Agile, Yucca Moths and Agent-Smith

How Nature and AI Teach Us Perfect Practice.

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

MANIFESTO FOR AGILE SOFTWARE DEVELOPMENT [1]

Those values look simple on paper, but mastering perfect practice, as Lombardi called it—takes more than ceremonies.

Practice does not make perfect. Only perfect practice makes perfect.
— Vince Lombardi

To see perfect practice in the wild, let's abstract from the official terminology and allow me to introduce the Yucca Moth. Over time, the yucca moth (*Tegeticula yuccasella*) evolved not as a general pollinator, but as a specialist— uniquely adapted to one plant: the yucca. Unlike other insects that stumble upon pollination by chance, the yucca moth performs it deliberately. It gathers pollen from one flower, flies to another, and actively places the pollen onto the stigma to ensure fertilization. Only then does it lay its eggs inside the flower's ovary, where the larvae will feed on a portion of the developing seeds.

FIGURE 1. YUCCA MOTH [2]

This tightly woven relationship, known as obligate mutualism, means neither the moth nor the plant can survive without the other. If the plant doesn't get pollinated, it won't produce seeds. If the moth lays eggs without pollinating, the flower will abort and both will lose. As a result, both species evolved to adapt together, balancing their survival through feedback, timing, and just-enough interaction. The moth doesn't exploit; it contributes just enough to keep the system healthy—thriving not through control, but through cooperation and iteration over generations.

In a changing ecosystem, this kind of co-evolved strategy offers resilience. It's not the most aggressive that survives, but the one that learns, adapts, and stays connected to its environment.

The yucca moth is the embodiment of perfect practice — refined over centuries through evolution, iteration, and feedback.

The moth thrives by pairing deliberate action with constant feedback—exactly where many Agile transformations stumble. People try to achieve this perfect practice in their work as they strive for perfection and they all believe they can do this by becoming *agile,* embracing the *agile methodology* and living the *agile lifestyle.*

It sounds easy, doesn't it. Small steps, small improvements, big rewards. However, most people do it wrong. A recent analysis by ValueX2 [3] found that nearly 75% of agile transformations fail to deliver meaningful business results. Why? Because many organizations are "buying ceremonies, not transformations". Let me explain this using an example. During the World War 2, allied forces-built airstrips and military bases on Melanesian islands, particularly places like Vanuatu or Papua New Guinea. Now these bases regularly receive cargo, such as supplies, food, and equipment. To these isolated islanders, these deliveries seemed magical or divine, a gift from the sky. After the war, the Allies left and so did the cargo. In response, some islanders built imitation airstrips out of sticks and stones. They created wooden control towers, wore makeshift uniforms, and mimicked the movements of soldiers with the hope that planes would return. Simply, they logically found a solution and implemented it; they went through a hell of an effort to make it work; but as you can assume, it brought them no success. It's tempting to see their effort as hopeless—yet many modern teams make the same mistake in digital form. They introduce the agile methodology incorrectly which gives them a false sense of progress, false sense of security.

FIGURE 2. MELANESIAN ISLANDERS [4]

The first principle is that you must not fool yourself-and you are the easiest person to fool.

— Richard Feynman, the man who first introduced the concept of the cargo cult

Feynman's warning applies to software too: follow the motions of Agile without evidence and you fool yourself first.

The problem goes deeper; according to several surveys [5], over 40% of Agile failures stem from cultural resistance and a lack of proper training. Teams lack the support, guidance, or time to evolve habits. Without reinforcement, retrospectives become routine, feedback loops are skipped, and continuous improvement stalls.

Now since we saw that implementing agile is so unexpectedly difficult and so many people fail; you could wonder if it's worth it? The decisive answer is yes, think back to the Yucca Moth who achieved perfect practice, and this is exactly what your competitors are trying to achieve. Over the period of 2016-2019 [6] the agile practice grew by 81%. The year 2020, remembered as the beginning of the Covid pandemic and rise of remote work, had a staggering increase of 43% alone. In the 2021-2022 gap, sectors not in IT have had a 46% adoption rise – June 10th, Villeroy stated that *European Central Bank will be as "agile" as needed in line with data flow* [7], which showed huge support for agile methodology. Then in 2023 agile had 68% of different business units implementing it.

Teams push on because, when done right, the payoff is real—evident in the numbers below.

64% of the companies witnessed increased capability to manage changing priorities efficiently.

47% of organizations felt that there was improved communication and collaboration between IT and business teams by implementing Agile.

42% of respondents who incorporated agile practices found a substantial increase in software product quality.

Implementing Agile saw companies enjoying a 47% increase in team productivity with efficient outcomes of delivery.

Adopting an agile enterprise architecture framework offered a 41% improvement in the predictability of software product deployment to companies.

The agile approach offered 40% improved project transparency across different departments of the surveyed organizations.

FROM EXPLORING THE LATEST TRENDS IN AGILE STATISTICS FOR 2024 [6] BY RAVIKUMAR PATEL

Just like the moth doesn't survive by chance, great teams don't thrive by accident. They succeed by being intentional—through feedback, adaptation, and shared purpose. Therefore, what can we do to implement the agile methodology effectively? To not build imitations of airstrips, to not mimic movements of the soldiers, to not be like the Melanesian islanders. While keeping in mind that 40% of human behavior is habitual [8], we can conclude that we do not like friction or change.

If 40 % of behavior is automatic, we need a helper that fits the habit loop rather than fights it. That fix is Agent Smith—a lightweight AI assistant bot with Slack integration on the roadmap that drops the busywork and lets you focus on building. In early trials, AI assistants like Smith have cut routine developer tasks by 30 – 50 percent [9], and Smith applies the same edge to team hygiene:

- Auto-create and tidy Jira tickets—no more chasing down missing fields or stale stories.
- Instant workload digests—Smith summarizes what's on everyone's plate before the stand-up even starts.
- Sprint housekeeping—from backlog grooming to end-of-sprint snapshots, the bot keeps your cadence clean.

In short, it smooths out the grind of checking old work, keeps history organized, and—most importantly—wins back hours you'd otherwise burn on repetitive admin. Today, Smith runs on a Chainlit server for fast iteration and real-world use. Below is a live prompt run through that UI, so you can see exactly how it pulls every ticket tagged "Agent Smith" in seconds.
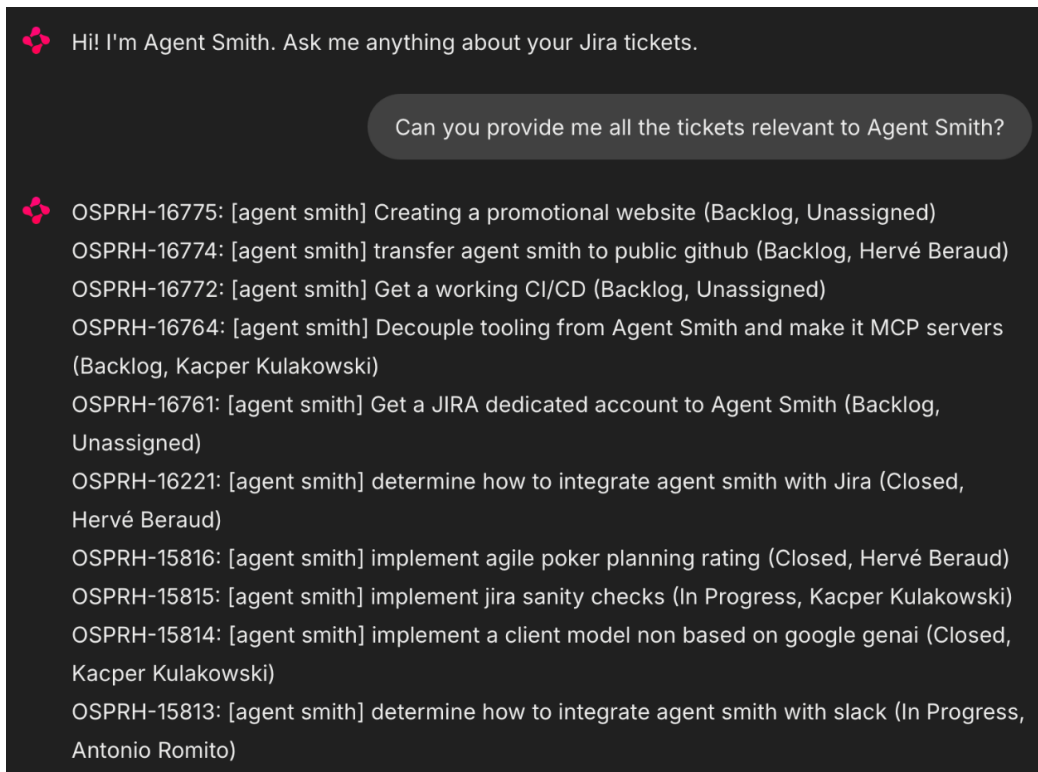

FIGURE 3. SAMPLE QUERY PASSED TO AGENT SMITH

Curious to see what Agent Smith can do? Try it out and help us shape where it goes next.

Agent Smith is still early in development, but it's already handling useful tasks like Jira ticket creation, cleanup, and digest summaries through a simple Chainlit web interface. AI tools like this have already shown that they can save dev teams serious time, now we're building one that works for you.

Clone the repo:

git clone https://gitlab.cee.redhat.com/eng/openstack/team/pidone/agent-smith

Run it locally:

Install dependencies and launch with Chainlit run chainlit_app.py. You'll get a browser UI where you can start chatting with Agent Smith.

Explore what's working now:

Ask it to find tickets, clean up old tasks, or summarize what's in progress. See how it handles the basics—and where it can go next.

Agent Smith is actively maintained by the Pidone group (rhos-dfg-pidone). If you have questions, ideas, or want to get involved, feel free to reach out to @hberaud on Slack. By contributing now, you'll help shape the future of Agent Smith—and move us all a step closer to achieving perfect practice, the only real path to meaningful progress.

References

[1] *Manifesto for Agile software development* https://agilemanifesto.org/.

[2] The Editors of Encyclopaedia Britannica (1998) *Yucca moth | Pollination, Coevolution & Mutualism*. https://www.britannica.com/animal/yucca-moth.

[3] ValueX (2025) *The uncomfortable truth about agile Transformation Failure Rates*. https://www.linkedin.com/pulse/uncomfortable-truth-agile-transformation-failure-rates-valuex2-3sjof

[4] Skogsberg, P. (2023) 'Cargo cults are everywhere: the hidden rituals that shape our world,' *Sképsis*, 29 September. https://philipskogsberg.substack.com/p/cargo-cults-are-everywhere-the-hidden.

[5] Digital.ai (2025) *The 17th State of Agile Report*. https://2288549.fs1.hubspotusercontent-na1.net/hubfs/2288549/RE-SA-17th-Annual-State-Of-Agile-Report.pdf.

[6] Ravikumar-Patel (2024) *42+ Most important agile statistics for 2024*. https://radixweb.com/blog/agile-statistics.

[7] Leigh Thomas (2025) *ECB will be as 'agile' as needed in line with data flow- Villeroy*. Edited by Dominique Vidalon. https://www.reuters.com/markets/europe/ecb-will-be-agile-needed-line-with-data-flow-villeroy-2025-06-10/.

[8] Palmer, C. (2021). Harnessing the power of habits. [online] Apa.org. Available at: https://www.apa.org/monitor/2020/11/career-lab-habits.

[9] JAIN, Y. (2025). AI is dramatically changing the landscape of software engineering. In the past few years, tools like OpenAI's ChatGPT and GitHub Copilot have burst onto the scene, proving that AI in software development isn't just hype — it's already here, reshaping the daily work of developers. [online] Linkedin.com. Available at: https://www.linkedin.com/pulse/how-ai-transforming-software-engineering-2025-tools-trends-yash-jain-cbb8f .