

In the dark midwinter of early 2001, a group of like-minded software professionals gathered at a remote ski lodge in Utah. It's the sort of setting which fires the imagination of Stephen King, but in this story our intrepid heroes come out on top. Over the course of a weekend, they drafted a document that changed the way we talk about software development. How did it happen? And what relevance does the Agile Manifesto still have today? To find out, DOC checks into...

THE LODGE

WORDS: Bruce Johnston ILLUSTRATION: billmconkey.co.uk





One man who survived the expedition to Snowbird was James Grenning, Agile Manifesto co-author and embedded software expert. DOC caught up with James to hear the inside story.

“Hey, I’ll be with you in a second. I just need to get some coffee – Monday morning came up on me pretty fast.” It’s 9am and I’m chatting with James Grenning on Zoom. He fixes himself a brew and tells me about the weekend he’s just spent in Chicago with his grown-up kids at the ball game. I get a little explainer on the geographical divide that separates the Cubs fans from their sworn enemies, the White Sox, and I’m reminded that James is good at explaining things – something I’d realised when I first met him after his keynote talk at CoDe-Conf 2017 in Copenhagen.

I’ve arranged the interview to get some insights into what happened at Snowbird back in February 2001 – the ski resort we now know as the birthplace of the Agile Manifesto. James was there and played his part, but over the years he’s developed a well-rehearsed line that makes light of his own input. Indeed, it’s the first thing he says once the conversation turns to the Manifesto: “I really wanted to go skiing. I’d been there before and I’ve told that story too many times to change it now. My expectations were to go and hang out with a bunch of guys who I knew and respected.”

I know James well enough to realise that he’ll make no grand claims for himself. Even if he’d authored the whole document on his own, he’d be too modest to admit it. What I’m hoping for is that James can shed some light on the true story behind the Agile Manifesto – what actually happened in Snowbird over the course of that weekend? My initial research turned up a small historical piece written shortly after the event by Jim Highsmith, another co-author, but it left me with as many questions as answers. I’m left wondering about the seeming ease with which the



IMAGE: The actual manifesto created at the event

manifesto came about; I struggle to organise a video call with more than two people in it. How exactly do you get 17 leading software engineers to a ski lodge in Utah and have them reach unanimous agreement on a document that would permanently change the way we approach software development? And over the course of a single weekend where they spent half the time skiing? I mean, come on!

I'm clearly missing something, but what? I'm intrigued by Jim Highsmith's description of the group as "organisational anarchists", so I start off by asking James the simplest question of all: how did they succeed in getting everyone together?

"It was probably because of the people involved. Bob [Martin] had good connections, as did Alistair [Cockburn], and they were the guys who were really trying to make it happen. So they had a good reach. They knew a lot of people in the industry who they thought were into the same kind of things, or maybe against the same kind of things."

I suppose I can understand how this might have been a big draw for everyone, with FOMO and all that. But they did more than just meet up and ski – how did all 17 of them

succeed in reaching an agreement? Was it a case of enjoying the slopes and scribbling down some manifesto stuff when they found some time?

"Yeah, let's write down our four bullet points and go have dinner. No, it wasn't quite that easy!" James explains. "It didn't come quite so easily as you'd think, but it did distill out. We had three days of those meetings... maybe more like half-days so we could go out and play. But unlike many debates we find in our culture today, we were looking for areas of agreement, not for cementing disagreements." I find myself nodding along. When put like this it seems obvious: areas of agreement, cooperation, collaboration, people – so many of the elements that informed the manifesto itself right there in the process. Of course. The medium is the message.

Extreme Programming

I want to know more about where the various personalities were coming from and we start to dive a little deeper. There had to be foundations, groundwork, or some shared platform to work on? It surely couldn't have all been spun up from scratch that weekend. From Jim Highsmith's account I read about a similar meetup at Red River Lodge in Oregon the previous year. Perhaps the preliminary work was done there?

“Let’s write down our four bullet points and go have dinner. No, it wasn’t quite that easy! It didn’t come quite so easily as you’d think, but it did distill out. We had three days of meetings... maybe more like half-days so we could go out and play”

James wasn't at Red River, but he's not overly convinced that too much came out of it. What he does want to draw my attention to is Extreme Programming.

"I like to say there wouldn't be an Agile without Extreme Programming. If you look through the names that signed the manifesto, a lot of the co-authors came from the Extreme Programming contingent. By calling it that it got everyone's attention, and when you read about it you'd say, 'That couldn't possibly work.' But then you try it and you realise that it does work.

"The Agile Manifesto had to soften that a little bit because the term 'extreme' scared people. It was always kind of a joke – do we need elbow pads and helmets for this? What's so extreme here? So that word got softened, but over the

last five years or so I've been trying to bring it back. I'll use 'Agile' but I'll switch to Extreme Programming as soon as I can because that's really the history of it. The creative problem-solving that Kent [Beck] and Ward [Cunningham] came up with amazes me to this day. They were not satisfied with the status quo of adding bugs to code and taking them out later."

Method

Extreme Programming – now we're really getting into it. But I also want to know if they worked together, either in a big open session or in small groups according to expertise? James disappears to go and look for something.

"One of the things we did as Extreme Programmers was to always have some of these." He comes back into shot a

"I really consider myself extremely lucky. It was awesome to be involved in it"

few moments later waving a small pile of notecards. “For whatever reason, we tossed out some notecards and looked for things we could agree on, or believe in, or whatever. In the 80s, at Teradyne, we studied problem solving and brainstorming techniques as part of Total Quality Management (TQM). One of the problems with the verbal brainstorm is the first idea will always get shot down by someone. Then either people stop talking or it just isn’t productive. But if we write our ideas down, everyone gets to express themselves without interruption and we can find out what we have in common. So there was that, a lot of whiteboard stuff, lots of back and forth... my memory of the event is not perfect.

“A friend told me several years after Agile became a big deal in the industry that I mentioned to him after the meeting I had offered the notecards. Tossing those cards out might’ve influenced things a bit, but this was a good group of strong-minded people – trying to find what we could agree on was definitely where we were headed.”

Terminology

One thing I’m really struck by is the importance of terminology running through everything: “Extreme” Programming, an “Agile” Manifesto. Why agile? And how different might things have been had it been called something else? It’s a question that must’ve been put to the

various co-authors thousands of times over the years and I’m no different. I ask James who came up with it. “I think the first person I heard say the words, ‘We need a manifesto’ was Martin Fowler and that sounded kind of communist to me!” I can’t help but smile at the collision between British and American English. A manifesto! Well done, Martin. But what about the word “agile”?

“Well, I don’t quite know, but one thing I can tell you is that ‘lightweight’ was dismissed really early. Ward said, ‘The first thing we need to do is come up with a better name because who wants to be known as a lightweight?’ There were a few words already out there, like ‘adaptive’, which I think would have also been great. I was sitting next to Kent and we’d settled on one of the other names, which I don’t remember now. But in the end we settled on a fine name, got our four bullet points, and we thought no one would care.”

But people did care. And they started caring pretty quickly. I ask James about when he started to realise that what they had done at Snowbird was a big deal.

Influence

“Agile was appearing on the names of all these conferences pretty much right away. Around 2001, Bob started the XP Universe conference in the Chicago area, then Alistair started another conference that was more on the business



“One thing I can tell you is that ‘lightweight’ was dismissed really early. Ward Cunningham said, ‘The first thing we need to do is come up with a better name – who wants to be known as a lightweight?’”

side. They kind of competed for a while before a grand unification around 2005.”

But what about today? Seventeen years on from Snowbird, what is the legacy of the Agile Manifesto? Is it still relevant for software in 2018?

“Unfortunately yes, because people read it in a really funny way. The part that was deemphasised is the only thing they can see. ‘People over processes and tools’ – so no processes and no tools. ‘Working software over comprehensive documentation’ – that’s taken to mean no documentation. The continual misunderstanding of it keeps it relevant.”

I’m slightly surprised and a little disheartened by James’ answer. The Agile Manifesto might be a well known text, but it seems that some of its core messages still aren’t getting through to the people who need to hear them. Too many engineers are spending their lives hating something called Agile that isn’t really agile at all. So, where is the real legacy of the manifesto? Is it preserved in DevOps?

“Yeah, I think it’s all very related. I’m looking at the manifesto now... we want to deliver something valuable to customers regularly. You could almost say, ‘Why come up with DevOps at all? It could easily have been End-to-End Agile’ or something like that.”

We’ve been talking for nearly an hour and I don’t want to keep James for much longer. His answers have really helped me understand how this important piece of software history came about, but there is one last thing I want to know.

I ask James what sort of effect being a co-author has had on his life and whether he feels a sense of pride at having been involved.

“I really consider myself extremely lucky. I mean, it was awesome to be involved in it, but my part was small. To have taken those great ideas that other people came up with and adapted them to the embedded world gives me a sense of pride. It would have taken a long time before somebody started to do this, so I think I’ve made some very important contributions there. And it’s been a lot of fun, too – it’s made for an interesting and fun life.”

With that I thank James for his contribution and we hang up the call. I take a few moments to digest the conversation. “I really wanted to go skiing,” he said. Maybe there’s more truth to that than I initially realised. After all, all work and no ski makes James a dull boy.

■ You can find out more about what James does at wingman-sw.com/about