scram™
SCHEDULE COMPLIANCE RISK ASSESSMENT METHODOLOGY

From Waterfall to Agile: Lessons in Safety Assurance and Cyber Security from the Trenches

Australian System Safety Conference 2017
Sydney Australia. 31 May — 2 Jun 2017

Angela Tuffley
SCRAM Principal and
Director, RedBay Consulting Pty Ltd

Elizabeth (Betsy) Clark
SCRAM Principal
President, Software Metrics Inc

Hi from Betsy

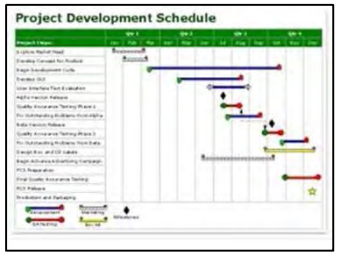
Topics

- SCRAM Overview
- Development Life Cycles
- Lessons Learned from the Trenches
- Conclusion

Schedule Compliance Risk Assessment Methodology (SCRAM) has been developed

To benefit decision makers, program managers and the acquisition community…

by providing a methodology that assists experienced engineers and subject matter experts…

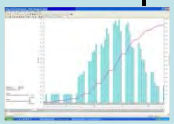


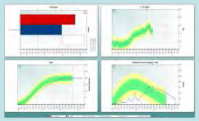
to consistently identify root causes of schedule slippage and recommend remedial action.

What is SCRAM?

An independent review to identify issues and risks to schedule

- Quantifies the schedule impact of issues and risks using scientific analysis techniques
  - Schedule Monte Carlo Simulation
  - Software Parametric Modelling





Embodies best practices

- Systems and software engineering
- Schedule development and project execution

Facilitates improved business practices

- Based on feedback from reviews
- Identification of systemic root causes / issues

SCRAM Usage

Sponsored by the Australian Department of Defence

- To improve Project Schedule Performance in response to Government concern as identified by the Australian National Audit Office (ANAO)
- Successfully applied to the F-35 JSF Program in the USA and has been used to monitor software development performance on the program (web search “F-35 Australian SCRAM”)

Diversity of SCRAM Reviews

Aerospace

Satellite Ground Stations

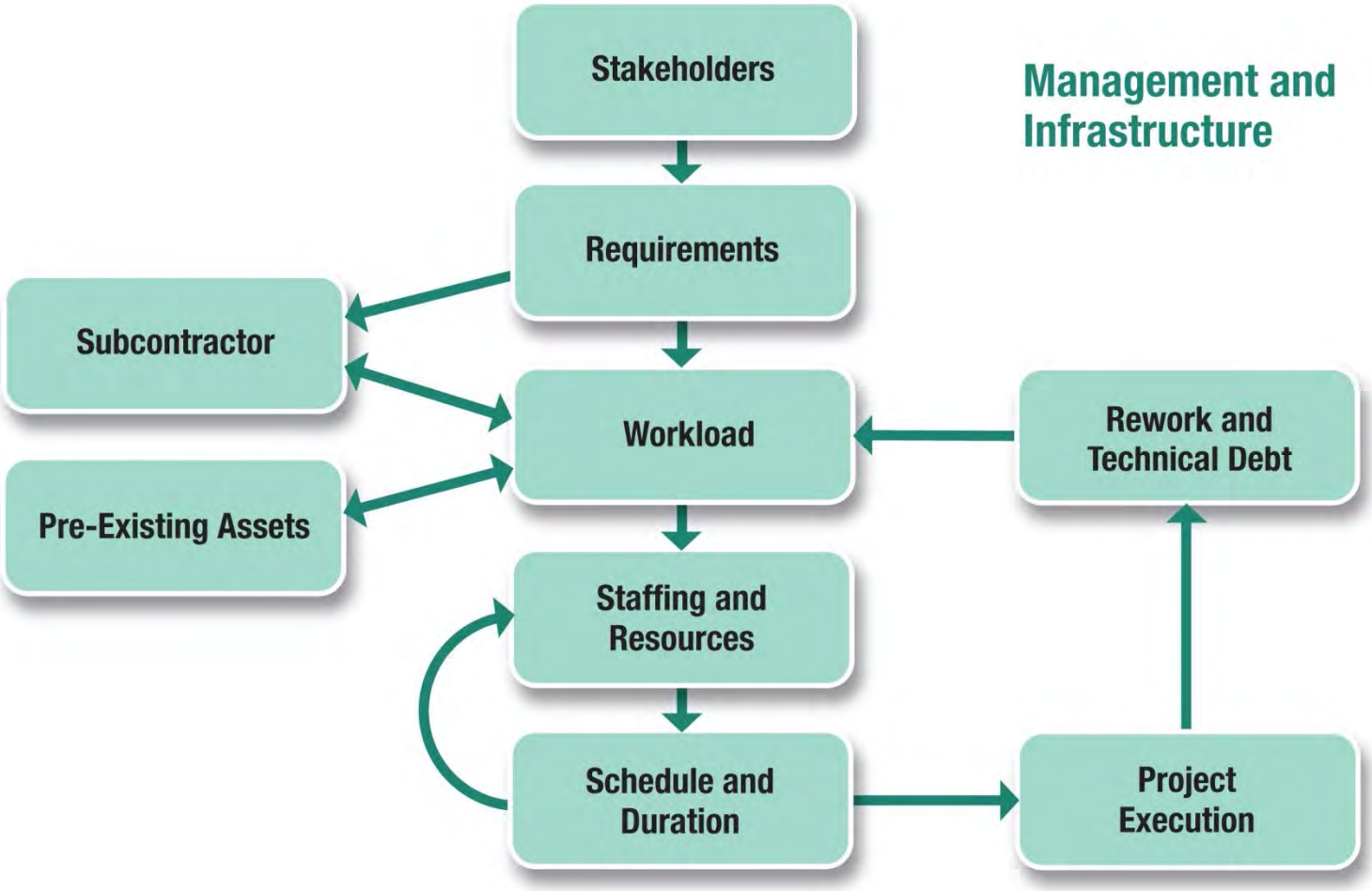
Maritime

Enterprise Resource Planning

Telecommunications

Training Systems

Root Cause Analysis of Schedule Slippage (RCASS) Model

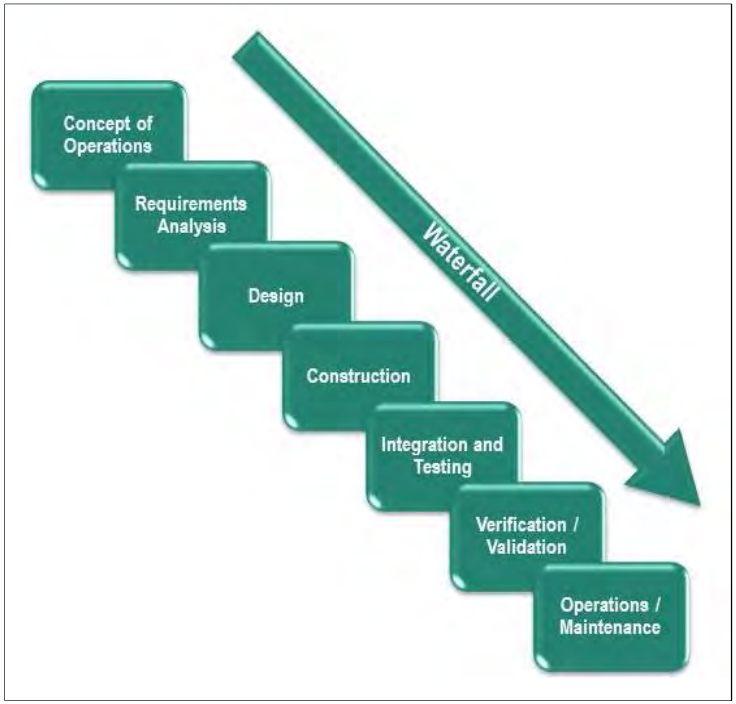


Topics

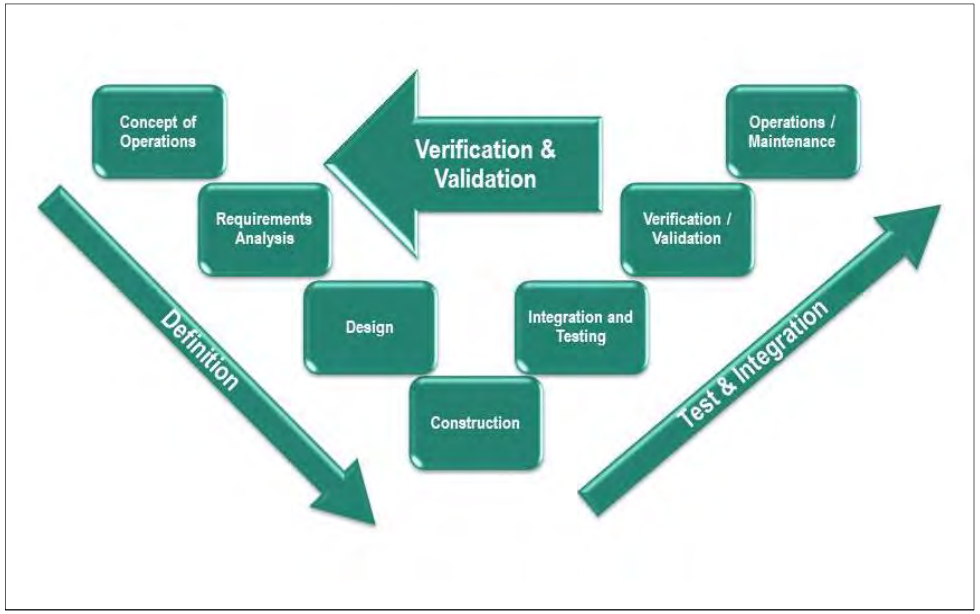
- SCRAM Overview
- Development Life Cycles
- Lessons Learned from the Trenches
- Conclusion

Traditional Development Life Cycles

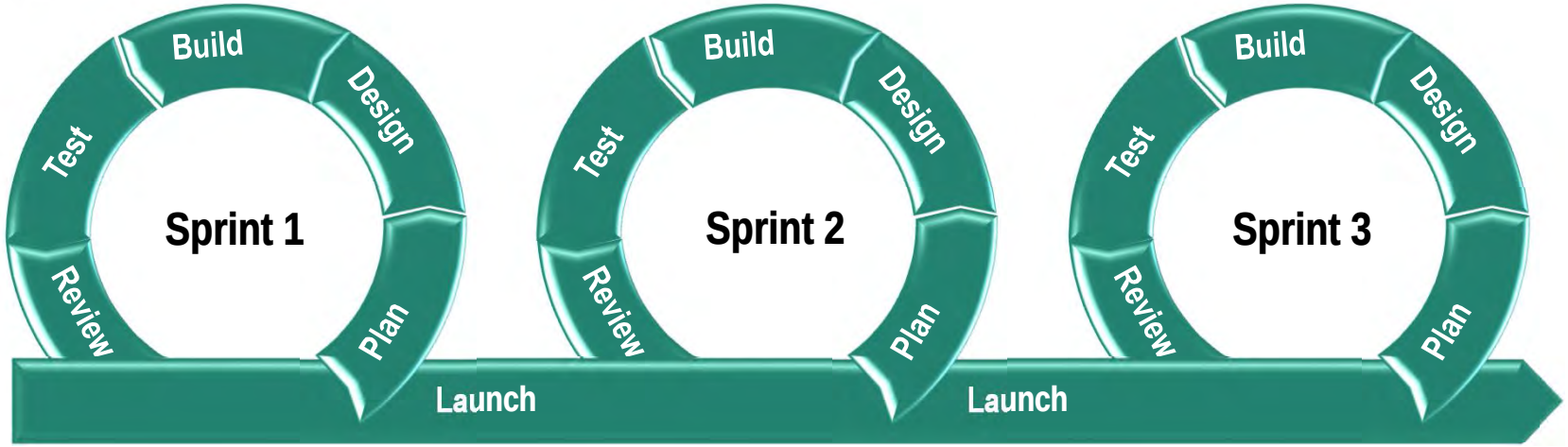
Waterfall



"V" Model



Agile Development Life Cycle



- The Agile Manifesto adopts four key values.
  - Individuals and interactions over processes and tools
  - Working software over comprehensive documentation
  - Customer collaboration over contract negotiation
  - Responding to change over following a plan
- Defence domain
  - Initially applying the traditional approach of bounding the solution
  - Followed by the Agile approach of sprints for the remaining phases

➢ **Modified Agile**

Topics

- SCRAM Overview
- Development Life Cycles
- Lessons Learned from the Trenches
- Conclusion

Five development projects

Project	Life Cycle	Phase
A	Traditional	Verification
B	Traditional	Operations & Maintenance
C	Traditional	Verification
D	Modified Agile	Verification
E	Modified Agile	Requirements/ Design/ Implementation

Five key areas impacting safety and security assurance

Project A

- Mature system developed in the 80s with a large international customer base
- Australian-specific enhancement resulted in additional safety critical features
- Lacked detailed documentation to adequately trace safety critical requirements through design to code to test cases
- Significant additional effort required to certify the system in Australia

Project B

- Rapidly aging legacy system
- Redesign and replace the system
- System configuration changes without appropriate approvals
- Not able to progress past design acceptance
- Conducting a Physical Configuration Audit to document the current configuration

Project A

- Delayed by a difference in expectations between local certification authorities and for Australian authorities
- Resulting in considerable additional effort.

Project D

- System to be hosted on global defence networks
- Failed to fully understand cyber-security certification requirements
- Additional software had to be developed to support system penetration testing
- Project had to work with several government agencies on how to test the security features of the design
- Total delay to the project was more than a year.

Project A

- Lack of a rigorous safety assurance process
- Quality of documentation lacked sufficient evidence of testing and traceability
- Required extensive corrective actions
- Addressing safety by software developers sitting with software safety expert to conduct detailed software safety analyses followed by independent peer reviews

Project C

- No agreed position System Acceptance requirements
- Main issue to pass FCA which required the safety case and cyber-security assessment
- Late “Penny Drop” moment about the requirement for a safety case
    - Safety engineer began working closely with the technical authority
    - Prior to this communication had only been via email.

Project D

- Incremental release upgraded two COTS operating systems
  - Required re-certification
- Re-certification several years after the previous one that led to a delay of more than a year
- Much smoother
  - Knew what to expect and could plan for it

Project E

- Security certification had never been granted for wireless COTS devices
- Breaking new ground
- To mitigate, working groups established with the signals directorate to understand requirements
- Proactively working certification risks early in development

Project E

- Security penetration testing of the system
- Conducting threat assessments and training prime contractor's team on defensive coding
- Conducting regular follow on "red team" attacks on the system
- Proactive approach to cyber security

Project X

- Not included in the original analysis as not a developmental project
- Certification of COTS products
- International prime subcontracted local SME during the tender phase to get it right

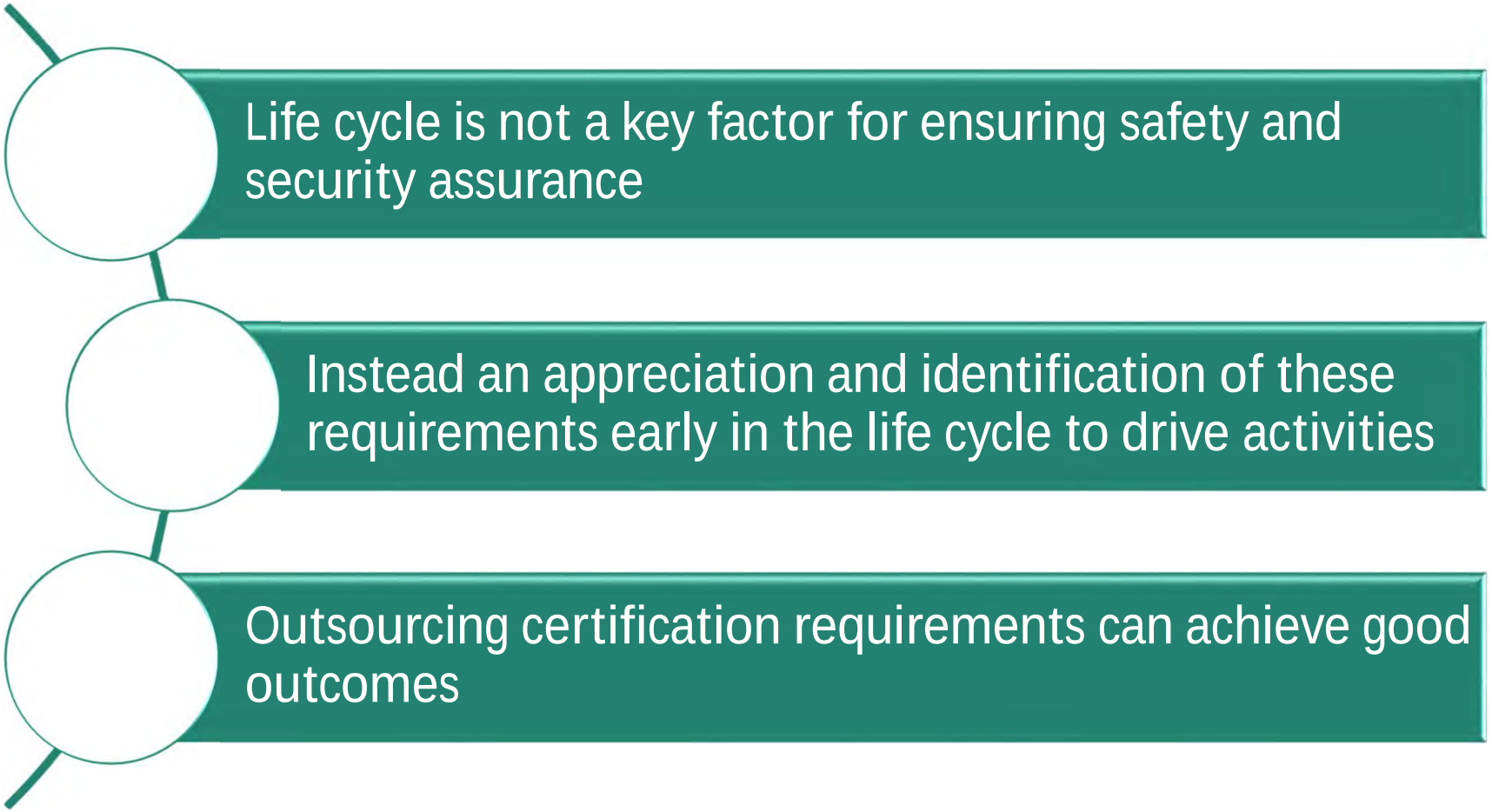
Topics

- SCRAM Overview
- Development Life Cycles
- Lessons Learned from the Trenches
- Conclusion

Five projects

Project	Life Cycle	Phase	Certification Impact on Schedule
A	Traditional	Verification	Delayed
B	Traditional	Operations & Maintenance	Delayed
C	Traditional	Verification	Delayed
D	Modified Agile	Verification	Delayed
E	Modified Agile	Requirements/ Design/ Implementation	No effect
X	Traditional	Requirements	No effect

Conclusion



BACKUP SLIDES

BER Berlin Brandenburg Airport
7 Jan 2017

Berlin Brandenburg Airport Timeline

		BER Opening Date
Sep 2006	Construction Started	30 Oct 2011
May 2010	Topping out	
Jun 2010	New Opening Date announced	3 Jun 2012
Nov 2011	Acceptance Tests commence	
May 2012	Acceptance Test failure	
Jun 2012	New Opening Date announced	17 Mar 2013
Jan 2013	Opening Date announcement	Not until 2014
Jan 2014	Opening Date	Not 2014!
Feb 2014	Opening Date	Unlikely before 2016!!
Aug 2014	Opening Date	Probably 2017 / 2018!!!
Oct 2016	Opening Date	Opening in 2017 impossible!!!!
Mar 2017	Opening Date	Pushed to 2018 or 2019 with 2020 a possibility!!!!

WHY?

- Failure (officially) of the fire protection system
  - Not built according to the construction permit
  - Failed mandatory acceptance tests
    - Interim solution to employ up to 700 human fire spotters which was rejected!
  - Flaws in the wiring, programming and implementation of the highly complex automatic control system for sprinklers, smoke extractors and fire doors designed by Siemens and Bosch
  - Terminal ceiling smoke extraction ducts
    - Designed to NOT exhaust to the rooftop
    - Instead smoke would be pumped from the ceiling into a shaft running down and through the basement below the structure
    - Reversing the natural rising behaviour of hot air in the shaft
      - thus far, this elaborate smoke extraction system has not worked as planned
      - large scale reconstruction work might be needed

WHY??

- More Fire Protection Woes
  - Cable conduits hold too many cables or are in incompatible combinations
    - Phone lines with high voltage wires
  - 60 km of cooling pipes allegedly installed with no thermal insulation
    - To correct this, demolition of numerous walls may be necessary
  - 18 km exhaust system to remove smoke from a fire is leaking
  - Some lightning rods were missing
  - Back-up generator powering the sprinkler system did not provide adequate power
  - 600 fire protection walls have to be exchanged
    - Erroneously built out of gas concrete blocks and are insufficient fire protection
    - The mortar inadequate as well

WHY???

- And the list goes on!
  - Incoming or departing trains may suck smoke into the underground railway station
    - Needs a redesign for the underground part of the fire exhaustion system
  - 3000 smoke detectors went missing
    - But were later found
  - The sprinkler system has sustained failures
    - The sprinkler heads were replaced for increased water flow, but the pipes are too thin to withstand the increased water pressure
- And after this photo was taken

  - Technical issues involving the electric doors (January 2017)
  - The transformer station exploded (March 2017)

And to throw fuel on the fire

- A former manager for BER was taken into remand for the alleged bribery in May 2015

- An Imtech manager is alleged to have given bribes in an envelope at a highway gas station in 2012 to a BER manager
  - Imtech built parts of the fire exhaustion system

- Imtech filed for bankruptcy in August 2015
  - The parent company of Imtech went also bankrupt a few days after its German dependency

- Angela Tuffley; Director - RedBay Consulting Pty Ltd, a.tuffley@redbay.com.au
- Betsy Clark, President – Software Metrics Inc, betsykclark@gmail.com

- Addendum
  - Thanks to Mr Jeff Morris
    - Retired Vice President from Lockheed Martin
    - In charge of the development of the Mission Systems software for the F-35 Joint Strike Fighter)
    - Worked a number of years in Australia
      - Jindalee Over the Horizon Radar Network (JORN)