

Table of Contents

[TUXQUOTE]

"It is not enough that management commit themselves to quality and productivity, they must know what it is they must do. Such a responsibility cannot be delegated." – W. Edwards Deming from Out of the Crisis, 1986 "Culture eats strategy for breakfast" – Peter Drucker "Operating a product development process near full utilization is an economic disaster." – Don Reinertsen "If you only quantify one thing, quantify Cost of Delay" – Don Reinertsen "Principles of Product Development Flow" "You can't scale crappy code." – Dean Leffingwell "Provide sufficient capacity margin to enable cadence." – Don Reinertsen "Principles of Product Development Flow" "Innovation comes from the producer." – W. Edward Demings. Could have been just as easily said by Steve Jobs. Related is "No useful improvement was ever invented as a desk" – Taiichi Ohno "Any inefficiency of decentralization costs less than the value of faster response time." – Don Reinertsen "Principles of Product Development Flow" "Today's development processes typically deliver information asynchronously in large batches. Flow-based processes deliver information in a regular cadence of small batches." – Don Reinertsen "When WIP and utilization become too high, you will see a sudden and catastrophic reduction in throughput." – Don Reinertsen "People are already doing their best; the problems are with the system. Only management can change the system." – W. Edwards Deming. Or "I like to assume that no one came to work with the explicit intention of pissing me off or doing the wrong thing." – Hans Samios "A common disease that afflicts management the world over is the impression that 'Our problems are different.' They are different for sure, but the principles that will help to improve the quality of product and service are universal in nature." – W. Edwards Deming "Understanding economics requires an understanding of the interaction amongst multiple variables." – Don Reinertsen "Principles of Product Development Flow" "Left to themselves, components (of a system) become selfish, independent profit centers and thus destroy the system." – W. Edwards Deming "All we are doing is looking at the timeline, from when the customer gives us an order to when we collect cash And we are reducing the timeline by reducing the non-value added wastes." – Taiichi Ohno "Development managers today face an important trade-off between steady state of performance and the system's ability to accommodate unanticipated changes in resource requirements without descending into the firefighting cycle. Taken collectively, the insights from operating beyond the brink imply that the current methods deployed for aggregate resource planning used at most organizations are woefully insufficient in preventing firefighting. Development managers and company leadership with a desire to avoid firefighting must rethink their approach to managing multi-project development portfolios." – Dantar P. Oosterwal in "The Lean Machine" "People who operate within the development system are rarely directly to blame. Individual mistakes happen. However, consistent lack of performance must be attributed to systemic issues. To improve the development system, it must first be recognized that a bad system will always undermine even the best efforts of good people. Bad systems beat good people every time!" – Dantar P. Oosterwal in "The Lean Machine" "A firefighting organization requires extraordinary people to achieve ordinary results. In an exceptional organization ordinary people produce extraordinary results routinely." – Dantar P. Oosterwal in "The Lean Machine" "There was in fact no correlation between exiting phase gates on time and project success. The data suggested the inverse was true." – Dantar P. Oosterwal in "The Lean Machine" "Any project that costs more than \$1 million in normalized labor is a waste of money, time, and resources." The Standish Group asserts this startling conclusion in its "Factors of Value" report (June 29, 2015). "Integration points control product development." – Dantar P. Oosterwal in "The Lean Machine" "The notion that 'our problems are different', is a disease that affects western management. They may be different, to be sure, but the principles that will help to improve quality of product and service are universal in nature." – W. Edwards Deming "Passion is inversely proportional to the amount of real information available" – "Benford's Law of Controversy", Gregory Benford, 1980 "The most important and visible outcropping of the action bias in the excellent companies is their willingness to try things out, to experiment. There is absolutely no magic

in the experiment... But our experience has been that most big institutions have forgotten how to test and learn. They seem to prefer analysis and debate to trying something out, and they are paralyzed by fear of failure, however small.” – Tom Peters and Robert Waterman, *In Search of Excellence* (Peters & Waterman, 1982) “Cease dependence on mass inspection to achieve quality. Improve process and build quality into the product in the first place.” – Deming “A project plan is like a lettuce. On the day you buy it, it looks firm and crispy; a week later its a bit wilted around the edges, and after a month its unrecognizable.” – Martin Fowler quoting a project manager. “If I have learned one thing it is this: there is no answer. There is never an answer; there are only better questions.” “... doing the wrong thing righter ...” – Russell Ackoff “There was in fact no correlation between exiting phase gates on time and project success...the data suggested the inverse might be true.” – Dantar P. Oosterwal in “The Lean Machine” “Most people do not listen to understand, they listen to reply” – Stephen Covey “In science it often happens that scientists say, 'You know that's a really good argument; my position is mistaken,' and then they would actually change their minds and you never hear that old view from them again. They really do it. It doesn't happen as often as it should, because scientists are human and change is sometimes painful. But it happens every day.” – Carl Sagan “A mind is like a parachute. It doesn't work if it's not open” – Frank Zappa “Study after study shows that we have little-to-no skill in when it comes to making predictions — but that sure doesn't keep us from trying.” “Today's development processes typically deliver information asynchronously in large batches. Flow-based processes deliver information in a regular cadence of small batches.” – Don Reinertsen “If you are going to quantify one thing, quantify cost of delay” – Don Reinertsen “Everyone has a plan, until they get punched in the mouth.” – 'Iron' Mike Tyson. “Performance of the individual cannot be measured, except on a long-term basis, for which I mean 15, 18, 20 years.” – W. Edwards Deming in a seminar for CEOs, “Quality, Productivity, and Competitive Position,” 1992. Printed in *The Essential Deming*, page 53. “Motivation – nonsense. All that people need to know is why their work is important.” – W. Edwards Deming, from a speech at General Motors in 1992: *Introduction to a System*. Reprinted on page 157 of *The Essential Deming*. “The idea of a merit rating is alluring. The sound of the words captivates the imagination: pay for what you get; get what you pay for; motivate people to do their best, for their own good. The effect is exactly the opposite of what the words promise. Everyone propels himself forward, or tries to, for his own good, on his own life preserver. The organization is the loser. The merit rating rewards people that conform to the system. It does not reward attempts to improve the system. Don't rock the boat.” – W. Edwards Deming, “The Merit System: The Annual Appraisal: Destroyer of People,” 1986. Reprinted in *The Essential Deming*, page 27. “We learn from history that man can never learn anything from history.” – George Bernard Shaw. Think this applies to estimation. “There's technical debt, then there's technical subprime mortgages with exploding balloon payments.” – Mark Imbriaco (@markimbriaco) via Steve Wingard “We are working on everything but we can guarantee you that it will not be delivered any earlier than if we were not working on it at all.” – Don Reinertsen commenting on pretending to work on everything when you have a delivery system constrained by capacity. “You don't get more planes to land at an airport by insisting on putting more planes in the air.” See “The Big Ideas Behind Lean Product Development” “Any fool can write code a computer can understand, but it takes real genius to write code a human will understand.” – Martin Fowler. “If done well, management is a tough job, which is why the pay is premium. However, there will always be those managers who want to get paid for the hard parts of management work without actually doing them.” Jerry Weinberg “Managing Teams Congruently” “The reasonable man adapts himself to the world: the unreasonable one persists in trying to adapt the world to himself. Therefore all progress depends on the unreasonable man.” – George Bernard Shaw, *Man and Superman* “Worried that TDD will slow down your programmers? Don't. They probably need slowing down.” – J. B. Rainsberger “In particular I always argue that high-level tests are there as a second line of test defense. If you get a failure in a high level test, not just do you have a bug in your functional code, you also have a missing unit test. Thus whenever you fix a failing end-to-end test, you should be adding unit tests too.” – Martin

Fowler "It's not the practices; it's the culture that makes you want to use these practices." Boeing Exec to Joe Justice. "The secret of getting ahead is getting started. The secret of getting started is breaking your complex overwhelming tasks into smaller manageable tasks, and then starting on the first one." – Mark Twain via Brun, Brian J "Any fool can make something complicated. It takes a genius to make it simple." – Woody Guthrie "Keeping the system working is more important than doing work" – Jez Humble on the attitude required for continuous integration and continuous delivery. "People under time pressure don't think faster" – Tim Lister. "The surest way to mismanage a project and jeopardize a product is to put too much emphasis on the schedule as it demoralizes the team and drives them to make stupid decisions despite their better judgments." – Steve Maguire "Debugging the Development Process" "While working with these teams, I discovered that they were all making the same fundamental errors and they were perpetually repeating those errors ... In every group I worked with, I found the project leads were spending nearly all their time writing code and almost none of their time thinking about the project. The leads didn't spend time trying to keep schedules on track, they didn't look for foreseeable problems so they could circumvent them, they didn't work to protect other team members from unnecessary work, they didn't pay particular attention to training other team members, they didn't set detailed project goals or create effective attack plans. The leads were spending too much time working when they should have been thinking." – in preface by Steve Maguire "Debugging the Development Process". "It is better to use imprecise measures of what is wanted than precise measures of what is not." – Russell L. Ackoff "Develop on Cadence. Deliver on Demand." – Dean Leffingwell "Never attribute to malice that which is adequately explained by stupidity" – Hanlon's Razor "In theory, theory and practice are the same. In practice, they're not." – Yogi Berra "FEATURE" – licence plate seen on a Volkswagen Beetle. "A single-point estimate is usually a target masquerading as an estimate." – Steve McConnell "There is a limit to how well a project can go but no limit to how many problems can occur." – Steve McConnell "A company shouldn't get addicted to being shiny, because shiny doesn't last." – Jeff Bezos on complacency "I think frugality drives innovation, just like other constraints do. One of the only ways to get out of a tight box is to invent your way out." – Jeff Bezos on innovation. "If you're competitor-focused, you have to wait until there is a competitor doing something. Being customer-focused allows you to be more pioneering." – Jeff Bezos on progress. "All businesses need to be young forever. If your customer base ages with you, you're Woolworth's." – Jeff Bezos – on growth. "Smart people learn from their mistakes. But the real sharp ones learn from the mistakes of others." – Brandon Mull via Devraj, Sujata. "Opacity when inspecting an Increment is like covering a thermostat with a cold, wet washcloth. The thermostat doesn't have the correct understanding of the actual room temperature, and would incorrectly initiate heating when it should be cooling." – Ken Schwaber and David Starr on Definition of Done. "Scrum is not a silver bullet; Scrum is a silver mirror" – Mike Dwyer on idea that Scrum helps identify improvement areas earlier rather than later and in and of itself does not improve any project woes. "I smell fried brains" – Tim Lister at afternoon keynote speech at Agile 2013 "The deal with engineering goes like this. Product management takes 20% of the capacity right off the top and gives this to engineering to spend as they see fit. Whatever is required to avoid, 'we need to stop features to rewrite code'. ... If you're in really bad shape today, you might need to make this 30% or even more of the resources. I get nervous when I find teams that think they can get away with much less than 20%." – Marty Kagan, Inspired "Do painful things more frequently, so you can make it less painful..." – Adrian Cockcroft, Architect, Netflix. This is the generally applicable approach. Further he says "We don't get pushback from Dev, because they know it makes rollouts smoother." which points to the education that is required so that people understand why they need to do this hard stuff. "Having a developer add a monitoring metric shouldn't feel like a schema change." – John Allspaw, SVP Tech Ops, Etsy on need to make it easy to help build systems that self-monitor "We found that when we woke up developers at 2am, defects got fixed faster than ever" – Patrick Lightbody, CEO, BrowserMob "If people don't understand the model they won't let it influence their behavior" – Unknown "You don't have to ask permission to do your job well" – attributed to Larry

Constantine "Lead people, manage things." – Peter Drucker "If the actions of management signal that meeting specifications is satisfactory, the product invariably falls short." – Donald J Wheeler and David S Chambers in "Understanding Statistical Process Control". "If you don't think it's going to work, it's already a waste of time. Your experiment can only fail if you believe in a specific outcome." – Eric Ries tweet. "Value is rare, extreme and obvious in retrospect" – Joshua J. Arnold (see Cost of Delay) "Estimate is just one additional metric that helps us make decisions, forecast and model the future." – Michael Dubakov from "Estimates in Software Development. New Frontiers." "If you don't like change, you are going to like irrelevance even less." – General Eric Shinseki "Don't be taken in by the Dopeler Effect. The tendency of stupid ideas to seem smarter when they come at you rapidly." – Terry Jones, On Innovation. Did Travelocity and Kayak. "The key, though, is not to manage by metrics but to use the metrics to understand where to have conversations about what is not getting done." Gary Gruver et al, "The Practical Approach to Large-scale Agile Development." "Computers are useless. They can only give you answers." – Pablo Picasso implying that the key to understanding anything is to ask the right question. "It is useful to do detail planning, but you always want to use it with a short time horizon because detailed planning is very perishable. You do not want to hold a large inventory of detailed planning because you are throwing money away when you do that." – Don Reinersten. "Yes, estimation is fraught. It is inaccurate, and politically dangerous. But we do have some knowledge and the project deserves to have it." – Ron Jeffries tweet 2013-03-18. "Agile is about simplicity. Agile is not simple. If you think it's simple, wait and see." – Ron Jeffries "The First Law of Wisdom Aquisition – It is much easier to identify another's foolishness than to recognize one's own." – Norman Keith from "Project Retrospectives" "If you want people to act like adults, you need to treat them like adults." – R. Semler "If it's hard but important to do, do it more often!" – No idea. Idea is that If something appears too hard, or too costly, or too slow—figure out a way to do it more often so that you actually get better at it. "Agile, Scrum, XP, Kanban, Lean: The same elephant, different points of view" – Ron Jefferies "Innovation simply doesn't happen in the tiny shards of time in the midst of incredibly busy work lives." – Gary Hamel "ARE YOU REALLY SERIOUS ABOUT INNOVATION?" "And I find it little bit odd that a 50 year old woman in Bangladesh often has an easier time getting experimental capital than the average first line employee in a global 1000 company." 1000's of microfinance projects are being done with no collateral, and little paperwork. "We say that code is of high quality when productivity remains high in the presence of change in team and goals." Ward Cunningham "Technical debt is the stuff around code that keeps it from being quality code- that makes it hard to change." – Dan Rawsthorne "If you can't do it with a card you'll really mess it up with a tool" – Dave Thomas, Keynote at Agile 2010 "Simple, clear purpose and principles give rise to complex, intelligent behavior. Complex rules and regulations give rise to simple, stupid behavior." Dee Hock, founder and former CEO of Visa "There is no such things as as an IT project; there are only business project that involve IT." – UK Cabinet Office Permanent Secretary. "If you're thinking people don't like to make changes to their behaviors, just watch a teenage girl get her first cell phone. The changes in her life will be profound and dramatic. And she'll enjoy every moment of it." – B.J. Fogg on perception that people don't like change. To which Jared Spool adds: "If people love change, why do our users complain when we make changes to our designs? Well, it's not because of their aversion. It's because we did it wrong." – Jared Spool "The define/build/test component team is the fractal on which agile development is based." – Dean Leffingwell through Dan Wuescher. "A computer does not substitute for judgment any more than a pencil substitutes for literacy." – Robert McNamara "A hallucination is a fact, not an error; what is erroneous is a judgment based upon it." – Bertrand Russell "Clean code is simple and direct. Clean code reads like well-written prose. Clean code never obscures the designer's intent but rather is full of crisp abstractions and straightforward lines of control." Grady Booch "I could list all of the qualities that I notice in clean code, but there is one overarching quality that leads to all of them. Clean code always looks like it was written by someone who cares. There is nothing obvious that you can do to make it better. All of those things were thought about by the code's author, and if you try to imagine

improvements, you're led back to where you are, sitting in appreciation of the code someone left for you—code left by someone who cares deeply about the craft.” – Michael Feathers “You know you are working on clean code when each routine you read turns out to be pretty much what you expected. You can call it beautiful code when the code also makes it look like the language was made for the problem.” Ward Cunningham “Clean code is not written by following a set of rules. You don't become a software craftsman by learning a list of heuristics. Professionalism and craftsmanship come from values that drive disciplines.” – Robert “Uncle Bob” Martin “Cost cutting is just one of the steps in the process toward extinction.” – Haiko Meelis “If you don't care about the number of defects, I can get it done on any schedule you'd like!” – Robert “Uncle Bob” Martin “For success, like happiness cannot be pursued; it must ensue and it only does so as an unattended consequence of one's personal dedication to a cause greater than oneself.” – Victor Frankl – Neurologist and Psychologist. “Your project, the whole project, has a binary deliverable. On scheduled completion day, the project has either delivered a system that is accepted by the user, or it hasn't. Everyone knows the result on that day. The object of building a project model is to divide the project into component pieces, each of which has this same characteristic: Each activity must be defined by a deliverable with objective completion criteria. The deliverables are demonstrably done or not done.” – Tom Demarco – Controlling Software Projects (1982) “Focusing is saying yes, right? No. Focusing is about saying no. You've got to say, no, no, no,” – Steve Jobs “Never mistake motion for action.” – Ernest Hemingway “Testing is the infinite process of comparing the invisible to the ambiguous in order to avoid the unthinkable happening to the anonymous.” – James Bach “I have always wished that my computer would be as easy to use as my telephone. My wish has come true. I no longer know how to use my telephone.” – Bjarne Stroustrup, the designer and original implementor of C++. “Do or do not. There is no try.” – Yoda. Interesting use of this quote by Ron Jefferies on dealing with priorities that change. “The conversation surrounding the estimation process is as (or more) important, than the actual estimate.” – Dean Leffingwell. “Simplicity is the ultimate sophistication” – Leonardo daVinci “We need to make sure that our people weed their own garden.” – Mike McQuerry, CEO Mindspring. Talking about the success of Mindspring coming from everyone taking responsibility for doing the right thing, not just a couple of managers. “Again, you can't connect the dots looking forward; you can only connect them looking backwards. So you have to trust that the dots will somehow connect in your future. You have to trust in something – your gut, destiny, life, karma, whatever. This approach has never let me down, and it has made all the difference in my life.” – Steve Jobs “When lost in the woods, if the map doesn't agree with the terrain, in all cases believe the terrain.” – Swiss Army Survival Guide “Product Management is like have 100 children at Christmas, but only 1 present under the tree – you are going to leave 99 children unhappy.” – Cory Von Wallenstein, Dyn Inc “Management is not focusing on Technical Debt and probably won't until it is too late. And we have seen what happens when people in charge don't focus on debt haven't we?” – Dan West. “A little debt speeds development so long as it is paid back promptly with a rewrite.” – Ward Cunningham, developer of the Technical Debt metaphor. Note that this is not necessarily a negative thing, only when you don't repay quickly is it a problem. “Fallor ergo sum” – St Augustine. “I err, therefore I am”. “Inertia is the residue of past innovation efforts. Left un-managed, it consumes the resources required to find the next generation innovation.” – Geoffrey Moore, Dealing with Darwin “Cost cutting is just one of the steps in the process toward extinction.” – Haiko Meelis “We can always shoot ourselves in the foot. The amazing thing is how quick we are able to reload and do it again.” – Bob Thurber “You can fight reality and lose, but only 100% of the time.” Byron Katie “Things are the way they are because they got that way.” – Gerry Weinberg on why it doesn't make sense to seek who to blame. “So much of what we call management consists of making it difficult for people to work.” – Peter Drucker “If you call a tail a leg, how many legs has a dog? Five? No, calling a tail a leg don't make it a leg.” – Abraham Lincoln. I used this in the context that just because you have a group of people working together and you call it a team does not mean you have a team. “No matter what, the cost of addressing technical debt increases with time.” – Chris Sterling “We're more likely to

get it 'right' the third time.” – Chris Sterling “Software and cathedrals are much the same – first we build them, then we pray.” – Sam Redwine 1988 “Prediction is very difficult, especially about the future.” – Niels Bohr, Danish Physicist (1885 – 1962) “No is easier to do. Yes is easier to say.” – Unknown. Quoted at 37 Signals. “Most software today is very much like an Egyptian pyramid with millions of bricks piled on top of each other, with no structural integrity, but just done by brute force and thousands of slaves.” – Alan Kay “There is nothing so useless as doing more efficiently what should not be done at all.” – Peter Drucker. “... manage for the normal and treat the exceptions as exceptional.” – W Edwards Demming. “In the business world, the rear-view mirror is always clearer than the windshield.” – Warren Buffett “It is a mistake to look too far ahead. Only one link in the chain of destiny can be handled at a time.” – Winston Churchill “... as we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns – the ones we don't know we don't know.” – Donald Rumsfeld on the unknowns we don't know. “Doing it better is not hard, it's easier, so do it better. Increase in code skill means code is better, so your job is easier.” – Ron Jeffries @ Agile 2010 “Many people outsource to a country that is further away than the space station – does this make sense?” – Chet Hendrickson @ Agile 2010 “The difference between a methodologist and a terrorist is that you can reason with a terrorist” – Dave West @ Agile 2010 “Calling a group of people a 'team' does not make it so.” – Esther Derby @ Agile 2010 “Law of crappy systems - if you have a crappy system even brilliant people will never be more that mediocre.” – Esther Derby @ Agile 2010 “Attempts to force non-deterministic systems to operate at greater than 80% efficiency will cause short bursts of stabilization followed by extreme periods of destructive and unpredictable variations from that goal.” – W. Edwards Deming. “The problem with single point estimate is that it says there is 100% chance that this will happen. It's a target masquerading as an estimate.” – Arin Sime @ Agile 2010 “Only certification drowned in waterfalls.” – Dave Thomas Agile 2010 Conference Keynote speech, Tuesday 10th August 2010 “If you can't do it on a card, then you will only do it worse in a tool.” – Dave Thomas Agile 2010 Conference Keynote speech, Tuesday 10th August 2010 “Don't show other people the back of your computer! Its rude ...” – Dave Thomas Agile 2010 Conference Keynote speech, Tuesday 10th August 2010 “Agile is FrAgile as it depends on sustainable leadership and discipline.” – Unknown “'Technical success' is a euphemism for failure.” – Mary Poppendieck on tying goals of a team to goals of a business @ Agile 2010 “In social psychology, the fundamental attribution error (also known as correspondence bias or attribution effect) describes the tendency to over-value dispositional or personality-based explanations for the observed behaviors of others while under-valuing situational explanations for those behaviors. The fundamental attribution error is most visible when people explain the behavior of others. It does not explain interpretations of one's own behavior—where situational factors are often taken into consideration. This discrepancy is called the actor-observer bias.” – From Wikipedia. “In preparing for battle, I have always found that plans are useless but planning is indispensable.” Dwight D. Eisenhower. Or to put it another way “No battle plan survives contact with the enemy.” – Field Marshall Helmuth von Moltke. “The hardest thing to explain is the glaringly evident which everyone has decided not to see.” – Ayn Rand. “People don't want a quarter inch drill. They want a quarter inch hole”. – Theodore Levitt. To our customers software our code is not an asset; its just a way of getting the job done. “We know this is a useful number because there is a decimal point in it.” – A little fun from Dave Nicolette on problems with numbers and comparing team velocities at How To Compare Elephant Herds. “Managers who don't know how to measure what they want settle for wanting what they can measure” – Russel Ackoff “Tell me how you will measure me and I will tell you how I behave.” – Eli Goldratt (1990) “Poor management can increase software cost more rapidly than any other factor.” – Barry Boehm, Software Engineering Economics, 1981 “Self-organization does not mean that workers instead of managers engineer an organization design. It does not mean letting people do whatever they want to do. It means management commits to guiding the evolution of behaviors that emerge from the interactions of independent agents instead of specifying in advance what effective behavior is.” – Philip

Anderson, The Biology of Business “Self-organizing is about the team determining how they will respond to their environment (and managers / leaders can influence that environment.” – Mike Cohn “What is the simplest thing that could possibly work?” – Kent Beck on a good design “Direction set in advance of experience has an especially short half life.” – Kent Beck “I have made this letter longer than usual because I lack the time to make it shorter.” – Blaise Pascal 1657, although what he actually said was “Je n'ai fait celle-ci plus longue que parce que je n'ai pas eu le loisir de la faire plus courte.” Message – it takes time / work / effort to make something simple. This has been also attributed to Mark Twain, Samuel Johnson and Winston Churchill. “I have not failed. I have just found 10,000 ways that won't work.” – Thomas Edison on experimentation. “Change your organization, or change your organization” – Martin Fowler on taking responsibility for pushing changes in an organization. “...we all safely interpret dangerous things in ways that don't require us to change our lives.” — Orson Scott Card “When you want your boat to go fast it is easier to cut anchors than to add horsepower.” – Luke Hohmann “They must understand, within the context of their specific product, the difference between excellence and perfection. No company can afford a perfect product, but building a product that delivers customer value and maintains technical integrity is essential to commercial success.” – Jim Highsmith from Agile Project Management “Be quick, but don't hurry.” – John Wooden. Further he said that “Life, like basketball, must be played fast – but never out of control.” “The agile triangle: value (releasable product), quality (reliable, adaptable) product, & constraints (cost, schedule, scope)” – Jim Highsmith “Bad news does not get better with age” – Joe Little from his “agile principles” “I know it when I see it.” – Judge Potter Stewart (adapted for Scrum use to understand why the inspect and adapt cycle works) “Without data, all managers can do is 'Motivate', and cross their fingers. When 'Motivation' fails, “Mandation” often follows. With little effect.” – Uncle Bob Martin (1995) “If a test is worth writing, it's worth automating, and it must always pass in the future. I often see teams that ignore failures in their CI tests because 'We have to get these new stories out, this is the PO's priority.' So the PO doesn't care whether what she prioritized in previous sprints still works or not? When the CI breaks, the team should stop the line and get it green.” – Lisa Crispin “Simple, clear purpose and principles give rise to complex, intelligent behavior. Complex rules and regulations give rise to simple, stupid behavior.” – Dee Hock a designer at Visa “We are risk adverse when we might gain.” – Piattelli-Palmarini “If you never erase the whiteboards, you might as well write on walls.” – Ron Jeffries “A mess is not technical debt. A mess is just a mess.” – “Uncle” Bob Martin on the concept of technical debt being used as an excuse for producing crap code in the name of expediency. “To be uncertain is to be uncomfortable but to be certain is to be ridiculous.” – Chinese proverb “Remember the law of accumulation: the sum of many little collaboration efforts isn't little” – Dan Zadra “Perfection is reached not when there is nothing left to add, but when there is nothing left to take away.” — Antoine de Saint-Exupéry “No one has to change. Survival is optional”, W Edwards Demming. “Plans based on average assumptions are wrong on average!” – Sam Savage from “The Flaw of Averages” “This leads us to the odd conclusion that strict control is something that matters a lot on relatively useless projects and much less on useful projects. It suggests that the more you focus on control, the more likely you're working on a project that's striving to deliver something of relatively minor value.” – Tom DeMarco in “Software Engineering: Idea Whose Time Has Come And Gone” “No matter how far down the wrong road you've gone, turn back.” – Turkish proverb “Bubbles don't crash.” – Bertrand Meyer. With the message Bertrand referred to all those engineers who have turned into Powerpoint or UML freaks. A Powerpoint slide or UML diagram will never crash which indicates there is a lot more that could be learned. “If you think you have a new idea, you are wrong. Someone probably already had it. This idea isn't original either; I stole it from someone else.” – Bob Sutton “Instead of being interested in what is new, we should be interested in what is true.” – Jeffrey Pfeffer' on feeling that people are more interested in the latest management fad rather than just using data to fix what they have. “Brooks's law is a principle in software development which says that “adding manpower to a late software project makes it later”. – Fred Brooks in his 1975 book The Mythical Man-Month. Thinking here is that new

people disrupt existing process and mean that more communication needs to happen thus delaying the project. "Here is Edward Bear, coming downstairs now, bump, bump, bump, on the back of his head. It is, as far as he knows, the only way of coming downstairs, but sometimes he feels that there really is another way, if only he could stop bumping for a moment and think of it. And then he feels that perhaps there isn't." – A. A. Milne "Ah well! I am their leader, I really ought to follow them!" – Alexandre Auguste Ledru-Rollin on servant leadership. "Customers can now have what they want at the project end, after they've learned, instead of getting what they wanted at the project start." – Kent Beck & Dave Cleal on "Optional Scope Contracts" "There are two kinds of inspection: 1. Inspection after the defect occurs 2. Inspection to prevent defects." – Shigeo Shingo who was a Japanese industrial engineer who distinguished himself as one of the world's leading experts on manufacturing practices and The Toyota Production System. Or as Tiffany Cooper quotes: "It is better to prepare and prevent than to repair and repent." Harvey Mackay "How come we believe bad news and data immediately but we don't believe any good news". – John Simpson commenting on reaction to velocity increase (improved productivity) we are seeing in Scrum teams. "Partnerships are not about cost reduction, they are not about risk reduction, nor are they about adding capacity. The fundamental reason for partnerships is synergy, people – and companies – can achieve better results through cooperation than they can achieve individually." – Mary Poppendieck from "Implementing Lean Software Development" "A world class 'varsity eight' (plus coxswain) can cover 2000 meters over the water in about 5.5 minutes. However, a single sculler can at best row the same distance in about 7 minutes. The difference is synergy, and if rowing faster were a matter of survival, the cooperators would be the fittest." – Peter A Corning, from "The Synergism Hypothesis: In the Concept of Synergy and Its Role in the Evolution of Complex Systems". "It's hard for Americans to understand the idea that a business organization cannot improve its long term financial results by working to improve its financial results. But the only way to ensure satisfactory and stable long term financial results is to work on improving the system from which those results emerge." – H. Thomas Johnson, winner of the Deming Medal of honor. "Purely people factors predict project trajectories quite well, overriding choice of process or technology. I found no interesting correlation in the projects that I studied among processes, language, or tools and project success. I found success and failures with all sorts of processes, languages and tools. A well-functioning team of adequate people will complete a project almost regardless of the process or technology they are asked to use (although the process and technology might help or hinder them along the way)." – Alistair Cockburn, "Agile Software Development", 2002 "For a new software system, the requirements will not be completely known until after the users have used it" – This is the "Humphrey's requirements uncertainty principle". Watts S. Humphrey. A Discipline for Software Engineering. SEI Series in Software Engineering. Addison-Wesley, 1995. "Just because Scrum doesn't say anything about breakfast doesn't mean you have to go hungry!" – Pete Deemer & Gabrielle Benefield Yahoo, 2006 on the fact that Scrum is lightweight and does not proscribe a practice for every situation. Basically message is "use your common sense". "A well functioning Scrum will deliver highest business value features first and avoid building features that will never be used by the customer. Since industry data shows over half of the software features developed are never used, development can be completed in half the time by avoiding waste, or unnecessary work." – Jeff Sutherland 2006 – "The Nuts and Bolts of Scrum" "In most companies, development is slowed down by impediments identified during the daily meetings or planning and review meetings. When these impediments are prioritized and systematically removed, further increases in productivity and quality are the result. Well run Scrums achieve the Toyota effect – four times industry average productivity and 12 times better quality." – Jeff Sutherland 2006 – "The Nuts and Bolts of Scrum" "I believe that the prevailing system of management is, at its core, dedicated to mediocrity, If forces people to work harder and harder to compensate for failing to tap the spirit and collective intelligence that characterizes working together at its best." – Peter Senge, MIT on traditional command and control approaches to management. "Any piece of software reflects the organizational structure that produced

it.” – Conway's Law. “Sit down with your team and have a dialog. Dialogs work better with 2 people. Dialogs with one person requires medication.” – Kent Beck on overcoming communication barriers.

“Glory and success are not a destination, they are a point of origin” – Chef Baglio. This is a comment of software craftsmanship and the attitude required to get better.” “A stronger 'definition of done' will always increase velocity and improve quality”. Jeff Sutherland at Agile 2008: Jeff Sutherland: Reaching Hyper-Productivity with Outsourced Development Teams “When it comes to code it never pays to rush.”

– Marick’s law “The task is then to refine the code base to better meet customer need. If that is not clear, the programmers should not write a line of code. Every line of code costs money to write and more money to support. It is better for developers to be surfing than writing code that won't be needed. If they write code that ultimately is not used, I will be paying for that code for the life of the system, which is typically longer than my professional life. If they went surfing, they would have fun, and I would have a less expensive system and fewer headaches to maintain.” – Jeff Sutherland, quoted from object-technology@yahoogroups.com Message 121 “Predictable outcomes are one of the key expectations that the marketplace imposes on companies ... Unfortunately, software development has a notorious reputation for being unpredictable, so there is great pressure to make it more predictable. The paradox is that in our zeal to improve the productivity of software development, we have institutionalized practices that have the opposite effect. We create a plan, and then act on that plan as if it embodies an accurate predication of the future. Because we assume predictions are fact, we tend to make early decisions that lock us into a course of action that is difficult to change. Thus, we lose our capability to respond to change when our predictions turn out to be inaccurate. ... Fundamentally, an organization that has a well-developed ability to wait for events to occur and then respond quickly and correctly will deliver far more predictable outcomes than an organization that attempts to predict the future.” – Mary and Tom Poppendieck, from “Implementing Lean Software Development” (pg 31-32). “In preparing for battle I have always found that plans are useless, but planning is indispensable.” – Dwight Eisenhower. Or as Mike Tyson put it “Everybody has a plan until they get punched in the mouth.” “We can't solve problems by using the same kind of thinking we used when we created them.” – Einstein “We never have time to do it right; we always have time to do it twice.” – Unknown “Unit testing is spell checking the word, not validating the sentence.” – Unknown “Plans are an ongoing dynamic activity that peers into the future for indications as to where the solution might emerge and treats the plan as a complex situation, adapting to an emerging solution.” – Mike Dwyer, IT Program Manager, American Healthways, Westborough, MA “One is a thug's game, played by gentlemen, and the other a gentleman's game, played by thugs.” – Old quote comparing Rugby with Soccer. Rugby is regarded as a violent game ... “You will pay for the cost of a face-to-face meeting regardless of whether you have it or not.” – Ken Pugh speech at Agile 2008. “Later = Never”. – LeBlanc law on all those things you leave in the code that you thing you are going to get to at some time in the future. “The best measure of clean code? The number of WTF's per minute when reading code.” – Robert Martin, keynote speech at Agile 2008. “A change in requirement late in the development cycle is a competitive advantage provided you can act on it.” – Mary Poppendieck “Design, development, test and documentation are activities NOT people.” – Bob Schatz during sprint planning training. “If you already know the project is going to fail, then a lot of traditional documentation in a project is about figuring out ‘who to blame’” – Bob Schatz during sprint planning training “User stories are not use cases” – Mike Cohn “Scrum is just the result of software developers going through a midlife crisis” – Unknown, on how everyone says scrum is just the “way they used to do software development”. “Waterfall is a software development process which works as if it were designed for the convenience of lawyers” – Unknown “It is amazing how much can be accomplished if no one cares who gets the credit.” – John Wooden, on team performance “... if a build is not breaking then it is doubtful the team is pushing itself hard enough and concerns should be raised” – Troy Magennis on the “continuous build / integration” process “Testing by itself does not improve software quality. Test results are an indicator of quality, but in an of themselves, they don't improve it. Trying to improve

software quality by increasing the amount of testing is like trying to lose weight by weighing yourself more often. What you eat before you step onto the scales determines how much you weigh, and software development techniques you use determine how many errors testing will find. If you want to lose weight, don't buy a new scale; change your diet. If you want to improve your software, don't test more; develop better." – Steve C McConnell, "Code Complete: A Practical Handbook of Software Construction", 1993 "It is typical to adopt the defined (theoretical) modeling approach when the underlying mechanisms by which a process operates are reasonably well understood. When the process is too complicated for the defined approach, the empirical approach is the appropriate choice." – B.A. Ogunnaike and W.H. Ray (Process Dynamics, Modeling and Control) on how to control processes in general, not just the software development process. "Of the organizations that are attempting to implement Scrum, probably 30-35% will successfully implement it. And that's because of this core problem. Most organizations don't want to be faced with what they don't want to see. And this puts it up there and says 'are you going to do something about it, or not?'" – Scrum et al by Ken Schwaber talking at Google. "Scrum is arguably the oldest and most widely applied agile and iterative method, with an emphasis on iterative and adaptive PM practices. It has been applied in thousands of organizations and domains since the early 1990s, on projects large and small, from Yahoo to Medtronic to Primavera, with great results when leadership commits to the deep required changes moving away from command-control and wishful-thinking – predictive management, and with poor results when leadership can't or won't make those changes. Scrum can be easily integrated with practices from other iterative methods, such as practices from the Unified Process and Extreme Programming, including test-driven development, agile modeling, use cases, user stories, and so forth. On the surface Scrum appears to be simple, but its emphasis on continuing inspect-adapt improvement cycles and selforganizing systems has subtle implications." – Certified Scrum Trainer Craig Larman "One of the industry statistics is that over 65% of the functionality that is delivered and then has to be maintained and sustained is rarely or never used" – Scrum et al by Ken Schwaber talking at Google. Quote refers to a study by the Standish Group. "Scrum is a "lean" approach to software development." – Jeff Sutherland 2006

From:

<https://www.hanssamios.com/dokuwiki/> - **Hans Samios' Personal Lean-Agile Knowledge Base**

Permanent link:

https://www.hanssamios.com/dokuwiki/quatable_quotes?rev=1445710857Last update: **2020/06/02 14:29**