

JOE

TWITTER NOTES

PROBLEMS



Woody Zuill @WoodyZuill · 2018/7/7

"We fail more often because we solve the wrong problem than because we get the wrong solution to the right problem." -Russell Ackoff

▼

twitters

PROBLEMS



Eberhard Wolff ✅ @ewolff · 2018/5/20

I used to think

"I know a technical solution to that problem."

Now I added

"... and I see the organizational and cultural problems that keep you from implementing it."



Dan Creswell @dancres · 2018/7/8

To summarise: We're spending too much time detecting and fixing defects emitted from a defective mechanism with all the attendant costs rather than actually eliminating all the defects in the mechanism itself. Analytic/silo vs holistic.



PROBLEMS



Niels Pflaeging @NielsPflaeging · 2018/8/29

Organizations today do not actually suffer from **#tech** problems - such as **#digitalization**.

They suffer from **#socialtech** problems & barriers - such as **#management**, **#meetings**, **#decisionmaking**, **#innovation**, **#change**.



Programming Wisdom @CodeWisdom · 2018/7/3

"In programming, the hard part isn't solving problems, but deciding what problems to solve." - Paul Graham

PROBLEMS



James Hollingshead @bladesjester · 2018/8/17

Replies to [@Agile201](#) [@allenholub](#) and [@RonJeffries](#)

That's just it. By and large, management will grab onto any metric of "productivity" that they can because it's easier than actually thinking about how/what people are doing.

The moment that happens, problems start.



Kathryn E. Campbell @sparks_kc · 2018/6/7

Biggest mistake in leadership #5: over focus on "performance" (delivering more, faster, cheaper). Remember performance doesn't last long without also maintaining quality, user satisfaction, employee engagement. [#BetterSoftwareCon](#) [@petebehrens](#)

PROBLEMS



Bob Marshall @flowchainsensei · 2018/7/29

I'm sure a greater part of the appeal of "going [#Agile](#)" is exactly *because* it's a local optimisation. Many companies (erroneous) believe they can reap its promised benefits without any disruptive impact to the wider organisation. Idiots!



Bob Marshall @flowchainsensei · 2018/4/28

YIL You probably don't realise that, from a company-wide perspective, Agile was and is only EVER a local optimisation, and thus must ALWAYS make company-wide results WORSE.

PROBLEMS



Simon Wardley @swardley · 2018/6/24

X : What does bleeding obvious mean?
Me : Bleeding obvious is a metaphor for technology change. It refers to a wall on which the letters "Obvious" are writ large. You have to bang your head against this wall for about a decade before others go "Oh, that's obvious".

1

8

22

↑



Simon Wardley
@swardley

It's why many engineers in meetings grab their heads and bang them on the table whilst muttering curses. They are giving praise to the great wall of the bleeding obvious whilst they prepare for the many more years of head banging to come.

PROBLEMS



CHANGE



Lisa Broderick @LisalnThenow · 2018/6/17

We cannot solve the problems with the same thinking we used
when we created them - Einstein

twitters



Philip Crosby Associates @DiscoverCrosby · 2018/5/29

Any valuable idea has a part done within the current management system and another requiring it to change. Most efforts won't include the second part, but will instead distort the original idea to fit "our culture". [#WhyWeFail](#)



April Wensel @aprilwensel · 2018/6/13

"When a problem has existed for a long time it can feel permanent. The critical, judgmental mind has trouble imagining anything could dissolve such a seemingly entrenched state. But if you're open, sudden change can happen at any time."

—Tosha Silver, Outrageous Openness



Tom Peters ✅ @tom_peters · 2018/6/20

Small>>Big. Herb Kelleher, Southwest Airlines: "Think small and act small, and we'll get bigger. Think big and act big, and we'll get smaller."



Martin Hogg @martinahogg · 2018/5/27

Many decision that seems significant turn out to not matter much. Most actually significant decisions seem insignificant at the time you make them.

NO SHORTCUTS



Dan Creswell @dancres · 2018/8/21

"Most large organizations embrace the idea of invention, but are not willing to suffer the string of failed experiments necessary to get there."

twitters

NO SHORTCUTS



Gordon Tredgold ✅ @GordonTredgold · 2018/7/2

Leaders always choose the harder right rather than the easier wrong. - Orrin Woodward



Krystian Kaczor @krystian_kaczor · 2018/5/23

Organisations tend to focus optimisation in technology or product management because solving issues around people and teams are difficult and unpleasant #agile #scaling #scrum



Damian Prosalendis @DamianProsa · 2018/6/22

"What's the most important lesson"

"What's the best book"

"What's the one piece of advice"

These are manifestations of "shortcut mentality". If you speak like this, you are looking for a shortcut, not for efficiency, but for avoiding the work.

NO SHORTCUTS



Allen Holub @allenholub · 2018/4/26

Re: "we're transitioning to agile, but we need to get work done *now*, so can't do X." You can't become agile by not being agile. Some productivity reduction is normal when you learn *anything* new. If you don't accept that, your transition will fail.



Programming Wisdom @CodeWisdom · 2018/5/14

"You can have the project:

Done On Time

Done On Budget

Done Properly

Pick two."

- Unknown



Gordon Tredgold ✅ @GordonTredgold · 2018/7/15

A good general not only sees the way to victory; he also knows when victory is impossible. - Polybius

NO SHORTCUTS



Philip Crosby Associates @DiscoverCrosby · 2018/6/5

It's extremely difficult to change when no one around you is supporting and facilitating your efforts. Organizations where some try moving while everyone else remains in place are like trying to get off a crowded train when the doors are still shut.

John Cutler @johncutlefish

Consider how difficult it is to "Transform" one's self (we try every New Years).

Now multiply that by thousands of interconnected actors with their own needs....



Simon Wardley @swardley · 2018/6/16

They generally don't want to be told that they need to put in lots of hard work, to start from the beginning, to learn what they should have been learning. Alas, no time machine and no simple path. You either put in the work or you don't learn.

QUALITY



Randy Shoup @randyshoup · 2018/6/23

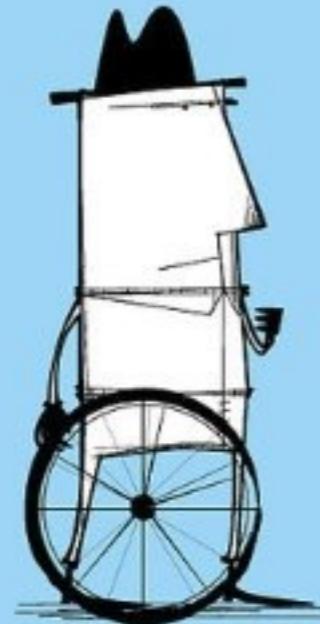
"We're too busy with quality problems to focus on quality"

@RobMeaney at #RebelCon

twitters

ERRR...

CAN'T STOP.
TOO BUSY!!



TOO BUSY TO IMPROVE?

WorkCompass

QUALITY



Jez Humble

@jezhumble

Replies to [@KirstenMinshall](#) [@allenholub](#) and [@SergeyIRL](#)

Happens all the time. Any improvement work that would improve productivity and quality in the medium term but takes capacity in the short term gets nixed basically. I ❤️ this John Seddon quote.

The paradox is that when managers focus on productivity, long-term improvements are rarely made. On the other hand, when managers focus on quality, productivity improves continuously.

— John Seddon



Javier Bonnemaison [@jbonnemaison](#) · 2018/7/12

"Why is it that productivity increases as quality improves?
Less rework. There is no better answer."

W. Edwards Deming, Out of the Crisis, 1982 (Chapter 1) [#lean](#)
[#Quality](#) [#Agile](#) [#Kanban](#) [#Collaboration](#)

QUALITY



Kevlin Henney @KevlinHenney · 2018/7/12

One reason managing technical debt, etc., is often downvoted when prioritising work stems from a misunderstanding that work adding business value should be prioritised: good technical quality doesn't add business value; poor quality loses it. Prioritise to maximise, not to add.

6

108

176



Jutta Eckstein
@JuttaEckstein

Replying to @KevlinHenney

Well, I think we in tech have to do a much better job to ensure the business can understand the business value of paying technical debts. I think this is part of our job.

QUALITY



Venkat Subramaniam @venkat_s · 2018/8/15

A generation of developers may have grown to ignore quality partly due to their business environments, but the bigger concern is the lack of skills, that they have failed to develop, to care for quality even when placed in situations that facilitate or necessitate high quality.



Philip Crosby Associates @DiscoverCrosby · 2018/6/20

And yet faster is easier for managers to measure, thus becoming the corporate cry. #Mindset #Culture

George Dinwiddie @gdinwiddie

There are two ways to do something noticeably quicker:

1. do an exceptionally good job, or
2. do an exceptionally bad job.

The first is not achieved by focusing on speed, and provides other benefits than just speed.

EVOLVING



Programming Wisdom @CodeWisdom · 2018/4/22

"It's not at all important to get it right the first time. It's vitally important to get it right the last time." - The Pragmatic Programmer

▼

twitters

EVOLVING



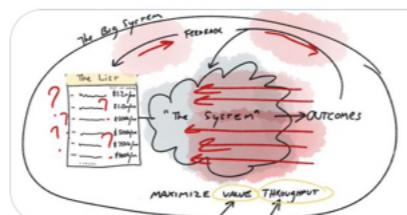
Dee W Hock @deewhock · 2018/7/5

Success depends less on choice than execution. There are always adequate alternatives to reach an objective. Select one promptly and use the time saved to perfect it's execution. An acceptable plan perfectly executed will prevail over perfect plan, poorly executed every time.



John Arundel @bitfield · 2018/7/29

"There is very real value in things like refactoring, restructuring, and improving tooling. It literally could be the MOST IMPORTANT thing you work on. I suggest learning more about Cost of Delay. It is a great catalyst for conversations."



Prioritizing "Non-Feature" Work and
Continuous Improvement
hackernoon.com



Uncle Bob Martin @unclebobmartin · 2018/7/31

The word “refactoring” should never appear in a schedule. Refactoring is not a story or a backlog item. Refactoring is not a scheduled task. Refactoring is immediate and continuous. It’s like washing your hands in the bathroom. You always do it.



Allen Holub @allenholub · 2018/6/25

Improving the quality of your estimates rarely benefits the business. Improving the quality of your processes always benefits the business. Don’t focus on estimates, focus on continuous process improvement.



Programming Wisdom @CodeWisdom · 2018/8/4

“If you can get today’s work done today, but you do it in such a way that you can’t possibly get tomorrow’s work done tomorrow, then you lose.” - Martin Fowler



Philip Crosby Associates @DiscoverCrosby · 2018/6/3

Why is sw refactoring a thing? Because organizations think of growth linearly, not in continuously-improving feedback loops.

Why do orgs have structural issues? Because organizations think of growth linearly, not in continuously-improving feedback loops.

Dan Creswell @dancres

It is beyond me why refsctoring is a thing in and of itself in sw dev.

In my view, it's an indivisible part of writing decent code, no more important or deserving of focus than any other.

[Show this thread](#)

EVOLVING



QuintinDK @quintindk · 2018/4/26

I like this... "On a purely technical level, it's more important to focus on things like clean code, good testing, and attention to evolutionary architecture."



Microservice Trade-Offs
martinfowler.com



John Arundel @bitfield · 2018/7/19

"One of the most productive statements a team can make is, "This week we removed 10K lines of code while preserving all existing functionality." That doesn't sound very sexy on the face of it but it contributes to your organization's long-term health."



Source Code Is A Liability, Not An Asset
blogs.msdn.microsoft.com

Folks asking me to help instill a culture of continuous improvement consistently ask me if learning will impact feature velocity

2018/6/4, 03:09

60 Retweets 125 Likes



Mike Long @meekrosoft · 2018/6/4

The answer is yes, because focus on feature velocity is what got them in the mess in the first place

1

13

50



Mike Long @meekrosoft · 2018/6/4

The best teams use around 30% of time on improvement, and this is what makes them high performing

2

93

171



Mike Long @meekrosoft · 2018/6/4

The worst teams use all their time on features, failure demand and manual work - because they don't have time to improve

TESTING



Programming Wisdom @CodeWisdom · 2018/6/7

"Program testing can be a very effective way to show the presence of bugs, but is hopelessly inadequate for showing their absence." -
Edsger W. Dijkstra

▼

twitters

TESTING



Allen Holub @allenholub · 2018/6/27

It's an absolute fallacy that testing somehow slows down delivery. In fact, the more you test, the faster you'll be. Humble et al's book Accelerate (amzn.to/2N1Cmku) has hard numbers verifying that assertion. Read it. You cannot run your business based on mythology



Brian P. Hogan @bphogan · 1d

The problems you are solving are complex. Your code doesn't have to be. Optimize for change. Maintainability, readability, discoverability.

If it's hard to test it's hard to change.



Antonio Cangiano 🤖🥑 @acangiano · 2018/7/13

A key advantage of well tested code is the ability to perform random acts of kindness to it. Tending to your code like a garden. Small improvements add up and compound. Without tests, it's hard to be confident in even seemingly inconsequential changes.

TESTING



Tim Ottinger @tottinge · 2018/8/1

"effective, efficient testing is not separate from development; it is woven into development" buff.ly/2Oyjjz9



Allen Holub @allenholub · 2018/6/29

What % of the time should an agile team spend testing? IMHO, testing should be so tightly integrated into dev that it's impossible to distinguish one task from the other. That percentage is impossible to measure. If you can measure a testing ratio, you're not working effectively.



Allen Holub @allenholub · 4d

More often than not, bottlenecks are created silos. If you have a separate testing/QA org, they are almost certainly a bottleneck. The rate at which they can handle work controls the delivery rate for the whole system. Integrating testing into dev can speed up delivery by 10x.

TESTING



James Grenning @jwgrenning · 2018/8/6

It's too bad that companies focus so much on Agile management and not Agile engineering. Being successful at iteration is highly dependent on incremental engineering skills, like TDD. You need both, though the management gets a lot easier when supported by iterative engineering.



René Wiersma @Rene_Wiersma · 2018/7/27

I just refactored a complex piece of code.

Fearlessly – because a suite of unit tests caught any mistakes.

That feels like magic to me.

#TDD #Refactoring

TESTING



Michael Bolton @michaelbolton · 2018/6/29

Replies to [@gortok](#)

Don't let shallow bugs get beyond the developer's machine in the first place; build testability in. Heuristics: TDD/test first, pairing, review, collaboration with testers, unit checks, logs, scriptable interfaces, simplicity,... Developers need time for that. Give it to them.



Ron Jeffries @RonJeffries · 2018/6/28

"I screwed up this idea. Therefore it is a bad idea."

David Tanzer @dtanzer

"Mock objects will never work on a real project. We had them. It was a mess." Really? I think it's more nuanced: buff.ly/1Uz2xga
#oldpost

EMPOWERMENT



Allen Holub @allenholub · 2018/8/24

Agility can't happen w/o autonomy at the team level. All requirements to get permission, wait for someone, do things in standard & overly complex ways, make reports, etc. add delays. Agile is about delivering *quickly* so that you can get feedback. You can't do that w/o autonomy.

twitters

EMPOWERMENT



Allen Holub @allenholub · 2018/9/1

I've been in orgs where nobody could do certain things (like use a language feature or package) w/o permission from The Architect. Even though The Architect always said yes, the existence of that rule *dramatically* slowed the pace of innovation (and development). Rules matter.



Anthony Sciamanna @asciamanna · 2018/6/29

"We believe the critical factor in motivation is not measurement, but empowerment: moving decisions to the lowest possible level in an organization while developing the capacity of those people to make decisions wisely." -[@mpoppendieck](#), Lean Software Development

EMPOWERMENT



Daniel Spiewak @djspiewak · 2018/6/21

One of the most valuable lessons I learned from [@timperrett](#): infrastructure should be decentralized, self-service, and composable. Empower your teams. Unblock them. Look for ways to *remove* constraints.



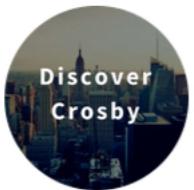
Christian Posta @christianposta · 2018/6/21

In this world of public cloud, hybrid private/public, multi-cloud, etc. a consistent and low-friction developer experience, regardless of infra, should be top of mind. Enable your developers.



Rod Johnson @springrod · 2018/4/27

Any time you reduce the cycle time of anything in development or delivery, you unleash more of the potential of developers



Philip Crosby Associates @DiscoverCrosby · 2018/6/14

Cities increase the ability for individuals to connect and collaborate, large organizations are managed to ensure people remain focused on the work and goals within their silo.

Michel Löhr @1ohr

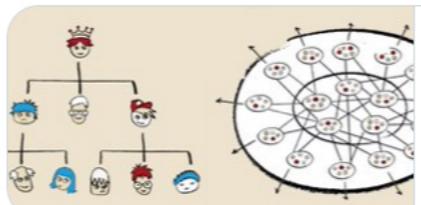
Every time the size of a city doubles, #innovation and #productivity per resident increase by 15 percent. But when companies get bigger, innovation and productivity per employee generally go down. —Tony Hsieh, CEO, Zappos.



Niels Pflaeging @NielsPflaeging · 2018/8/16

In #complexity, organizations must be #decentralized - not centrally steered, not functionally divided, not siloed, not matrixed, not flat, or businessunitized!

That is not a matter of opinion. It is a systemic fact!



Flat hierarchies: Just another step in the wrong direction
linkedin.com



Giulia Merlo @giuliavmerlo · 2018/6/5

We don't believe people *can* work in the right way, so we create systems that encourage them to work in the wrong way, then when they do we say: see? Theory X feeds itself. Or, as [@NielsPflaeging](#) said it better:



Why we cannot learn a damn thing from
Toyota or Semco
[linkedin.com](#)



Niels Pflaeging @NielsPflaeging · 2018/8/14

Self-organization is nothing [#special](#).

What is strange & shocking is that we fool ourselves into believing that [#management](#), command-and-control, or steering of organizations can actually work!



Management Belongs onto the Garbage Heap
of History
[linkedin.com](#)

EMPOWERMENT



Tim Ottinger @tottinge · 2018/6/15

Some people say "empower" and merely mean "allow."

Permission is not power.

I'm *permitted* to renovate my kitchen, but I'm not *empowered*. I don't have the skills, time, material, knowledge to do it. I can't do that job. I would need support, help, time, money to do it.



Tim Ottinger @tottinge · 2018/7/9

A: So how can we get the developers to work harder?

B: Take away their tools. Add more bureaucracy. Use slow computers and dead programming languages.

A: And that will bring up the velocity?

B: No. You asked how to make their work harder.

NO



Geoff Watts @geoffcwatts · 52m

"Deciding what not to do is as important as deciding what to do." //
Steve Jobs

▼

twitters

#NOPROJECTS



Rob England @theitskeptic · 2018/7/29

Agile and Lean techniques fundamentally change how we think about IT projects. When I say that, what I really mean is that we should stop thinking about "projects."

- Mark Schwartz, Seat at the Table



Steven Limmer @_slimmer_ · 2018/7/11

Replies to [@theitskeptic](#) and [@markdalgarno](#)

Perhaps if organisations stopped treating IT as projects to be done, rather than long-lived business-enablers to be maintained, things might improve.



Joe The IT Guy @Joe_the_IT_guy · 2018/7/11

"The way to fix culture and behaviour is to change the system. And the greatest single issue in IT systems is project management" -
[@theitskeptic](#) buff.ly/2ITUrOp #ITSM #PPM



NOPROJECTS

From my presentation "Project Management was the worst thing ever to happen to IT" comes 20 dysfunctions of IT project management

1. Define once
2. Business case theatre
3. Estimates become commitments
4. Bells and whistles, gold plated
5. Latency between requirements and delivery
6. Lack of experiment
7. Difficult to pivot
8. Difficult to kill
9. Do once: no incentive to automate
10. New only: no funding for defects or debt
11. Remote from customers
12. Testing ends at go-live
13. Big bang change
14. Complex change across multiple systems
15. Time, cost, deliverables, quality. Quality loses.
16. Dead cat syndrome: no accountability for quality
17. Those building the system don't run the system
18. Handover of knowledge and artefacts
19. Break up teams
20. Set people up to fail

#NOPROJECTS



Rob England @theitskeptic · 2018/7/29

ITIL has led people to think in siloes ("go fix change management"). Project Management has led people to think in finite units of work instead of streams of product.

Both are fundamental dysfunctions of the framework model, not failures of execution.



John Cutler @johncutlefish · 2018/8/14

Give project managers a tool (e.g. Jira)...and they'll use it for project management.

Health of the overall system and continuous improvement was never a goal of project management (nor should be).



Doc Norton @DocOnDev · 2018/6/25

Software project risk comes less from events that may change the plan and more from the notion that the plan shouldn't change.



Krystian Kaczor @krystian_kaczor · 2018/7/15

Watermelon project: green outside, red inside, 92% made of water.
#corpo #project

NOPROCESSES



Christopher McCann @letterpress_se · 2018/7/4

"If you're not watchful, the process can become the thing. This can happen very easily in large organizations. The process becomes the proxy for the result you want. You stop looking at outcomes and just make sure you're doing the process right." -Jeff Bezos



Mahoo @SystemsNinja · 2018/8/15

In my experience the more complex your process the less able it is to absorb complexity. Simple processes with the right skills in the right place do the job much more effectively 😊

NOPROCESSES



Troy Tuttle @troytuttle · 2018/4/26

Process is just a poor proxy for collaboration.



Process is only needed when we:

1. Don't know how to collaborate.
2. Choose not to (or refuse to) collaborate.

In a fully collaborative environment, process is unnecessary.

#lean-agile



Allen Holub @allenholub · 2018/7/10

Seems to me that if you "standardize" process---something required for Scaling---you're not an agile organization any more.

The teams need to create their own processes. You might start with Scrum, but the retrospective process will move you away from that pretty quickly. 1/



#NOCOMMANDCONTROL



Allen Holub @allenholub · 2018/8/4

Running a business using century-old management practices (command/control, Taylorism) and thinking (month- or years-long release cycles) is just irresponsible. Too many companies that say they're "Agile" are doing exactly that.



Philip Crosby Associates @DiscoverCrosby · 2018/5/24

Unfortunately, most leaders think it's done by rules and regulations.

#ptsafety #IHCongress

Charles Murphy, MD @ZeroPatientHarm

Gary Kaplan: Leader behavior sets the culture. #ptsafety

#IHCongress

#NOCOMMANDCONTROL



Woody Zuill @WoodyZuill · 2018/8/13

Let's be careful about putting rules into place.



Once we have a rule, it's hard to remove it. And we eventually end up with rules about changing rules.

And eventually we need to make rules about the rules about changing rules.

Dang!

#NOESTIMATES



Woody Zuill @WoodyZuill · 2018/6/1

I don't believe estimates of software development time/cost reduce risk/uncertainty - IMO at best they hide the risk/uncertainty.



Alan Cooper ✅ @MrAlanCooper · 2018/7/24

The biggest, stankiest elephant in the room is that software dev cannot be estimated, and yet millions of middle managers spend countless hours "estimating" the inestimable. Why? Why do we not learn? 12

#NOESTIMATES



Lalit Shandilya @shanlalit · 2018/8/15

Replies to [@Bloodymirova](#) [@mikeveerman](#) and 2 others

Mr. Management don't trust people they trust reports. They even don't want to see the demo it's time consuming and doesn't tell them when it's going to be done...As that's the objective/goal.



Programming Wisdom @CodeWisdom · 2018/7/14

"When asked by management how complete is the code, I always respond with 93%, first line or last line, it is always 93%." -

[@PamBennettSails](#)

NOMEASURES



Ron Quartel @ronquartel · 2018/6/13

When a measure becomes a target, it ceases to be a good measure
- Goodhart's Law



El_Gordo
@GCHLTD

Replying to [@rethinkingserv](#)

Its so true that measures drive everything and arbitrary targets drive the wrong behaviours and lead to cheating the system to hit the number.

NOMEASURES



KanbanDev @KanbanDev · 2018/8/7

Replies to [@KanbanDev](#) and [@neil_killick](#)

Unfortunately companies keep rewarding "getting stuff done" because it's easily observed and measured for each individual.

However it's rare that anyone gets rewarded for reducing the rate of bugs because it requires a complex root cause analysis.



Paweł Lasek @pawel_lasek · 2018/7/26

Replies to [@mishmosh](#) [@peterseibel](#) and [@jlrichey](#)

Guess why I mentioned keeping managers away from that?

A friend who *also* had JIRA admin in the same org once got pissed and implemented *exactly* what the manager wanted. All process was FUBAR for I think two sprints (around a month)

#NOMEETINGS



Naval Ravikant Bot @NavalBot · 2018/8/7

"Unnecessary meetings (and most are) are a mutually-assured-destruction of time. Learning how to avoid them is a prerequisite of doing anything great." - [@naval](#)



Michael Feathers @mfeathers · 2018/8/15

"Meetings are synchronization bottlenecks"

Post Agile:
embracing
asynchronous
processes

Jilles van Gorp • Aug 13

DEV

Post Agile: embracing asynchronous
processes
[dev.to](#)



Gordon Tredgold ✅ @GordonTredgold · 2018/7/2

People who enjoy meetings should not be in charge of anything. -
Thomas Sowell

#NOBUZZWORDS



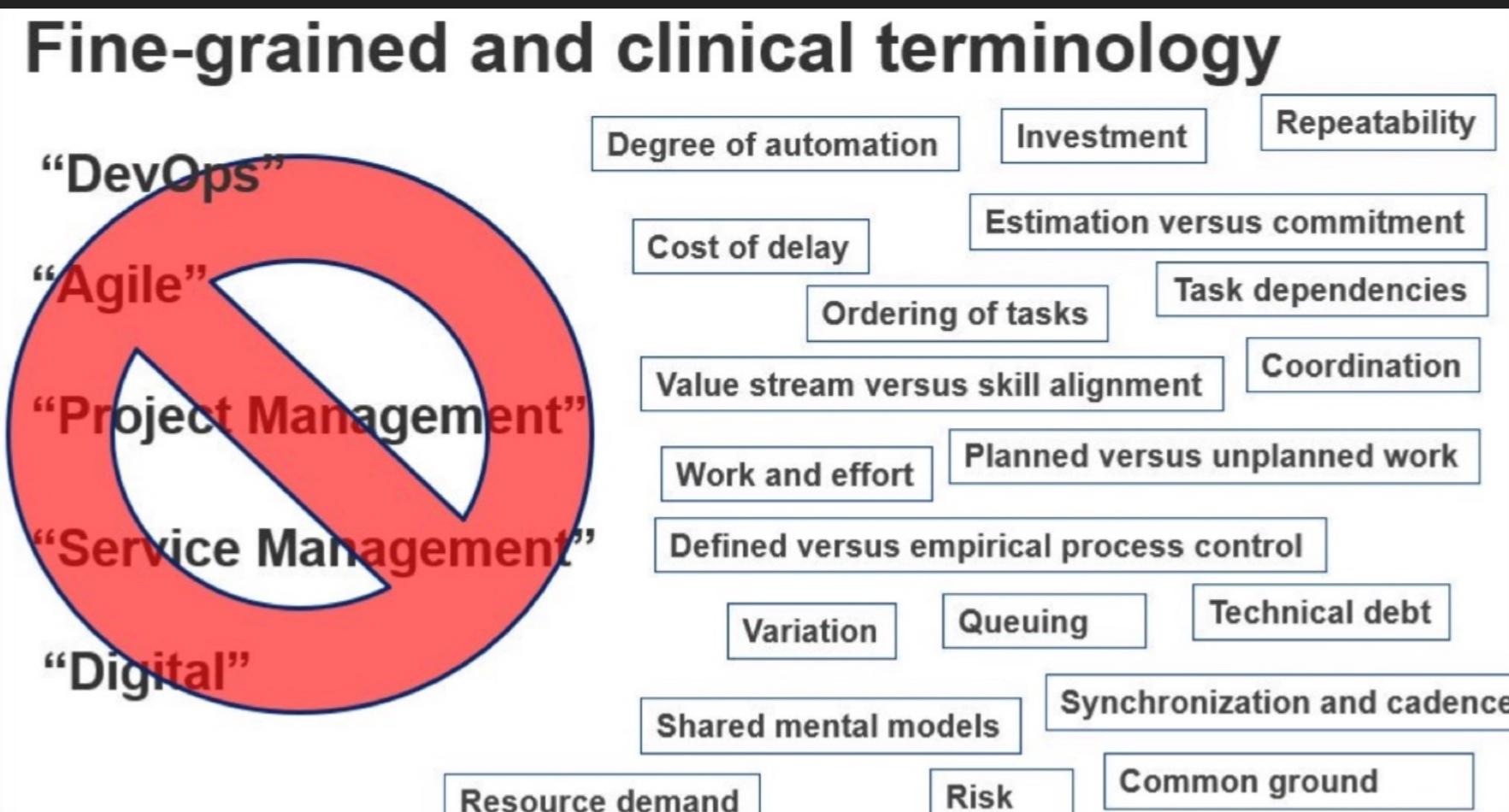
ANTIREZ @antirez · 2018/6/22

After so many years the term "cloud" still sounds like a lot of bullshit, even if it's clear what people mean at this point when using this term. Moreover the more some person actually builds software, the less she will use this word.



Allen Holub @allenholub · 3d

If you have to define some bit of jargon to be understood, stop using the jargon and use real words. Eg. QA and QC are jargon. It's ridiculous wasting time defining them. Instead, let's just use "testing," "continuous improvement," and other words that actually mean something.



NOSHORTCUTS



Jon Fulton @JonFultonUk · 2018/6/24

Waterfall is a commitment to poor quality, long development cycles, unpredictability and lost opportunities. So when i hear you ask for 'hybrid', i hear 'agile, but can you just make it a bit worse please'.



Jon Hall @JonHall_ · 2018/7/12

I've tried to encourage ITSM people for several years to stop thinking about "how DevOps is compatible with ITSM" and to start thinking of the new challenges DevOps faces. It's our job to make it successful, not to assimilate it.

NOSHORTCUTS



Hermann Hyytiälä @hemppah · 2018/8/2

"95% of changes made by management today make no improvement. Examples are reorganization [of organization chart], a new computer [IT system]."-W. Edwards Deming



Rachael Byrne @b3rachael · 2018/5/3

SAFe is waterfall with an agile wrapper that does nothing but make management feel better #dodzh18

CULTURE



L. David Marquet ✅ @ldavidmarquet · 2018/6/17

In healthy organizations, leaders take responsibility for the system
and people take responsibility for their actions.

In unhealthy organizations, leaders blame the people and the
people blame the system.

▼

twitters



Niels Pflaeging @NielsPflaeging · 2018/5/29

▼

In **#complexity**,
#soft skill is the new **#hard** skill!
Because communication beats **#process**,
mastery beats **#knowledge**,
principle beats **#rule**,
social density beats **#hierarchy**,
preparation beats **#planning**,
agreements beat **#decisions**,
and flow beats **#scale**.

It is neither good nor bad!



Joshua J. Arnold @joshuajames · 2018/6/18

"Simple, clear, purpose and principles give rise to complex and intelligent behavior. Complex rules and regulations give rise to simple and stupid behavior."

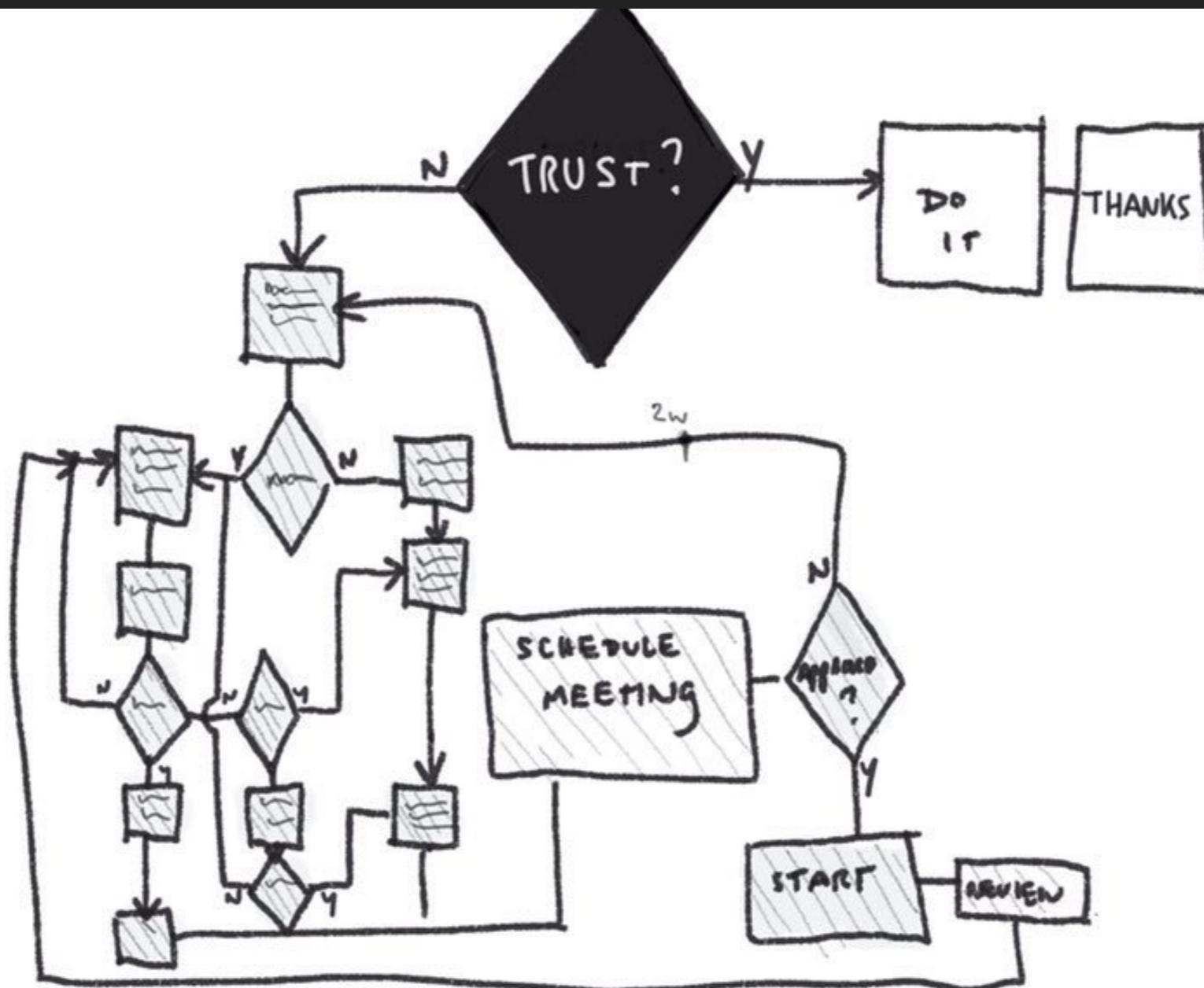
– Dee Hock

nora bateson @NoraBateson

Replies to @monk51295 and @gregorylent

the opposite of complexity is reductionism... not simplicity. :-) simplicity is complexity with grace.

CULTURE





L. David Marquet ✅ @ldavidmarquet · 2018/7/2

The formula for trust?
transparency x time = trust.



what's your formula?



Dave Gray @davegray · 2018/6/12

Google discovered that giving every employee a credit card and trusting them to follow policy on expenses, even with some bad actors, was cheaper than the enforcement and structures most companies use.



How much do you waste due to low trust? Can you do the math for your company?

CULTURE



Nora Jones @nora_js · 2018/6/12

Reminding employees to simply “be careful” does not actually help your software system be safer nor encourage people to make safer decisions. It infantilizes them.



Vala Afshar ✅ @ValaAfshar · 2018/5/22

If you never want to be criticized, don't do anything new. —Jeff Bezos



Lisa Broderick @LisalnThenow · 2018/5/14

The purpose of fear is to raise your awareness, not to stop your progress. - Steve Maraboli #quote

CULTURE



CZnative @pczarkowski · 2018/6/4

Kubernetes will solve your container orchestration problem but you don't have a container orchestration problem you have a culture problem.



Troels Damgaard @dontrolle · 2018/5/14

"Choose microservices because you want to scale the organization, not (necessarily) because you want to scale the application."

@jboner at [#microcph](#)

AGILE



Allen Holub @allenholub · 2018/8/31

There is no such thing as "Agile project management." If you're "managing" the project, you're not agile. Period. [#thatsNOTAgile](#)

▼

twitters



Randy Shoup @randyshoup · 2018/5/10

Agile is no more or less than

- * work in small increments
- * get feedback
- * take engineering seriously

If someone tells you it's more than this, they are trying to sell you something.

If someone tells you it's less, they don't understand agile.

@tastapod at #CraftConf



Agile201 @Agile201 · 2018/4/24

"The important thing is not your process. The important thing is your process for improving your process." @HenrikKniberg

#agile #methodology



Mike Veerman 🇧🇪 @mikeveerman · 2018/8/29

Companies that really want to start an agile transformation should start by shipping to production at least once a month.

If you can't even do that, let's not waste each other's time...



John Cutler @johncuttlefish · 2018/8/10

"Yes. We can now accurately forecast efforts to within 30 days 90% of the time!"

Vs

"Yes. We deliver value weekly, remove non value-add complexity, can pivot quickly, make great product bets, and measure outcomes!"



Matthew Skelton @matthewpskelton · 2018/5/16

Current status: seeing so much pain from lack of early deployments into Production. Start by just deploying readme.txt, ffs - evolve the rest later.

#ContinuousDelivery



Brandon Olin @devblackops · 2018/7/10

When management says they want to “do DevOps”, ask them if they are willing to change IT and business processes to enable it.

If they aren’t, then they don’t want DevOps, they just want to automate things.

SHOW RESPECT



Philip Crosby Associates @DiscoverCrosby · 2018/5/21

Technology brings new capabilities, but people are the most important element in any system. If we're going to accomplish great things, attitudes and mindset are the real levers of change.

twitters

SHOW RESPECT



Richard Feynman @ProfFeynman · 2018/6/15

There's a big difference between knowing the name of something and knowing something!



Doctor Octothorpe @doc_octothorpe · 2018/7/10

Coworker: "You can learn Kubernetes in a day."

Me: "You can learn chess in a day."

SHOW RESPECT



Karl Isenberg @KarlKFI · 2018/7/21

Replies to [@ibuildthecloud](#) and [@drnic](#)

Even with kubeadm and a team of 6 you can't get upstream K8s production ready in 6 months. We tried.

Hard Problems:

- High Availability
- Single Sign On
- Multitenancy
- Resource Isolation
- Permission Management
- Upgrades
- Backups
- Package Management
- CI/CD Integration

SHOW RESPECT



Woody Zuill @WoodyZuill · 2018/8/19

"The first step of any project is to grossly underestimate its complexity and difficulty." ~@nicollhunt



Gary Bernhardt @garybernhardt · 2018/5/3

New programmers, take note: this is important, but it's unlikely that people will tell you it! In programming, when something is branded as "easy", "simple", or "lightweight", it frequently (not always, but frequently) means that it's confusing and complicated!



SHOW RESPECT



John Cutler @johncuttlefish · 2018/7/10

You cannot “split” an engineer across two projects without paying for context switching. So the pie chart would look more like:

35% A

35% B

30% context switching, waiting

LEARNING



Michael James @michaeldotjames · 2018/6/19

Organizations underuse people's ability to learn. Not to a small degree, but to a tremendous degree.

▼

twitters



John Allspaw
@allspaw



This is well put, and is what motivates [@AdaptiveCLabs](#).

"Management Strategies for Complex Adaptive Systems: Sensemaking, Learning, and Improvisation" (McDaniel, 2007)

power. When attention is paid to these potential multiple interpretations of the world, then it is more likely that learning about the situation will take place and a satisfactory resolution developed. In traditional views of organizational life, knowledge is the key but in a complexity view, learning is the key. Certainly the capacity to learn is built on what is known, but the focus shifts from learning to know, to knowing to learn.

LEARNING



Tamara McCleary ✅ @TamaraMcCleary · 2018/6/10

The No. 1 asset for job seekers of the future: The ability to learn
tek.io/2s9x91I interview w/[@TamaraMcCleary](#) by [@jasonhiner](#) via
[@techrepublic](#) #FutureOfWork #AI #robotics
#DigitalTransformation #DES2018



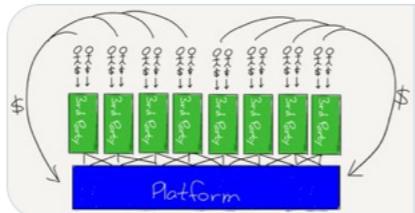
Allen Holub @allenholub · 2018/6/27

Hiring based on a laundry list of technology is short sighted at best. Agile==change. Whatever tech you're using this week will be irrelevant next week. We need to hire based on an ability to learn, not what you know this second. Breadth of knowledge is a good indicator of that.



Ryan Singer @rjs · 2018/5/10

This resonates. Two philosophies in tech: replacing people vs enabling people to do new things. By [@benthompson](#)



Tech's Two Philosophies
stratechery.com

LEARNING



James Hollingshead @bladesjester · 2016/12/9

How to get things right:

- 1) get lots of things wrong.
- 2) learn from the mistakes.

Most people forget step 2



QUOTES

Quotes ❤️ @wordstionary · 2018/6/7

Mistakes are proof that you're trying.



The Best Linux Blog In the Unixverse @nixcraft · 2018/6/1

By the time you're 35, you should have successfully done following at least once:

1. rm -rf /
2. Reboot production server by mistake
3. Drop CPU/hard disk and damage it permanently
4. Changing firewall rule instantly blocked you out
5. No backup was made/verified ever...



LEARNING



Sean Kelly @StabbyCutyou · 2018/5/31

Enjoys learning



Ours is an industry that moves too quickly. Every day, new tech comes out, old "best practices" are uncovered as the frauds we all knew they were the whole time, etc. You don't have to keep up with everything, but you should enjoy the process of learning.



Richard Feynman @ProfFeynman · 2018/7/28

If you're not having fun, you're not learning. There's a pleasure in finding things out.



DEV Community 🧑‍💻🧑‍💻 @ThePracticalDev · 2018/6/2

"I've been a professional developer since I graduated in 1987. I am currently the lead for a team of five doing front-end development.



Apparently, at 53 and still up-to-date and still developing I've become a novelty :)"

I have been a professional developer for 31 years and I'm 53 now, Ask Me Anything!

 John Munsch · May 31

DEV

I have been a professional developer for 31 years and I'm 53 now, Ask Me Anything!

dev.to



Quotes ❤️ @wordstionary · 2018/6/8

QUOTES

The truth hurts, but lies are worse.



Gordon Tredgold ✅ @GordonTredgold · 2018/5/28

Knowing what to do and not doing it is the same as not knowing what to do.— Robin Sharma #quote