com/en/blog/2017/08/littles-law-and-system-stability [Accessed 28 June 2025].
21. Kanban University (2021) The Official Guide to The Kanban Method [Online]. Available at: https://kanban.university/new-to-kanban-get-the-official-guide-to-the-kanban-method/ (Accessed: 29 June 2025).
22. *Gilb, T. (2005) Competitive Engineering: A Handbook for Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage. Oxford: Elsevier Butterworth-Heinemann. Also available at: https://bit.ly/TomGilbCompEng*
23. Maassen, O., Matts, C. and Geary, C. (2013) Commitment: A novel about managing project risk. The Netherlands: Happy About.
24. Vacanti, D. S. (2015) Actionable Agile Metrics for Predictability: An Introduction. United States: ActionableAgile Press.
25. Vacanti, D. S. (2023) Actionable Agile Metrics for Predictability Volume II: Advanced Topics. United States: ActionableAgile Press.