Scrumfall: The fall of Scrum

The Origin of Scrum

In his book **Scrum: The Art of Doing Twice the Work in Half the Time,** Jeff Sutherland introduce us to the idea and origins of Scrum. He brings us to a time when he was serving in the airforce during the Vietnam war as a pilot. They had a clear directive OODA, **observe**, **orient**, **decide** and **act**. It is a four-step approach to decision-making that focuses on filtering available information, putting it in context and quickly making the most appropriate decision while also understanding that changes can be made as more data becomes available. The only way that this could be done was by **empowering** individuals to follow these core values, later known as the pillars of Scrum: Transparency, Inspection and Adaptation.

Drawing from this experience, and inspired by the development of the first Roomba, he put the first Scrum team in action. How was this team similar to the vacuum cleaner we use today? Well, the Roomba is blind. It starts in a direction opposite of its base a drives until it hits an impediment it cannot cross. Then it inspects by slightly changing direction and trying to slowly move that way; if it is clear it starts to increase velocity until it hits an obstacle again. But, it also records where, or how far from the first obstacle it traveled so the next time it knows where the two barriers to its path are relative to the base; it slows down before touching them the next time around, to again reorient and drive in a different direction.

Ok, this is Scrum, this is what we have been doing all along - you might react. The answer, for some (or rather most) of us sadly, is no.

Paradox: What have we be doing?

The tendency of nature is to take its most stable state; the tendency of men is to take their most secure one. This is most evident in the anti-pattern called "Information Silo", that is completely opposite of empowered individuals, the driving force of Scrum. So what happens is that these hierarchical structures transforms the Scrum flow into Scrum fall, a Water-Scrum-Fall scandal.

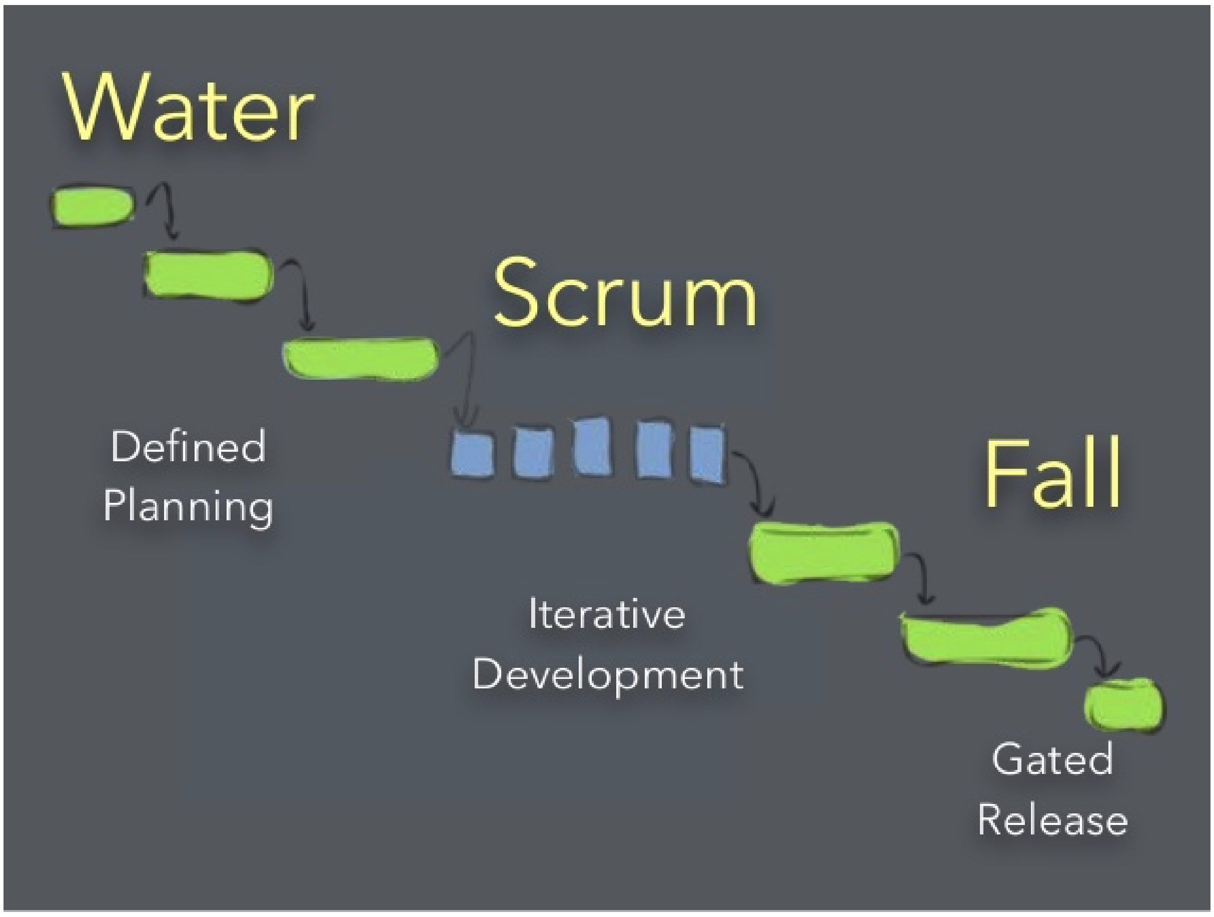
What is it and how did we come to it?

I started my management career in Mechanical Engineering where we did standard waterfall planing, Gantt charts, budgets allocations, and whatnot, creating beautiful charts and graphs that were never true but as my CEO said it smiling, “Colors for Directors”.

One day my line manager came to me and said, where is your sprint backlog? You see, he was inspired by the IT department of the company doing Scrum and delivering twice the value for half the time. I had no idea what he is talking about. So he enrolled me in a Scrum course where I got my **Professional** Scrum Master certificate - I had still no clue what Scrum is, but … oh well, I was a Scrum “**Master”** now.

The birth of the Water - Scrum - Fall framework wrapper

So the natural next step was to bring the rest of the team up to speed, to implement what I’ve learned about iterations and reviews, changing direction to bring most value, to inspect and adapt the process. The problem was, we were already found in a mid-project phase of implementation, so the scope, the delivery plan, budget and resources, etc. were already defined. What we started doing was Scrum inside a Waterfall project, for the “development” phase. It looks something like this:



So we decided on 3-week iterations, finishing with a demo and collecting feedback from SMEs, acting on that feedback and improving the design until we reached a satisfactory state of the product that we pitched to the board of innovations, awaiting approval (The Gate).

But this was Mechanical Engineering, and we were developing Hardware, so it was possible working like this. We had to plan upfront, to have a conceptual design that can be evaluated by a third vendor that will need to manufacture it to give an estimated cost, manufacturing and delivery time; as well as other third party equipment that had to be integrated with the main construction.

Using this method, we developed a “test” machine that had to be put in operation to measure certain signals that will dictate the looks and performances of the final design. We developed a prototype of the final design as well, that could have been easily modified and adapted after learning a thing or two from the “test” unit. This again is in broad sense the spirit of Agile, put in hardware.

Where things went haywire

One interesting thing to mention here is the reality of the “Second-system effect”, a term coined by Fred Brooks, saying that the there is a tendency of small, elegant and successful systems to be succeeded by over-engineered, bloated systems, due to inflated expectations and overconfidence.

Jeff Sutherland says throughout his book that it is not the people’s fault, rather its the system’s fault, or us allowing the systems to guide us, and not the other way around. This has been proven right again and again. Seeing the scrum-fall process (and it is a process now, not a framework like Scrum) works for a specific scenario doesn’t mean it is good. I’ve seen this process camouflaged in “pure Scrum” in an IT department when working for a digital marketing company.

This is how it happened:

- The “Top Management” people had an idea that was put into “Requirements”.
- There was an R&D team that developed that idea to its finest details, providing mockups, wireframes and solution proposals.
- They presented this idea to the IT department representatives (Solution Architects and PMs).
- After some back and forth, the idea was refined and accepted, it went into “Design” phase.

- The UI/UX team produced detailed designs while the PM team crafted the stories:
    - The designs were beautifications on the mockups.
    - The stories were ‘almost’ copy-pastes of the “Design Documents” produced by R&D.
    - Everything needed to get a “sign-off” by someone up the ladder.
- The development team, working in Scrum, followed a step-by-step tasks and delivered the requirements in successive order.
- There wasn’t a Review or Demo session. It was not needed, and Stakeholders were too busy producing another idea and design documents.
- The PM team produced Deployment Reports informing the users that there is a new feature, where it can be found and how it can be used. It didn’t matter if they wanted it, needed it, liked it or had some concerns.
    - Well, concerns were documented, placed into the backlog and rotted there for few years before they were deemed obsolete and removed completely.

When I first saw this, you can safely guess that I was in shock. This is not Scrum, this is not even Agile, this is “pure Waterfall” or an unsuccessful try at Scrum-fall. It was brutal and completely demotivating to the development team, but when the concern was raised the reply was “This is how it has been always done” of course. When even a small deviation from the plan, or a try at moving closer to the actual Scrum were made, all hell broke loose and I even received a comment “You are not working as Agile as before”. Oh well…

What we need to be wary of!

Scrummerfall. n. The practice of combining Scrum and Waterfall so as to ensure failure at a much faster rate than you had with Waterfall alone.

Sometimes failing fast is a good thing, it yields good results, it gives an insight what and how much we are wrong. But we need to track its signals, measure it, analyse it and act on it (again the Observe, Orient, Decide and Act. Only when we are empowered to voice our concern and to correct it when it is clearly destructive we can make a change for the better.

It is damaging when are blind, we are not seeing that we are failing, because it is not transparent. And it is not transparent because the hierarchy of influence hides it in an ever increasing information silos, securing and strengthening people’s positions and importance in the company.

This is the deadliest poison to the agile mindset. We need to create a system, a structure, a process that will work towards brining value for the customer, liberated of gates and sign-offs, change requests and approvals that will only bring value to the disruptors, the information siloists and the executioners of agile, the fall of Scrum.