

F/A-18 Super Hornet & Growler HUMS

Agile Software Development to Deliver a Deployable Engine Life Management Capability

Rob Findlay
BAE Systems Australia
rob.findlay@baesystems.com

25 February 2019
Defence Science and Technology International Conference on Health and Usage Monitoring Systems (HUMS 2019)

Agile Software Development - Deployable Engine Life Management

Overview

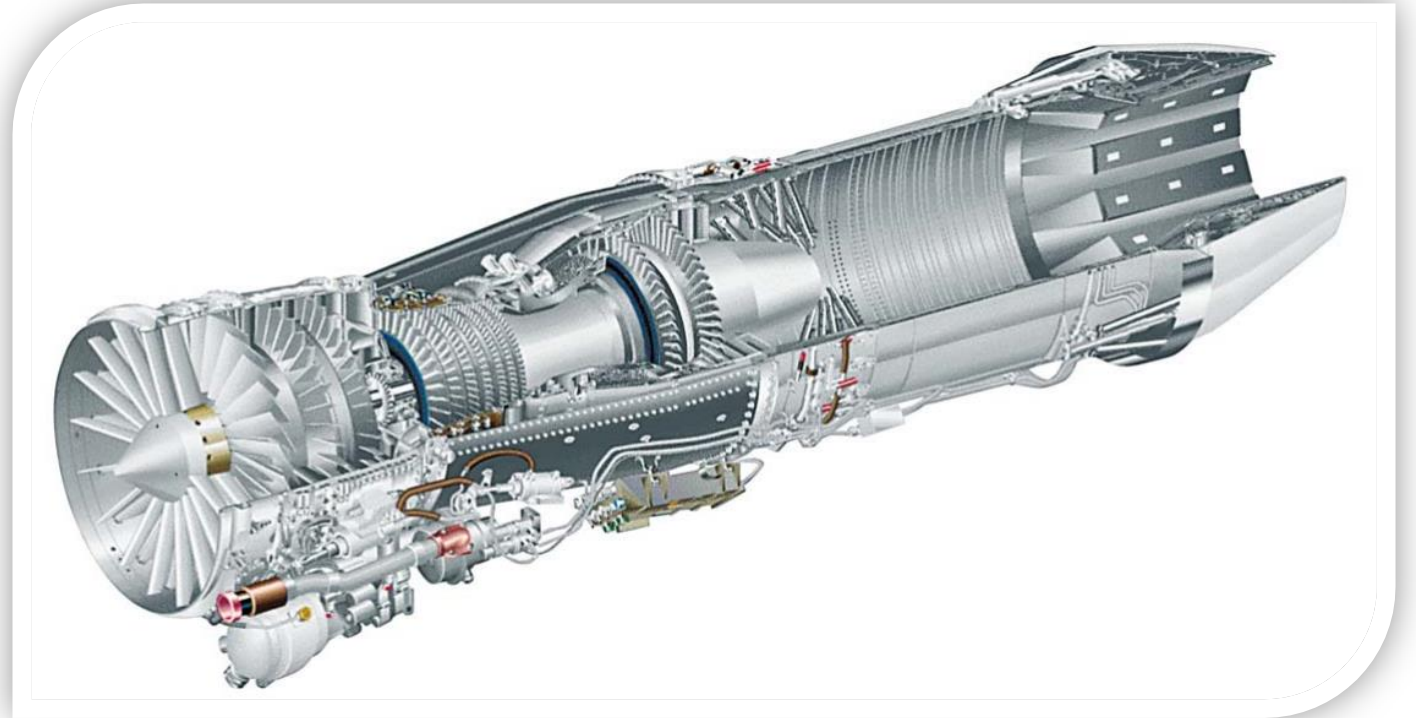
- Background
- Problem
- Solution
- Software development methodology
- Implementing Agile
- Summary
- Questions



Agile Software Development - Deployable Engine Life Management

Background

- Super Hornet / Growler:
Twin F414-GE-400 engines
- Fatigue critical components
- Usage monitoring system tracks accrued fatigue vs. limits
- For maximum availability and minimum cost, engines are flown as close to limit as possible
- Fortnightly cycle of data gathering, processing and reporting



Agile Software Development - Deployable Engine Life Management

Problem

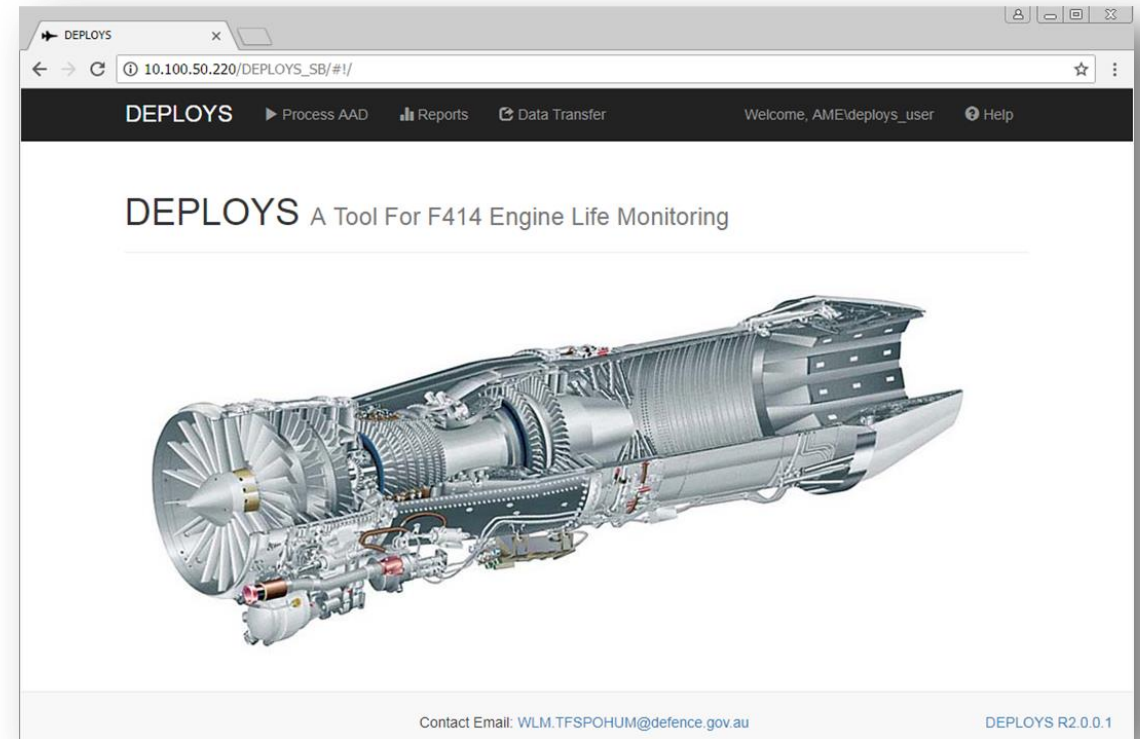
- Low life engines: weekly, daily, every flight
- Turnaround times, exacerbated by deployment
- Lessons learnt from F/A-18F Okra deployment in 2014
- F/A-18F to return to Okra in 2017



Agile Software Development - Deployable Engine Life Management

Solution

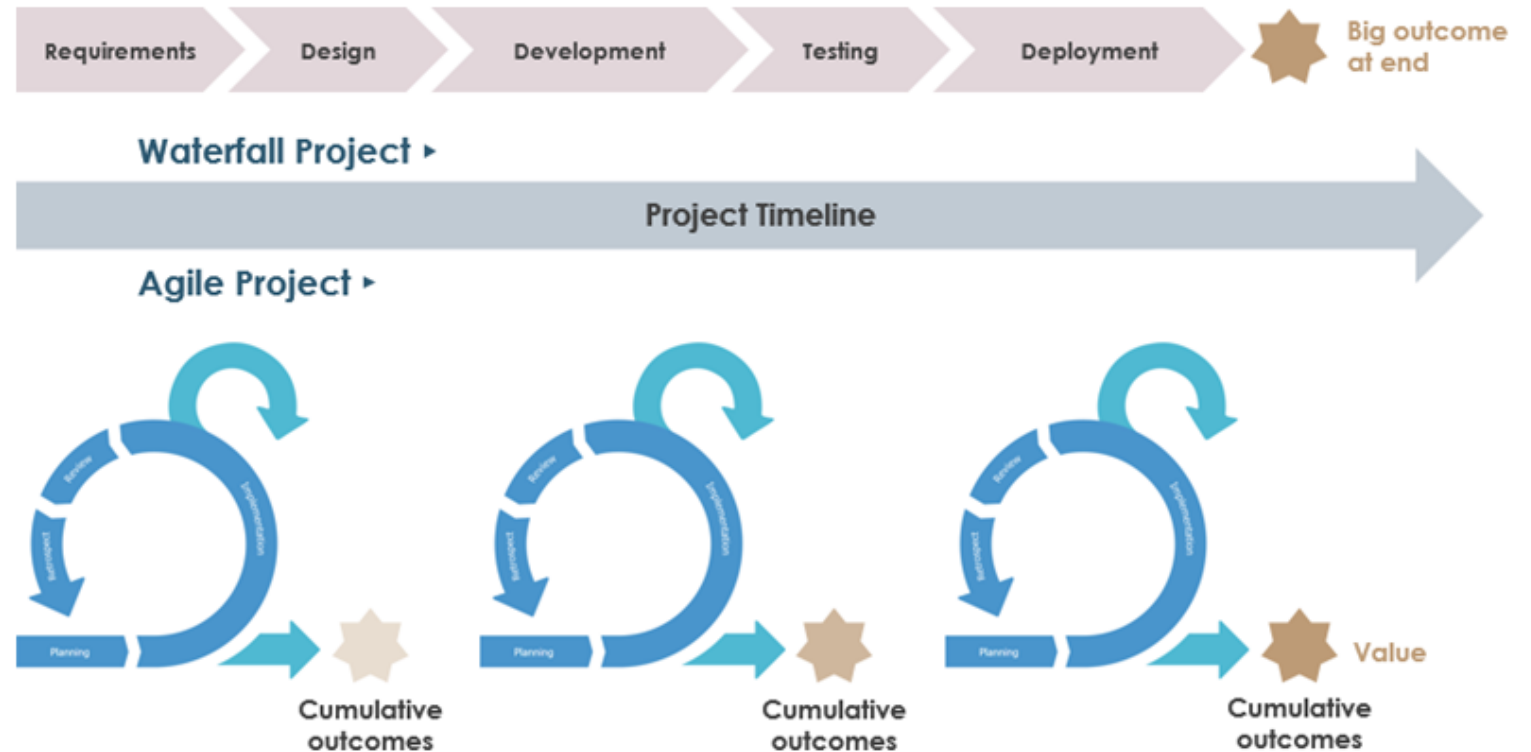
- Develop software for existing deployed AME kit to allow maintainers to process aircraft data locally after every flight to provide up to date fatigue tracking
- Aggressive schedule, no slippages acceptable
- Eliminate non-critical capability
- Prioritise task
- Identify and mitigate development risks
- Involve design signatories and stakeholders early and continually



Agile Software Development - Deployable Engine Life Management

Software development methodology (Waterfall)

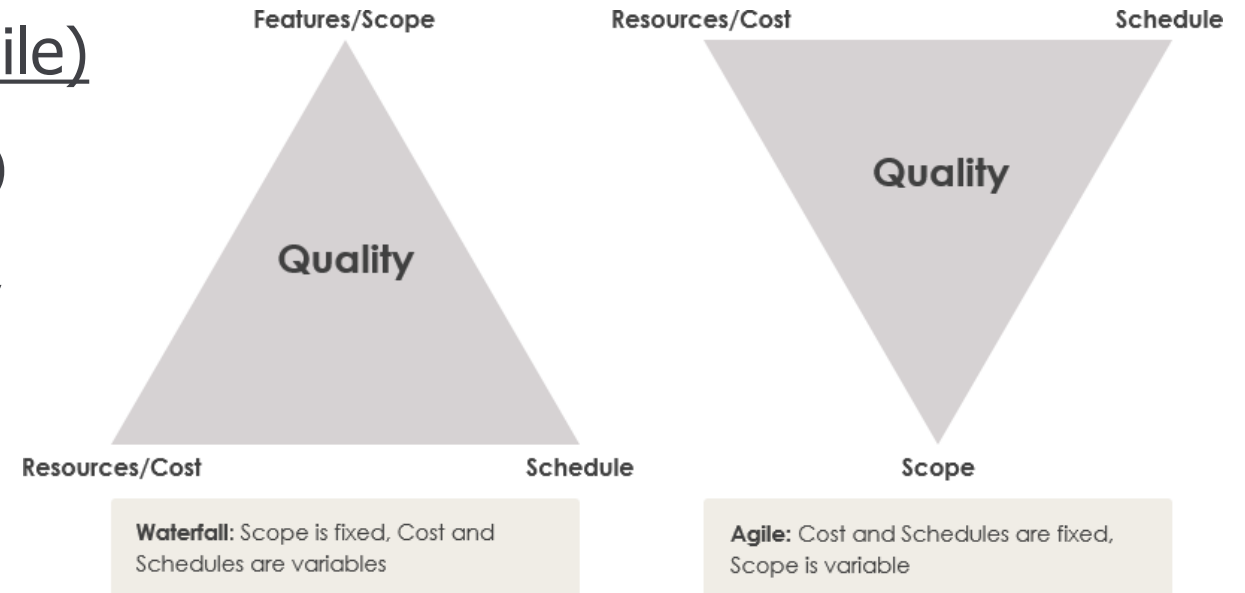
- We use a traditional Waterfall approach
- Given uncertainty around schedule and ability to tailor scope depending on available time, felt this was not the best approach for this task
- Need to explore alternatives such as an Agile methodology



Agile Software Development - Deployable Engine Life Management

Software development methodology (Agile)

- With a fixed schedule (upcoming deployment) and resources, but the ability to tailor scope to include only the essential features plus any desirable features within available time, Agile is attractive
- Particularly given the increased probability of success that it offers



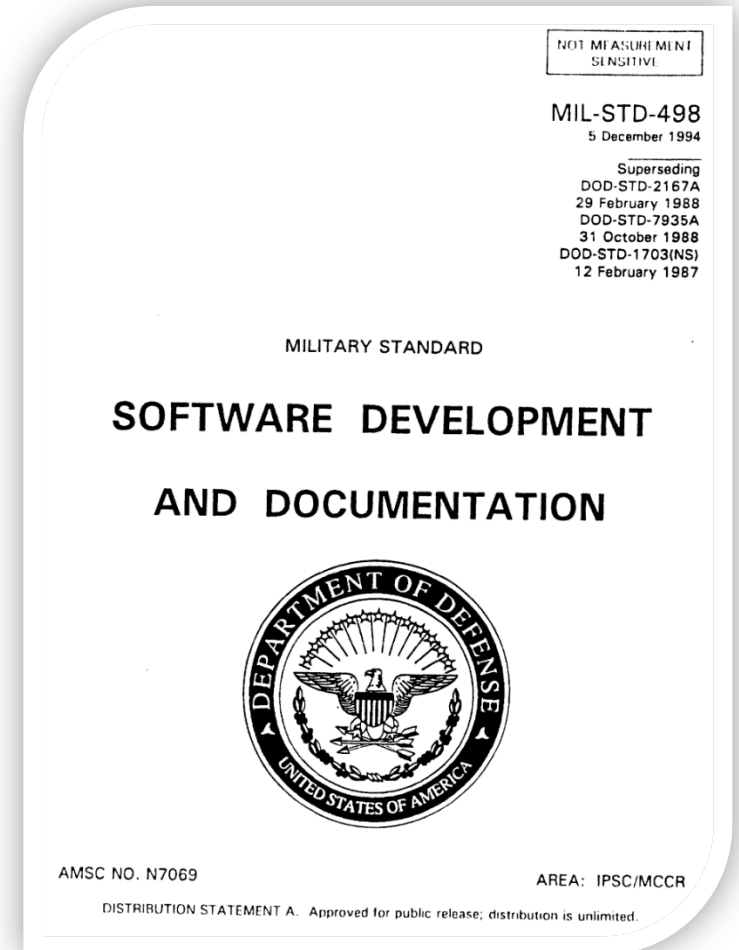
Method	Successful	Challenged	Failed
Agile	42%	50%	8%
Waterfall	26%	53%	21%

(Source: vitalitychicago.com – [Comparing Waterfall and Agile Project Success Rates](#))

Agile Software Development - Deployable Engine Life Management

Implementing Agile (potential pitfalls)

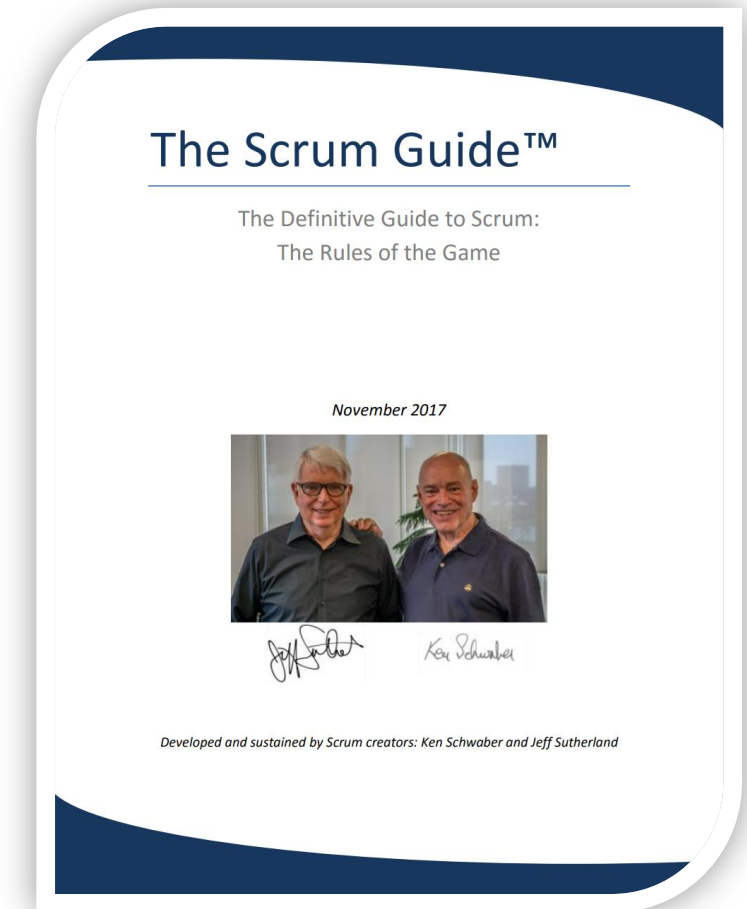
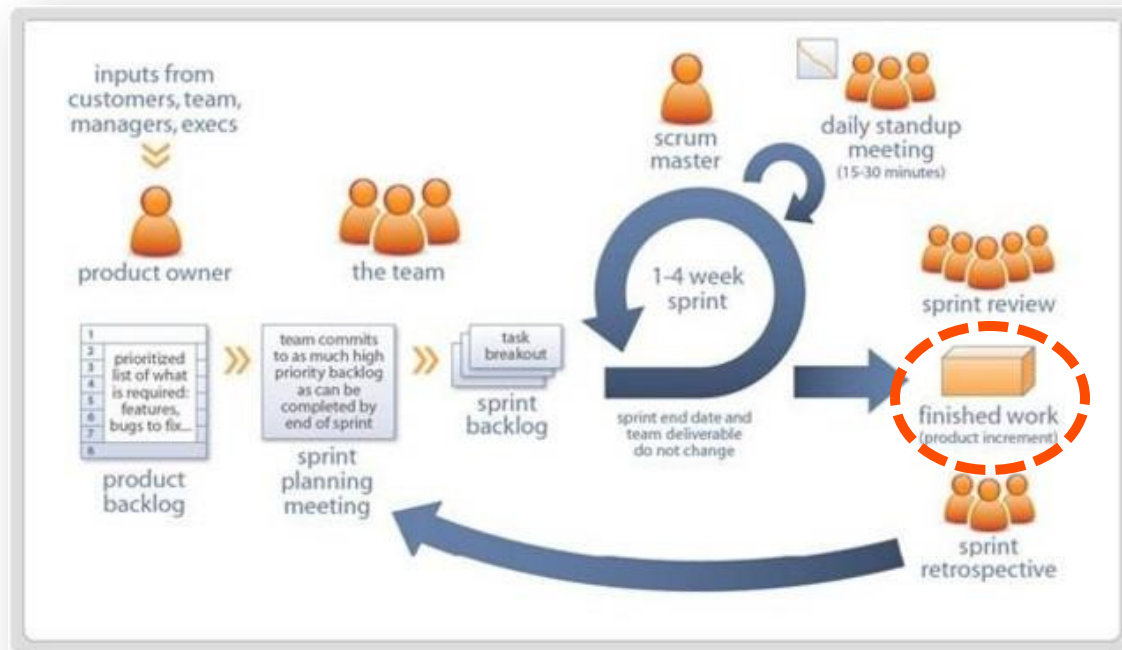
- Contracted to ISO/IEC 12207 (Software lifecycle practices)
- Document design i.a.w. MIL-STD-498 (Software development and documentation)
- Engineering Management System and culture based on historic alignment with ADF TMM and AEO certification
- Software produces reports that have a safety / airworthiness impact
- Every sprint in Agile (3-4 weeks) is supposed to result in releasable software
- How to gain the benefits of Agile within the constraints of our environment?



Agile Software Development - Deployable Engine Life Management

Implementing Agile (tailoring Scrum)

- *"Scrum is a framework for developing, delivering, and sustaining complex **products**"*

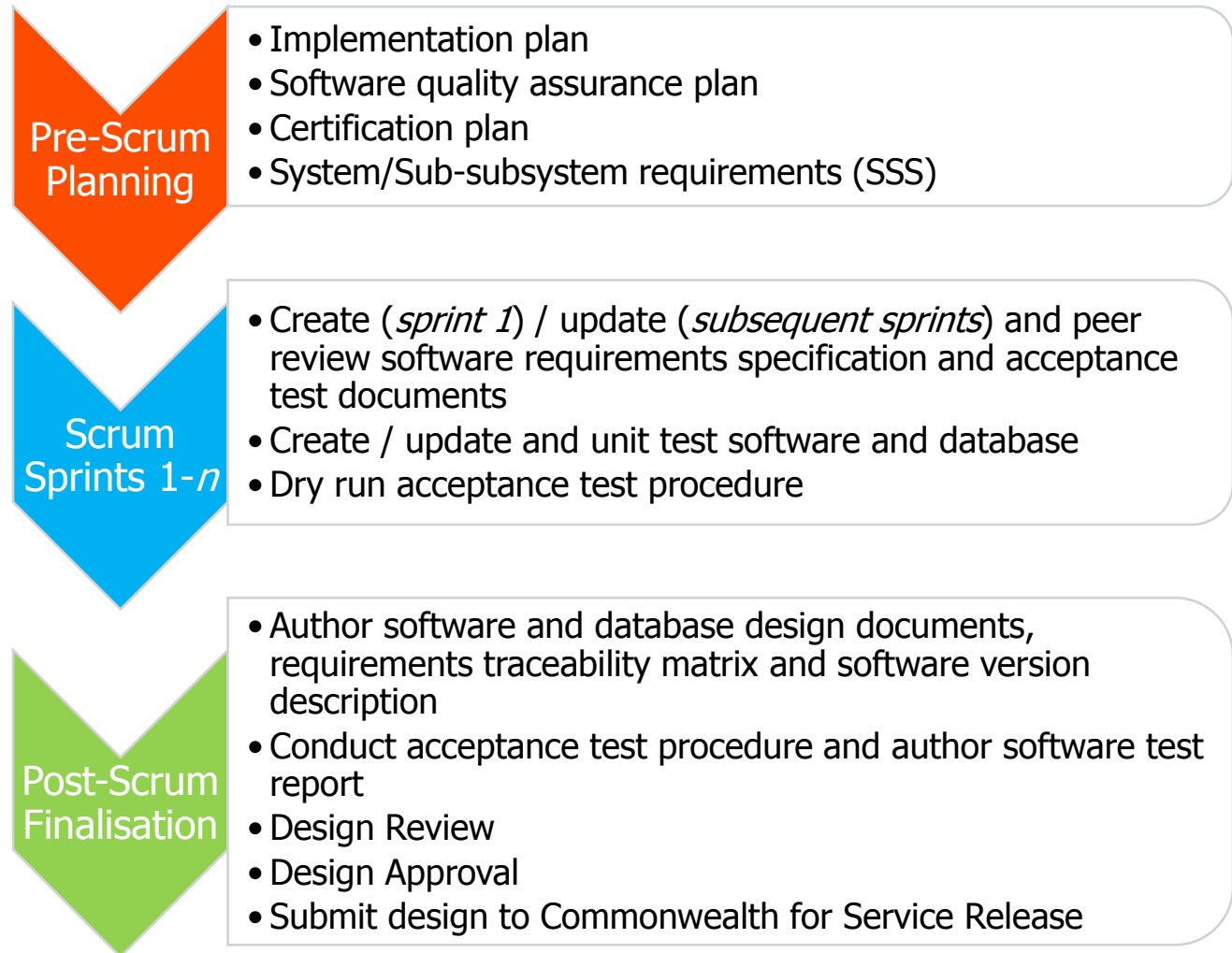


(Source: <https://www.scrumguides.org/>)

Agile Software Development - Deployable Engine Life Management

Implementing Agile (tailoring Scrum)

- We tailored Scrum to suit our needs, combining the elements we could with existing work practices
- During the sprints, we implemented Scrum in accordance with *The Scrum Guide*
- We didn't have a releasable product at the end of each sprint (the whole point of Scrum)



Agile Software Development - Deployable Engine Life Management

Implementing Agile (tailoring Scrum)

- **But** we were able to take advantage of many of the benefits of Scrum
 - Sprint planning: Tailor scope to the given time and resources
 - Daily stand-up: Increased communication between remote personnel
 - Sprint review: Early feedback from stakeholders on developed product
 - Sprint retrospective: Adjusting the plan based on the team's performance and changing environment (groomed backlog)



Agile Software Development - Deployable Engine Life Management

Implementing Agile (learnings)

- Ignore Scrum advocates who say that Scrum is not tailorable (but be aware of the impact of your tailoring)
- Look upon challenges as an opportunity to implement change in your work practices and culture
- Scrum pushes the team to live the *Scrum values* of commitment, courage, focus, openness and respect – which increases enjoyment and productivity in the workplace



Agile Software Development - Deployable Engine Life Management

Summary

- Software delivered, accepted and installed on-time
- Operational for first deployed flight with no significant problems
- Maintainers can now process data immediately post-flight to determine serviceability for next sortie, giving them greater flexibility in executing critical missions
- Have since implemented update to the software to allow them to track low life engines at home

"The product meets an important operational need (learnt from the first operational deployment) that provides significantly improved deployed flexibility through better engine asset management ... the product was both fit for purpose and on time; it has been fielded and is meeting the operational need."

– WGCDR Scott Parry, Chief Engineer

Agile Software Development - Deployable Engine Life Management

