

# Table of Contents



- “A mess is not technical debt. A mess is just a mess.” – “Uncle” Bob Martin
- “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.
- “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
- “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.
- “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
- “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
- “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
- “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, It 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](mailto: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.” – Mary and Tom Poppendieck, from “Implementing Lean Software Development” (pg 31-32).
- “Everybody has a plan until they get punched in the mouth.” - Mike Tyson
- “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.
- “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's law
- “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
- “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
- “Trying to improve software quality by increasing the amount of testing is like trying to lose weight by weighing yourself more often. ... 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)
- “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.
- “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

funny, quote

From:

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

Permanent link:

[https://www.hanssamios.com/dokuwiki/quotable\\_quotes\\_am](https://www.hanssamios.com/dokuwiki/quotable_quotes_am)

Last update: **2023/03/07 09:32**

