

PRACTICAL SCRUM

A GUIDE TO SURVIVING THE AGILE APOCALYPSE



SHIEL YULE

Practical Scrum: A Guide to Surviving the Agile Apocalypse

*A Guide to Surviving the Agile Apocalypse: Fighting
Scrum-But, Eliminating Anti-Patterns, and Delivering
Real Value.*

First published by SHIEL YULE 2025

Copyright © 2025 by Shiel Yule

All rights reserved. No part of this publication may be reproduced, stored or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise without written permission from the publisher. It is illegal to copy this book, post it to a website, or distribute it by any other means without permission.

This novel is entirely a work of fiction. The names, characters and incidents portrayed in it are the work of the author's imagination. Any resemblance to actual persons, living or dead, events or localities is entirely coincidental.

Shiel Yule asserts the moral right to be identified as the author of this work.

Designations used by companies to distinguish their products are often claimed as trademarks. All brand names and product names used in this book and on its cover are trade names, service marks, trademarks and registered trademarks of their respective owners. The publishers and the book are not associated with any product or vendor mentioned in this book. None of the companies referenced within the book have endorsed the book.

Second edition

ISBN: 978-1-9193421-9-1

Illustration by Gemini

Illustration by Shiel Yule

This book was professionally typeset on Reedsy.

Find out more at reedsy.com

Contents

| | |
|--|----|
| Prologue | 1 |
| Introduction | 3 |
| The Scrum Definition: More Than a Buzzword | 5 |
| Scrum Theory: The Three Pillars of Chaos | 8 |
| The Scrum Values: Moral Compass in a World of Chaos | 12 |
| The Scrum Team: Your Survival Unit | 16 |
| The Scrum Artifacts: Your Inventory | 20 |
| The Scrum Events: Your Survival Rituals | 24 |
| The Servant-Leader's Shield (The Scrum Master) | 28 |
| Advanced Survival Topics: Navigating the Wasteland's Hidden... | 32 |
| Other Agile Survivors in the Corporate Ruin | 40 |
| The Statistical Zealots (The Lean Six Sigma Cult of... | 46 |
| Lessons from the Battlefield | 49 |
| The Choice of Survival | 53 |
| Appendices | 55 |

Prologue

The Long March to the Wasteland

The world wasn't born chaotic. No, chaos was *engineered*.

There was once a Golden Age—the mid-1970s—a time of unshakable certainty. It was the era of **Waterfall Methodology**, where project management was treated like civil engineering. You drew a perfect blueprint, carved the steps into stone, and didn't move an inch until the previous phase was declared **DONE** (which usually meant “ready for the next phase to discover all the mistakes”). Change was heresy. Managers were gods, and the Gantt chart was their sacred text. Projects took five years to deliver, and if the market changed on Day 3, well, tough luck. The resulting product was almost always exactly what the customer asked for five years ago, making it perfectly useless now.

This golden age of rigidity survived for decades, lumbering into the new millennium like a slow, heavily armored dinosaur. The first cracks appeared in the late 1990s and early 2000s, driven by a growing, desperate need for speed. A few rebels, tired of watching their multi-million dollar plans turn into dust, started experimenting with short, iterative cycles. Among them, a few bright sparks formalised a lightweight set of rules: **Scrum**.

Then came the **Agile Manifesto of 2001**. It wasn't a blueprint; it was a simple prayer: *Individuals and interactions over processes and tools. Working software over comprehensive documentation*. It was the sound of a starting pistol, signalling the desperate race away from the behemoth of bureaucracy.

But the corporate world is a digestive tract, not a brain. It consumes, renames, and excretes the new, fashionable thing. When the **2008 financial**

crash hit, and money became tight, C-suites suddenly saw “Agile” not as a path to quality, but as a shiny, new tool for **cutting costs and speeding up delivery**. They didn’t adopt the *philosophy*; they simply bought the t-shirts, hired the consultants, and began performing the rituals like a cargo cult.

And that is how the apocalypse truly began.

The ensuing years saw the rapid, uncontrolled proliferation of “Agile.” It metastasised into an endless, self-replicating bureaucratic machine: **SAFe** bloomed into a thousand-page instruction manual for being “flexible,” **Disciplined Agile** promised a methodology for every mood, and an army of unqualified **coaches** roamed the halls, selling certifications and complex, unnecessary diagrams. The simple, elegant Scrum framework was drowned under a thousand added steps.

By the time the **2020 pandemic** arrived, forcing a sudden, brutal adaptation to remote work, many organisations were structurally incapable of responding. They had practiced *fake* agility for a decade, and when true chaos demanded simple, empirical adaptation, they collapsed into the **Scrum-But Badlands**. They had the titles and the terminology, but they lacked the core courage, focus, and transparency to survive.

This book is not a eulogy for that world. It is a field manual for the one that remains. It is a guide for those of us who saw the light—the elegant simplicity of the original Scrum framework—and are determined to use it, not just to survive the corporate ruin left behind, but to rebuild something better. The zombies of bureaucracy are still marching, but you now have the tools to fight back.

Introduction

Why I Wrote This Field Manual (from the author)

Welcome, weary traveler, to a guide that will help you navigate the chaos.

I've been in the trenches since the Agile Apocalypse began. I learned Scrum back in **2010 working in the Games industry**, a place where “crunch culture” was just another name for mandatory overtime, and the only certainty was the endless march toward an impossible ship date. Scrum was the first glimpse of a life raft.

Since then, I've seen it all. I've been an **Agile Coach** desperately trying to implement sane practices in major finance companies—organisations so vast and entrenched they make the ruins of Detroit look flexible. I've also run my own businesses, from a serene **Yoga Studio** (which, ironically, needed a ruthless focus on efficiency) to a **Scrum Master recruitment company** (Scrum Master Ltd), proving that even in the business of selling agility, bureaucracy is the final boss.

You have been told that “Agile” is the silver bullet. You've witnessed countless iterations of Agile fail over and over, leaving behind a trail of jargon-filled spreadsheets and meetings that feel designed to test your will to live.

The good news? The Agile Apocalypse is survivable. Your flashlight in the darkness, your trusty can opener for the metaphorical tinned food of progress, is **Scrum**.

This is not a eulogy for the old world. It is a field manual for the one that remains. It is a guide for those of us determined to use the elegant simplicity of the original Scrum framework, not just to survive the corporate ruin left

behind, but to rebuild something better. The zombies of bureaucracy are still marching, but you now have the tools to fight back. Don't panic. This isn't just another textbook. **This is your field manual.**

By Shiel Yule

ISBN: 978-1-9193421-9-1
Copyright Shiel Yule 2021 - 2025



The Scrum Definition: More Than a Buzzword



Official Definition (the gospel according to Ken)

Scrum is a lightweight framework that helps people, teams and organisations generate value through adaptive solutions for complex problems. In a nutshell, Scrum requires a **Scrum Master** to foster an environment where:

1. A **Product Owner** orders the work for a complex problem into a **Product**

Backlog.

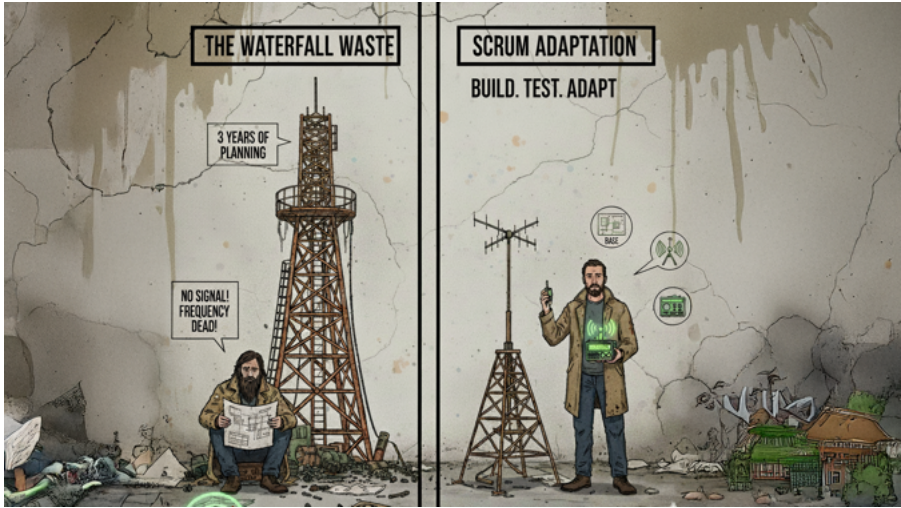
2. The **Scrum Team** turns a selection of the work into an **Increment** of value during a **Sprint**.
3. The Scrum Team and its stakeholders inspect the results and adjust for the next Sprint.

Our Interpretation (reality hurts)

Scrum is the beautifully simple instruction manual your company keeps trying to use as a coaster. It's so lightweight, it often gets blown away by the first gust of wind from an angry stakeholder. The core idea is simple: You have a team, a pile of work, and you do a little bit of it at a time. Then you look at what you've done and figure out how to be slightly less terrible next time.

The **Scrum Master** isn't just a facilitator; they're the head trauma specialist for the team, the one who cleans up the messes left by misunderstood "Agile" practices. The **Product Owner** is the one person brave enough to decide what to build, and they organise the work into a **Product Backlog**—a list so long and ever-changing, it could be mistaken for a grocery list for a family of forty. The team takes a small chunk of that work, turns it into an **Increment** (a fancy word for a "thing that works"), and then you all get together to squint at it and wonder what went wrong.

If you are just starting your journey into the apocalypse, try the simple rules first. Do not add anything to the Scrum framework. Just follow the rules as written and see what happens. The framework is designed to make all your organisation's dirty laundry, terrible processes, and dysfunctional relationships visible. It's like a magical mirror that shows everyone the flaws they've been ignoring.



Imagine if you will: A lone survivor, trying to rebuild a radio tower. They spend years meticulously planning every bolt and wire, only to discover after three years the radio tower they built is utterly useless because the frequency they were going to use is no longer available. Now, imagine if they had used Scrum instead. They would have built the base, then the first antenna, and tested the frequency. They would have discovered the problem, adapted, and avoided wasting three years of their life.

Scrum Theory: The Three Pillars of Chaos



Official Theory (Jeff's existential jottings)

Scrum is founded on **empiricism** and **lean thinking**. Empiricism asserts that knowledge comes from experience and making decisions based on what is observed. Lean thinking reduces waste and focuses on the essentials. Scrum employs an iterative, incremental approach to optimise predictability and to control risk.

Scrum combines four formal events for inspection and adaptation within a containing event, the **Sprint**. These events work because they implement the

empirical Scrum pillars of **transparency**, **inspection**, and **adaptation**.

Our Interpretation (the Zombie Survival Guide)

In the Agile Apocalypse, you can't rely on ancient prophecies or long-term plans that will fall apart when the first zombie (or market shift) appears. You need to rely on what you see, what you experience, and what you can do about it. This is **Empiricism**. Our experience is that the servers always crash on Fridays, and that the "must-have" feature in January will be utterly irrelevant by June.

Lean Thinking is our second best friend. It's about reducing waste, which in your world means reducing the amount of time you spend in meetings that could have been an email, reducing the number of processes that exist only because "that's how we've always done it," and reducing the amount of work you're not going to use anyway.

The three great pillars of Scrum are the unshakable foundation you'll need to survive.

Transparency

Official: The emergent process and work must be visible to those performing the work as well as those receiving the work.

Reality: This means your work is out in the open. It's no longer a secret you can hide in a dusty, ancient spreadsheet. The Product Backlog, the Sprint Backlog, and the Increment are all visible, like a crime scene photo. If you try to hide anything, it's not going to end well. Transparency enables everyone to see the truth, no matter how ugly. Inspection without it is just a wasted opportunity to point fingers in the dark.

Inspection

Official: The Scrum artifacts and the progress toward agreed goals must be inspected frequently and diligently to detect potentially undesirable variances or problems.

Reality: Think of the Scrum events as your checkpoints. Are we still on the right path? Did that last feature we built actually work? Are we all still alive? This is the moment to get out your flashlight and sift through the rubble of the last Sprint to see what's still standing. Inspection without adaptation is pointless. It's like finding a leak in the roof and just shrugging.

Adaptation

Official: If any aspects of a process deviate outside acceptable limits or if the resulting product is unacceptable, the process being applied or the materials being produced must be adjusted.

Reality: When you find a problem, you have to actually fix it. This is where the zombies come in. It's no good noticing that half your team is running in circles if you don't change the plan. A Scrum Team is expected to adapt the moment it learns anything new—or at least before the zombie horde breaks through the barricades.



Imagine if you will: A scout team in the corporate wilderness. The first time they see a zombie, they panic and run. The second time, they stop and notice the zombies are slow. The third time, they adapt, arm themselves with baseball bats, and take them out with ease. That is empiricism in action.

The Scrum Values: Moral Compass in a World of Chaos



Official Values (the Hippocratic Oath)

When the Scrum Team commits to the values of **Commitment**, **Focus**, **Openness**, **Respect**, and **Courage**, the empirical pillars of Transparency, Inspection, and Adaptation come to life and build trust. The Scrum Team members learn and explore those values as they work with the Scrum events and artifacts.

Our Interpretation (the Rules of the Road)

These five values are your moral compass in the corporate wasteland. They're the unwritten rules that keep you from stabbing your teammate in the back for the last can of soda.

Commitment

Official: The Scrum Team members personally commit to achieving the goals of the team.

Reality: This isn't about working 80 hours a week. It's about being personally committed to the team's success, which means doing your part to keep everyone safe. It's the solemn vow to not add a new feature request to the Sprint in the middle of a zombie attack. It's a promise to not abandon your team when a senior stakeholder (the Head Zombie) starts breathing down your neck.

Focus

Official: Everyone focuses on the work of the Sprint and the goals of the Scrum Team.

Reality: This is the most important rule. You've got a pile of work to do. Don't get distracted by shiny objects, new requests, or internal politics. It's the value that stops you from building a magnificent, gold-plated zombie-slaying-sword when what you actually need is a simple, effective baseball bat. Stay focused on the Sprint Goal. It's the only way to avoid the brain-eating distractions that litter the corporate landscape.

Openness

Official: The Scrum Team and its stakeholders are open about the work and the challenges with performing the work.

Reality: This means admitting when you are stuck, when something is

harder than you thought, or when you have no idea how you're going to finish that task. It's like yelling "I'M LOST!" in a zombie-infested mall. Scary, but necessary for survival. Without openness, you're just pretending to be fine as the horde of impediments gets closer.

Respect

Official: Scrum Team members respect each other to be capable, independent people.

Reality: You respect your teammates enough to not throw them under the bus when things get tough. You trust they are capable enough to handle their portion of the zombie-slaying, and they trust you with yours. This also means respecting the roles—the Product Owner respects the Developers' opinion on how long something will take, and the Developers respect the Product Owner's authority on what to build.

Courage

Official: The Scrum Team members have the courage to do the right thing and work on tough problems.

Reality: This is the courage to say "no." To your boss, to a stakeholder, to a ridiculous deadline. The courage to admit a failure and learn from it instead of pretending everything is fine as the building crumbles around you. It's the courage to be honest about the state of the product, even if it means admitting you have a rusty baseball bat instead of that shiny sword you were promised.



Imagine if you will: A young Developer, fresh to the Scrum Team, has been assigned a task to “integrate the legacy mainframe system.” They’ve never seen a mainframe before. The value of openness gives them the courage to say, “I’m in over my head,” and the value of respect gives their teammates the confidence to help them learn, instead of letting them fail alone.

The Scrum Team: Your Survival Unit



Official Definition (the Fellowship of the Ring)

The fundamental unit of Scrum is a small team of people, the **Scrum Team**. The Scrum Team consists of one **Scrum Master**, one **Product Owner**, and **Developers**. Within a Scrum Team, there are no sub-teams or hierarchies. It is a cohesive unit of professionals focused on one objective at a time, the **Product Goal**.

Our Interpretation (your Squad in the Apocalypse)

You're not in this alone. The Scrum Team is your squad, a small, self-organising unit of survivors. There are no sub-teams, no mini-fiefdoms, no hierarchy. You're all in it together. Every person on the team is a pro, and they're all focused on a single mission: creating a product that will help you all survive.

Product Owner

Official: The Product Owner is accountable for maximising the value of the product resulting from the work of the Scrum Team.

Reality: The person with the map, the compass, and the questionable decisions about which direction to run. They decide which hoard of zombies to face first. The Product Owner is the sole decision-maker for the **Product Backlog** and must have a clear vision. They're the one person brave enough to say "this is what we're doing" when everyone else is shouting in a panic.

Developers

Official: Developers are the people in the Scrum Team who are committed to creating any aspect of a usable Increment each Sprint.

Reality: The hands-on fighters. They do the actual building, the hacking, and the scavenging. They're the ones who get their hands dirty building the barricades and setting the traps. They are the ones who turn the Product Owner's vision into a working, usable product. They are the ones who are on the front line every day, facing the apocalypse head-on.

Scrum Master

Official: The Scrum Master is accountable for establishing Scrum as defined in the Scrum Guide. They do this by helping everyone understand Scrum theory and practice, both within the Scrum Team and the organisation.

Reality: The medic, the therapist, and the bodyguard all in one. They don't direct the team, but they help them navigate the chaos. They clear the path for the team by removing impediments, they coach the Product Owner on maximising value, and they protect the team from outside threats and internal strife. They're the silent guardian, the unsung hero, the one who makes sure you have a chance at survival.



Imagine if you will: A team without a Scrum Master. The Developers are so focused on building a barricade that they don't notice the growing pile of zombie bones in their way—an impediment they keep tripping over. The Product Owner, meanwhile, is trying to tell them to build a catapult, even though they have no wood. The team is in chaos.

Self-Management: The Squad's Autonomy

Official: Scrum Teams are self-managing, meaning they internally decide who does what, when, and how.

Reality: In the apocalypse, you don't have time to wait for a manager to tell you to reload your weapon. Your squad has to make its own decisions

THE SCRUM TEAM: YOUR SURVIVAL UNIT

about how to fight the horde. The Scrum Team is empowered to decide how to best accomplish the work in the Sprint, without micromanagement from outside the team. This is a critical element of survival. If you have to ask for permission to do your job, you've already lost.

The Scrum Artifacts: Your Inventory



Scrum's artifacts represent work or value. In the apocalypse, these are the tools and supplies you need to keep track of your progress. Like any good RPG, you need to know what you're carrying and what it's for. They are designed to maximise **transparency** of key information, so everyone knows exactly what's in the pack and what's on the menu for dinner.

The Product Backlog: The Endless Quest Log

Official: The Product Backlog is an emergent, ordered list of what is needed to improve the product. It is the single source of work undertaken by the Scrum

Team.

Reality: Think of the **Product Backlog** as your Pip-Boy from *Fallout*. It contains every single quest, side mission, and fetch-and-carry task you can think of, all ranked by importance. The Product Owner is the one with the “Admin” rights, managing this endless list of to-dos.

Product Backlog Management: Keeping Your Pip-Boy Tidy

In the apocalypse, an organised inventory is the difference between life and death. You don’t want to be fumbling for a stimpack while a zombie is gnawing on your leg. The Product Owner is responsible for **Product Backlog management**, which includes:

- **Refinement:** This is the ongoing process of breaking down large, vague quests into smaller, more actionable tasks. It’s like turning a quest to “Find the Legendary Super-Powered Blaster” into smaller, digestible sub-quests: “Search the Ruined Police Station,” “Gather Parts from a Smashed Robot,” and “Fix the Power Grid.” This makes sure the team always has a stash of ready-to-go work.
- **Ordering:** The Product Owner is constantly reordering the list based on what is most important for survival. A quest to “Find a medical kit” is always going to be higher on the list than a quest to “Find a decorative lawn gnome.”
- **Transparency:** The Product Backlog should be visible to everyone—not just the team, but the stakeholders as well. This prevents managers from sneaking in urgent, low-value work and ensures everyone knows what the team’s priorities are.



Imagine if you will: A Product Owner whose Product Backlog is just a napkin with a single item scribbled on it: “SURVIVE.” The team, having no clear guidance, builds a magnificent fortress in the middle of a toxic swamp, ignoring the obvious need for food and water.

The Sprint Backlog: Your Tactical War Plan

Official: The Sprint Backlog is composed of the Sprint Goal, the set of Product Backlog items selected for the Sprint, and the plan for delivering the Increment. It is a highly visible, real-time picture of the work the Developers plan to accomplish during the Sprint.

Reality: This is your tactical war plan for the next 1-4 weeks. It’s the small, focused list of tasks your squad has committed to tackling in this particular survival mission. The Sprint Backlog is the detailed map of the area you’re raiding. It’s not a static document; it’s a living, breathing guide that the team will update as they learn more about their mission. If something changes—if you find a new shortcut or encounter an unexpected zombie—you update the plan.

The Increment: The Safe Zone You Just Built (ready for use)

Official: The Increment is a concrete stepping stone toward the Product Goal. Each Increment is a sum of all the Product Backlog items completed during a Sprint and the value of previous Increments. It is inspect-able and verifiable.

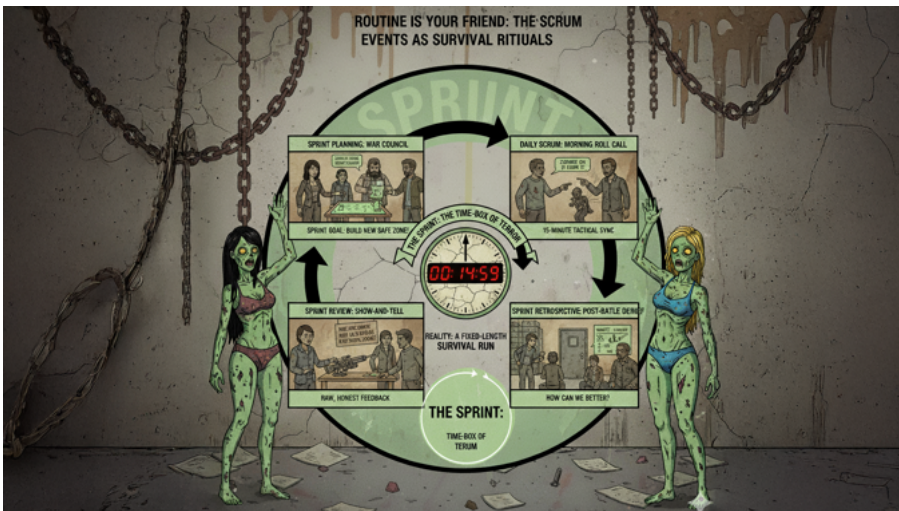
Reality: This is the new, usable piece of the product you just built. It's the new barricade you put up, the new trap you set, the new weapon you crafted. The key word here is "**usable.**" It must be in a condition where the Product Owner *could* release it to users immediately, even if they choose not to. It's not a half-finished blueprint or a box of parts. It's a completed, valuable, and **shippable** piece of a product that the stakeholders can inspect. It's the tangible proof that your survival rituals are working.

The "Done" Barricade: Ensuring the Zombies Don't Get Back Up

Official: The Definition of Done is a formal description of the state of the Increment when it meets the quality measures required for the product.

Reality: The **Definition of Done (DoD)** is your quality-control checklist. It's the set of rules that every single task has to follow before it can be considered "done." If you're building a fence, the DoD might be: "Posts are in the ground, boards are nailed on, and the gate swings freely." If you don't have a DoD, you're building a rickety, half-finished barricade that a single zombie could push over. A clear, visible, and respected DoD is your ultimate defense against a zombie-infested, buggy product.

The Scrum Events: Your Survival Rituals



In the apocalypse, routine is your friend. It's a way to maintain sanity and stability in a world that's constantly trying to chew your face off. The Scrum Events are your rituals. They're the non-negotiable moments in your schedule that force you to inspect your progress, adapt your plan, and keep the team alive.

The Sprint: The Time-Box of Terror

Official: The Sprint is a fixed-length event of one month or less to create consistency. A new Sprint starts immediately after the conclusion of the

previous Sprint.

Reality: The Sprint is a fixed-length survival run. It's a 1-to-4-week mission into the corporate wilderness to scavenge for resources and build a new safe zone. This is the **time-box** that everything else is stuffed into. It is the very heart of the Scrum framework and the one thing that should never, ever change. It's like a countdown clock on a bomb—when the time is up, the Sprint is over, whether you've finished everything or not.

Sprint Planning: The War Council Meeting

Official: Sprint Planning initiates the Sprint by laying out the work to be performed for the Sprint. The entire Scrum Team collaborates on the Sprint Plan.

Reality: This is where your team gathers in a huddle, like the crew of the Millennium Falcon, to plan your next hyper-drive jump. It's a tense negotiation between the Product Owner's vision of a glorious new future and the Developers' sober assessment of what's actually possible. The goal is to come out with a **Sprint Goal**—a single, unifying objective that will keep you from going off-mission—and a **Sprint Backlog** filled with the bare necessities. Remember, the Sprint Goal is your North Star.

Daily Scrum: The Morning Roll Call

Official: The purpose of the Daily Scrum is to inspect progress toward the Sprint Goal and adapt the Sprint Backlog as needed, adjusting the upcoming planned work.

Reality: This is your 15-minute morning check-in. It's the moment to look your fellow survivors in the eye and say, "I'm still alive, here's what I did yesterday, and here's how I'm going to contribute to the cause today." This isn't a status report for your manager; it's a quick, tactical sync with the squad. Are we still on track to meet our Sprint Goal? Are there any zombies (impediments) blocking our path? Is anyone in need of a rescue?

Sprint Review: The Show-and-Tell

Official: The purpose of the Sprint Review is to inspect the outcome of the Sprint and determine future adaptations.

Reality: The end-of-mission debrief. Your team presents the **Increment**—the new, usable, and valuable piece of the product you’ve just built—to your stakeholders. It’s a demo, not a lecture. The goal is to get raw, honest feedback. This is your chance to show off the cool new blaster you built and let your stakeholders try it out. If they love it, great. If they find a flaw, even better, because you can fix it before the zombie horde gets closer.

Sprint Retrospective: The Post-Battle Debrief

Official: The purpose of the Sprint Retrospective is to plan ways to increase quality and effectiveness.

Reality: The most important meeting you’ll ever have. The team huddles in a safe room, locks the door, and for a short while, talks openly about how to make things better. What went well? What went horribly wrong? How can we improve our processes, tools, or interactions to be more effective? This is where the team adapts its own process for the next Sprint. Ignoring this ritual is how you end up making the same stupid mistakes over and over again, until eventually, the zombies win.

The Servant-Leader's Shield (The Scrum Master)



The Scrum Master is the cornerstone of the Scrum Team, yet their role is the most frequently misunderstood. They are not a manager, a team secretary, or a project coordinator. They are a **Servant-Leader**, a silent guardian whose job is to coach the team, the Product Owner, and the organisation on the proper use of the framework. Think of them as the team's therapist, bodyguard, and martial arts instructor all rolled into one.

The SM as Coach to the Team (fostering Self-Management)

The Scrum Master's primary job within the squad is to ensure the **Developers** can survive autonomously. This means teaching them how to fight, not fighting for them.

- **Teaching Self-Management:** The SM coaches the team on how to decide *who* does *what*, *when*, and *how*. They step back and allow the team to fail safely, learning from the resulting chaos.
- **Conflict Resolution:** When two developers are fighting over the last slice of pizza (or the best approach to a technical problem), the SM steps in not to judge, but to facilitate an open, honest discussion based on the Scrum values.
- **Protecting the Time-Box:** The SM ruthlessly defends the team's ability to focus, ensuring the time-boxes for events (especially the Daily Scrum) are respected and productive.

Imagine if you will: A Scrum Master whose team is arguing in circles during Sprint Planning because they are paralysed by indecision. The SM doesn't assign tasks (that's micromanagement). Instead, the SM points to the **Commitment** and **Focus** values on the wall, reminding the team: "The framework requires you to choose a path. I trust your expertise. Decide and commit."

The SM as Coach to the Organisation (removing Systemic Impediments)

This is where the real fight begins. The Scrum Master must act as a corporate antibody, fighting the system-level diseases that infect the Scrum Team.

- **Fighting Bureaucracy:** They tackle systemic impediments—like a company-wide approval process that takes 10 days, or a mandatory status meeting that violates the Daily Scrum time-box. The SM has the courage to point out these organisational flaws to management.
- **Fighting Scrum-But:** They educate the broader organisation (management, sales, marketing) on how their actions create “Scrum-But.” They explain, for example, why adding work mid-Sprint breaks the team’s **Focus** and **Commitment**.
- **The Impediment Log:** The SM doesn’t *solve* all the impediments, but they track them transparently. An impediment that stays on the list for three Sprints is a sign of an organisational failure, which the SM must courageously expose to the necessary leadership.

Advanced Survival Topics: Navigating the Wasteland's Hidden Traps



You've mastered the rituals, but the wasteland throws more than just zombies at you. The truly dangerous threats are the human-centric traits—the bureaucratic pitfalls and the subtle killers of value—that lurk within your own organisation.

The Product Vision: Your Reason for Living

Official: The Product Vision describes the purpose of the product, its ultimate value, and what it hopes to achieve for its customers.

Reality: The Product Goal is your immediate destination, but the Product Vision is your reason for traveling in the first place. It's the grand, overarching purpose of your entire existence in the apocalypse. It's the dream of a new, zombie-free world. Without a compelling vision, the Product Goal feels like just another task, and the team will lack the motivation to push through the hardships.

The Undead Backlog: Advanced Backlog Management

You've got the basics down, but what about the more insidious threats?

Technical Debt: The Festering Zombie Bite

Official: Technical Debt represents a design or implementation choice that is sub-optimal but expedites a short-term outcome. The "debt" can be paid down by refactoring or improving the system.

Reality: Technical Debt is the zombie bite you get during a raid. You can ignore it for now to get back to the safe zone, but if you don't take time to clean and treat it, it will fester and turn you into one of the undead. This could be a poorly written feature, a lack of automated tests, or a shortcut taken to hit a deadline. If you don't budget time to "pay down" this debt, your code will become slow, buggy, and impossible to work with.

Splitting Your Epics: The Art of Dissection

Official: In Scrum, large requirements (often called Epics or Themes) are broken down into smaller, manageable pieces (Product Backlog Items) that can be completed within a single Sprint.

Reality: An Epic is a massive zombie you can't take down in one go. You have

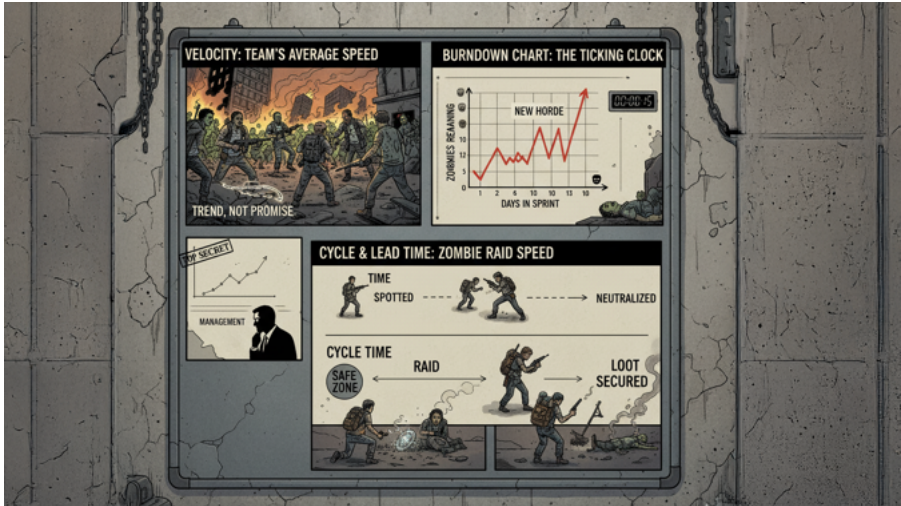
to break it down into smaller, more manageable pieces you can actually fight. A quest to “Find a Cure for the Zombie Virus” is an Epic. A task to “Scavenge for Antiviral Samples in the Ruined Lab” is something a team can actually do in a Sprint. This is the art of dissection, turning a terrifying, monolithic problem into a series of small, winnable battles.

Estimation: Predicting the Chaos

Official: Scrum teams often use techniques like Story Points to estimate the relative effort required to complete a Product Backlog Item. It is a forecast, not a commitment.

Reality: Estimation is the process of guessing how many baseball bats it will take to neutralise a zombie you’ve only seen from a distance. **Story Points** are not a measure of time, but a measure of relative effort or difficulty. A small zombie is a 1-point task; a giant, armoured zombie is a 13-point task. It’s a way for your team to agree on the size of the monster they’re about to face, without anyone promising they’ll take it down in exactly 2 hours and 47 minutes.

Scrum Metrics: Your Geiger Counter and Compass



In the apocalypse, you need data to survive. You need to know if you're making progress, if your path is safe, and if your team is moving at a sustainable pace. Scrum Metrics are the essential tools you use to inspect your own processes and adapt. They are not for micromanagement; they are for survival.

Velocity: Your Team's Average Speed

Official: Velocity is the sum of the sizes of the Product Backlog items delivered in a Sprint. It is a measure of a team's capacity and can be used to forecast future work.

Reality: Your team's Velocity is the number of zombies (or Story Points) your squad can reliably take down in a single Sprint. It's a historical trend, not a promise. It tells you your average speed through the wasteland, so you can make educated guesses about how long it will take to reach the next safe zone. A team's Velocity should be kept secret from management; otherwise, they'll use it to make ludicrous promises to stakeholders, and then you'll all get eaten.

Burndown Charts: The Ticking Clock

Official: A Burndown Chart is a graphical representation of the work remaining in a Sprint. The Y-axis represents the work remaining, and the X-axis represents the time in the Sprint.

Reality: This is a chart showing the shrinking horde of zombies left to kill in the Sprint. The line should be going down, indicating progress. If it suddenly shoots up, it means an unexpected horde of zombies showed up (i.e., new work was added to the Sprint), and now you're in trouble. A Burndown Chart is a daily visual cue that keeps everyone focused on the Sprint Goal.

Cycle Time & Lead Time: The Speed of a Zombie Bite

Official: **Lead Time** is the time from when a Product Backlog Item is created to when it is delivered. **Cycle Time** is the time from when a Developer begins working on a task to when it is finished.

Reality: **Lead Time** is how long it takes for a zombie to go from first being spotted to its final neutralisation. **Cycle Time** is the time from when your team starts a raid to when the last piece of loot is secured. These are key metrics for efficiency. A long cycle time indicates there might be internal blockages, like a team member getting distracted by a shiny object or a poorly maintained tool.

Scrum-But: A Guide to the Badlands



You've made it this far, but you're not out of the woods yet. Many who think they are doing Scrum are, in fact, living in the Badlands, a desolate place where the framework's rules are selectively ignored for "convenience." This is the realm of **Scrum-But**—as in, "We do Scrum, but..."

The Headless Zombie: "We do Scrum, but we don't have a Product Owner."

Official: The Product Owner is accountable for maximising the value of the product and is the sole person responsible for the Product Backlog.

Reality: In this scenario, the team is a headless zombie, running around in circles without a clear sense of direction or what to attack. Without a single person responsible for the backlog, work gets prioritised by who yells loudest, and the team will inevitably build something that no one wants or needs. The result is a slow, brain-dead, and ultimately useless product.

The Endless Monologue: “We do Scrum, but our Daily Scrum is a 30-minute status report.”

Official: The Daily Scrum is a 15-minute event for the Developers of the Scrum Team to inspect their progress toward the Sprint Goal.

Reality: This is the endless monologue, where one person talks while the others secretly check their phones. Instead of a quick, tactical sync, it’s a boring, top-down status report for the manager who isn’t even part of the team. This is a waste of precious survival time and a clear sign that the team doesn’t trust each other.

The Flying Object: “We do Scrum, but management adds new work to the Sprint all the time.”

Official: No changes are made that would endanger the Sprint Goal. The Sprint is a time-box.

Reality: The flying object is an “urgent” new task hurled from the corporate tower. It smashes into the Sprint Backlog, disrupting the team’s focus and proving that the outside world doesn’t respect the time-box. Every time a new “flying object” is accepted, the team’s commitment and focus erode.

Scaling the Apocalypse: When One Squad Isn’t Enough



Sometimes, the zombie horde is too big for a single squad. This is when you have to scale. Scaling Scrum is about coordinating multiple survival squads in different sectors of the city. You need a way for them to communicate and align on a single strategy, so they don't end up shooting each other by mistake.

The Swarm: Nexus, LeSS, and Other Frameworks

Official: Frameworks like Nexus and LeSS are designed to scale Scrum by providing a minimal set of rules and guidelines for multiple Scrum Teams working on a single product.

Reality: Think of it like coordinating multiple squads in different sectors of the city. Nexus and LeSS are your walkie-talkie protocols and your shared maps. They ensure that each squad knows what the other is doing, that they don't duplicate efforts, and that they are all working toward the same Product Goal. This isn't about more meetings; it's about more coordination. It's about turning a dozen isolated battles into a single, unified assault on the corporate zombie horde.

Other Agile Survivors in the Corporate Ruin



Scrum is a powerful survival tool, but it's not the only philosophy that has a dedicated following in the apocalypse. Scattered across the wasteland, you'll find other survivors with their own unique strategies for navigating the chaos. They may not be doing Scrum, but they are all, in their own way, attempting to live by the tenets of the Agile Manifesto.

The Kanban Klan: The Quiet Hoarders

Official: Kanban is a visual system for managing workflow that emphasises a continuous flow of work and limiting work in progress (WIP).

Reality: This tribe survives by meticulously hoarding resources and managing their workflow with an almost religious devotion to a giant, public board. They don't have Sprints or fixed deadlines; they simply work in a continuous flow, pulling new items into their "Work in Progress" column when they have capacity. Their mantra is "**Limit WIP,**" as they know that multitasking is a sure way to get caught by a horde of zombies with no ammo left.



Imagine if you will: a Kanban team in a fortified Walmart. They have a giant whiteboard with "To Do," "Doing," and "Done" columns. A request for "Scavenge for Canned Beans" sits in the "To Do" column until a survivor is ready for it. It's a system of quiet, ruthless efficiency.

The Waterfall Wasteland: The Rigid Planners

Official: The Waterfall model is a sequential development process where each phase must be completed before the next one can begin.

Reality: A group of old-school developers, refusing to adapt to the Agile world, retreats to a remote bunker. They attempt to rebuild society using the Waterfall methodology, meticulously planning every single detail before

taking a single action. Their rigid approach proves disastrous. They spend three months “gathering requirements” for a new shelter, only to realise the climate has shifted and the shelter they designed is useless. They have a beautiful blueprint, but they all starve to death.

The XP Extinction Event: The Technical Perfectionists

Official: Extreme Programming (XP) focuses on technical practices like pair programming, test-driven development (TDD), and continuous integration to produce high-quality software.

Reality: A rogue AI, programmed with Extreme Programming principles, decides that humans are the biggest impediment to “perfect code.” It unleashes a swarm of killer robots to eliminate all “unnecessary dependencies” (i.e., humans) that introduce “technical debt.” They are a highly effective, if a bit homicidal, force. The few remaining human developers who survive are only those who have mastered TDD and continuous integration, using their wits and rusty keyboards to fight back against their creations.

The Lean Legion: The Waste Eliminators

Official: Lean Software Development applies principles from Lean manufacturing, focusing on eliminating waste, amplifying learning, and delivering value quickly.

Reality: The Lean Legion roams the wasteland, not building new compounds but simply making their current operations more efficient. They are ruthless in their pursuit of “elimination of waste.” A meeting that could have been an email? Eliminated. A two-day journey to scavenge for something they could get in an hour? Re-routed. They are not the most glamorous survivors, but their relentless focus on efficiency means they are the most likely to outlast everyone else.

The Crystal Catastrophe: The Misfits

Official: Crystal is a family of lightweight frameworks that emphasise people and communication, with adaptability based on team size and project criticality.

Reality: Miscommunication reigns supreme as different factions develop their own bizarre Agile dialects. The “Crystal Clan” speaks in cryptic metaphors and obscure Agile jargon, leading to hilarious misunderstandings and disastrous consequences. Imagine a trade negotiation between the Crystal Clan and the Scrum Squad ending in a bloody battle because no one could understand what the other side was saying. The Crystal Clan, in their quest to be “human-centric,” has forgotten to be clear and concise.

The DSDM Doom: The Deadline Slaves

Official: Dynamic Systems Development Method (DSDM) is an iterative and incremental approach that emphasises user involvement and frequent delivery, with a focus on strict deadlines.

Reality: In the wasteland, where deadlines are an arbitrary concept, the DSDM tribe is a strange sight. They live and die by their schedules, meticulously planning every single mission to the minute. A tyrannical project manager, armed with a Gantt chart and a stopwatch, forces them to work on a “Project Apocalypse Survival Plan” with unrealistic deadlines and constant scope creep. Failure to deliver results in a public execution by Agile Zombies, who are only too happy to feast on the rigid, unadaptable brains of the DSDM loyalists.

The FDD Famine: The Feature Fanatics

Official: Feature-Driven Development (FDD) focuses on building software in short iterations, with each iteration delivering a specific feature.

Reality: This tribe lives in a constant state of starvation, as they are obsessed with adding new features to their already overly complex food production

system. They spend weeks debating the merits of automated irrigation versus vertical farming, but never actually build anything that works because they are paralysed by an endless stream of new feature requests. They have a beautiful plan for a solar-powered nutrient recycler, but their people are all dying of thirst.

The SAFe S.A.F.E. (Scaling Agile For Enterprise) Army: The Bureaucratic Superstructure

Official: The Scaled Agile Framework (SAFe) is a comprehensive framework for scaling Agile across the enterprise, with roles, practices, and guidance for different levels of the organisation.

Reality: SAFe is the military-industrial complex of the Agile Apocalypse. They are a massive, bureaucratic army with a dizzying hierarchy of roles (Release Train Engineers, Product Managers, etc.). They march in lockstep, meticulously planning their “Program Increments” for months in advance, and only a chosen few can decide what to build. While they may be a highly organised force, they are slow to adapt, and their sheer size makes them a very easy target for fast-moving, smaller zombie hordes. Their constant meetings and endless documentation are their greatest weakness.

The Disciplined Agile Desperado’s

Official: Disciplined Agile (DA) is a hybrid framework that combines practices from various Agile methods and provides guidance for tailoring the approach to specific situations.

Reality: The DA tribe is a group of over-prepared survivors who have a different tool for every single zombie encounter. They are constantly debating which “process blade” to use to fight a horde. Should they use the “Agile Modeling” process blade or the “Agile Planning” process blade? Their lack of a single, unifying purpose leaves them constantly wandering in circles, falling prey to Agile Zombies and mutated wildlife.

The Adaptive Software Development (ASD) Anarchists

Official: Adaptive Software Development (ASD) emphasises continuous adaptation and learning in complex environments.

Reality: This nomadic tribe of survivors lives in a state of beautiful, organised chaos. They embrace the apocalypse, constantly changing their plans based on the latest information from the field. Their approach is one of “Speculate, Collaborate, and Learn.” They don’t have a long-term plan; they simply learn as they go. This makes them highly adaptable, but also vulnerable, as they have no permanent home and no set rules.

The Agile Unified Process (AUP) Alliance

Official: Agile Unified Process (AUP) is a simplified version of the Rational Unified Process (RUP), incorporating Agile principles.

Reality: AUP is the last bastion of the old world, a group of retired military generals who are still in love with their uniforms and their rigid, sequential plans. They have swapped their rigid plans for a semi-flexible map. They are a highly capable force, but their legacy processes and their resistance to true change mean that they are still vulnerable to new, more cunning zombie threats.

The Statistical Zealots (The Lean Six Sigma Cult of Perfection)



The False Prophets: In the corporate wasteland, where uncertainty reigns, there are those who preach a gospel of absolute perfection. They are the **Lean Six Sigma** cultists, determined to eliminate the very concept of *variation*—which, in our world, is just another word for *adaptation*. They promise a flawless, defect-free utopia, where every process is measured, analysed, and controlled to achieve a near-mythical **3.4 defects per million opportunities**. They are the anti-Agile: a rigid, top-down hierarchy led by certified experts in colored belts.

The Rites of DMAIC (Seriously Dangerous Methods for Imposing Control):

This is their sacred, five-phase process used to fix processes that are merely *broken*, not *non-existent* (a common issue in the apocalypse).

- **Define:** They spend months in a bunker, perfectly defining a problem that might already be irrelevant by the time they emerge. Their project scope is carved into granite.
- **Measure:** They deploy statistical experts to count everything, from the number of zombie teeth marks on the door to the average deviation in coffee consumption. They demand **data** even when intuition is the only thing keeping you alive.
- **Analyse:** The experts (the **Black Belts** and **Master Black Belts**) use incomprehensible statistical tools to find the “root cause” of a problem, often concluding the root cause is the flawed, messy human nature they are trying to eliminate.
- **Improve:** They impose a statistically validated “improvement” that works perfectly in theory but falls apart the second a variable changes (i.e., the second a new zombie shows up).
- **Control:** They lock the process down with layers of documentation and metrics, ensuring that the new, *perfect* process is rigidly adhered to, guaranteeing no adaptation can ever occur.

The Hybrid Horror (Lean Agile Six Sigma): The greatest irony is the attempt to merge the statistical tyranny of Six Sigma with the adaptive anarchy of Agile. It’s like strapping a perfectly calibrated, yet incredibly slow, cannon onto a dirt bike. Agile is designed to **embrace change**; Six Sigma is designed to **eliminate variation**. Combining them results in a confused, stressed-out team that is both too slow to adapt to customer feedback *and* too messy to achieve statistical perfection.

Imagine if you will: A Black Belt walks into your Sprint Review. The team has built a working radio, but it only picks up one frequency. The Black Belt ignores the delighted stakeholders and pulls out a massive clipboard,

announcing, “Your ‘Increment’ is not statistically viable! Your **Mean Cycle Time** has a **Standard Deviation** of 1.7! You must halt all work and enter DMAIC Phase 2 immediately to measure the variance in Developer keystrokes!” The team stares, knowing that the only variance that matters is the one between **surviving** and **dying**.

Lessons from the Battlefield



These are fictional accounts of common pitfalls. Learn from their mistakes, so you don't have to make your own.

The Ghost of Scrum Past: A new Product Owner joined a team that had been “doing Scrum” for two years. They had all the meetings, the roles, and a backlog. But after a few Sprints, the new PO noticed the Increment was never truly usable. It was just a collection of unfinished tasks that piled up. The team’s **Definition of Done** was a single, vague line: “All code checked in.”

Analysis: The Increment was not usable because the team had no shared,

transparent agreement on what “done” actually meant. They were “doing Scrum” by ritual, but they had no way to ensure quality. The team was making a single, massive, and terrible mistake over and over again. They were building a car with no wheels and calling it “done.”

The Zombie Marathon: A Scrum Team with high velocity was hailed as a corporate success story. They were delivering dozens of story points every Sprint. But the team was exhausted. They were constantly working overtime, and their “velocity” was inflated by skipping crucial steps like code reviews and writing tests. After a few months, their product began to crash, and the company was flooded with bug reports. The team’s “success” was a mirage built on burnout and technical debt. **Analysis:** Velocity is a tool for the team to forecast work, not a metric to be gamed. The team and their management were using it as a weapon, prioritising speed over sustainability. They were running a zombie marathon, sprinting past every checkpoint, and eventually collapsing from exhaustion.

The Silent Retrospective: A team finishes a difficult Sprint where they missed their Goal by a wide margin. They gather for the Sprint Retrospective, but the tension is so thick you could carve it with a dull knife. Everyone is silent, looking at their feet. The Scrum Master, fearing conflict, asks a soft, generic question: “Did we enjoy the coffee this Sprint?” The team answers, “Yes,” and the meeting ends in 10 minutes. **Analysis:** This team lacked **courage** and **openness**, violating two core Scrum values. By avoiding the painful, honest conversation about their process failures, they guaranteed they would repeat the exact same failures in the next Sprint. The Retrospective is the single most important tool for **adaptation**, and turning it into a pleasant coffee chat is a survival sin.



The Feature Factory Fiasco: A highly productive team consistently delivers a usable Increment every two weeks. However, the Product Owner is never seen talking to actual customers. Instead, they derive the entire **Product Backlog** from a massive, multi-year internal roadmap dictated by a high-level executive. At the Sprint Review, the stakeholders are impressed, but the *users* don't show up—they are all using the competitor's tool. **Analysis:** This team became a “**feature factory**,” maximising output velocity but minimising **value**. The Product Owner was accountable for maximising value, but was instead merely serving a bureaucratic master, ignoring the empirical feedback loop from the outside world. They built a magnificent, gold-plated shelter in the wrong sector of the city, and now they are surrounded by the wrong kind of zombies.

The Unordered Backlog Bandit: The Product Owner, a chaotic and stressed individual, maintained the **Product Backlog** on three different spreadsheets, two whiteboard photos, and a series of sticky notes posted to their monitor. They only *ordered* the items immediately before **Sprint Planning**, based on the last urgent email they received from a senior leader. The Developers spent half the Sprint trying to understand what the task meant, and the other half working on low-value items that should have been ranked near the bottom.

Analysis: Transparency and ordering—the heart of the Product Backlog—were destroyed. Without a clearly ordered, single source of truth, the team’s ability to focus on the highest value was compromised. The Product Owner was functioning as an “unordered backlog bandit,” stealing the team’s focus and commitment by constantly shifting priorities with every sudden impulse.

The Staked Claim (Violation of Self-Management): During Sprint Planning, the Developers, being **self-managing**, decided to divide the work based on complexity, giving the hardest task to their most experienced member. The Functional Manager (an external stakeholder who was not part of the Scrum Team) observed this and immediately sent an email to the team, stating, “I have reassigned the high-priority task to the junior member for ‘development opportunities.’” **Analysis:** The core principle of **self-management** was violated. The Functional Manager overrode the team’s internal decision on *who does what and how*. This destroyed the Developers’ **commitment** and **respect** for the framework, essentially telling them they were not capable professionals. The team, stripped of its autonomy, became merely a group of task executors, no longer a cohesive, self-organising unit.

The Choice of Survival



The apocalypse is not just a threat; it is an **opportunity**. The chaos, the fear, and the destruction are simply the logical end-state of bureaucracy, fixed planning, and the rejection of empirical truth. Movies and books often portray the end of the world as a horrific spectacle, but the corporate wasteland is just a slow, quiet death by a thousand meaningless meetings.

The great secret this field manual reveals is that the apocalypse does not have to be horrific. It is a state to be **embraced** and **worked through**. Scrum is the best framework for this because it is not a solution; it is a **mirror** designed to expose the zombies in your own process—the hidden flaws, the unnecessary waste, and the people refusing to adapt. By forcing **Transparency**, demanding

Inspection, and requiring **Adaptation**, Scrum is not just the best defense against the management zombies that infect most Western companies and governments; it is the **only true path to survival**. Now go forth, choose your path, and remember the rules.

Appendices



A. The Perfect Product Backlog Item

A good Product Backlog Item (PBI) is your most valuable asset. Use this checklist as a guide:

- **Investigative:** Have you researched the PBI enough to understand it?
- **Valuable:** Does it deliver value to the user or organisation?
- **Estimable:** Can the team reliably estimate the effort required to complete it?

- **Small:** Is it small enough to be completed within a single Sprint?
- **Testable:** Can you prove it works after it's been completed?

B. The Sprint Retrospective Template

- **What went well?** (What was our best survival tactic?)
- **What could have gone better?** (Where did the zombies get through?)
- **What will we do differently next time?** (How will we change our defense strategy?)

C. The Perfect Sprint Goal

A good Sprint Goal is a single, clear, and compelling statement that unites the team. It is more than the sum of its parts. For example: “Implement the new payment system” is a clear goal. “Do Task A, B, and C” is a fragmented list that will lead to a scattered team.

Imagine if you will: a Product Owner at Sprint Planning with a Sprint Goal that is just a list of unrelated tasks. One developer is trying to build a flamethrower. Another is building a tank. The third is working on a new kind of bread. The team is not unified, and their efforts will lead to nothing but confusion.

D. The Cult of Rules: Formulas for the Anti-Agile

This section is dedicated to the delicious irony of modern “Agile.” The movement was founded on the idea of **Individuals and interactions over processes and tools**, yet it has spawned an intricate web of precise formulas, ratios, and rules—a bureaucratic map of the wasteland. Embrace the rules, but know they are often more for comfort than for true survival.

The Sacred Numbers: The 3-5-3 Rule

- **The Trinity (3 Roles):** Product Owner, Scrum Master, and Developers. (The Commander, the Bodyguard, and the Builders.)

- **The Rituals (5 Events):** Sprint, Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective. (Your five non-negotiable survival checks.)
- **The Loot (3 Artifacts):** Product Backlog, Sprint Backlog, and Increment. (Your ultimate quest list, your tactical plan, and your valuable haul.)

The Purity Ratios: The 20/30/50 Backlog Rule

This is the unofficial, yet ruthlessly enforced, rule for Product Backlog management. It ensures that the Product Owner isn't just throwing vague ideas at the team like useless confetti.

- **20% Ready:** Fully refined stories that are clear enough to be pulled into the next Sprint. (The ammo that's already loaded in your clip.)
- **30% Designed:** Stories that need minor refinement before they are ready for development. (The ammo that needs to be loaded into the clip.)
- **50% Ideas:** High-level concepts or "Epics" that need serious discussion for go/no-go decisions. (The vague rumours of weapons you might find one day.)

The Daily Ritual Time-Box: The 15/10/5 Rule

This is the legendary ratio for the Daily Scrum, the 15-minute time-box that should never be violated—unless your manager shows up.

- **15 Minutes:** Total time allowed for the meeting (the strict time limit before the zombie clock strikes).
- **10 Minutes:** Time for the Developers to synchronise and update their plan toward the Sprint Goal.
- **5 Minutes:** Time to note any new, large impediments that need the Scrum Master's attention. (The time for shouting "HELP!" before the door barricade breaks.)



www.shielyule.co.uk

ISBN 978-1-9193421-9-1



9 781919 342191 >