COUNCIL OF MILITARY EDUCATION COMMITTEES OF THE UNIVERSITIES OF THE UNITED KINGDOM

THE CHALLENGE OF COMPLEXITY

By David Bradley, Edward Simpson & David Dawson

NOVEMBER 2020

COMEC OCCASIONAL PAPER. No 13.

The Challenge of Complexity

The Role of Design Tools in Establishing & Managing Functional Outcomes

David Bradley[†], **Edward Simpson**[†] **& David Dawson**

An occasional paper for presentation to Tayforth Universities MEC

Executive Summary

Systems of all types are becoming increasingly complex and depend for their operation on the integration of a wide range of technologies, often with differing development lead times. This leads to particular issues and complications in relation to the design, development and implementation of large scale military systems where the time from the initial concept to entry into service may extend over one or more decades. This means that concept development and planning must inevitably be based on predictions, themselves based on a range of assumptions, as to the nature of the in service role.

Not only does this place a particular emphasis on the requirements capture to get things right, and hence be able to do things right, it also brings into question a project management structure in which individuals are, for reasons of career progression, rotated into and out of roles over what are relatively short time periods in relation to the overall project timeframe[‡].

This paper therefore has its origins in a series of discussions as to the role formalisation of the design process might have in relation to the procurement of military systems. This led to questioning whether it would be the case that a better use of resources could be achieved if the design process; its role, operation, methods, procedures and limitations, was better understood, particularly in ensuring that user needs are properly, and fully, expressed and understood by all parties. Such an understanding is essential if the resulting systems are to match and conform to real user needs, rather than those needs as imagined by the designer and manufacturer, issues which involve not only technical considerations but issues of diplomacy, empathy and, in particular, trust.

The paper therefore sets out to set out and describe the main features of the early stages of the design process and their impact on that process. Particular consideration and emphasis is given to the approach to requirements capture and the elicitation of actual, as opposed to perceived, user need, and the integration of these within an overall system structure which includes issues of maintenance, refurbishment, replacement and upgrading over the service life of a system. This to include the use of formal methods in support of the design process along with a consideration of the costs associated with remediation and change and the role of gatekeepers in managing technological progression and innovation in the context of the project.

Other design oriented factors included in the discussion are the need to future proof designs by ensuring that a programme of refurbishment, upgrade and replacement is integrated within the design, and that that is supported by mechanisms for evaluating and supporting the introduction of new technologies such as those based around the concepts of Cloud computing and the Digital Twin.

It is also the case that whatever approach is adopted it must of necessity involve a consideration of the career pathways of service personnel, balancing these with the need for a continuity of oversight across the entirety of service life of the systems.

May 2020

[†] David Bradley & Edward Simpson are both members of the Tayforth Universities MEC.

[‡] Eliasson G (2017) *Visible Costs & Invisible Benefits: Military Procurement as Innovation Policy*, Springer

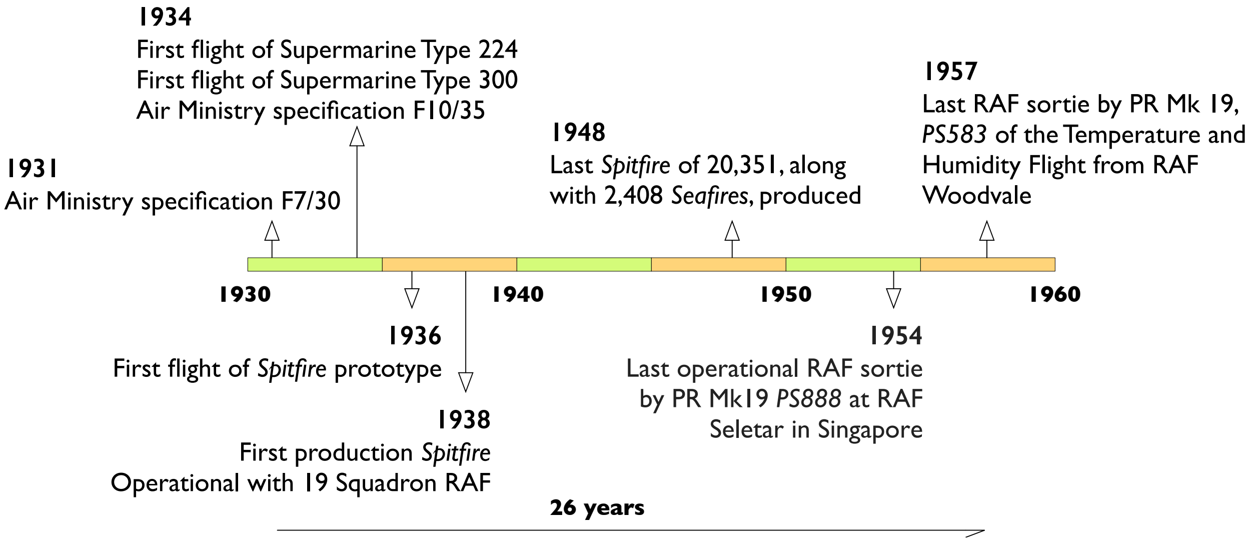
The Challenge of Complexity
The Role of Design Tools in Establishing & Managing Functional Outcomes

David Bradley†, Edward Simpson† & David Dawson

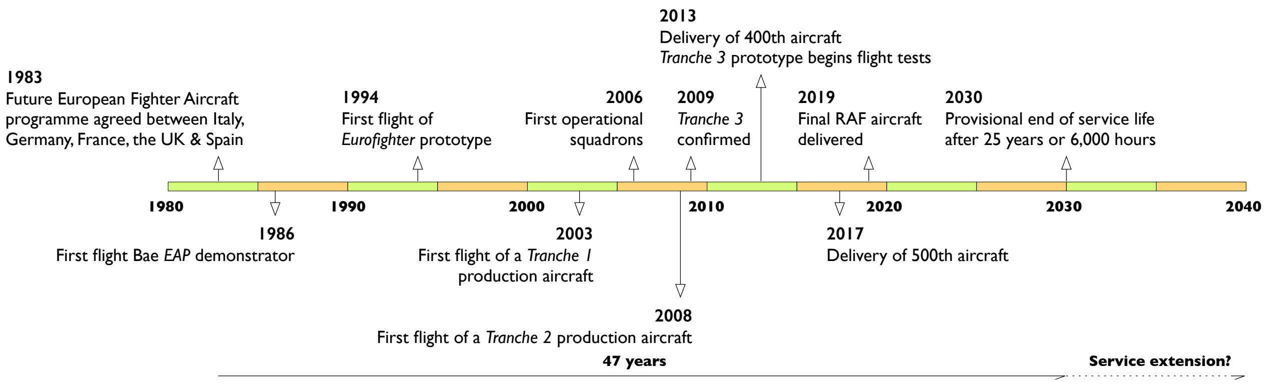
An occasional paper for presentation to Tayforth Universities MEC

Introduction

As systems become ever more complex with development lead times in some cases extending over several years, designers and clients are faced with the problem of developing systems to meet unknown goals associated with future, and generally unforeseen, events using technologies which in some cases have either not been developed or which were intended for a radically different purpose[1]. In this context, consider first Fig. 1 which shows the development and service timelines for the Supermarine *Spitfire*, a total time span of 26 years, and the Eurofighter *Typhoon* with a time span of 47 years plus which includes a period of 23 years from the initiation of the programme to the aircraft entering service. Figure 2 then gives an indication of the rate of change of technical development over the period from 1970 to 2020, from which it can be seen that the most significant advances in technology have been those associated with computing and information technologies followed by electronics and the associated hardware. Indeed, it was only from the mid-1960s on that systems such as fly-by-wire became practical or possible as a consequence of technological developments in electronics and computing.[2]



(a) Supermarine *Spitfire*



(b) Eurofighter *Typhoon*

Figure 1: Development and service timelines for Supermarine *Spitfire* & Eurofighter *Typhoon*

Taken together, the situation illustrated by Figs 1 & 2 places a burden on both the user and designer to approach their respective tasks so as to be able to accommodate the changes that will inevitably occur in projects with such long lead times. This implies a need to future proof the resulting systems once in service by supporting the introduction of upgrades, which may themselves take the form of either or both of replacement and enhancement. It also means that certain elements of the design, as for instance the airframe in the case of an aircraft or the hull form for ships, will be

† David Bradley & Edward Simpson are both members of the Tayforth Universities MEC.

frozen relatively early on in the design process to allow construction to begin while other areas such as the electronics and software components may well still be under development. This means that project costs may be highly vulnerable as a result of the need to accommodate change[3].

In this context, consider the situation as described by Figs 3 & 4. Figure 3 shows in general form the relationship between the allocation of project costs and the actual spend, from which it can be seen that the Requirements Capture & Analysis and Design & Development phases will typically determine the allocation of some 80% of the overall project costs, while being responsible for spending only some 20% of these[4]. While these norms are those generally associated with successful product development, it needs to be understood that a failure to invest properly in the requirements capture and design and development stages of a design oriented project is likely to impact significantly on the success of the resulting system. It also needs to be pointed out that costs in these early stages are often considered as invisible costs and hence are subsumed in general overheads rather than being allocated to specific projects when, by doing so, a more effective control can be maintained over the project.

Figure 4 then shows the cost of remediation of errors. Thus for the remediation of an error associated with Requirements Capture & Analysis which remains undiscovered until the start of Production, the cost will be of the order of 100 times greater than if it had been discovered during Requirements Capture & Analysis, resulting in significant cost overruns. In illustration, consider the issues involved when a decision was made to modify the design of the *Queen Elizabeth*-class aircraft carriers to accommodate a catapult system[5], a decision eventually abandoned because of the cost and construction delays that would have been incurred.

It should also be noted that the conditions of Fig. 4 apply equally to sub-assemblies and sub-systems as well as to the final, operational, system and that attempting to introduce a new technology into the design has the potential to impact on costs in a manner similar to error remediation depending on how and when that new technology is introduced.

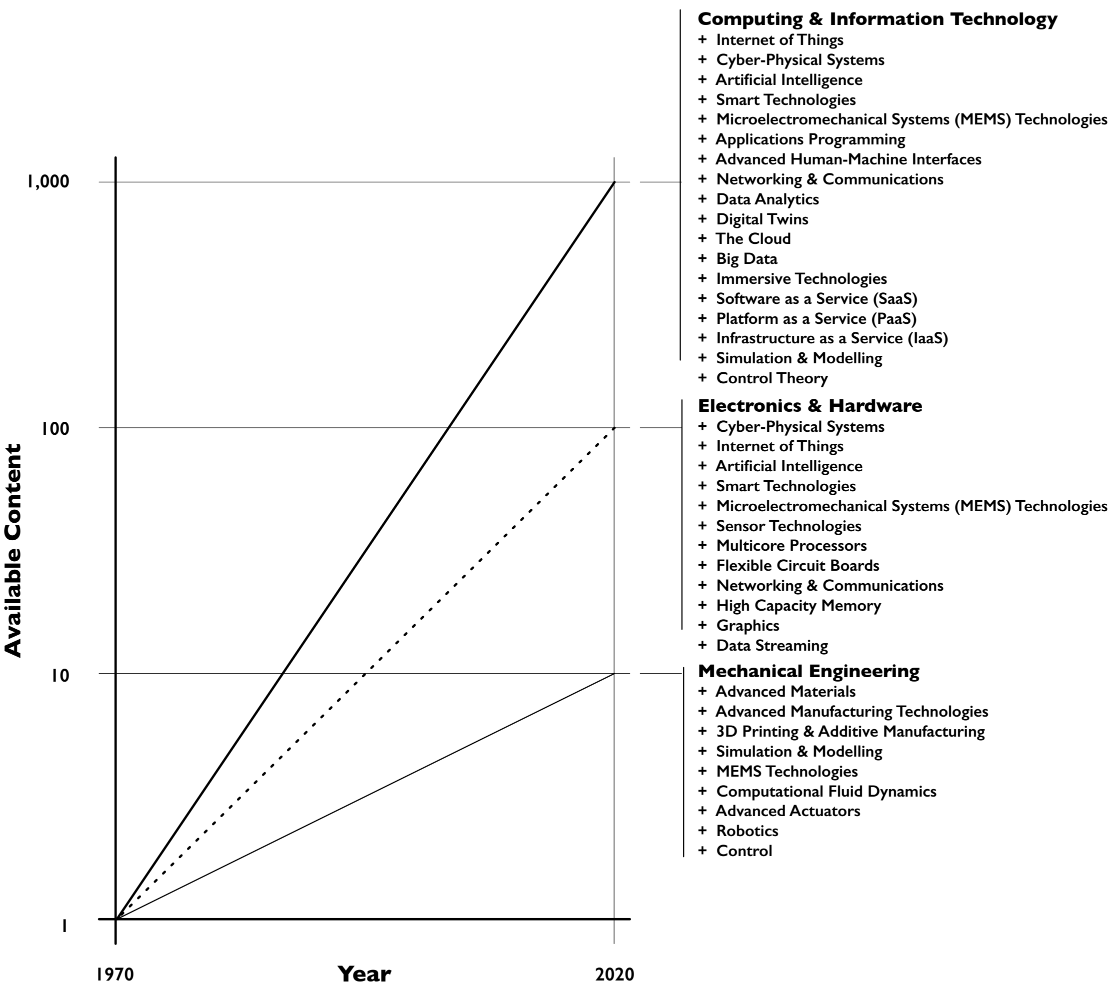
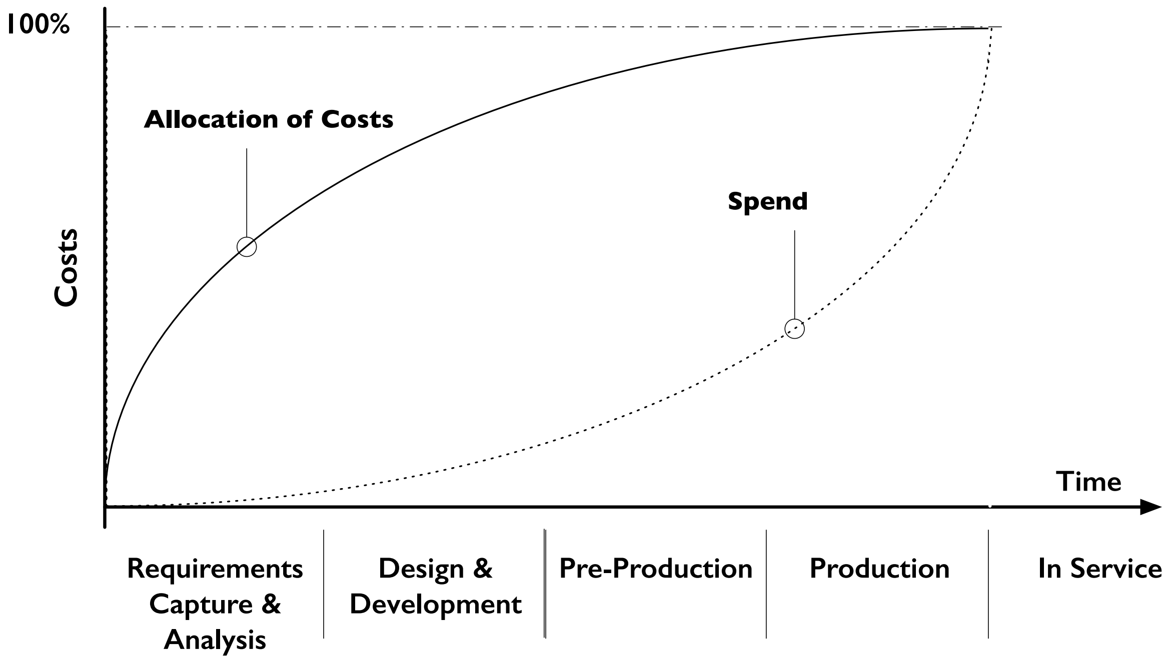


Figure 2: Developments in technology



3: Allocation of project costs in relation to project spend

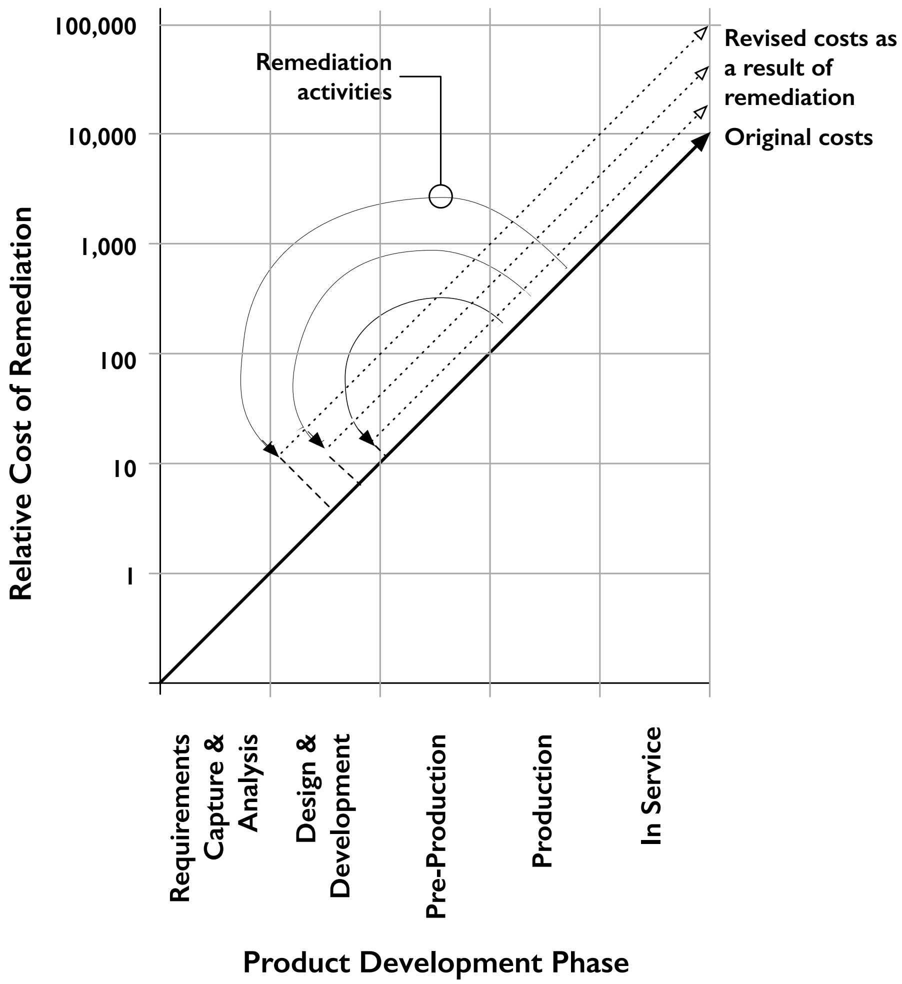


Figure 4: Cost of remediation[6]

For military procurement the requirements of operational systems are likely, to a very significant degree, to have been determined by the perception at the time of project inception of the likely threat or threats to be encountered on the system entering service. This then presents a significant challenge to the requirements capture process, particularly in relation to the choice of technologies.

Consider therefore Fig. 5 which sets out Technology Readiness Levels from observation of basic principles to readiness for full scale deployment[7]. This transition will typically be associated with a time span of some 30 to 50 years, as is observable across a wide range of technologies[8]. For instance, in the case of computers, Alan Turing published his paper *On Computable Numbers* setting out the basic structure of a digital computer in 1936[9] while the first IBM personal computer was not produced until 1981. In relation to space flight, Tsiolkovsky published *The Exploration of Cosmic Space by Means of Reaction Devices* in 1903, 66 years before *Apollo 11* reached the Moon![10] These examples serve to make the point that decisions about the viability of a technology may need to be made early on in the design phase, and the initial technological constructs then frozen in line with those decisions.

Referring again to Fig. 5, this suggests that for a technology to be considered as viable for the first systems to enter service then at the inception of a build phase extending over several years that technology should be at least at level 5 on the readiness scale. and preferably higher. This then implies a necessity to future proof systems by incorporating options for upgrade, refurbishment, refit and replacement over the life of the physical structures that they support. Thus in the case of a ship or an aircraft, while the physical structures such as a hull or fuselage may have an in service life of decades, the equivalent for the embedded technologies may be only a few years, as for instance in the case of a computer operating system, implying a regular programme of refit[11].

Now consider the V-Model of Fig. 6 which sets out in a simple diagrammatic form the basic system development processes[12]. Embedded within basic model, but not shown, are design oriented issues such as manufacturing methods and technology, procurement, supply chain and maintenance. A key feature of the model is that it emphasises the requirement that the evaluation processes associated with the Test & Integration leg need to be considered and defined during the Definition & Design phase, and that this should then include the future-proofing strategies to be adopted over the life of the system.

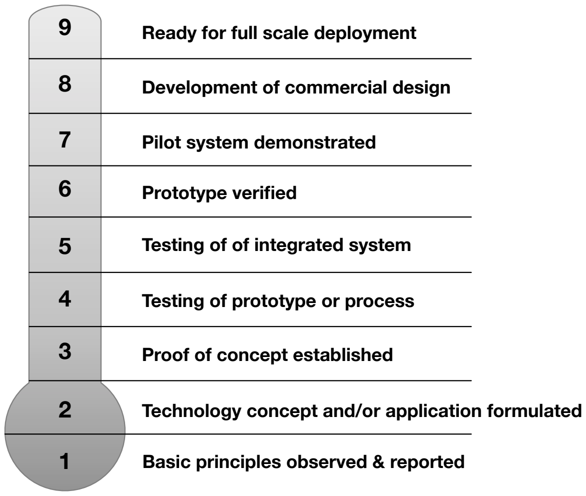


Figure 5: Technology readiness levels[7]

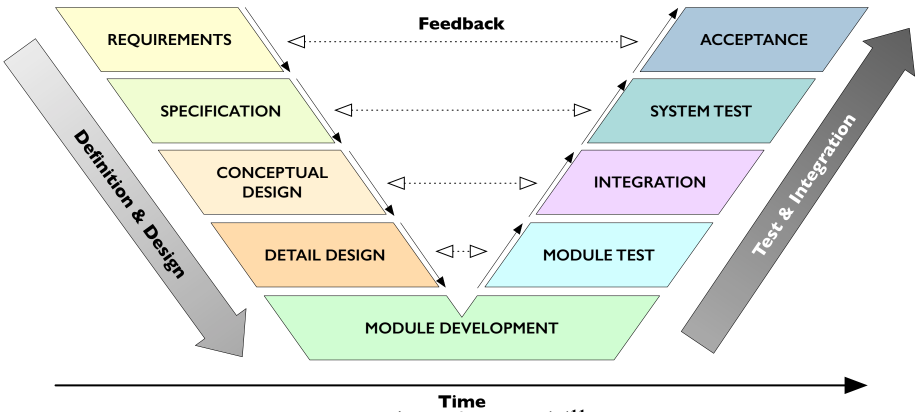


Figure 6: V-Model[11]

Omitted from the above discussion, but of significance to the overall outcome, is the political context. As the process of translation from concept to entering service is likely to extend over several political cycles which may involve different political parties who may wish to change the operational, and hence the functional, parameters associated with the system when it enters into service. While it is unlikely that a change in political direction can be accommodated in the initial stages of the design process, there needs to be an awareness of the potential for such political impact over time.

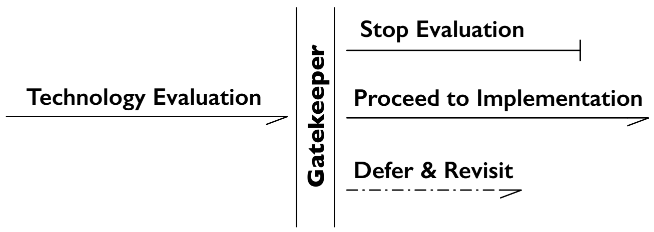
Gatekeepers

In a dynamic and evolving technological environment the role of the gatekeeper assumes significance in managing the implementation process. At the most basic level, the gatekeeper task is to act as a gateway in the technology evaluation process as suggested by Fig. 7(a) where the gatekeeper seeks to determine which technologies are to be taken forward to implementation. There are typically three basic options associated with this process namely:

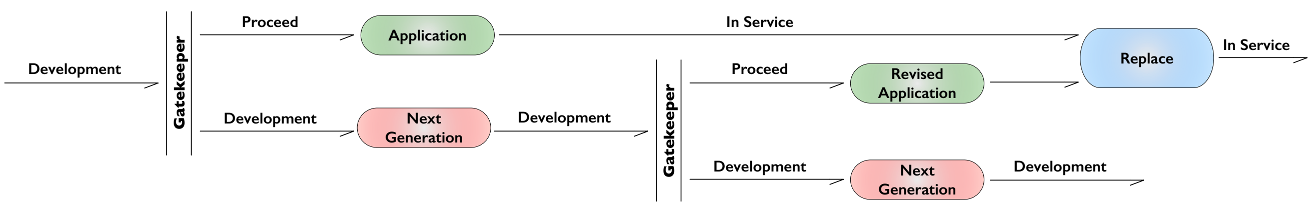
1. **Stop evaluation** – The technology is assessed as having little value within the current context.
2. **Defer & revisit** - The technology is considered as of interest, but not yet at a point where it could be applied in the current context. The evaluation is placed on hold for a stipulated period after which it is revisited and re-evaluated.
3. **Proceed to implementation** - The technology is assessed as being of value in the current context and is taken forward from the evaluation phase to the implementation phase.

Figure 7(b) then illustrates the role of the gatekeeper during the development and in-service phases. In this instance, the role is that of determining when to freeze a particular development timeline in order to take it forward to the target application within the desired timeframe. Development can then continue in parallel with a target timeline set for either upgrading or replacing existing systems. In either case, the gatekeeper role is of significance in achieving an effective progression from concept to application.

This however creates problems in relation to military personnel who might expect to be in a specific role for a period of no more than 2 years before rotating out to another posting as part of their career progression. Such rotations can however negatively impact on roles such as those of gatekeepers where a familiarity, and hence continuity of input, are often an important element. A further element to be taken into consideration in relation to military hardware is the continually changing nature of the role of the military, and hence the demands made on its systems, often resulting in systems having to be deployed in relation to tasks for which they were not designed[13]. With the design lead times involved in major systems it would be a rare occurrence when the role as conceived precisely matches that required on deployment, and this must also be factored in to the gatekeeper's role.



(a) Technology evaluation process



(b) Technology review and update[14]

Figure 7: Gatekeepers

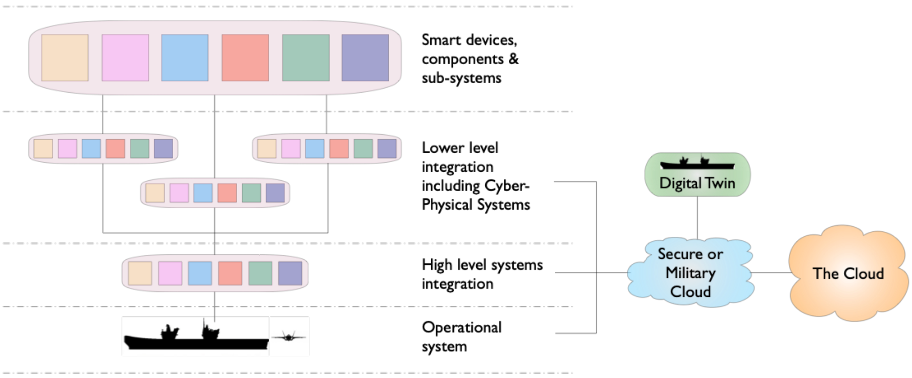


Figure 8: System of systems

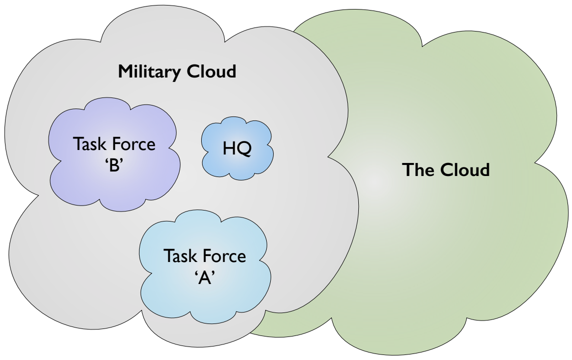


Figure 9: Linking the Military Cloud and The Cloud

System of Systems
All major military systems are essentially systems of systems. Referring to Fig. 8 firstly there are the basic devices, components and sub-systems around which everything else is configured. Increasingly, these are considered as '*smart*' implying that they have certain built in capabilities such as communications and self-monitoring, as for instance in support of the gatekeeper role[15]. Next, these are integrated to form more complex systems, often incorporating integral AI structures to create what are referred to as Cyber-Physical Systems with their own decision making capabilities.

These may increasingly linked to an external Cloud-like environment in the form of the Military Cloud. Here, each Task Group may well have its own local Cloud which links all members of the group, with several such Task Groups then being connected via the Military Cloud, which itself might in turn have links to the wider, generic, Cloud all as suggested by Fig. 9. Further, a '*digital twin*', a model of the system which receives all system performance and related data and may be available which receives, in real-time, performance data from on-board systems and which can be used for diagnostics, performance evaluation and prediction using techniques such as Big Data Analytics[16]. Higher level systems then bring these together within the operational context of the final system to help both diagnose and correct weaknesses and also to facilitate the identification of future system requirements.

Self-Organising Systems - As the complexity of the systems increases, so does the ability of the designer to understand them and their way of working, particularly when such operation involves the analysis and integration of large volumes of data from multiple sources. Such self-organising systems can therefore function in ways not understood by their users, including their underlying decision making processes. In a military context, this issue is being brought to the fore in relation to what are referred to as Lethal Autonomous Weapons (LAWs) where the decision processes determining the actions to be taken are embedded in the on-board systems software, raising significant ethical issues in relation to their deployment[17].

Further, many of the systems of systems that may be found in military hardware may become increasingly self-organising. For instance, a new or replacement module describes itself to the system which then autonomously reconfigures to take account of the capabilities of the introduced system[18], presenting additional challenges for both the system specifier and the system designer[19].

The Role of Formal Methods
Establishing user need has long been a key issue in the design process, resulting in the introduction of a number of formal methods to capture user thinking and to isolate the core needs. In this context, Heilmeier's Catechism[20] provides a useful first step towards understanding the process. This poses the following questions to both user and designer with the aim of making all involved both think and talk about the project along with their individual and collective aims,

goals and objectives.

- What are you trying to do?
- How is it done today, and what are the limits of current practice?
- What's new in your approach and why do you think it will be successful?
- Who cares, and if you're successful, what difference will it make?
- What are the risks and the payoffs?
- How much will it cost?
- How long will it take?
- What are the midterm and final '*exams*' to check for success?

To which must be added the extremely important question of who is paying?

When approached from the perspectives of both the user and the designer the goal is that of ensuring a common and agreed awareness of the problem, the associated issues and the success criteria at the very start of the process[21]. Once this in place, work can be begin to establish user requirements and need. In order to support this process, there are a number of formal methods available as summarised in Table 1[22].

Table 1: *Formal methods in requirements capture and analysis*

Method	Notes
Viewpoint Analysis	Splits the system into Functional Viewpoints, namely those directly with the operation of the system at all levels, and Non-Functional Viewpoints generally associated with factors such as layout and appearance. This can be a very powerful tool when used in a focus group like environment for engaging users with the thinking processes associated with design.
Function/Means Analysis	Breaks the system down into its functional elements and then considers the means by which the desired level of functionality can be achieved. Complements viewpoint analysis in many respects and can be effective in establishing '*hidden*' functionalities such as maintenance strategies.
Data Flow Diagrams	Developed as a software engineering tool, these can be extended to understand the form of data, and hence information, in the system, including the necessary links to human decision making.
QFD - *House of Quality*	The so called *House of Quality* represents the top level of the Quality Function Deployment methodology and supports the identification of the relationships between specific functions, and the importance placed upon these by users. It also supports a comparison at the functional level between alternative solutions.
Controlled Requirements Expression (CORE)	Developed by BAE to enable the relationships between systems, sub-systems and components to be established, CORE maps these in terms of their source, associated inputs, actions, outputs and destinations.

Software
It may be argued that software is a special case in relation to the previous arguments as it is likely to be subject to a process of continuous development throughout the lifetime of the primary system. For the purpose of the discussion, software is considered here under two headings namely:

1. *Platform* – The hardware on which the software runs.
2. *Code* – The core of the software package intended to achieve and provide the required functionality and performance.

Now consider each in turn.

Platform
The platform should be subject to the same decision making processes as all other physical systems, including the use of gatekeepers with the proviso that it is likely to be the physical component that is replaced or upgraded moist often over the course of the in service lifetime. However, a simple reliance on Moore's Law of Fig. 10 to add both capability and capacity over time is not necessarily an effective indicator of outcomes[23].

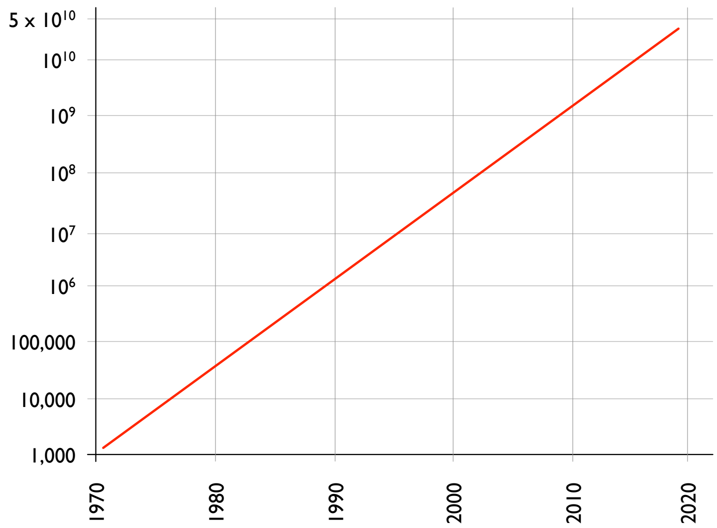


Figure 10: Moore's Law – Number of transistors on a chip doubles approximately every 2 years

Code
Increasingly, software is developed on a continuous basis using updates to enhance system performance to the limits of the hardware platform. Referred to as Agile Software Development this follows the pattern set out in Table 2.

Table 2: *Precepts of Agile Software Development*

Drivers	Notes
Individuals & Interactions → Processes & Tools	People are the most important element in the process.
Working Software → Comprehensive Documentation	Documentation, though helpful in understanding how the software functions, is secondary to the development of the software.
Customer Collaboration → Contract Negotiation	It remains essential to work closely with clients to establish their needs properly.
Respond to Change → Following a Plan	Need to accommodate changes in technology as well as in client needs & priorities

From which the need to work closely with the end user, in the form of the customer, is seen to be a key feature.

These precepts are supported by a set of twelve principles as set out in the *Manifesto for Agile Software Development* as follows:[24]

1. Customer satisfaction by early and continuous delivery of valuable software.
2. Welcome changing requirements, even in late development.
3. Deliver working software frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, who should be trusted
6. Face-to-face conversation is the best form of communication (co-location)
7. Working software is the primary measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity - the art of maximizing the amount of work not done - is essential
11. Best architectures, requirements, and designs emerge from self-organizing teams
12. Regularly, the team reflects on how to become more effective, and adjusts accordingly

While the intent of an agile approach is that of a sustainable development and progression of the software components, it still requires, as identified in Table 2, working closely with users to establish their need. In the context of the types of system under consideration here, this implies both a robust initial requirements capture process supported by a process of feedback from the operational systems to the developers to support upgrade and enhancement.

Information Flow and Vertical Integration
Referring back to Fig. 8 it is clear that a particular issue is that of dealing with and managing the vertical integration that exists between the various hierarchical layers in the system, including Cloud-based software. In this respect, consider the situation of Fig. 11 which shows the relationship between the flow of procedural based information or

WHAT is to be done, and process based information or **HOW** things are to be done.

Each level in the figure represents an action layer associated with the execution of a particular task or function. At the lowest level are the fundamental components, systems and sub-systems from which the entirety of the system is constructed, the operating time frame for which would typically be milliseconds or less while at the highest level are the Cloud-based systems with potential operating time frames of hours or longer. Each of these layers then has associated with it a series of process oriented '*instructions*' or '*commands*' which define the way in which the task or function associated with that level are to be executed, the **HOW**. Procedural information then flows upwards and downwards between these execution layers, defining the actions to be taken at each individual layer, the **WHAT**.

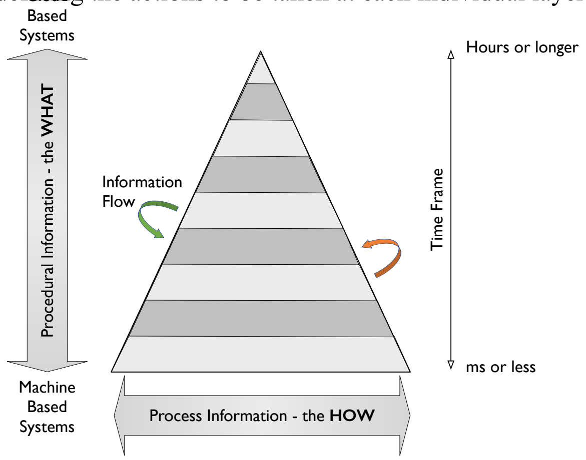


Figure 11: Information flows

At the lowest level, vertical transmission is essentially machine-to-machine while at the highest level it is software-to-software. Middle levels often involve human-based transmission, making them particularly vulnerable to misinterpretation, particularly as humans are very good at reasoning with incomplete data and filling in gaps, either real or perceived, in such data. The problem here is that some of the associated interpretation may be dependent on factors such as an individual's prior experience, which may differ from individual to individual[25]. One of the roles of the requirements capture process must therefore be to establish the nature of these communications and act to minimise interpretive errors.

Other Issues

Security - Security is obviously a major concern in any system such as those indicated, particularly when there is any form of external or Cloud-based links involved. The security of all systems must therefore be an important consideration in the design process.

Design for Manufacture - The way in which something is to be manufactured can significantly influence its design and is something that needs to be accommodated during the requirements capture phase of the design process.

Procurement & Supply Chain - This is not only important during the build and production stages of a project but as part of its whole-life plan. Components, and particularly electronic and software components, which were near state of the art at the time of build will become obsolescent and need to be replaced, implying that thought needs to be given during design to managing such transitions, including potential suppliers[26].

Maintenance & Repair - Military systems often have to operate remotely from any major facility which means that they have to become essentially autonomous in respect of maintenance and repair in. order to remain on station. This means that design for maintainability becomes a significant element within the design process. The application of design strategies such as modularity and *plug and replace* therefore take on a particular significance along with the adoption of techniques such as additive manufacturing[27] to enable a wide range of components to be produced locally on demand, alleviating the need to carry a wide range of specific items as spares.

Summary

Military systems are increasingly complex requiring long development and construction lead times. These place additional pressures on the requirements capture process and the procedures adopted for the management of change, during requirements capture, design, project build and in service operation and maintenance. In particular, there is the issue of integrating a knowledge led development process within a procedure driven and hierarchical organisational structure in order to allow for the proper choice of technology and its integration with and within operational requirements. This is particularly the case when it comes to the gatekeeper role where the requirements of a military career progression and promotion may mean that role may fall, at least in part, on someone with limited experience of its function as well as a limited exposure to the technologies involved. Gone are the days when a captain could be appointed as the keel was being laid to see the ship through its construction, fitting out, trials and introduction into service.

When dealing with the technologies there are of course issues in communicating and understanding between the user and those responsible for implementation, something of particular concern where software is involved. The goal here is to achieve a situation where the contractor, be it for hardware or software, provides the user with what they need, rather than what they think they need. Assuming of course that the user knows what they actually need, as opposed to what they think they need! All of this being associated with requirements capture and project definition. There are of course numerous issues and problems associated with achieving this, not least in ensuring that all likely, or indeed is some instances all possible, conditions of use are considered and incorporated into the design process, an unlikely scenario in its own right. Here it needs to be recognised that the project definition document (PDD) resulting from requirements capture and analysis, and the precursor of the detailed specification, is not a static document but one which should guide the whole of the development process. Too often the PDD is written, used to create the specification and then placed into the filing cabinet and forgotten.

In illustration, consider the following salutary tale. In February 2007 a flight of F-22A *Raptor*s were deploying from Hickam AFB in Hawaii to Kadena AFB in Japan. As the aircraft crossed the International Date Line (IDL), their on-board computer systems crashed. When attempts to reboot the systems failed, the F-22s had to be led back to Hawaii by their escorting tanker aircraft. In an interview with CNN, retired Airforce Major Gen Don Sheppard stated that[28]:

> "*... at the international date line, whoops, all systems dumped and when I say all systems, I mean all systems, their navigation, part of their communications, their fuel systems. They were — they could have been in real trouble. They were with their tankers. The tankers – they tried to reset their systems, couldn't get them reset. The tankers brought them back to Hawaii. This could have been real serious. It certainly could have been real serious if the weather had been bad. It turned out OK. It was fixed in 48 hours. It was a computer glitch in the millions of lines of code, somebody made an error in a couple lines of the code and everything goes*."

The problem, the essentially instantaneous transition from 179.9°W to 180°E on crossing the IDL, was something which no-one had thought about[29].

These, and other errors involving communication between individuals, may also be associated with what is referred to as cognitive bias, where an individual's assessment of a problem or situation is influenced by their perceptions of a situation rather than its reality[30]. This in turn may lead to a situation where the domain expert assumes that they know the nature of the solution required, and hence minimise, or even ignore, user inputs[31,32]. This can lead to a tendency on the part of the user to overstate their needs, and the provider to overstate capability.

It is argued that a properly implemented and understood approach to requirements capture and analysis, and the engagement through this of all stakeholders, taken here as being all participants in the full design and development process, while not eliminating all such problems, may well have significant impact on their minimisation and hence on mitigating their effect. This would include giving consideration to modified career pathways for military personnel involved in such projects in order to ensure that any skills and expertise that they develop that are linked to the project are retained by the project. This also suggests that the close integration of the military component of the project management team with the relevant domain experts is likely to be a key factor in achieving success.

Annex 1 – Design Case Studies

It should be stressed that the arguments presented in the paper are intended to highlight and emphasise issues which have the potential to impact on the procurement process, and that these issues are by no means unique to the procurement of military equipment, and indeed are prevalent across and through all types of industry. The following industry based case studies are therefore provided to illustrate issues, both positive and negative, that are often associated with the design process, and the understanding or otherwise of that process.

1) **Metal Fabrication**

A small company offered a range of metal fabrication services mainly, but not exclusively, for the agriculture sector. They also offered a bespoke design service utilising a small, but highly experienced, team located within the factory. As the members of the design team were permanent employees, and despite their work being almost full time on client designs, they were charged as overheads to the business.

Historically, designs had been agreed with clients, who would not have been charged up to that point, and then they would be fabricated in house. The company then found that clients were taking the (free) designs and having them made elsewhere at lower cost.

The company therefore re-evaluated their business model and realised that the design team was responsible for much of the added value associated with the construct elements of the business. The company therefore started to charge for the design while reducing the build costs so that the overall nett cost to the client of design and build was essentially the same as under the old model. The result was a restoration of the fabrication business along with additional, design oriented, business being generated.

2) **Construction steelwork**

A large construction company installed an advanced robotic plant to cut and weld structural steelwork. However, they found it necessary to install a second facility to carry out remedial work on this same steelwork as a result of design changes being requested or implemented by the client and architect, sometime in the period between the steelwork leaving the robotic plant and arriving on site.

From the construction companies perspective the installation of the robotic plant enabled them to reduce costs, and hence lower their quotation for a job, while the real costs, and the company's profits, were embedded in the remedial work that they had to carry out.

3) **Technology evaluation**

A company involved in a range of technological areas found that it was having difficulty making decisions as to the choice of technological avenues it needed to pursue in order both to maintain and to grow its business, and indeed found that there might often be parallel investigations into a technology going on across the company. It therefore initiated a special technology assessment and evaluation group within the company whose role was to examine new technologies in the context of the business while also acting as gatekeepers for those technologies.

The effect was to reduce costs and to speed up the introduction of a technology into the company.

4) **Design integration**

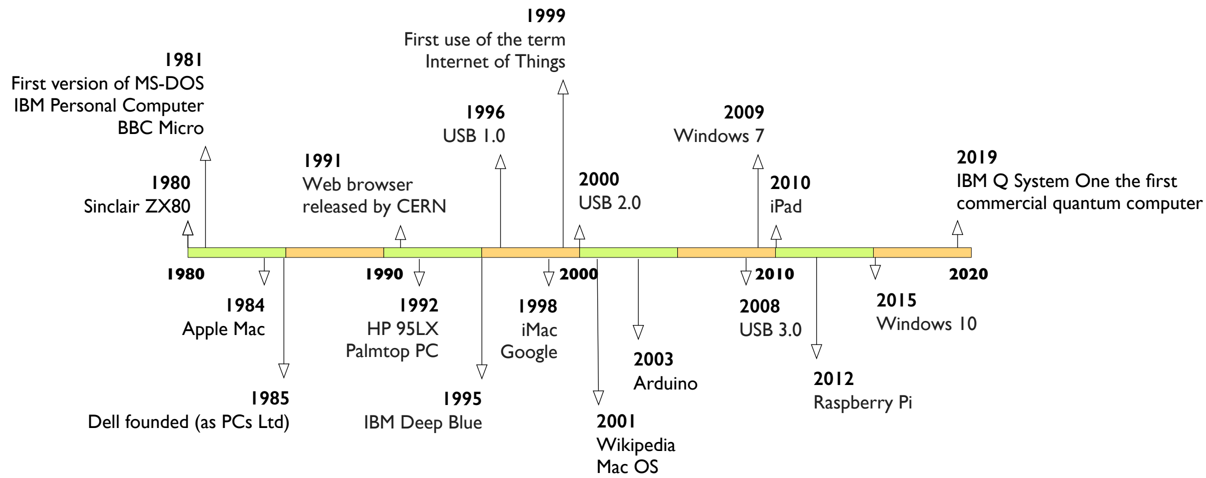
A company's products were driven by a combination of aesthetics and technology requiring the introduction and use of advanced manufacturing technologies with potentially long lead times, as for instance in the production of the moulds for injection moulding machines. By creating integrated design teams involving all aspects of the business from aesthetics to manufacture, lead times, and hence production costs, were reduced as was the time to market for new designs.

Notes

1 As for instance following the collapse of the Soviet Union.

2 Personal communication and memoires of WR Scarfe, Head of Systems Development @ BAe Warton in the 1990s.

3 In this context, the figure below summarises developments in computing technology over the lifetime to date of the Eurofighter *Typhoon*.



4 An example of the *Pareto Principle* or 80:20 rule.
See also:
Andreasen MM & Hein L (2000) *Integrated product development*, IPU

5 Thereby reversing an initial decision which itself seems to have been made on cost grounds and which essentially eliminated operating options such as cross-decking while requiring the adoption of a less capable, in terms of range and payload, version of the primary aircraft.

6 As for instance the issues associated with the design and construction of hybrid-fuel ferries for Caledonian MacBrayne by the Ferguson shipyard.

7 Hirshorn S & Jefferies S (2016) Final Report of the NASA Technology Readiness Assessment (TRA) Study Team @ ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170005794.pdf (accessed April 2020)

8 Hanna el al (R Hannah, R Gross, J Speirs, P Heptonstall & Gambhir, *Innovation timelines from invention to maturity. A rapid review of the evidence on the time taken for new technologies to reach widespread commercialisation,* 2015, UK Energy Research Centre) found that for 14 innovations considered, the average time from invention to widespread commercialisation was 39 years. Lithium batteries were the fastest at 19 years while cars (according to the definition used) took 70 years in the US.

9 AM Turing, On computable numbers, with an application to the Entscheidungsproblem, *J. Mathematics*, **58**(345-363), 1936, 230-265

10 And of course rockets were known in China in the 13th century, and possibly earlier.

11 Recent and current examples of such an '*under the skin change*' include the *Type 23* Power Generation and Machinery Control and Surveillance (MCAS) System Update (PGMU) Project in line with achieving extended Out of Service Dates (OSD) and the Eurofighter *Typhoon Tranche 3* development.

12 The V-model provides a visual representation of a systems development lifecycle which sets out the high-level activities associated with product development. It is one of several such models and is used here for its simplicity and visual form.
Various versions exist including a US Government standard form. See the Defense Acquisition University (www.dau.edu) & Redshaw MC (2010) Building on a legacy: Renewed focus on systems engineering in defense acquisition, *Defense AR J.*, **17**(1), 93 - 110

13 As for instance the current need to deploy high value hulls for the interdiction of drugs trafficking in the Caribbean.

14 An instance of this process of continuous upgrade can be seen in the B-52 *Stratofortress*. This first flew in 1953 and entered service in 1955 with production ending in 1962. The aircraft currently remains in service as the B-52H variant and plans are in place to re-engine the aircraft to further extend its service life.

15 In the broader sense, the term *smart* has come to mean systems with these capabilities, but not necessary incorporating any built-in intelligence or decision making capability.

16 See:
Glaessgen E & Stargel D (201) The digital twin paradigm for future NASA and US Air Force vehicles, Proc. *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conf/20th AIAA/ASME/AHS Adaptive Structures Conf./14th AIAA*, 1818 - 1831
Boschert S & Rosen R (2016) Digital Twin—The Simulation Aspect, *Mechatronic Futures* (Hehenberger P & Bradley D (eds)), Springer
Fuller A, Fan Z & Day C (2019) Digital Twin: Enabling Technology, *Challenges and Open Research*, *arXiv preprint arXiv:1911.01276 @* arxiv.org/pdf/1911.01276.pdf (accessed April 2020)

17 Department of Defense Directive 3000.09, Autonomy in Weapon Systems (2012) @ www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodd/300009p.pdf (accessed April 2020)
Also:
Righetti L, Pham QC, Madhavan R & Chatila R (2018) Lethal Autonomous Weapon Systems [Ethical, Legal, and Societal Issues], *IEEE Robotics & Automation Magazine*, **25**(1), 123-126
Schmitt MN & Thurnher JS (2012) Out of the loop: autonomous weapon systems and the law of armed conflict, *Harvard National Sec. J.*, **4,** 231 - 281
Altmann J & Sauer F (2017) Autonomous weapon systems and strategic stability, *Survival*, **59**(5), 117-142

18 An advanced form of plug & play.

19 Muccini H, Sharaf M & Weyns D (2016) Self-adaptation for cyber-physical systems: a systematic literature review, *Proc. 11th Intl. Symp. Software Engineering for Adaptive and Self-Managing Systems*, 75-81

[20] G. Heilmeier, *Some Reflections on Innovation and Invention*, Founders Award Lecture, National Academy of Engineering, Winter 1992 @ isi.edu/~johnh/TEACHING/CS651/ARCHIVE/Heilmeier92a.pdf (accessed Mach 2020)

[21] An interesting, and often revealing, exercise is to get both the user and the designer to answer the same questions from their individual perspective.

[22] This list is neither comprehensive or exclusive but covers major approaches and techniques.

[23] There is increasing evidence to suggest that Moore's Law is reaching, or has indeed reached, its molecular limit.

[24] At web.archive.org/web/20100614043008/http://www.agilemanifesto.org/principles.html (accessed February 2020)

[25] See for instance:

Molesworth BR & Estival D (2015) Miscommunication in general aviation: The influence of external factors on communication errors, *Safety Science*, **73**, 73-79

Paek T (2003) Toward a taxonomy of communication errors, *ISCA Tutorial & Research Workshop on Error Handling in Spoken Dialogue Systems,* 53-58

[26] See for instance Huawei and 5G technologies.

[27] Also referred to as 3D-printing.

[28] www.defenseindustrydaily.com/f22-squadron-shot-down-by-the-international-date-line-03087/ (accessed February 2017)

[29] Other examples could of course be found as for instance in the design and introduction of the M-16 rifle and its problems of reliability in the jungles of Vietnam or the problems encountered by the SA80 when operated in extreme environments such as the Arctic or a desert.

[30] Kahneman D & Frederick S (2002) Representativeness revisited: Attribute substitution in intuitive judgment, *Heuristics & Biases: The Psychology of Intuitive Judgment*, **49**, 81

[31] Hinds PJ & Pfeffer J (2003.) Why organizations don't "know what they know": Cognitive and motivational factors affecting the transfer of expertise, *Sharing Expertise: Beyond Knowledge Management*, 3-26

[32] The two situations are sometimes referred to, both collectively and individually, as '*The Curse of the Expert*'.