18th Practical Software and Systems Measurement Users' Group Meeting and Workshops

"Measurement in a Complex Environment"

June 12-16, 2017 Arlington, Virginia

Software and Systems Measurement - UK Studies

Dr Antony Powell & Dr John Murdoch

yorkmetrics

Overview

UK MOD

- Issues in Defense Software Acquisition

PICASSOS

- Trials of Formal Methods in Automotive Systems

Approach

Insights

Questions

SECT-AIR

- Benchmarking Aerospace Software Practices

RR Metron

- Metrics Dashboards in Aeroengine Controls

MOD Software Acquisition

In 2004, the Committee of Public Accounts described the original procurement of the Chinook Mk2 Helicopter as "***one of the worst examples of equipment procurement***" that it had seen. (NAO Report on Chinook Mk3)

Software implicated in many NAO reports, Haddon Cave and DSAC studies.

What are the costs to MOD of software acquisition?

What's MOD's 'value at risk' of software acquisition?

Deliver evidence-based policy
(not policy-based evidence)

Project Name	IPA RAG
Complex Weapons	Amber
Operational Information Services	Red
Lightning II Programme	Amber
Successor SSBN	Amber/Red
Army Basing Programme	Amber/Red
The Materiel Strategy	Amber
Armoured Cavalry 2025	Amber
Armoured Infantry 2026	Amber
GRAPEVINE 2	Amber
Contracting, Purchasing and Finance	Amber/Red
EMPORIUM	Amber
Future Beyond Line Of Sight	Amber
GRAPEVINE 1	Amber
Astute Boats 1-7	Amber/Red
Army Reserve Development Programme	Amber/Red
Merlin Programme	Green
Core Production Capability	Amber
New Employment Model	Amber
A400M	Amber/Green
Airseeker	Amber/Green
MARSHALL	Amber/Green
Maritime Sustainment Programme	Amber/Green
PUMA	Green
Carrier Enabled Power Projection	Amber
Logistics Commodities Services Transformation	Amber
Crowsnest Programme	Amber/Green
CHINOOK (incl. Project Julius)	Green
Queen Elizabeth Programme	Amber
Wildcat Programme	Amber/Green
WATCHKEEPER	Amber
Spearfish Upgrade Programme	Amber/Green

Source: Ministry of Defence, Government Major Projects Portfolio data, September 2015

yorkmetrics

Method

Case #

Causes

Start of analysis

Causes

Software Problem

Scope, define problem

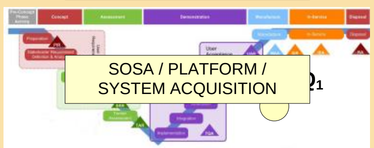
Locate problem WHERE, WHEN, WHO

Evidence of problem

Causes of Software Problem

time

OPS / SERVICES



SOSA / PLATFORM / SYSTEM ACQUISITION

SOFTWARE ACQUISITION

SUPPLY CHAINS

depth

Identified causes in sw acq and devlpt domains

Identified indicators/ metrics of cuases

Evidence of causality

Costly Event or Condition

Consequences of SW Problem: how is/ was the pain felt?

Types of Damage

Evidence of damage, if occurred; evidence of risks, if unfolding

Cost estimates of damages

Corrects or mitigates

Corrective Strategies/ Risk Mitigations

Identified corrective or mitigating actions

Evidence of beneficial effect

Estimated cost of actions

Implementation proposal, associated indicators/ metrics

informs

Justification for Investment in Improved SW Acq Capability

Planning the investigation

Issues in software acqusition management, as expressed by stakeholders

Select set of cases for study

Strategic Improvements

Pareto/ prioritisation of mitigation actions across cases

Coherence of interventions

Scaling of interventions

Opportunities for interventions

Time phasing of improvements

yorkmetrics

Findings - Causes

	Cause														
Initial	Requirements - Contract			■							■	■		■	
	Requirements - System	■				■		■		■			■		
	Architecture									■			■		
	Interfaces					■									
	Design		■			■		■	■	■					■
	Code	■			■			■				■			
	Application									■					
	Integration				■	■					■	■	■		
	Monitoring	■	■						■	■				■	■
	Software Performance				■	■						■			
	Estimation			■									■	■	
	Testing														
	SQEP	■	■	■	■	■	■	■	■	■	■	■			
	Compliance		■	■									■		
	Communications	■		■	■		■								
	COTS	■	■		■										
	Support														
	Project Management		■		■		■			■					
	Constraints	■	■	■			■						■		
	Budget	■		■			■				■	■	■	■	
Emergent	Requirements - Supply	■													
	Commercial	■	■			■									
	Risk Management	■	■	■	■								■		
	Complexity - Technical				■		■	■		■		■	■		■
	Complexity - Organisational				■	■	■	■				■			
	Complexity - International		■	■				■						■	
	Data Errors										■	■			
	System of Systems												■		

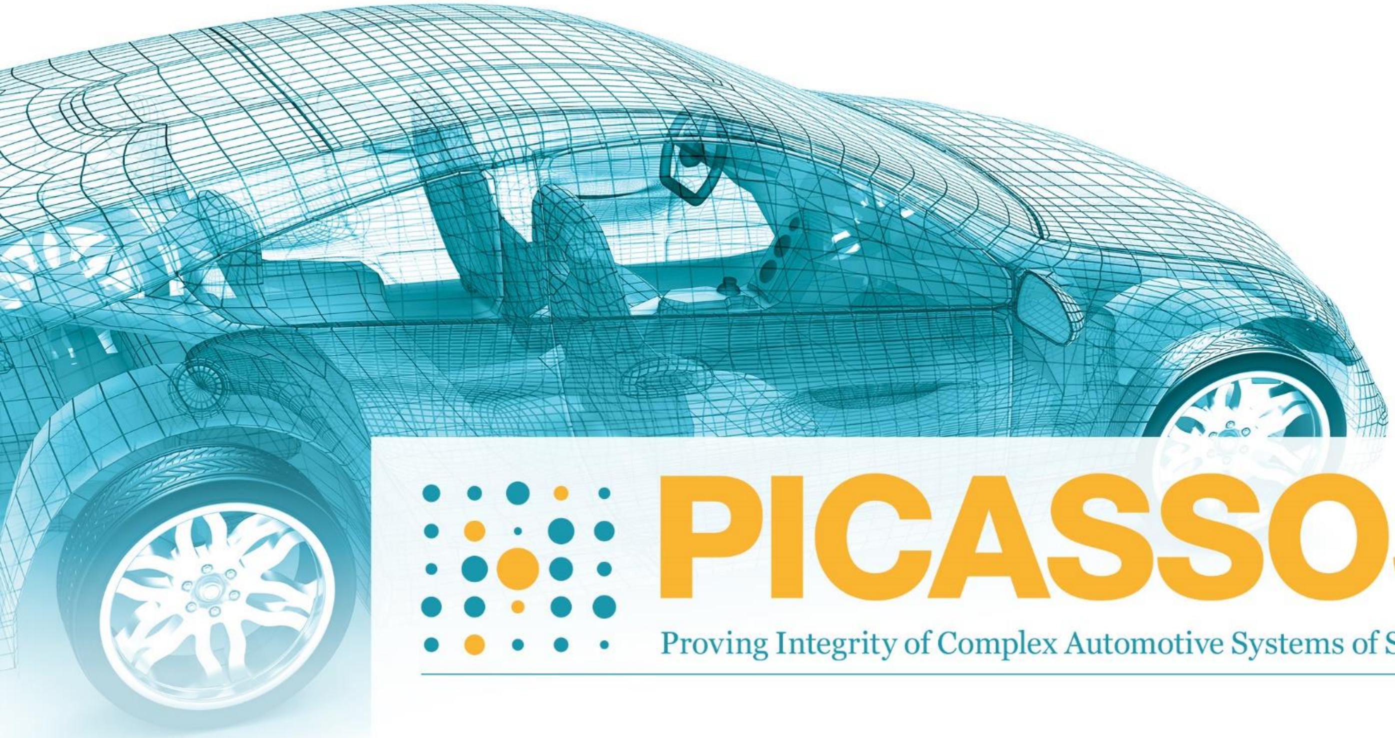
yorkmetrics

Findings - Effects

Effects														
Cost - Development		■		■	■	■					■	■	■	
Delays - Entry into Service					■	■			■		■	■		
Cost - Upgrade/Maintenance									■	■				■
Delays - Upgrade/Maintain									■					
Capability - Delivered								■			■		■	
Capability - Upgrade/Maintain	■								■					
Capability - Unavailability	■								■					■
Operation - Failure Event		■	■	■	■		■	■						
Security														
Compliance - Interface/Architecture					■							■		
Compliance - Safety/Security	■	■				■		■	■	■				
Performance		■	■	■	■									
Operation - Safety Event							■							■
Operation - Usability	■			■	■									
User Training														
Support Materials														
Information Access						■		■	■					
Cost - Through-Life														

Observations

- The broad cost to MOD of software-related problems is very high
- The combination of system and organisational complexity coupled with long project duration, staff turnover, project re-baselining and little metrics/data makes this problem difficult and time consuming to analyse, understand and reliably measure
- Causes identified tend to lie throughout the product specification and software development and acceptance lifecycle
- A failure at one point anywhere in the lifecycle can make all other good work worthless (“weakest link” issue)
- Undertaking a process or action without adequate understanding of the benefits and consequences will not reduce risk



PICASSOS

Proving Integrity of Complex Automotive Systems of Systems

Proving Integrity of Complex Automotive Systems of Systems

www.picassos.info

PICASSOS Objectives

- Introduce new (formal) verification methods
- Improve coverage
- Automate regression testing
- Use in early development phases
- Use at a system level
- Provide verification evidence to establish standards compliance
- Practical

PICASSOS
Proving Integrity of Complex Automotive Systems of Systems

Questions About New Verification Method

- **Assessment of engineering benefits**
  - Increased confidence/assurance in requirements, design specifications and as-built products; reduced residual risks in work flow products and in the final product?
  - At OEM, Tier 1 and supply-chain levels?
  - Generation of evidence compliant with ISO 26262?
  - Improved capability to develop more advanced products at similar cost/residual risk?
- **Assessment of process benefits**
  - Reduced overall development / lifecycle costs, schedules?
  - Reduced overall safety process / ISO 26262 compliance costs, schedules?
  - Reduced inter-working costs, schedules in supply chains?
- **Assessment of potential costs**
  - Additional effort to apply new methods? Who takes the costs, who benefits?
  - Preparatory training and specialist support costs?
  - Costs of transition from existing work flows; modification of other (upstream and downstream) activities?
  - Tool licensing and support costs?
  - Tool qualification costs, where needed?

Trials Process

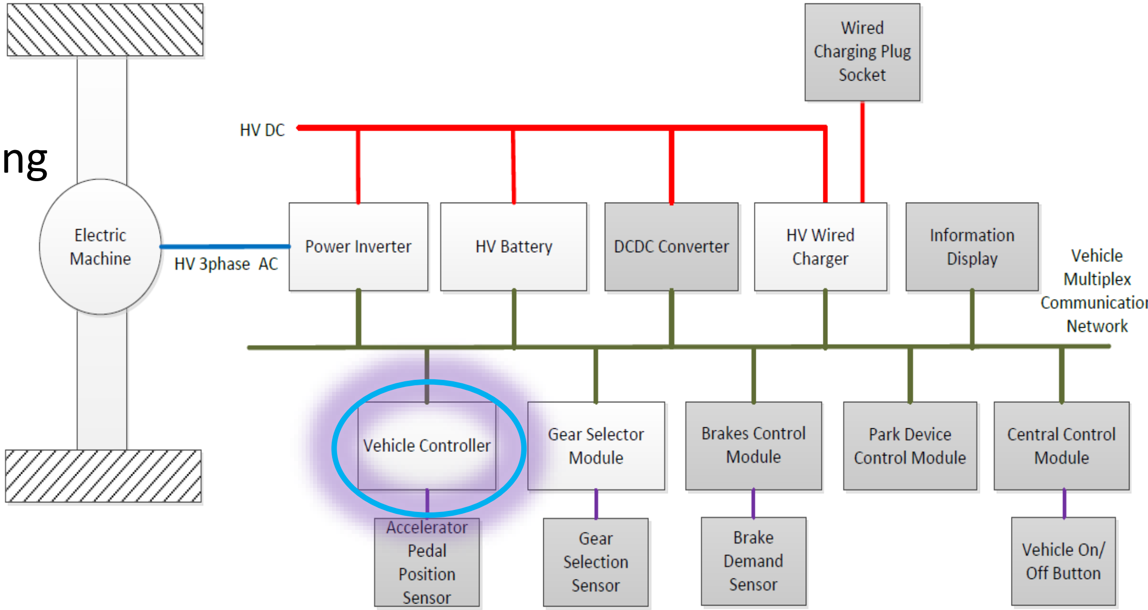
Trials conducted to enable comparison between:

- A **baseline process**, representative of current methods and tool chains
- **Delta processes**, which include the new methods/tools to be investigated.

An example product was taken through these processes. Error seeding is used as part of the testing of the effectiveness of the baseline and delta verification methods.

The software level trials looked at 6 software modules in the vehicle controller related to charging

These were defined by 83 requirements and their implementation took approx. 500 SLSF blocks



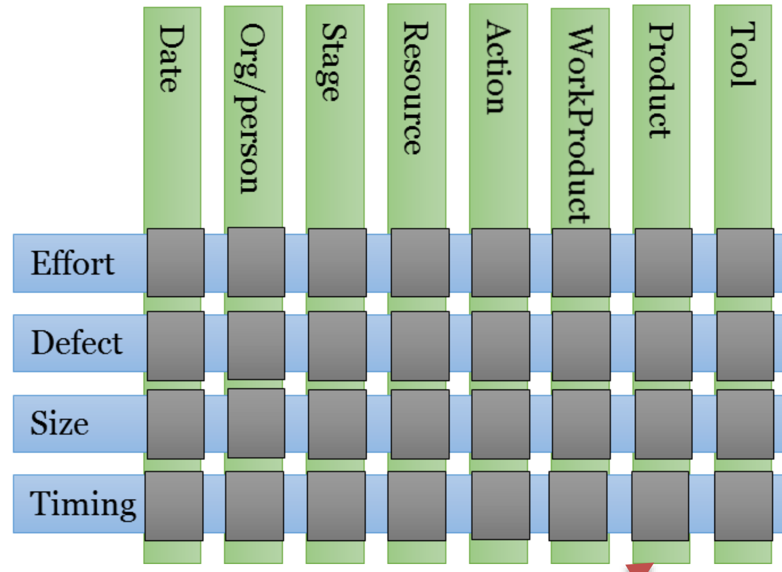
Measurement

Context-based Measurement

- Fine-grained context of activity
- Taxonomies evolve over time
- Allows post-hoc interpretation

Integration of results

- Intersection across metrics
- Quantitative and qualitative



Date	Org	User	Stage	Action	Process	WorkProduct	Product	Tool	Hours	Phase
02/02/2017	Org C	Person D	T1 Delta1.4(MWv4)	Produce	Model Formal Verification	Intermediate Language Requirements	SW Feature CL	MS Word	0.67	MW SWRs Translation
15/02/2017	Org C	Person D	T1 Delta1.4(MWv4)	Produce	Model Formal Verification	Intermediate Language Requirements	SW Feature CL	MS Word	0.53	MW SWRs Translation
15/02/2017	Org C	Person D	T1 Delta1.4(MWv4)	Perform	Model Formal Verification	Unit Test Setup - host	SW Feature CL	Mathworks Model Advisor	0.37	MW Configuration
15/02/2017	Org C	Person D	T1 Delta1.4(MWv4)	Produce	Model Formal Verification	Unit Test Setup - host	SW Feature CL	Mathworks Simulink	0.17	MW Configuration
15/02/2017	Org C	Person D	T1 Delta1.4(MWv4)	Produce	Model Formal Verification	Unit Test Setup - host	SW Feature CL	MS Word	0.52	MW Configuration
15/02/2017	Org C	Person D	T1 Delta1.4(MWv4)	Modify	Model Formal Verification	Unit Test Setup - host	SW Feature CL	MS Word	0.32	MW SWRs Translation
15/02/2017	Org C	Person D	T1 Delta1.4(MWv4)	Perform	Model Formal Verification	Unit Test Setup - host	SW Feature CL	Mathworks Model Advisor	1.32	MW Configuration

PICASSOS
Proving Integrity of Complex Automotive Systems of Systems

Results - Defects

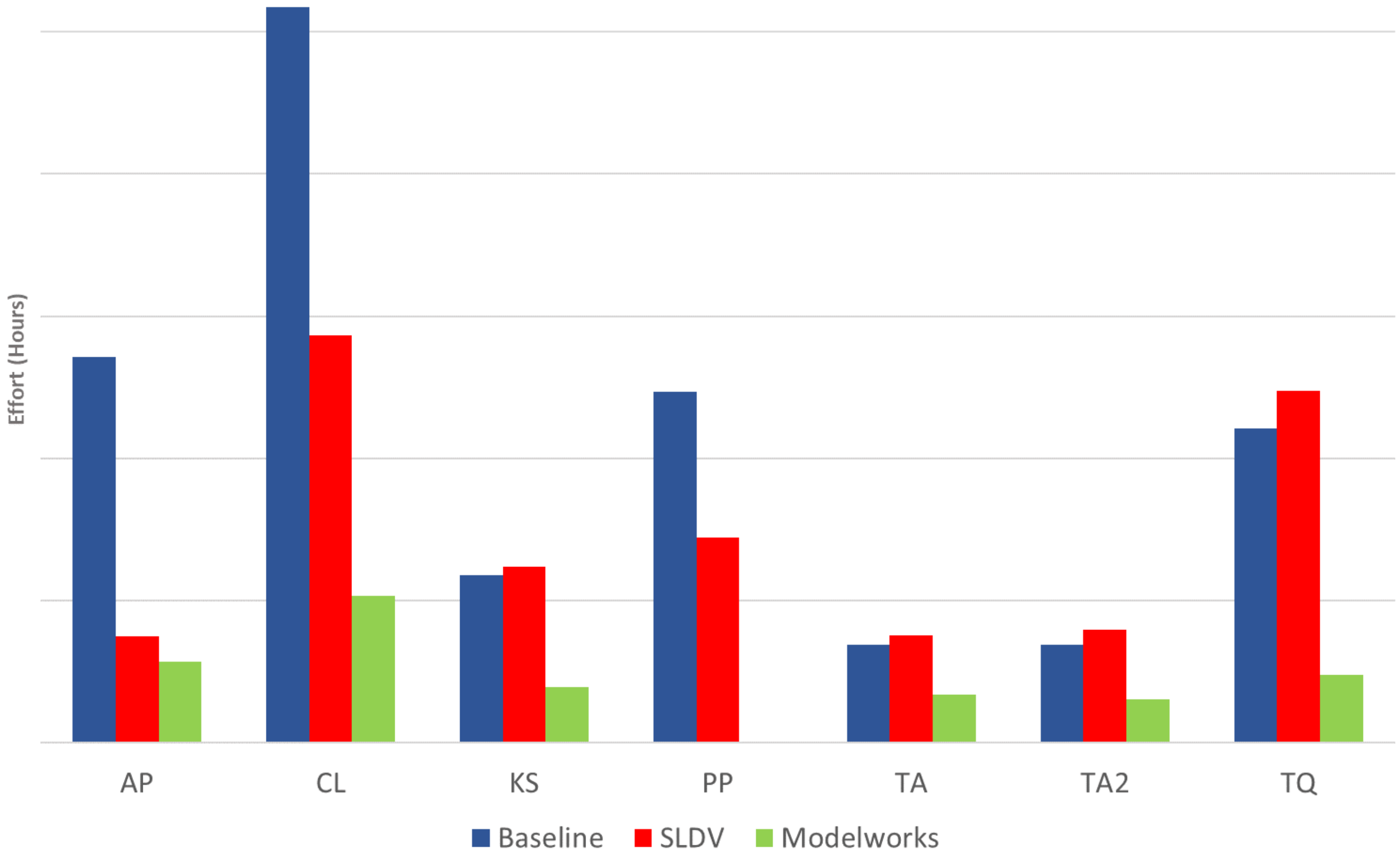
Venn diagram of Errors Found

- The error found only by the Delta process (using model checking) was a "real error" (it was not introduced by the error seeding)
- The error found only by the Baseline process is best described as a "Missing requirement" (this was also not error seeded)

PICASSOS
Proving Integrity of Complex Automotive Systems of Systems

Results - Effort

Comparative Effort per Model for Verification of Requirements Satisfaction



SECT-AIR:

Aerospace Software* Cost and Timescale Reduction

Dr Antony Powell, YorkMetrics
Dr Mike Bennett, Rolls-Royce
Prof John McDermid, University of York

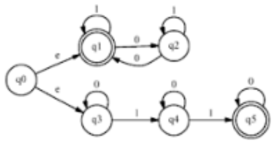
*and complex electronic hardware

SECT-AIR

"To deliver a step-change improvement in the affordability of aerospace software. This is required to secure and develop the UK as a world leader in critical and complex systems development and enable UK aerospace to build new products."

yorkmetrics

Technical Themes



More formal, modelled and less ambiguous requirements

Assess current industry state and develop roadmap

Develop UK aerospace guidance

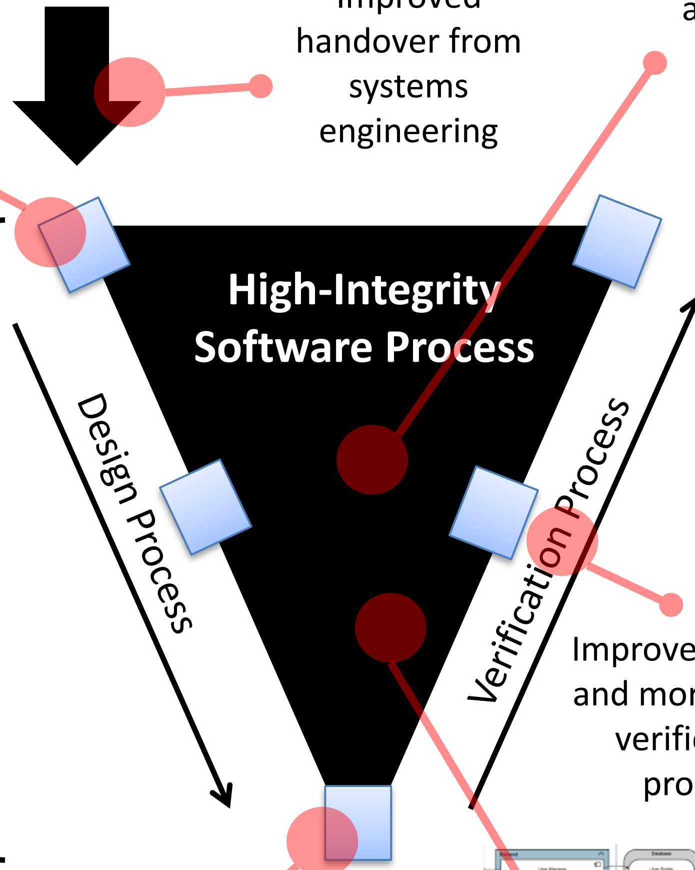
Create forum for ongoing exchange of best practice

Metrication

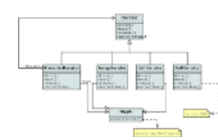
Improved and lower-cost certifiable hardware

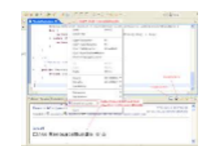
Improved handover from systems engineering

Automated translation of artefacts throughout the lifecycle.

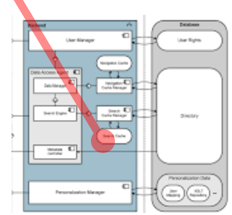


More intelligent, integrated and lower cost tooling





Improved testing and more formal verification process



More flexible and modular architectures

yorkmetrics

SECT AIR Benchmarking Goals

- To generate and share information about performance levels in software development, support and outsourcing, in order to:
  - To support sharing of ‘good practice’ information
  - Enable judgements to be made about current performance
    - of a project, enterprise or supply chain
  - To support possible business cases for improvement
  - To support competitive innovation in product design and processes
  - To improve supply chain performances and visibility for end-customers
  - To minimise supply chain partnering control and cooperation costs
  - Improve accuracy of estimation.

Benchmarking Task

An iterative method has been adopted, involving the seven participating firms

- Round 1 – questionnaire – baselining
  - Spread of responses
    - Diversity of practices
    - Legacy vs current projects
  - Difficult to discern a way forward
    - Different starting contexts
    - Different opportunities

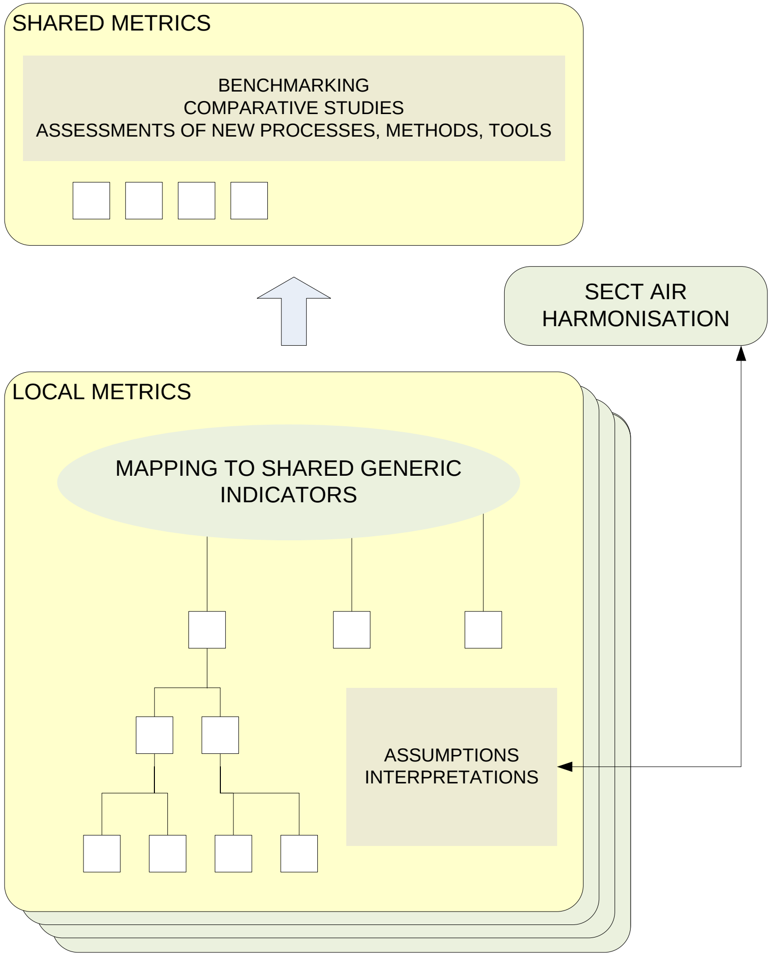
- Round 2 – initial interviews
  - Understanding the 'dimensions' of local context
  - Parallel efforts: (1) support local measurement, respecting the particular context, (2) 'translate' to shared metrics that take into account the local differences and enable meaningful comparisons/ benchmarking
  - Develop measurement guidance that is sensitive to local contexts

- Round 3 - local data sharing plus translation to shared common metrics for comparison purposes

Dimensions of Local Context

- Performance is interpreted and measured differently in different settings across several 'dimensions':
  - Productivity
  - Schedule
  - Quality
  - Agility/responsiveness to change
  - Innovation

- These differences:
  - shape local business cases (for a process/method/tool improvement)
  - have to be taken into account when comparing performances between teams



yorkmetrics

Early Findings

- Firms exhibit a range of current process 'starting points'
  - Implying different opportunities to achieve benefits
  - Mixed capability maturity and measurement across projects
  - Different engineering and organisational contexts
  - Notable difference in software management and leadership
  - Software treated variously as an opportunity or a cost/risk
- Software metrics
  - Mixed picture – some used effectively, others patchy
  - General opinion was "*we could be doing it better*"
  - Often lots of collection but little effective usage
- Good practices identified including
  - Strong leadership and voice for software in wider enterprise
  - Evidence-driven trade-offs within enterprise decision making
- Different types and degrees of Software-Systems team interaction
  - Some 'over the wall', others co-located, cross-functional
  - Clearly some change from MBSE and Agile

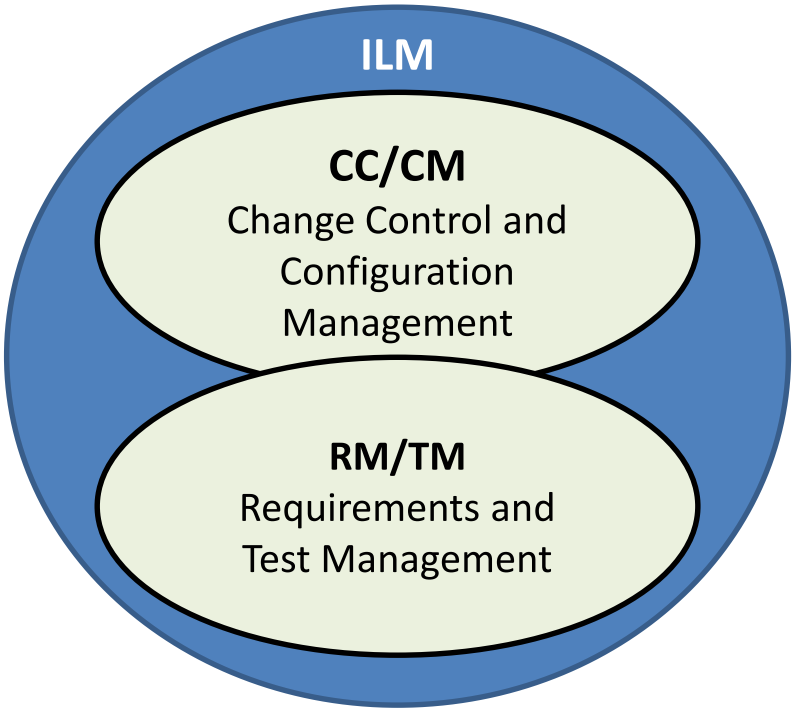
Metron

Greg Holland

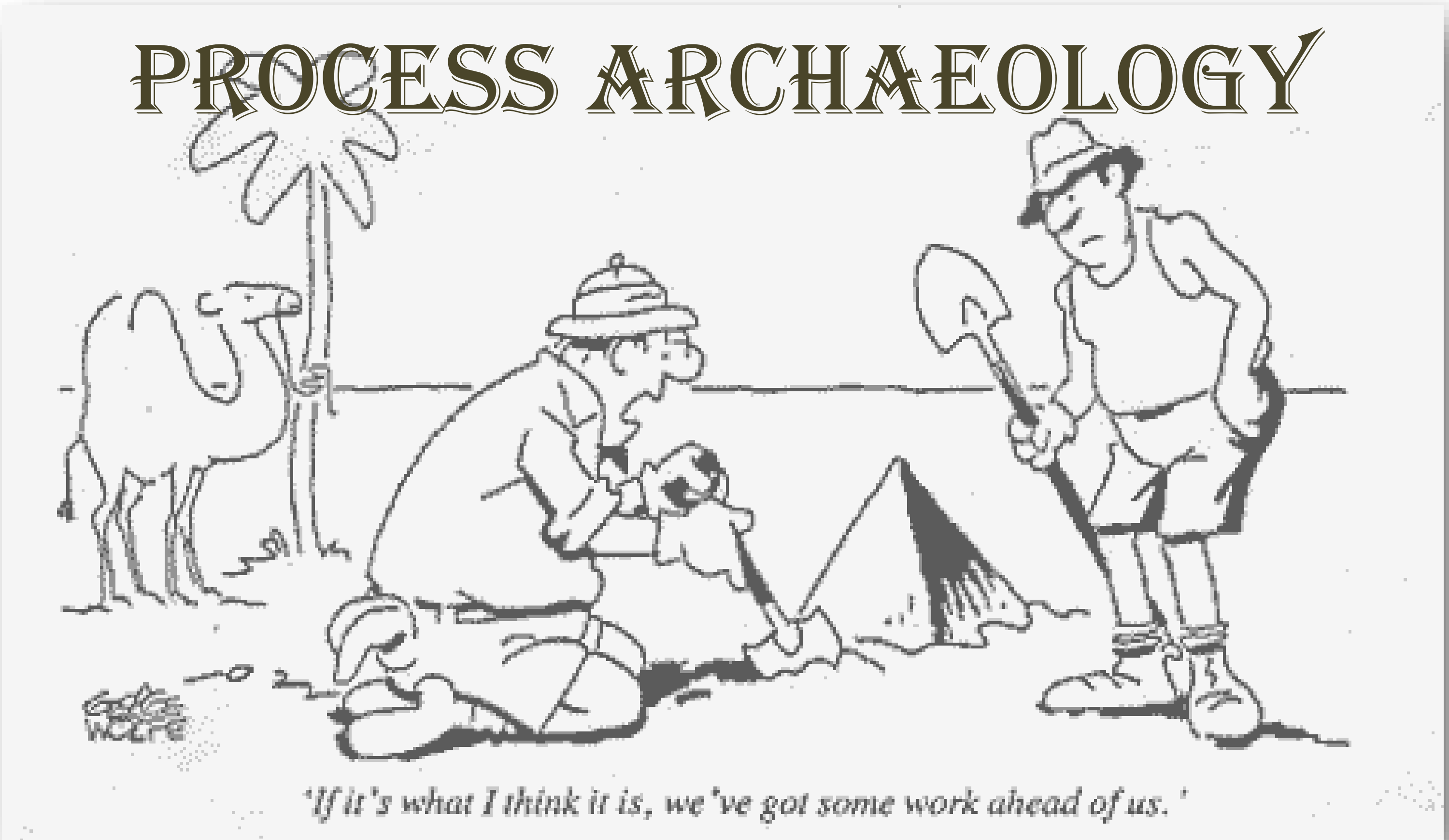
ALM (Application Lifecycle Management) Lead

Our ALM solution

- **Integrity LM is a COTS tool provided by PTC**
  - PTC also provides the Modeler (Artisan) modelling tool
- **We have customised Integrity to accommodate our processes**
- **Integrity comprises two principal components: CC/CM and RM/TM**



Uncovering reality

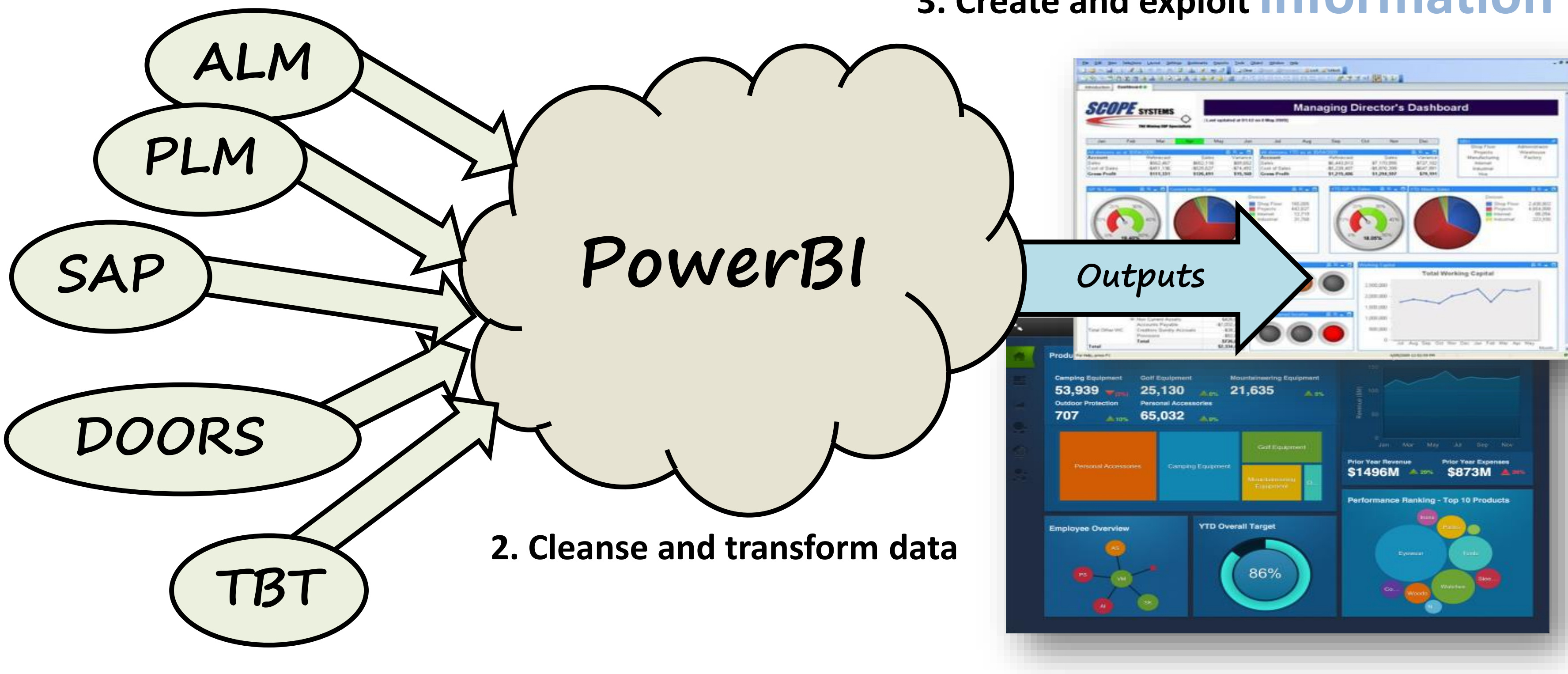


Integrity RTM enforces process rules

Rolls-Royce

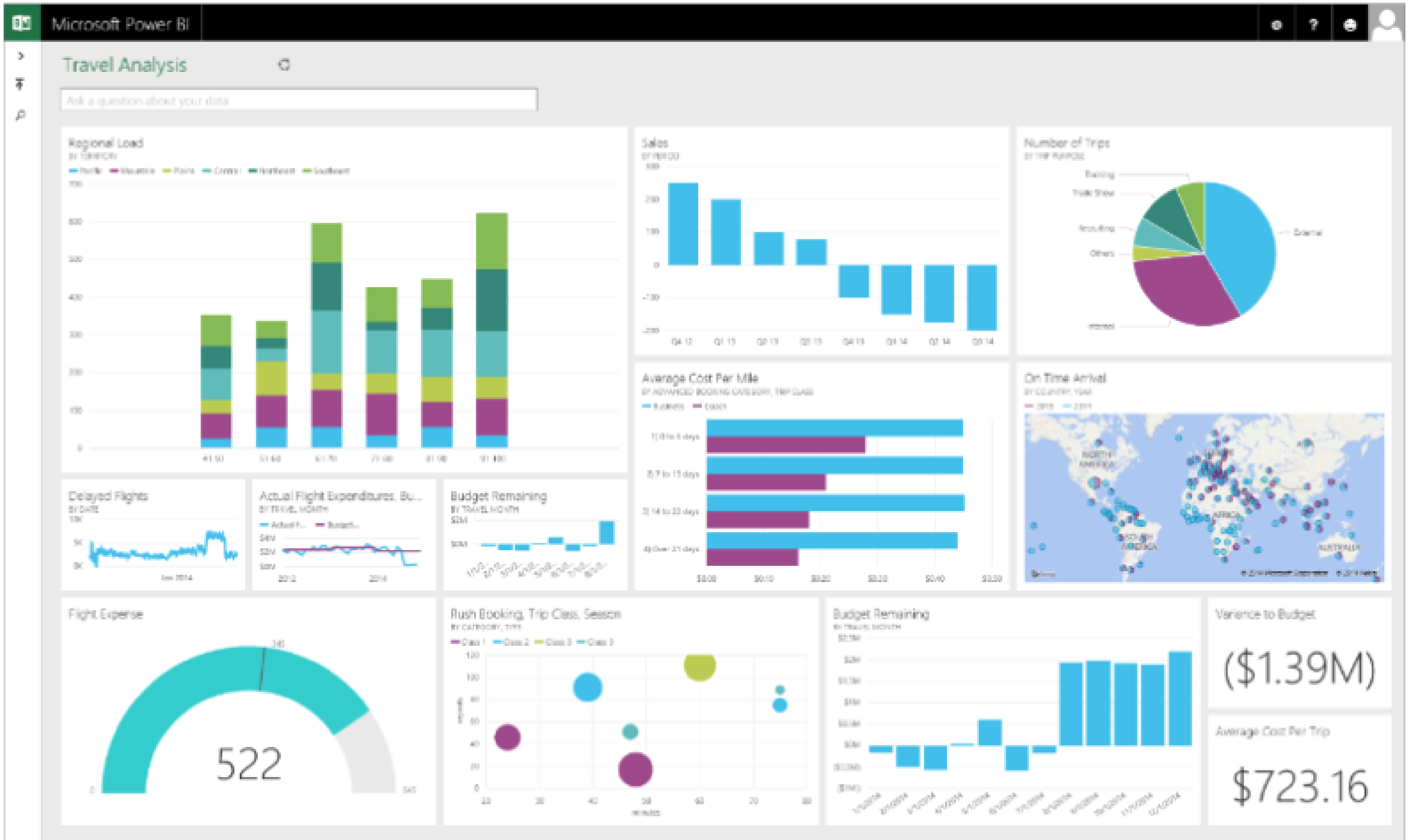
1. Harvest data from tools

3. Create and exploit information



2. Cleanse and transform data

Example dashboard (from a quick web search)



Conclusions - Trends

- **Trends in Software Acquisition**
  – Period of depletion of software capability in acquirer organisations, shrinking budgets, but with ongoing responsibilities
  – Issues include:
    - Achieving sufficient visibility of development efforts in supply chains
    - Market / COTS/ supply-side standardisation
    - Evolving practices, tools in supply-side system and software development
    - Through-life ownership requirements and costs
- **Trends at Integrator Level**
  – Reliance on major system suppliers but retaining integration & test and compliance responsibilities
  – Evolving standardisation & practices at system & component supplier levels
  – Market competitive pressures
  – Industry conformance with safety standards (e.g. ISO 26262)
  – Evolving system technologies, hybrid, electric vehicles, more automation
- **Trends in Aerospace & Defence Software**
  – Interest in benchmarking and learning between firms

Conclusions - Issues for Measurement

- In supply chain settings, arranging for measurement information to be required and provided between parties, with sufficient trustworthiness for the acquirer
- Evolving size and model bases on which cost is to be estimated
- Measurement to support cost estimation and value for money assessments in compliance/assurance activities
- Measurement applied to /embedded in model-based tool chains
- Measurement of model checking, testing and related assurance methods and tools
- Measurement information communicated in supply chains
- Establishing the bases for comparison between measurement data collected in different local contexts
- Establishing the bases for comparison between causal models to support improvement actions in different local contexts

Conclusions – The Future of Software and Systems Measurement

- Supply chain collaborations
  - Open
  - Honest
  - Importance of measurement & evidence
  - Across boundaries

- Shifts Underway
  - Methods
  - Model-driven
  - Formal methods
  - Agile
  - Organisations
  - Growth and Complexity

18th Practical Software and Systems Measurement Users' Group Meeting and Workshops

"Measurement in a Complex Environment"

June 12-16, 2017 Arlington, Virginia

Workshop:
The Future of Software & Systems Measurement

Dr Antony Powell, Dr John Murdoch

antony.powell@yorkmetrics.com
john.murdoch@yorkmetrics.com

YorkMetrics Ltd
Moyola House
York
YO31 7YJ UK

Tel: +44 3333 03 111 9

yorkmetrics