SCRUM - The Art Of Doing Twice The Work In Half The Time by Jeff Sutherland

The Takeaway from Chapter One: The Way the World Works Is Broken

Planning Is Useful. Blindly Following Plans Is Stupid. It's just so tempting to draw up endless charts. All the work needed to be done on a massive project laid out for everyone to see—but when detailed plans meet reality, they fall apart. Build into your working method the assumption of change, discovery, and new ideas.

Inspect and Adapt. Every little while, stop doing what you're doing, review what you've done, and see if it's still what you should be doing and if you can do it better.

Change or Die. Clinging to the old way of doing things, of command and control and rigid predictability, will bring only failure. In the meantime, the competition that is willing to change will leave you in the dust.

Fail Fast So You Can Fix Early. Corporate culture often puts more weight on forms, procedures, and meetings than on visible value creation that can be inspected at short intervals by users. Work that does not produce real value is madness. Working product in short cycles allows early user feedback and you can immediately eliminate what is obviously wasteful effort.

The Takeaway from Chapter Two: The Origins of Scrum

Hesitation Is Death. Observe, **O**rient, **D**ecide, **A**ct. Know where you are, assess your options, make a decision, and act!

Look Outward for Answers. Complex adaptive systems follow a few simple rules, which they learn from their environment.

Great Teams Are. They are cross-functional, autonomous, and empowered, with a transcendent purpose.

Don't Guess. Plan, **D**o, **C**heck, **A**ct. Plan what you're going to do. Do it. Check whether it did what you wanted. Act on that and change how you're doing things. Repeat in regular cycles, and, by doing so, achieve continuous improvement.

Shu Ha Ri. First, learn the rules and the forms, and once you've mastered them, make innovations. Finally, in a heightened state of mastery, discard the forms and just be—with all the learning internalized and decisions made almost unconsciously.

The Takeaway from Chapter Three: Teams

Pull the Right Lever. Change Team performance. That has much more impact—by several orders of magnitude—than individual performance.

Transcendence. Great teams have a purpose that is greater than the individual; e.g., burying General MacArthur, winning the NBA championship.

Autonomy. Give teams the freedom to make decisions on how to take action—to be respected as masters of their craft. The ability to improvise will make all the difference, whether the unit is reporting on a revolution in the Middle East or making a sale.

Cross-functional. The team must have every skill needed to complete a project, whether the mission is to deliver Salesforce.com software or capture terrorists in Iraq.

Small Wins. Small teams get work done faster than big teams. The rule of thumb is seven team members—plus or minus two. Err on the small side.

Blame Is Stupid. Don't look for bad people; look for bad systems—ones that incentivize bad behavior and reward poor performance.

The Takeaway from Chapter Four: Time

Time Is Finite. Treat It That Way. Break down your work into what can be accomplished in a regular, set, short period—optimally one to four weeks. And if you've caught the Scrum fever, call it a Sprint.

Demo or Die. At the end of each Sprint, have something that's done—something that can be used (to fly, drive, whatever).

Throw Away Your Business Cards. Titles are specialized status markers. Be known for what you do, not how you're referred to.

Everyone Knows Everything. Communication saturation accelerates work.

One Meeting a Day. When it comes to team check-ins, once a day is enough. Get together for fifteen minutes at the Daily Stand-up, see what can be done to increase speed, and do it.

The Takeaway from Chapter Five: Waste Is a Crime

Multitasking Makes You Stupid. Doing more than one thing at a time makes you slower and worse at both tasks. Don't do it. If you think this doesn't apply to you, you're wrong—it does.

Half-Done Is Not Done. A half-built car simply ties up resources that could be used to create value or save money. Anything that's "in process" costs money and energy without delivering anything.

Do It Right the First Time. When you make a mistake, fix it right away. Stop everything else and address it. Fixing it later can take you more than twenty times longer than if you fix it now.

Working Too Hard Only Makes More Work. Working long hours doesn't get more done; it gets less done. Working too much results in fatigue, which leads to errors, which leads to having to fix the thing you just finished. Rather than work late or on the weekends, work weekdays only at a sustainable pace. And take a vacation.

Don't Be Unreasonable. Goals that are challenging are motivators; goals that are impossible are just depressing.

No Heroics. If you need a hero to get things done, you have a problem. Heroic effort should be viewed as a failure of planning.

Enough with the Stupid Policies. Any policy that seems ridiculous likely is. Stupid forms, stupid meetings, stupid approvals, stupid standards are just that—stupid. If your office seems like a Dilbert cartoon, fix it.

No Assholes. Don't be one, and don't allow the behavior. Anyone who causes emotional chaos, inspires fear or dread, or demeans or diminishes people needs to be stopped cold.

Strive for Flow. Choose the smoothest, most trouble-free way to get things done. Scrum is about enabling the most flow possible.

The Takeaway from Chapter Six: Plan Reality, Not Fantasy

The Map Is Not the Terrain. Don't fall in love with your plan. It's almost certainly wrong.

Only Plan What You Need To. Don't try to project everything out years in advance. Just plan enough to keep your teams busy.

What Kind of Dog Is It? Don't estimate in absolute terms like hours—it's been proven that humans are terrible at that. Size things relatively, by what breed of dog the problem is, or T-shirt size (S, M, L, XL, XXL), or, more commonly, the Fibonacci sequence.

Ask the Oracle. Use a blind technique, like the Delphi method, to avoid anchoring biases such as the halo effect or bandwagon effect or just plain stupid groupthink.

Plan with Poker. Use Planning Poker to quickly estimate work that needs to be done.

Work Is a Story. Think first about who'll be getting value from something, then about what it is, and then why they need it. Humans think in narratives, so give them one. As an X, I want Y, so that Z.

Know Your Velocity. Every team should know exactly how much work they can get done in each Sprint. And they should know how much they can improve that velocity by working smarter and removing barriers that are slowing them down.

Velocity × Time = Delivery. Once you know how fast you're going, you'll know how soon you'll get there.

Set Audacious Goals. With Scrum it is not that hard to double production or cut delivery time in half. If you do it in the right way, your revenue and stock price should double as well.

The Takeaway from Chapter Seven: Happiness

It's the Journey, Not the Destination. True happiness is found in the process, not the result. Often we only reward results, but what we really want to reward is people striving toward greatness.

Happy Is the New Black. It helps you make smarter decisions. Plus, when you're happy, you're more creative, less likely to leave your job, and more likely to accomplish far more than you ever anticipated.

Quantify Happiness. It's not enough just to feel good; you need to measure that feeling and compare it to actual performance. Other metrics look backward. Happiness is a future-looking metric.

Get Better Every Day—and Measure It. At the end of each Sprint, the team should pick one small improvement, or kaizen, that will make them happier. And that should become the most important thing they'll accomplish in the next Sprint.

Secrecy Is Poison. Nothing should be secret. Everyone should know everything, and that includes salaries and financials. Obfuscation only serves people who serve themselves.

Make Work Visible. Have a board that shows all the work that needs to be done, what is being worked on, and what is actually done. Everyone should see it, and everyone should update it every day.

Happiness Is Autonomy, Mastery, and Purpose. Everyone wants to control their own destiny, get better at what they do, and serve a purpose greater than themselves.

Pop the Happy Bubble. Don't get so happy that you start believing your own bullshit. Make sure happiness is measured against performance, and if there is a disconnect, be prepared to act. Complacency is the enemy of success.

The Takeaway from Chapter Eight: Priorities

Make a List. Check It Twice. Create a list of everything that could possibly be done on a project. Then prioritize it. Put the items with the highest value and lowest risk at the top of that Backlog, then the next, and then the next.

The Product Owner. She translates vision into Backlog. She needs to understand the business case, the market, and the customer.

A Leader Isn't a Boss. A Product Owner sets out what needs to be done and why. How the team accomplishes it and who accomplishes it is up to the team.

The Product Owner: Has knowledge of the domain and the power to make final decisions. He or she is available to answer questions and is accountable for delivering value.

Observe, Orient, Decide, Act (OODA). See the whole strategic picture, but act tactically and quickly.

Fear, Uncertainty, and Doubt. It's better to give than to receive. Get inside your competition's OODA loop and wrap them up in their own confusion.

Get Your Money for Nothing, and Your Change for Free. Create new things only as long as those new things deliver value. Be willing to swap them out for things that require equal effort. What in the beginning you thought you needed is never what is actually needed.

The Takeaway from Chapter Nine: Change the World

Scrum Accelerates All Human Endeavors. The type of project or problem doesn't matter—Scrum can be used in any endeavor to improve performance and results.

Scrum for Schools. In the Netherlands, a growing number of teachers are using Scrum to teach high school. They see an almost immediate improvement in test scores of more than 10 percent. And they're engaging all sorts of students, from vocational to gifted.

Scrum for Poverty. In Uganda, the Grameen Foundation is using Scrum to deliver agricultural and market data to poor rural farmers. The result: double the yield and double the revenue for some of the poorest people on the planet.

Rip Up Your Business Cards. Get rid of all titles, all managers, all structures. Give people the freedom to do what they think best and the responsibility to be accountable for it. You'll be surprised at the results.

IMPLEMENTING SCRUM—HOW TO BEGIN

1. Pick a **Product Owner**. This person is the one with the vision of what you are going to do, make, or accomplish. They take into account risks and rewards, what is possible, what can be done, and what they are passionate about. (See Chapter Eight: Priorities for more.)

2. Pick a **Team**. Who will be the people actually doing the work? This team needs to have all the skills needed to take the Product Owners' vision and make it a reality. Teams should be small, 3 to 9 people is the rule of thumb. (See Chapter Three: Teams for more.)

3. Pick a **Scrum Master**. This is the person who will coach the rest of the team through the Scrum framework, and help the team eliminate anything that is slowing them down. (See Chapter Four: Waste for more.)

4. Create and prioritize a **Product Backlog**. This is a list at a high level of everything that needs to be built or done to make that vision a reality. This backlog exists and evolves over the lifetime of the product; it is the product road map. At any point, the Product Backlog is the single, definitive view of "everything that could be done by the team ever, in order of priority." Only a single Product Backlog exists; this means the Product Owner is required to make prioritization decisions across the entire spectrum. The Product Owner should consult with all stakeholders and the team to make sure they are representing both what people want and what can be built. (See Chapter Eight: Priorities for more.)

5. Refine and Estimate the **Product Backlog**. It is crucial that the people who are actually going to complete the items in the Product Backlog estimate how much effort they will take. The team should look at each Backlog item, and see if it is actually doable. Is there enough information to complete the item? Is it small enough to estimate? Is there a Definition of Done, that is, everyone agrees on what standards must be met to call something "done"? Does it create visible value? Each item must be able to be shown, to be demonstrated, hopefully to be potentially shippable. Do not estimate the Backlog in hours, because people are absolutely terrible at that. Estimate by relative size: Small, Medium, or Large. Or even better use the Fibonacci sequence and estimate the point value for each item: 1, 2, 3, 5, 8, 13, 21, etc. (See Chapter Six: Plan Reality, Not Fantasy for more.)

6. **Sprint Planning**. This is the first of the Scrum meetings. The team, the Scrum Master, and the Product Owner sit down to plan the Sprint. Sprints are always a fixed length of time that is less than a month. Most people now run one- or two-week Sprints. The team looks at the top of the Backlog and forecasts how much of it they can complete in this Sprint. If the team has been going for a few Sprints, they should take in the number of points they did in the last Sprint. That number is known as the team's **Velocity**. The Scrum Master and the team should be trying to increase that number every Sprint. This is another chance for the team and the Product Owner to make sure that everyone understands exactly how these items are going to fulfill the vision. Also during this meeting everyone should agree on a Sprint Goal, what everyone wants to accomplish with this Sprint.

   One of the pillars of Scrum is that once the team has committed to what they think they can finish in one Sprint, that's it. It cannot be changed, it cannot be added to. The team must be able to work autonomously throughout the Sprint to complete what they forecast they could. (See Chapter Four: Time and Chapter Six: Plan Reality, Not Fantasy for more.)

7. **Make Work Visible**. The most common way to do this in Scrum is to create a **Scrum Board** with three columns: To Do, Doing, Done. Sticky notes represent the items to be completed and the team moves them across the Scrum board as they are completed, one by one.

   Another way to make work visible is to create a **Burndown Chart**. On one axis is the number of points the team has taken into the Sprint, on the other is the number of days. Every day the Scrum Master tallies up the number of points completed and graphs them on the Burndown chart. Ideally there will be a steep downward slope leading to zero points left on the last day of the Sprint. (See Chapter Seven: Happiness for more.)

8. **Daily Stand-up** or **Daily Scrum**. This is the heartbeat of Scrum. Each day, at the same time, for no more than fifteen minutes, the team and the Scrum Master meet and answer three questions:

   • What did you do yesterday to help the team finish the Sprint?

   • What will you do today to help the team finish the Sprint?

   • Is there any obstacle blocking you or the team from achieving the Sprint Goal?

   That's it. That's the whole meeting. If it takes more than fifteen minutes, you're doing it wrong. What this does is help the whole team know exactly where everything is in the Sprint. Are all the tasks going to be completed on time? Are there opportunities to help other team members overcome obstacles? There's no assigning of tasks from above—the team is autonomous; they do that. There's no detailed reporting to management. The Scrum Master is responsible for making the obstacles to the team's progress, or impediments, go away. (See Chapter Four: Time and Chapter Six: Plan Reality, Not Fantasy for more.)

9. **Sprint Review** or **Sprint Demo**. This is the meeting where the team shows what they have accomplished during the Sprint. Anyone can come, not only the Product Owner, the Scrum Master, and the team, but stakeholders, management, customers, whoever. This is an open meeting where the team demonstrates what they were able to move to Done during the Sprint.

   The team should only demo what meets the Definition of Done. What is totally and completely finished and can be delivered without any more work. It may not be a completed product, but it should be a completed feature of one. (See Chapter Four: Time for more.)

10. **Sprint Retrospective**. After the team has shown what they've accomplished during the last Sprint—that thing that is "done" and can potentially be shipped to customers for feedback—they sit down and think about what went right, what could have gone better, and what can be made better in the next Sprint. What is the improvement in the process that they, as a team, can implement right away?

    To be effective, this meeting requires a certain amount of emotional maturity and an atmosphere of trust. The key thing to remember is that you're not seeking someone to blame; you're looking at the process. Why did that happen that way? Why did we miss that? What could make us faster? It is crucial that people as a team take responsibility for their process and outcomes, and seek solutions as a team. At the same time, people have to have the fortitude to bring up the issues that are really bothering them in a way that is solution oriented rather than accusatory. And the rest of the team has to have the maturity to hear the feedback, take it in, and look for a solution rather than getting defensive.

    By the end of the meeting the team and the Scrum Master should agree on one process improvement that they will implement in the next Sprint. That process improvement, sometimes called the kaizen, should be put into the next Sprint's backlog, with acceptance tests. That way the team can easily see if they actually implemented the improvement, and what effect it had on velocity. (See Chapter Seven: Happiness for more.)

11. Immediately start the next Sprint cycle, taking the Team's experience with impediments and process improvements into account.