

Austal BDD

Case Study



MECHANICAL
ROCK

The Client

Austal is the Australian shipbuilder, defence prime contractor and maritime technology partner of choice; designing, constructing and supporting revolutionary defence and commercial vessels for the world's leading operators.

In a push to differentiate itself from existing and potential new players in the market, MARINELINK-Smart captures data onboard ships through IoT and sends data to the cloud for consumption by analytics and machine learning services.

MARINELINK-Smart provides better outcomes for ship operators in areas such as fuel efficiency and reduced maintenance costs.

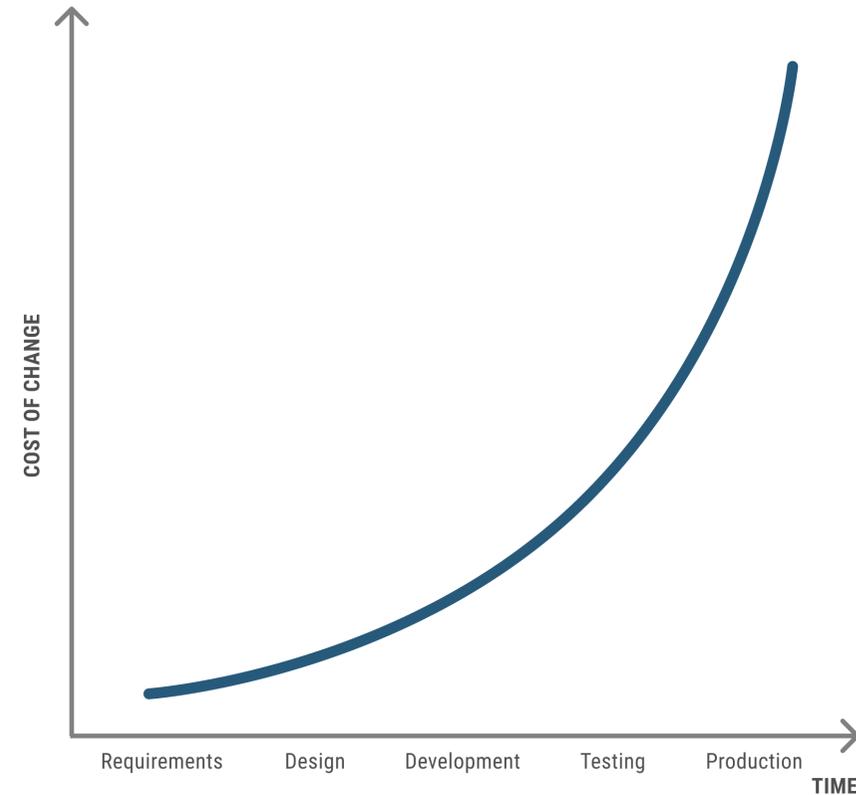


The Problem

MARINELINK-Smart covers a complex problem domain, which meant the development team sometimes encountered problems late in the development cycle. This led to an increased cost to deliver software.

Learning lessons when the software is built is a costly exercise and the cost to address increases exponentially with time.

One example is deciding when to start and end a ship's journey. The user story had seemed simple at first and the team developed a solution based on a lot of assumptions. The assumptions were subsequently disproven and real world scenarios that had not been considered violated business rules, leading to production issues.



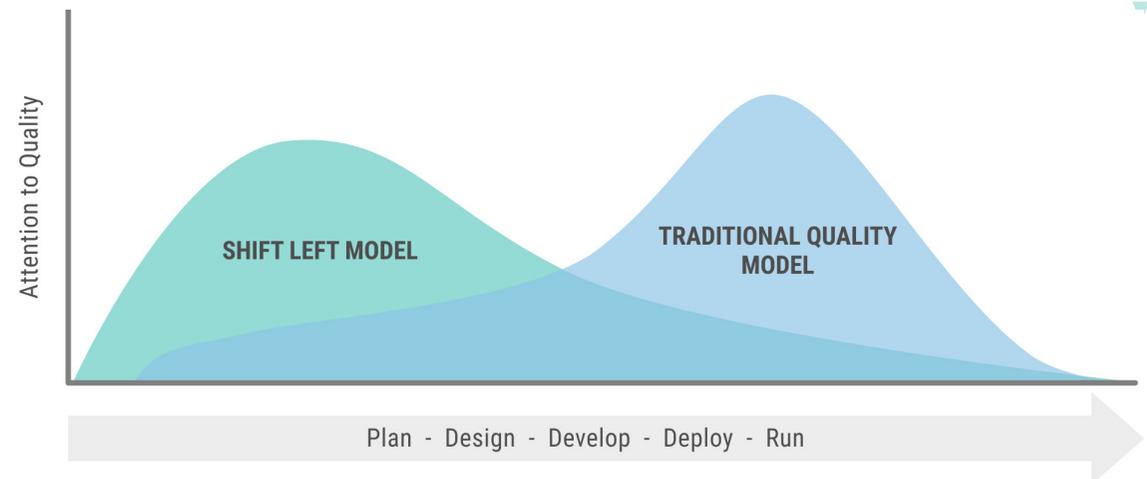
Reference: <http://www.ambyssoft.com/essays/whyAgileWorksFeedback.html>

The Solution

In order to reduce mis-communication and optimise their value chain, Austal attended a **two day BDD Kickstart training course**, hosted by Mechanical Rock.

In traditional, waterfall, development processes, the cost of resolving misunderstanding rises exponentially over time. BDD and example mapping empowers teams to “*shift learnings left*”. Using concrete examples and structured conversations enables product owners and the development team to discover business rules together. BDD removes waste, leading to improved team velocity and reduced development cost.

Over the course of the two days, the experienced trainers coached Austal on how to apply their learnings after the training completed.



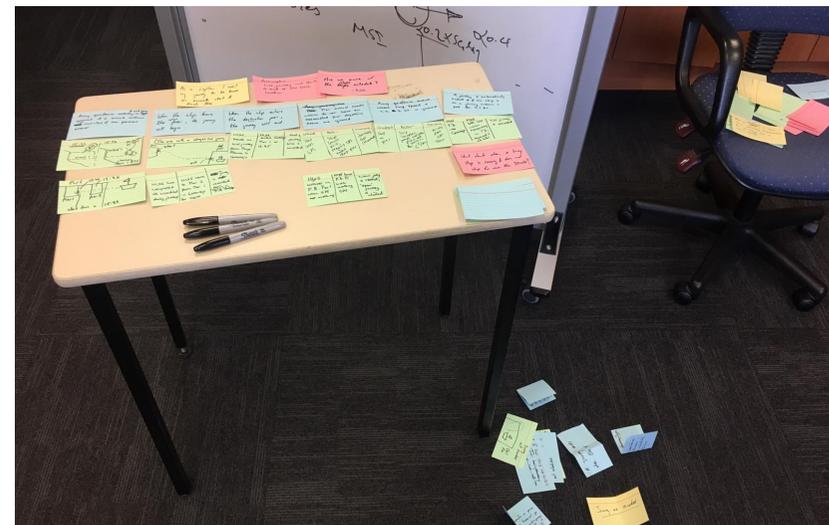
The Solution

The team chose to revisit the story of the start and end of a ship's journey.

Following a 25 minute example mapping session, exploring the business rules using concrete examples, the team discovered that the feature was not ready for development: further research was required.

After a period of reflection, the team reformed and reduced scope by considering the start and end of the journey separately.

The team identified a single business rule with examples that fully specified the user story needs. No questions arose and everyone agreed the story was ready for development. The examples were formulated into an executable specification and signed off, ready for implementation.

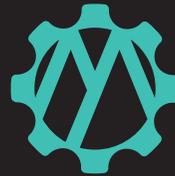


The Benefits

- **Reduced the cost of delivery**
Over 7x reduction in the time for requirements elaboration through a “*shift-left*” approach. The team **saved over two weeks by avoiding miscommunication** and building the wrong thing.
- **Common understanding**
A quorum was reached over the course of one hour, split over two sessions.
- **Innovation and experimentation**
Using concrete examples, the team were able to **formulate and test hypotheses**, without the need to write any code.
- **Improved feedback loops by over 2000x in order to fail fast**
Disprove hypotheses in minutes on paper rather than during implementation or testing.
- **Focus**
A **verifiable, agreed, definition of done** was captured as an executable specification before any development started.
- **Controlled scope**
A visual approach highlighted the complexity of the domain, and enabled the team to **split the problem more effectively**.

“Our first example mapping session was a game changer. Got to the bottom of a problem that’s been plaguing us for a while.”

Brian Foody, Lead Engineer



Think we can help with your project?

Get in touch so we can chat about your plans over a coffee

contact@mechanicalrock.io

www.mechanicalrock.io