The following presentation was given at:

SCAF Vendor & Service Provider Day

Tuesday 14th November 2017
The BAWA Centre, Filton, Bristol, BS34 7RG

Arke Ltd

Burchelli Consulting Limited

SCAF Vendor Day 2017

Mission Systems Software Cost Forecasting in an Airborne Environment

Presented by
Andy Mills – Arke Ltd
Spencer Woodford – Burchelli Consulting

1. Background
2. The Requirement
3. The MS Model Scope and Scale
4. Levels of Autonomy?
5. Any Questions
6. Demonstrations Available

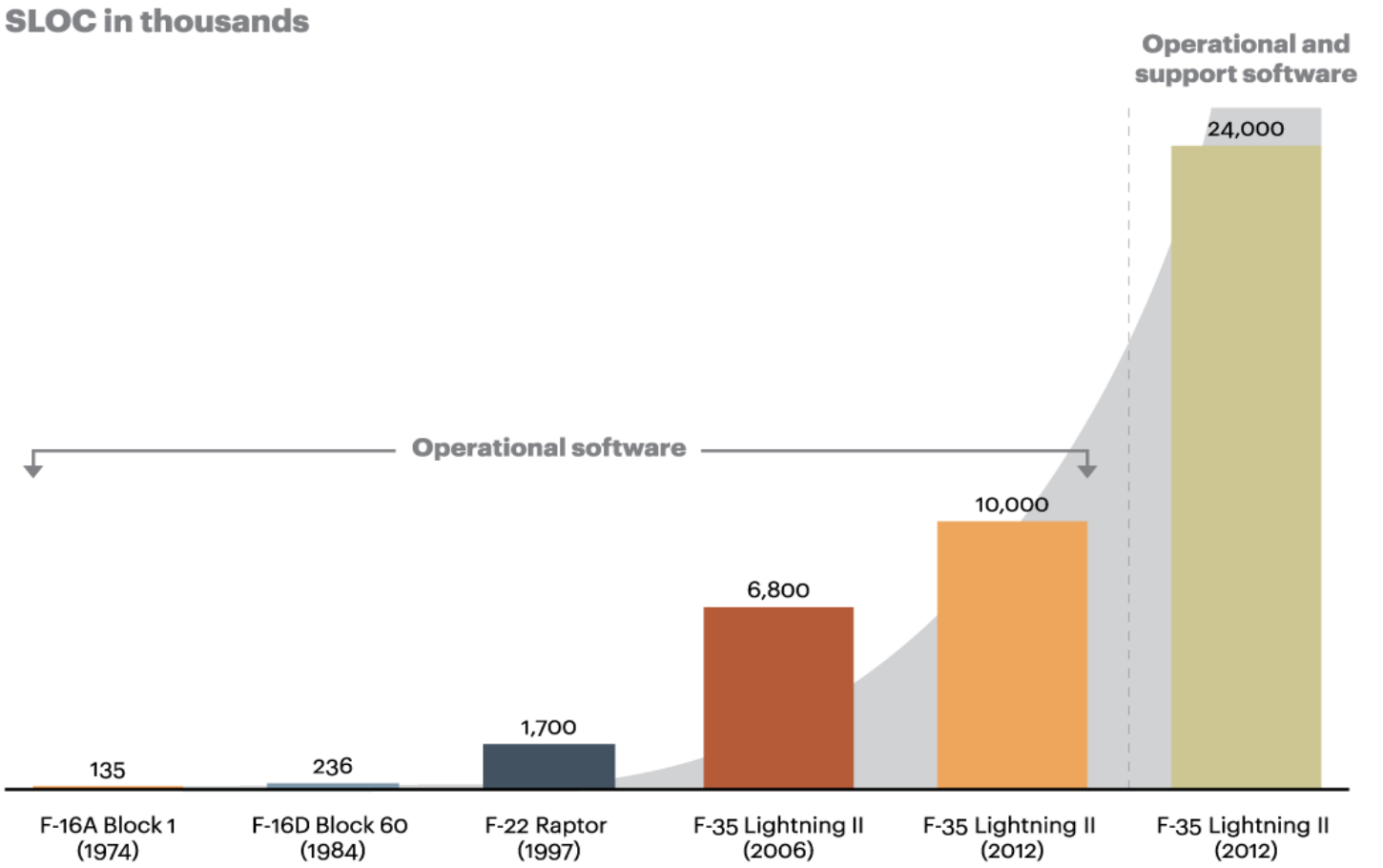
Background

- Direction of Travel
  - The Future Combat Air System (FCAS) programme is addressing the UK requirement for Combat Air (CA) capability in 2030+
  - Dstl needs to investigate the Whole Life Cost (WLC) and Cost Drivers of Manned and Unmanned Air System concepts in these timelines
  - FinMilCap-FCAS needs decision support for an operational demonstrator vehicle being pursued jointly with France and the delivery of alternative combat air systems, including low-cost concepts
  - Support the need for independent generation & challenge of cost estimates currently being provided by industry for concept programme such as the UK/FR UCAS Demonstrator Programme

- Current Capability
  - Dstl is utilising the Activity-Based Aircraft Costing and UtiliSation (ABACUS) WLC model – limited software modelling functionality
  - Current parametric software estimating tools do not incorporate capability for specifically understanding the impact of autonomy, open systems and unmanned operations on software development and support

AND

Software is expensive

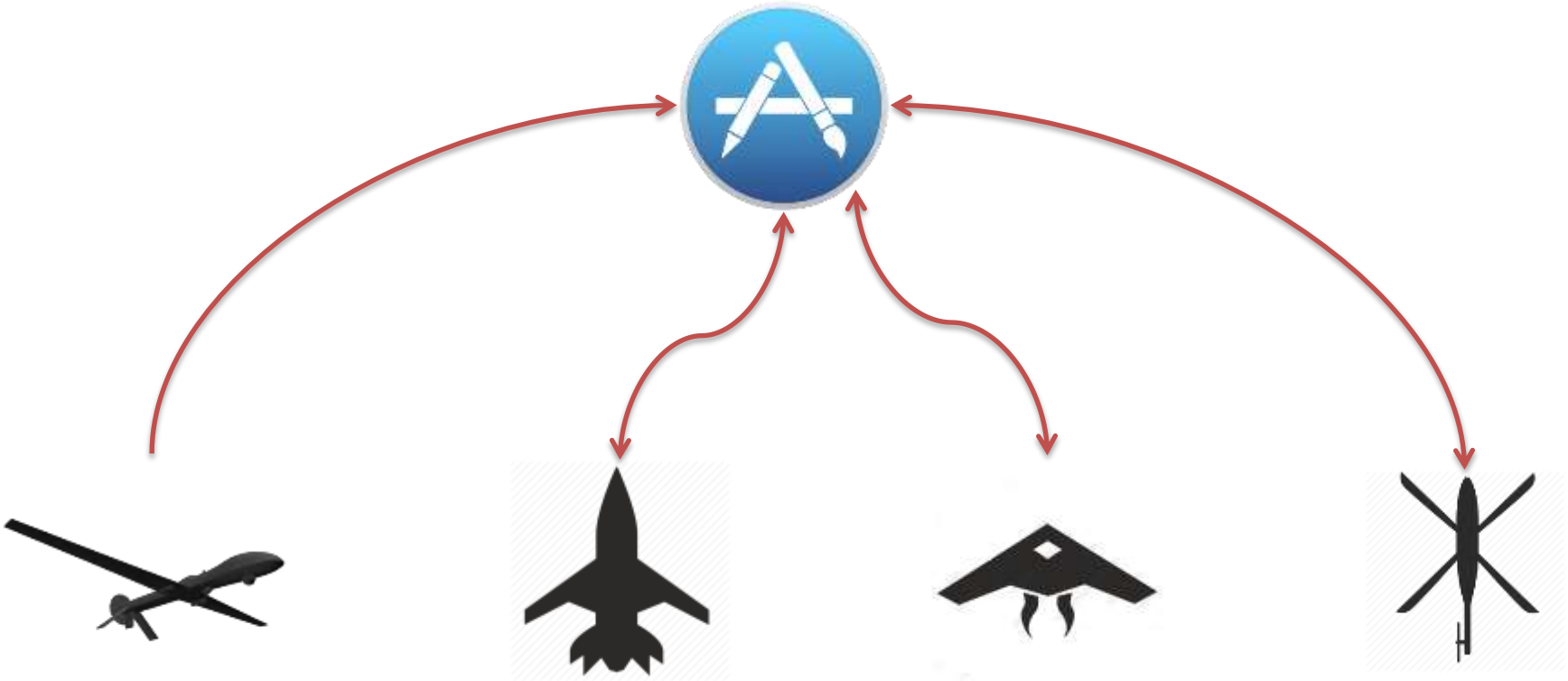


Notes: SLOC for F-16 and F-22 are at first operational flight. F-35 SLOC figures are from first test flight and current estimates/sources.

Sources: P. Judas and L.E. Prokop, "A historical compilation of software metrics with applicability to NASA's Orion spacecraft flight software sizing," Innovations in Systems and Software Engineering, vol. 7 issue 3, September 2011. p. 161–170; Andrea Shalal-Esa, "Pentagon focused on resolving F-35 software issues," Reuters, March 2012; Robert N. Charette, "F-35 Program Continues to Struggle with Software," IEEE Spectrum, September 2012

So if we could re-use software

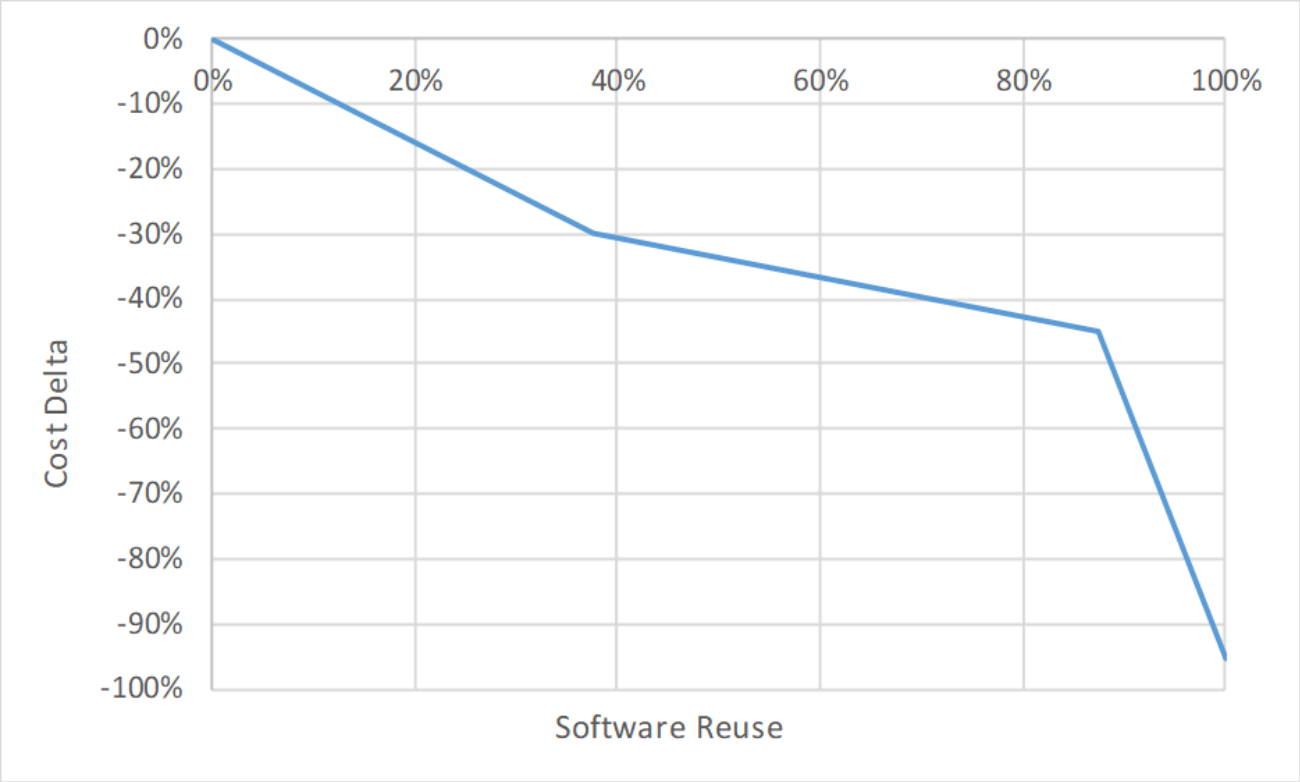
App Store



Software Re-use:

Time

Could we save money?

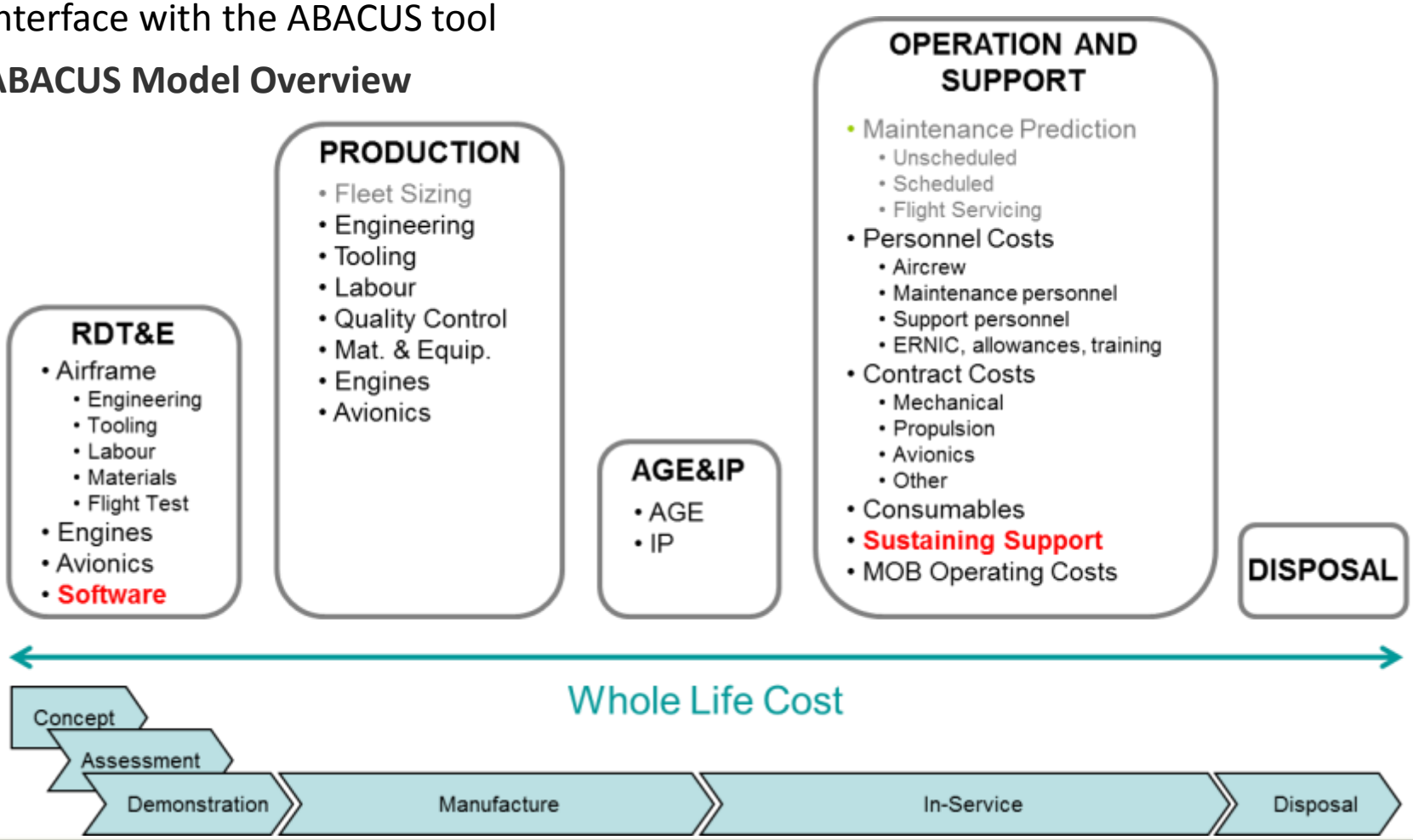


The data is based on a 2954 module sample from the technical environment of NASA unmanned spacecraft control, and has been treated as broadly analogous. This data also informs the estimation of re-use within the COCOMO II open source cost model, which has become an 'industry standard' used in many domains, including US and UK defence contractors.

Requirement 1

Develop a Mission Systems Software Cost Forecasting Model ("the MS model") to interface with the ABACUS tool

ABACUS Model Overview



The MS Model Requirement

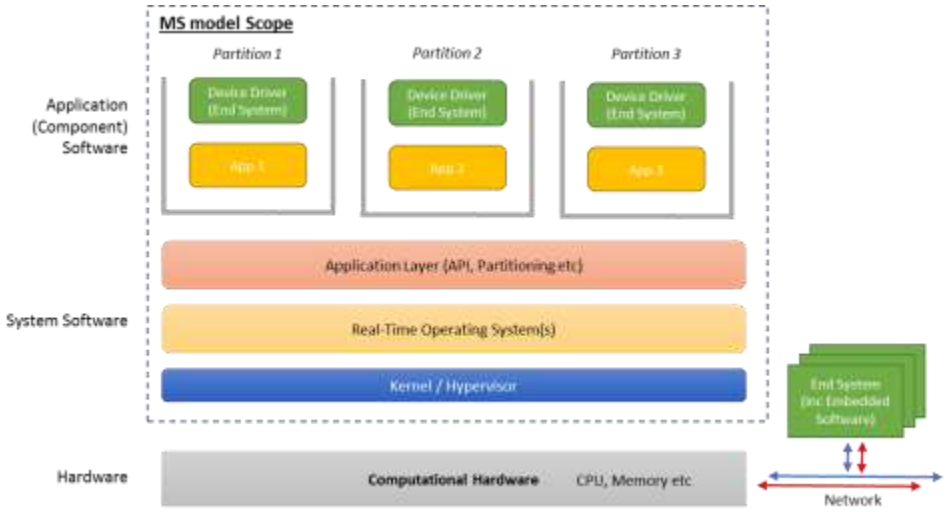
Key MS Model Requirements

- Can model differing levels of Autonomous operation (manned to unmanned)
- Can assess the of Impact of Open Systems on mission SW and potential for re-use across different A/C, roles and ground control systems
- Can accommodate uncertainty in input data (stochastic modelling)
- Can be supported by visible documented assumptions
- Can use simulation to develop the results compliant with ABACUS
- Can be updated as UK Unmanned Air Systems outturn software costs (SLOCs) emerge

The MS Model Scope and Scale

On-Aircraft Scope

- End systems Device drivers - end systems are discrete sub-systems, sensors, effectors etc;
- Application Protocol Interface (API) - application partitioning etc;
- Real-Time Operating Systems (RTOS) – code generated could have a high degree of reuse or be Commercial Off-The Shelf (COTS);
- Kernel and/or hypervisor – provides separation and partition of multiple on-aircraft operating system instantiations.



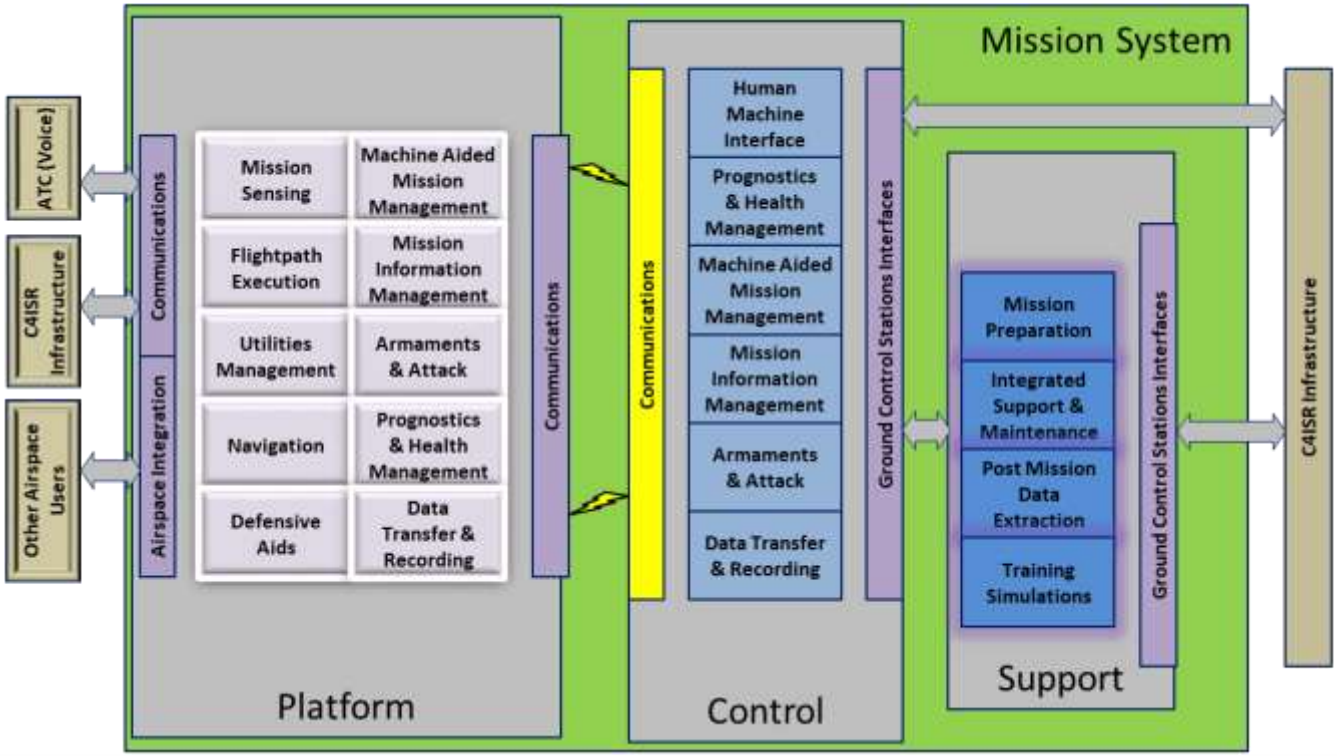
Ground Based Scope

- Mission Systems Software Applications (or Components);
- Underlying operating system(s);
- Kernel / Hypervisor;
- API layer inc. partitioning;
- Human Machine Interface (HMI) device software.

Mission Systems Scope (typical)

The mission system includes software components residing within the:

- Ground station / Unmanned Control System (UCS);
- The aircraft;
- Support software – e.g. Mission planning, Post Mission data extraction, maintenance and support, etc.



For manned aircraft the majority of the Mission System is assumed to be hosted on the aircraft.

MS Model Software Estimating Principles

- Iteration 1
  - The MS model was developed around the commonly used parametric based effort formula used by Off-The-Shelf models operated by the MOD (e.g. True Planning) using the following formula
    - Effort= A·⟦Size⟧^B•R
  - Where:
    - Effort is Person months
    - A is a productivity factor
    - Size is in KSLOC (thousand lines of code)
    - B is a scale (economy / diseconomy of scale based on software type
    - R is software reuse factor (1 + %saving of effort for x% reuse)
- Scale, Productivity and reuse factors used are specific to the use cases and requirements for the MS model.
- Iteration 2
  - The underlying equation and factors have been refined and developed further, based on data collected from open sources and UK specific projects and programmes.
  - The Size estimate within the Iteration 1 model was simple user input. Additional functionality has been added in Iteration 2 to support automated sizing (based on Aircraft characteristics), drawing on data collected from UK specific projects and programmes.

Key MS Model User Interactions and Interfaces

- Globals
  - Constants
  - CERs
  - Certification
  - Effort Profiling
  - Scale Drivers
  - Reuse
- Options
  - Aircraft Definition
  - Project Definition
  - Mission System
  - MS Reuse
- Calculation
- Simulation
  - Setup
  - Correlation
  - Analysis
- Outputs
  - Results
  - Simulated

Aircraft Definition

e.g.
- Type
- Automation Level
- Payload
- Op Env

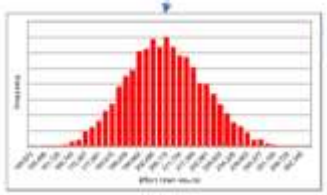
Project Definition

e.g.
- Experience
- Architectures & Standards
- Team

$$Effort = A \cdot Size^{B} \cdot R$$

Mission System

e.g.
- Reuse
- CER
- Safety Level
- Security Level



Outputs
- Monte Carlo Simulation
- Simulated 10th, 50th 90th

Levels of Autonomy?

Who needs pilots?

Levels of Autonomy

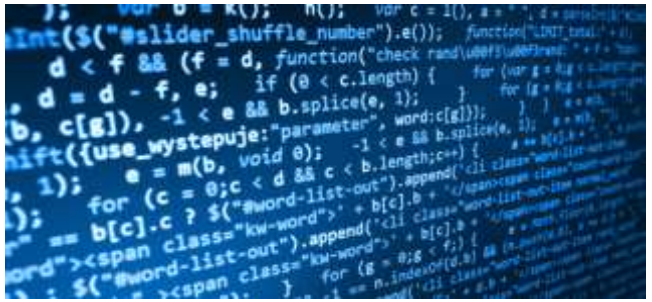
1. Human Operated (RPAS)
2. Human Assisted
3. Human Delegated
4. Human Supervised
5. Mixed Initiative
6. Full Autonomous (Note)

Note

Current UK policy is that the operation of weapons will always be under human control as an absolute guarantee of human oversight and authority and of accountability for weapon usage. The UK does not possess armed autonomous aircraft systems and it has no intention to develop them."

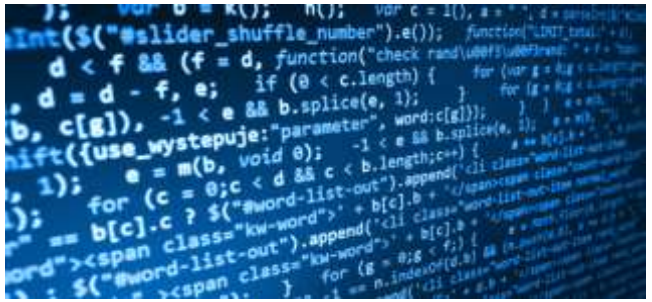
Who needs pilots?

Software



This software is:

- Highly Complex
- Safety Critical
- Real-time
- Learning
- Adaptive
- COSTLY



Any Questions?