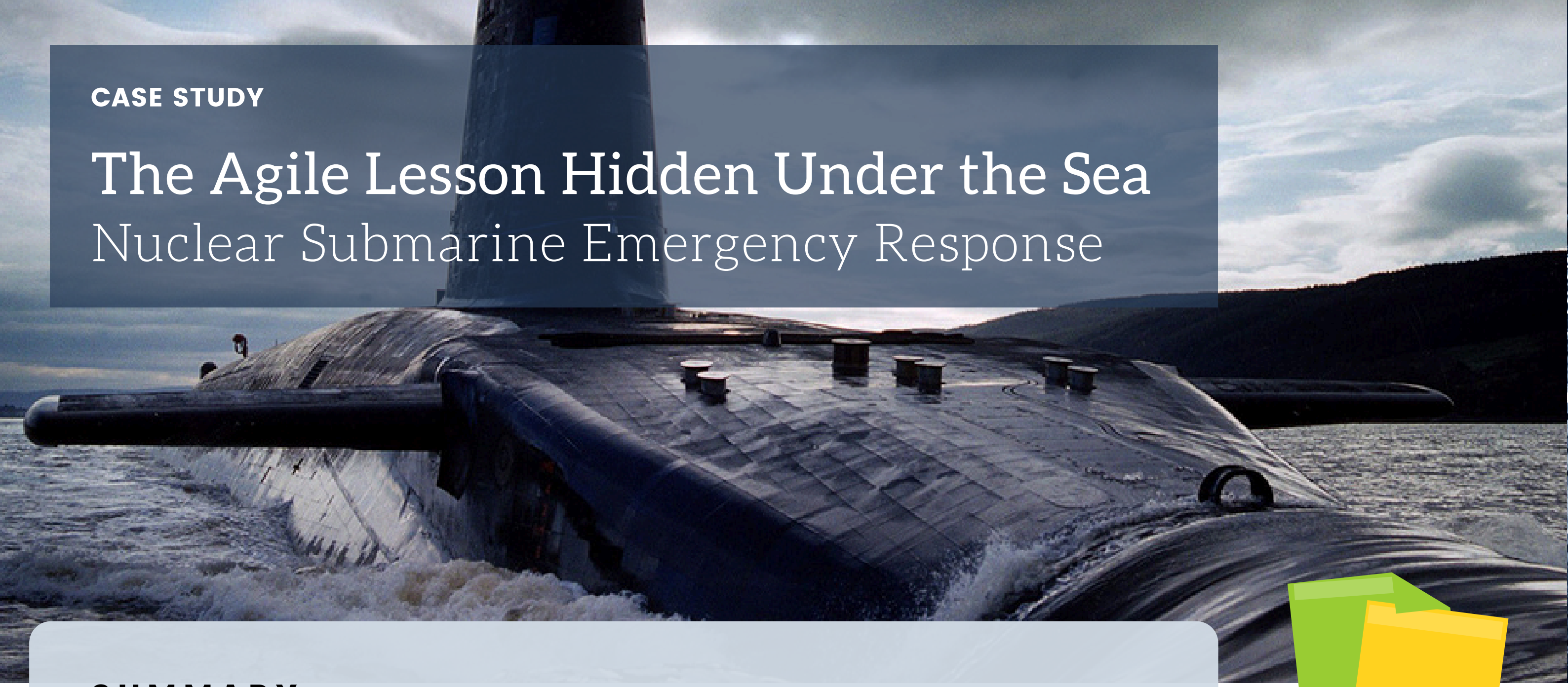
CASE STUDY

The Agile Lesson Hidden Under the Sea

Nuclear Submarine Emergency Response



SUMMARY

You're under pressure to do more with less, timescales are slipping immeasurably, you're late and over budget and the customer perception is that your delivery team is too slow and there's no light at end of the tunnel. The app is never going to ship...and you're holding the rudder!

This situation is all too similar. At Radical we see this scenario play out time and time again, and whilst there could be a gazillion potential reasons why you're in the mess you're in, in our experience the root cause is always the same.

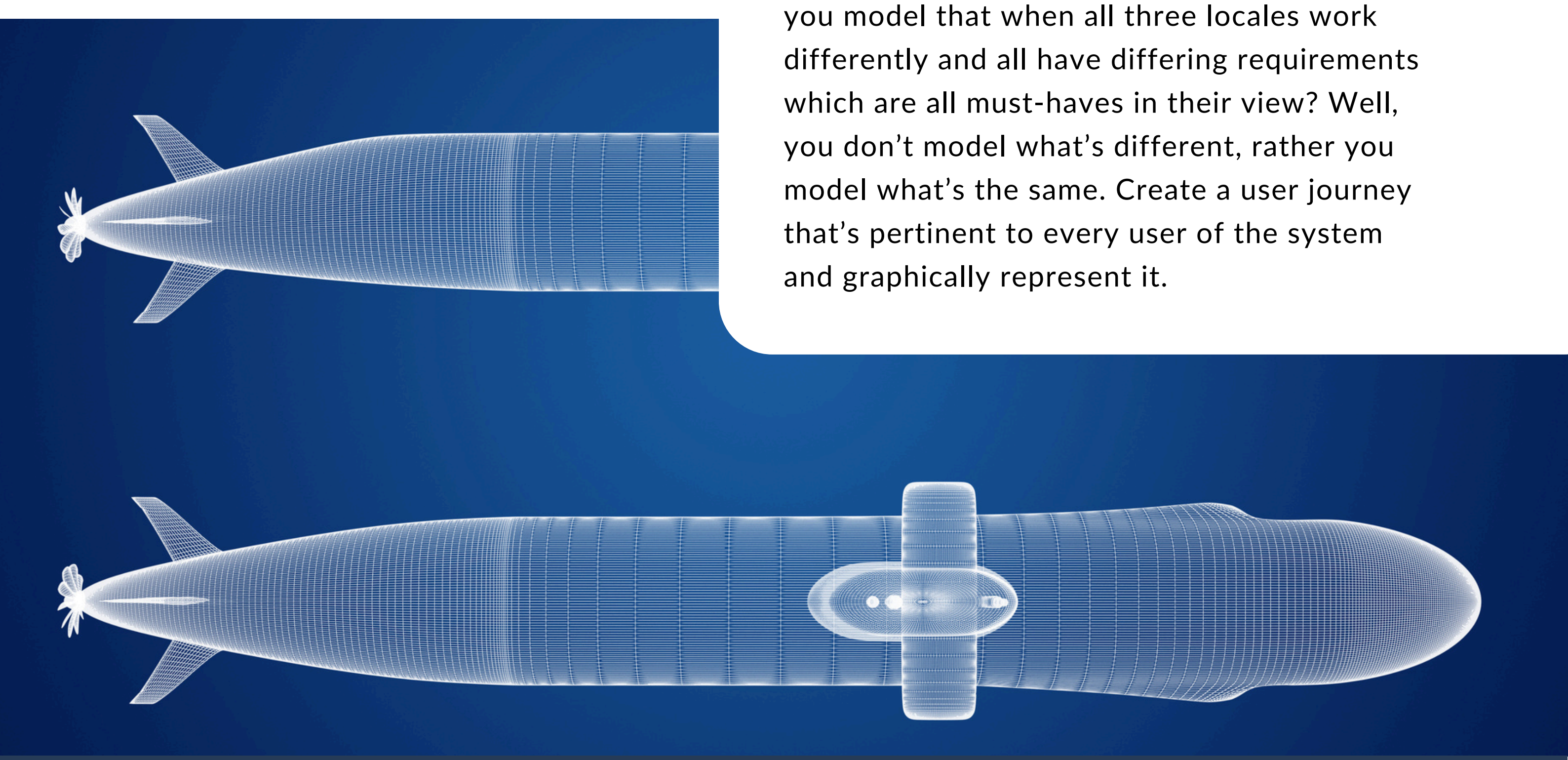
If you ask customers what they want, they'll give you a huge smorgasbord list of must-have features which you've got no way of delivering within your forecast time and budget. Instead, help customers discover what they really need and make that your MVP. This will expedite your delivery and delight your customer and stakeholders. But how do you do that easily with conflicting customer requirements?

Following the Chernobyl disaster each country with a nuclear capability (either power stations or defence related) understood the very real need for a comprehensive system to manage a nuclear emergency. Radical IT were asked to support the build of the Defence system to safeguard the U.K.'s nuclear submarine fleet and the transportation of nuclear weaponry and materials to and from nuclear submarines in port.

There are essentially three locales where our nuclear submarine fleet operates from, namely Faslane, Portsmouth and Plymouth. The online system we built supports all three locales within a single system. When we landed on the project, we were met by the inevitable huge JIRA backlog which was never going to get completed within the short deadline we had.

For example, there are many forms (situation reports, countermeasures, COBR output etc.) involved in the military management of a nuclear incident. Given the existing team's velocity, we estimated that just creating the forms alone, would take circa two years. And that's without any of the work required to model plumes of radioactivity based on weaponry type, trajectory, as well as health physics which calculate how long recovery personnel can remain in the danger zone. As Nick Hine, the Second Sea Lord at the time, said to us...' this IS rocket science, and many lives are at stake'. A sobering reality which called for a total rethink.

So, the first thing we did was help the customer discover what they really needed and create an MVP which could be delivered within an acceptable timeframe. But how do you model that when all three locales work differently and all have differing requirements which are all must-haves in their view? Well, you don't model what's different, rather you model what's the same. Create a user journey that's pertinent to every user of the system and graphically represent it.



Ignore the 'what if' brigade, for now. These are a well-meaning, although sometimes not very constructive, group of people who can always point to differences in their own processes and procedures and clearly explain why whatever you've created 'its great but...it won't work for us'. Capture all their edge cases and what if's but map out the one true happy path for now. We'll come back to their edge cases later.

Ask the following questions to sanity check your storyboard: Does it make sense, what could be improved, what isn't needed and is anything missing?

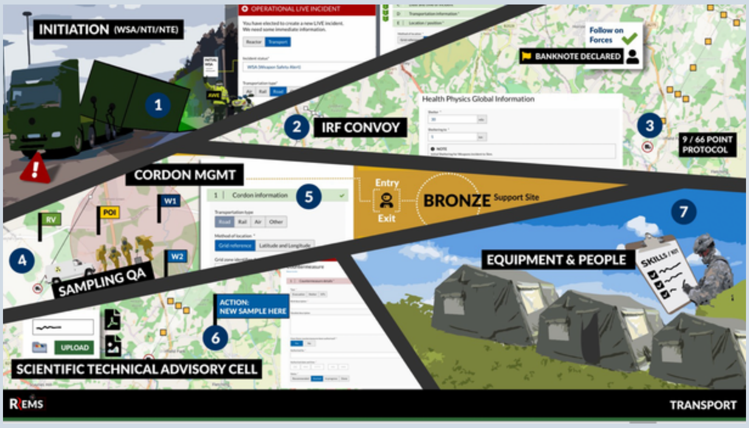
At Radical we use a simple user journey storyboard to model and map out the happy path journey to share with users and stakeholders.

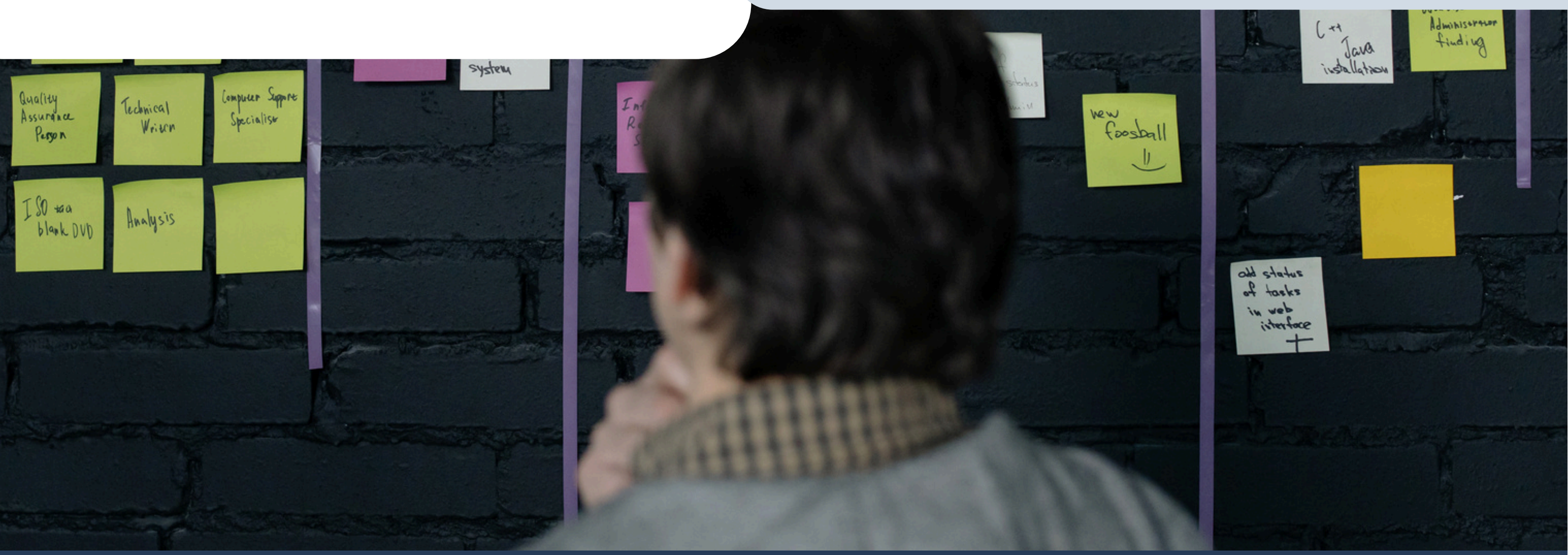
Figure 1 is an example of the mapped process for the recovery of Dangerous Nuclear Material (DNM) or Nuclear Weaponry during transportation to dockside.

Regardless of local i.e. Faslane, Portsmouth of Plymouth this graphic represents the common agreed process. For example, all three locales initiate an incident, deploy an Initial Response Force (IRF), execute cordon management around a site, take samples and monitor radioactivity etc.

You can download free copies of our templates from radicalit.co.uk.

Figure 1





Once this is mapped out visually, share with users and stakeholders to explain that this is your initial MVP and get feedback.

Explain that MVPs are key because they help you build and launch faster, get early feedback, validate any assumptions you made during modelling phase – are we on track, engage stakeholders early in the build process to get buy in and to help mitigate any potential risks. You need to get your app or system in the hands of users and stakeholders to get this invaluable feedback. And that's exactly what an MVP does for you.

The agreed, mapped out single process now becomes your north star. All your teams' efforts should be focused on this delivery and demonstrating the end-to-end process to stakeholders.

From this process it's now much easier to create a Sprint Schedule covering 90 days i.e. six times two-week sprints.

A Sprint Schedule is a plan-on-a-page of the work required to deliver the MVP within six two-week sprints.

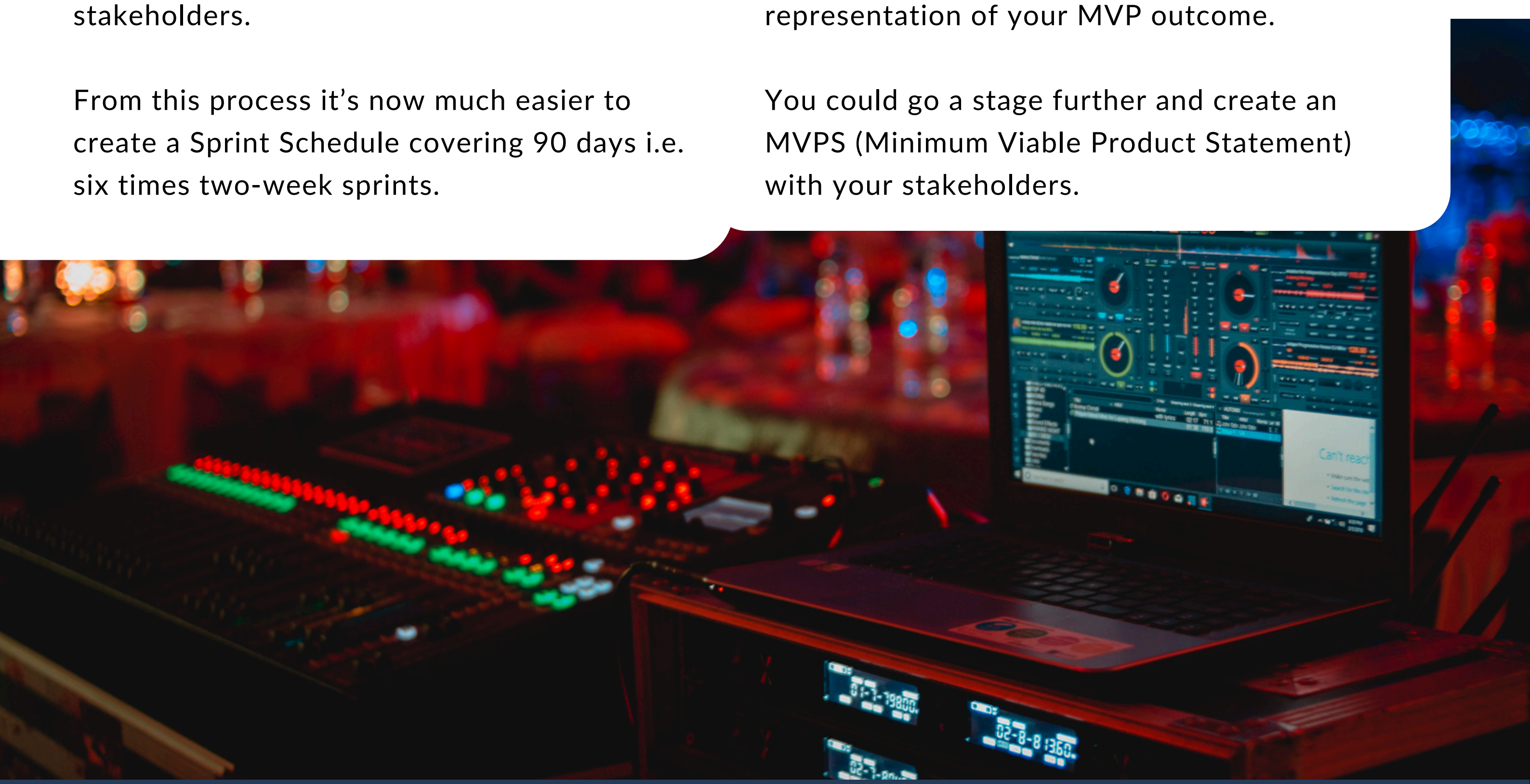
Timebox your teams MVP effort to ninety days.

Why ninety days? Well, this is just about enough time to create an MVP of any value whilst keeping your stakeholders engaged and onboard. With your team, brainstorm all the big-ticket items that would need to be there to deliver the MVP.

And no more. At Radical we call them buckets of work as they may contain more than one user story. Focus here is key.

This is why it's important to have a visual representation of your MVP outcome.

You could go a stage further and create an MVPS (Minimum Viable Product Statement) with your stakeholders.



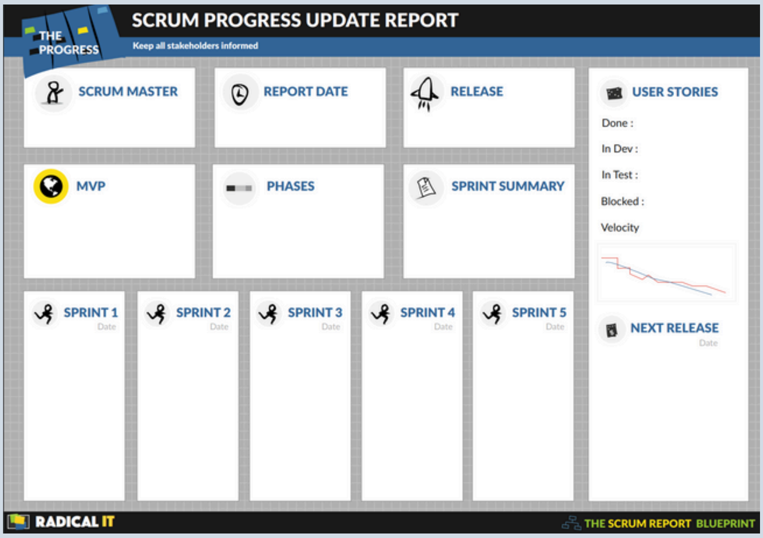
In the book ‘Think, Do, Show: the agile 2.0 secrets to building software people love to use’, Radical IT founder gives this explanation of an MVPS:

> **An MVPS, then, is a single sentence that encapsulates the purpose of the MVP and the desired result. It should detail precisely the specific value being delivered in a user journey. It should also detail how the delivered value can be measured in some way.’**

An MVPS is an excellent resource for keeping everyone on mission and avoiding scope creep. Put it front and foremost in all your comms. It should be the first thing stakeholders see at Show & Tell and the first thing they see at Sprint Planning to act as a handrail for what’s needed for the MVP. It’s a guardrail for discussions and planning too ‘Do we really need this feature for our MVP given that this MVPS is what we’re focusing on’. We’ve even seen people add it to their email signature, as a constant reminder of what everyone should be focusing on.

When creating your Sprint Schedule think in terms of sprint goals and tangible milestones at the end of each sprint. What’s the minimum we can build in two weeks that progresses us towards the MVP. Remember however that teams can get into a pickle here because they concentrate on the word ‘minimum’ and forget about the word ‘viable’. If the product isn’t viable and can’t be used to execute even the happy path, then it’s of no business value.

Figure 2 is an example of a Sprint Schedule from the nuclear submarine emergency response system.



THE PROGRESS

SCRUM PROGRESS UPDATE REPORT

Keep all stakeholders informed

SCRUM MASTER

REPORT DATE

RELEASE

USER STORIES

Done :

In Dev :

In Test :

Blocked :

Velocity

MVP

PHASES

SPRINT SUMMARY

SPRINT 1 Date

SPRINT 2 Date

SPRINT 3 Date

SPRINT 4 Date

SPRINT 5 Date

NEXT RELEASE Date

RADICAL IT

THE SCRUM REPORT BLUEPRINT

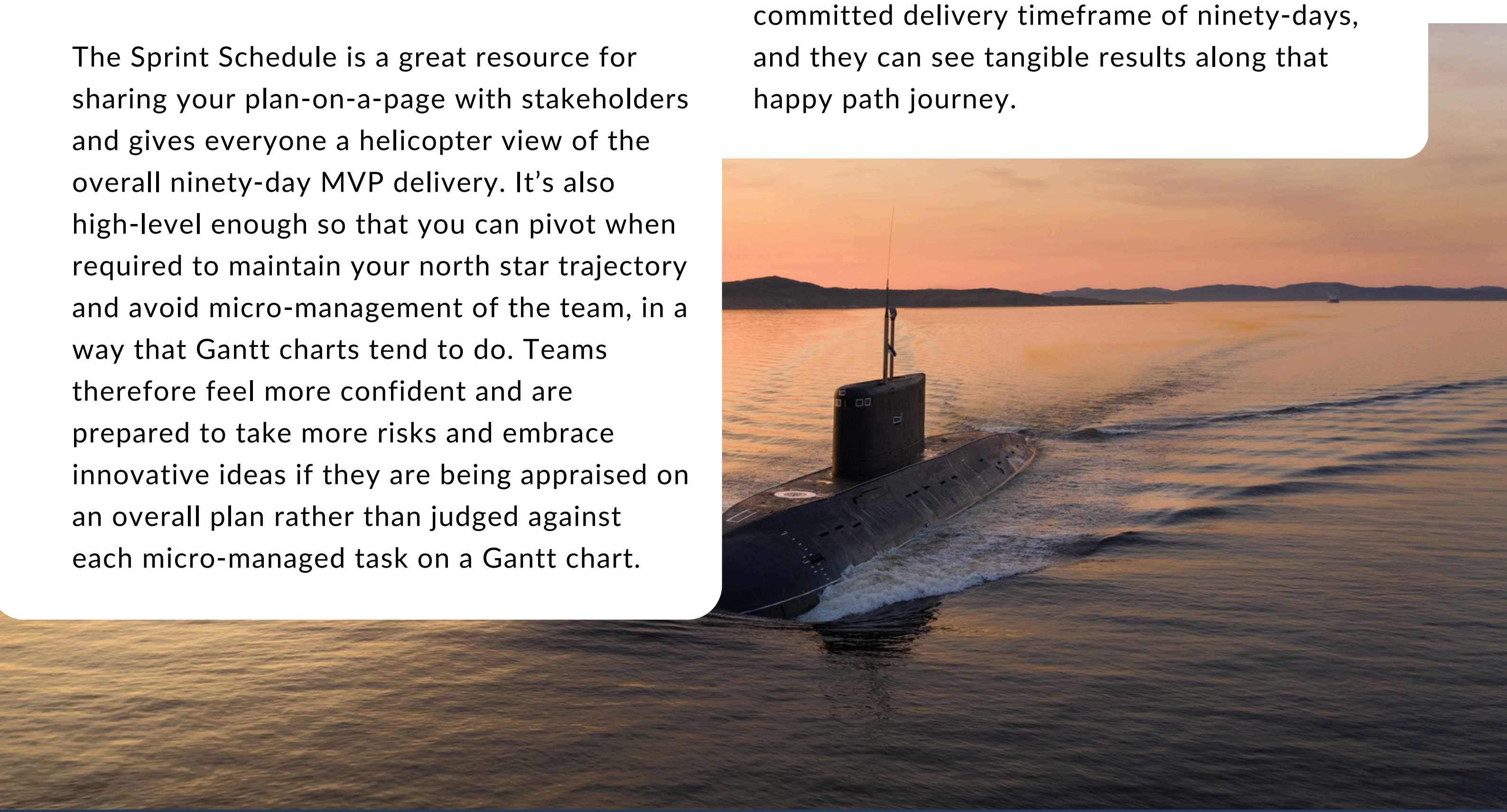
For the nuclear submarine emergency response system (codename RREMS) we mapped out a sprint schedule that had our buckets listed and a goal for each sprint. A great tip here to aid this brainstorming and mapping process is to ask the question 'at the end of this sprint what will we demo to stakeholders...what is it we will show stakeholders that delivers the sprint goal and clearly demonstrates progress on the happy path solution.'

To answer this question the delivery team must first imagine a successful outcome and then reverse engineer the steps they took to get there. This helps the team to focus and concentrate on what's really needed and ignore the superfluous.

The Sprint Schedule is a great resource for sharing your plan-on-a-page with stakeholders and gives everyone a helicopter view of the overall ninety-day MVP delivery. It's also high-level enough so that you can pivot when required to maintain your north star trajectory and avoid micro-management of the team, in a way that Gantt charts tend to do. Teams therefore feel more confident and are prepared to take more risks and embrace innovative ideas if they are being appraised on an overall plan rather than judged against each micro-managed task on a Gantt chart.

Having created the Sprint Schedule, now is the time to crank the delivery handle. End of sprint Show and Tells are key here. Imagine putting yourself in the shoes of a stakeholder. You attend Show and Tell and you're presented with a storyboard of the happy path process and a Sprint Schedule that details the buckets of work the team need to complete an MVP in ninety days. You then see a real demonstration of part of that storyboard, and you're asked for feedback. Finally, you're presented with the part of the process the team will demonstrate at the next Show and Tell.

This is a paradigm shift for most stakeholders. They can clearly see the end goal from the MVP/MVPS and Sprint Schedule. They have a committed delivery timeframe of ninety-days, and they can see tangible results along that happy path journey.



And this is when the real magic begins. Remember those edge cases we conveniently side-lined? True to the 80/20 rule you'll see that a good 80% of the 'we can't possibly do without x' simply fall by the wayside.

They're no longer required or even talked about by users and stakeholders.

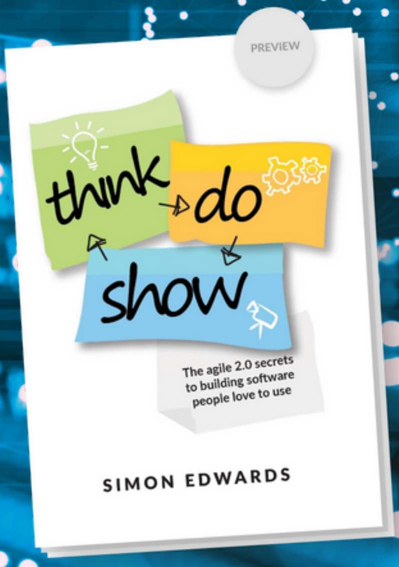
There will still be the 20% that does need addressing, but those you can pick up in the first sprints of the next ninety-day iteration.

One of the key side effects of working this way is the confidence in the team that it creates in stakeholders' minds. The team are moving at pace now, and there is light at the end of the tunnel.

The dynamic completely changes. The team are trusted and valued, collaboration becomes much easier and much of the communication friction dissipates. It's a much better place for everyone to be.

Using this process Radical IT were able to complete the build of RREMS in three ninety-day iterations, including all the forms (we devised an innovative way to generate the code required for the forms to vastly accelerate coding), health physics, plume modelling, mapping integration and even COBR meeting management.

It's important to add at this point that the U.K. has never had a nuclear incident and having worked in the industry now, we can vouch for the scrupulous vigour in which nuclear safety is taken so seriously. At Radical IT we are always struck by the irony of the RREMS project. In that whilst the team put everything into RREMS, and are extremely proud of the work we did, our genuine hope is that no one ever has to use it in a real incident.


PREVIEW
think
do
show
The agile 2.0 secrets to building software people love to use
SIMON EDWARDS