_GUIDEBOOK *for* DIGITAL ENGINEERING *and* e-SERIES

the SPOON BENDING

"Dicula Nulla Est."

WILL ROPER

01 19 21

***"Please just listen. I know why you're here, Neo. I know what you've been doing.… It's the question that drives us, Neo. It's the question that brought you here."* _Trinity, *The Matrix* (1999)**

I. INTRODUCTION. (Spoiler alert for *The Matrix*!)

I do indeed know why you're here. Whether by direct experience, online research, or extrapolating from *The Matrix*, you've glimpsed what digital acquisition might achieve. You've read *There is No Spoon* and tried applying its digital trinity – agile software, digital engineering, and open architecture – inside your programs. But remaining questions still drive you: What is digital engineering and an e-Series, *really*? Do we need them? What are their criteria? And how far must we go to effect a digital transformation for the Air Force and Space Force?

Many transformative ideas – the electric lights of the world – resist mass adoption initially because the oil lamps they'll inevitably replace presently occupy that mass. Transitioning from one to the next happens gradually at the edge, where devils in the details matter.

Transitioning from analog to digital acquisition is a devils-in-the-details task too. Open architecture and agile software are more easily considered, being either "on" or "off" in portions of a program. Digital engineering is more challenging, a dimmer switch of varying degrees. Not everything at the edge is worth digitizing, especially for extant systems, requiring value judgment from you on how much effort is actually illuminating.

"There is no spoon?" _Neo
"…it is not the spoon that bends. It is only yourself." _Spoon Boy, *The Matrix* (1999)

This companion guide to *There is No Spoon* will equip you for those value judgments and help you pursue spoon-bending results for both digital engineering and *e*-Series. Specifically, it goes deeper on the modeling and infrastructure requirements to effect several tenants of *There is No Spoon*: "eCreating before Aviating" and owning and furnishing the tech stack. Though written to stand alone, its insights will make more sense if read as a sequel.

I mentioned in *There is No Spoon* that digital engineering — done right — is both art and science. This is a good sentiment to upload while reading this guide. Art is also a pursuit. And just like those driving questions for *e*-Series, precisely what art is — and what its criteria are — defy rigid definition. But art does have fundamental, widely-accepted principles that evolve over time. Digital engineering does and will too. Look for spoon emojis throughout this document for important ideas and spoon-bending principles. You can also find them summarized in the Appendix.

Let's put the most important Air Force and Space Force digital engineering principle right up front: 🥄 **Digital Engineering must achieve a measure of *authoritative virtualization* that replaces, automates, or truncates formerly real-world activities. *e*-Series, remarkably so.**

This is the *art* of digital engineering — unlocking the impressive performance we've witnessed in bellwether programs like NGAD, GBSD, *e*T-7A, and the A-10. And yes, I'll be explaining more about what this means, starting now!

II. *A NEW DIGITAL ARTFORM*

***"I imagine right now you're feeling a bit like Alice, tumbling down the rabbit hole? Hmm?"* _Morpheus*, The Matrix* (1999)**

It goes without saying, "defense procurement" and "art" have rarely occupied the same sentences. Though our Cold War process *does* produce world-leading military systems, its escalating timelines and costs are unsustainable byproducts. The stark contrast with commercial industry puts our military at the "wonderless" end of the rabbit hole.

Thankfully new commercial technology called *digital engineering* is already lending a legitimate art form to military weapons-buying with wonderful, even Wonderland results. As presaged in *There is No Spoon*, the Air Force and Space Force did create an *e-Series* designation for digitally-engineered aircraft, satellites, and munitions.

But Matrix-like computerization is only part of the *e*-Series equation. The real art is observed in the real world. Just as architects capture physical structures, digital engineers capture physical systems and processes virtually — learning, perfecting, and even automating them — so that costly trial and error happens cheaply on computers.

Having this virtual rewind button has *seriously* fast-forwarded real-world results thus far. Take the digitally-designed *e*T-7A trainer jet: designed and built in just 36 months – a feat not accomplished since the 1950s with third-generation fighters. The same digital approach birthed our most advanced sixth-generation flight demonstrator years ahead of expectation.

Science fiction movies, like *The Matrix,* help us imagine the underlying technology. But this technology is equally an art, a new way to capture reality, but one with dire consequences if misapplied. Like architecture, both safety and success rely on sound methodologies to certify designs will faithfully translate to reality. We show our trust in these methodologies every time we enter a newly-constructed building. Digital engineering and *e*-Series simply entail analogous methods: how to trust reality reflects digital design.

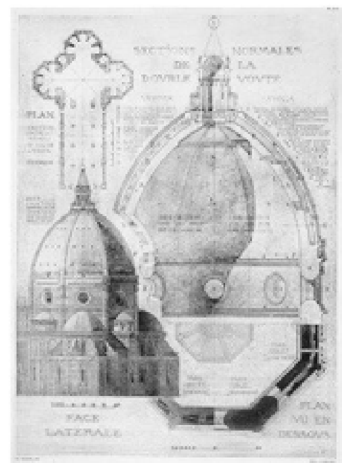
2a. _THROUGH THE LOOKING GLASS OF HISTORY

"Tank, find a structural drawing for this building."
_Morpheus*, The Matrix* (1999)

The conceive-it-then-construct-it nature of architecture has historically connected it to technologies that improve engineering execution. Filippo Brunelleschi used mirrors and geometry to generate 3D drawings with perfect linear perspective. Leonardo da Vinci meticulously studied physics to create modern technical drawings of complex systems. Frank Gehry employed computer-aided fabrication to achieve his physics-defying, Daliesque buildings. And in a dystopian sci-fi future, artificial intelligence uses a neural-interactive computer simulation to construct the *Matrix*.

Though not *yet* capable of *Matrix*-like modeling, digital engineering does take computer creation technology to the next level, rendering not just the design of complex systems, but their assembly, environment, and even physical performance in highpowered virtual reality (VR). Prominent modern architect Mies van der Rohe once observed that “whenever technology reaches its real fulfillment, it transcends into architecture.” Digital engineering is transcending into a type of four-dimensional architecture — one that designs three-dimensional systems *and* time-driven processes that govern them in realistic VR, long before their physical twins are built.

“Did you know the first Matrix was designed to be a perfect human world? … It was a disaster.” **_Agent Smith*, The Matrix* (1999)**

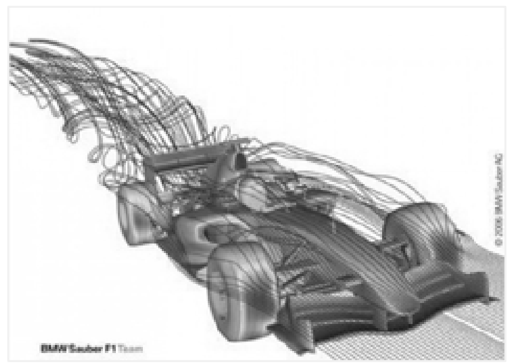


Architecture and engineering have endured a hit-or-miss relationship when plans transition to implementation. In constructing the world’s largest masonry dome in 1420, Brunelleschi devised new engineering marvels to complete his 150 foot Florentine masterpiece (pictured right): nesting two domes to avoid buttresses, laying bricks in novel self-reinforcing patterns, even inventing cranes and pulleys so ingenious they were later studied by da Vinci.

Other projects saw disasters. The bursting St. Francis Dam, collapsing Tacoma Narrows Bridge (pictured left), and falling windows of John Hancock Tower are cautionary examples of architectural design overlooking engineering reality. Despite the heavy use of computer models, Gehry’s MIT Stata Center did not account for drainage, mold growth, or snowfall. And even future AI failed to get the *Matrix* right on its first go-round. No wonder Brunelleschi said that in building, “only practical experience will teach that which is to be followed.” Even with computers, *true* reality is truly hard to capture.

Until recently.

Industries like the automotive were the first to replace Brunelleschi’s practical experience with digital ones. Computer-aided design tools had been widely used since the 1960s but never replaced the “rubber meeting the road” of physical prototypes and testing. Since then, the trillion-fold boost in computer processing has morphed those early blueprint tools into today’s powerful digital engineering models — called *digital threads*[1] and *digital twins*[2] — that replace real-world prototyping and testing with authoritative virtual sources of truth.



A good case and point is Formula 1 racing, where there are no physical prototypes today. Every car feature and all physics governing it — even the rubber literally meeting the road — is painstakingly virtualized and anchored by authoritative test data (airflow example pictured right). The end result is hundreds of digital cars being explored each racing season, even optimized for individual racetracks, all without bending a single piece of metal. Real-world checkered flags attest just how authoritative these “*e*-Cars” can be. Our military *e*-Series can be the same.

III. BENDING THE SPOON

***“Throughout human history, we have been dependent on machines to survive.”* _Morpheus*, The Matrix* (1999)**

As you read in *There is No Spoon*, a similar breakthrough has occurred for military systems. Our *e*T-7A successfully virtualized production, constructing hundreds of *e*-planes digitally to optimize their physical assembly. Our new ICBM used virtual design to explore six billion *e*-missile variants in mere months. And our advanced sixth-generation aircraft, also an *e*-plane, is adopting

---

[1] *Digital threads* are extensible analytic frameworks to *connect* models — and all associated data, software, and functional support that govern system lifecycle phases — to create an authoritative, digital source of truth with one-to-one real-word traceability.

[2] *Digital twins* are authoritative, one-to-one models of *individual* real-world systems that coevolve via data feedback loops.

Formula 1-style practices to out-iterate pursuing adversaries using a Digital Century Series approach.

Each of these hints at how transformative the art of digital engineering — perfecting reality by exercising its digital equivalent — can be for our military and hopefully nation.

So let's get more specific about how to begin. Given the most important principle of digital engineering centers on *authoritative virtualization*, we need to define both it as well as the *digital foundation* on which it is built. This is where things get heavier technically.

3a. LAYING THE DIGITAL FOUNDATION

"Get some rest. You're going to need it." _Morpheus
"For what?" _Neo
"Your training." _Morpheus, *The Matrix* (1999)

As with any new construct, the process begins with the foundation on which it is built. **Our Digital Foundation is the infrastructure, policy, training, and culture that enables digital acquisition, digital engineering, and *e-Series*.**[3]

A strong digital foundation, alone, is worthy of its own guide, but one mostly for others to write. Thankfully, Air Force Materiel Command and Space Force Space Systems Command have already begun. Department-wide infrastructure to provide tools and connectivity, policies to democratize data and digital tools as underlying technologies change, training to employ those tools effectively, and the workforce culture to make it happen are all groundwork tasks. Without them, there is nothing on which to build. Just like its physical counterpart, a digital architecture begins with a firm digital foundation.

Once this digital foundation is in place, we can create the core construct for digital engineering and *e*-Series: authoritative virtualization.

[3] Digital acquisition also contains business management, operations, software, and other non-engineering lifecycle disciplines that may be digitally transformed in addition to digital engineering.

3b. AUTHORITATIVE VIRTUALIZATION & OWNING THE TECH STACK

***"This is the construct. It's our loading program. We can load anything from clothing to equipment, weapons, training simulations — anything we need."* _Morpheus, *The Matrix* (1999)**

To dig into this subsection, let's start by defining its core construct. 🥄 **Authoritative Virtualization is a digital model of a system that renders its inputs, operational environments, and internal functions and behaviors — along with all subsystems necessary to capture them — such that outputs can be certified as *predictive*.**

There's a lot to unpack in this definition, including multiple important principles, to ensure virtual spoon-bending leads to real-world results.

First, because inputs affect outputs, this definition could imply they must also be authoritatively virtualized, likely the case in many complicated system of systems. But it could also be their properties are understood via physics (e.g., environmental effects) or empirical data (e.g., manufactured part tolerances or software runtimes). 🥄 The key principle is that **every system virtualization has a starting point — its basic building blocks — and those building blocks must be quantitatively understood**. Otherwise, your digital thread was never digital from the start.

If your program is *not* a new acquisition, this is an especially important insight for you. Your starting block is a legacy system, which doesn't preclude you from serious digital artistry. But it does require you nail down your starting point analytically. For the B-52 Commercial Engine Replacement Program, that starting point was the podded mount for the digital engines to "attach" to the physical wing, as well as the measured center of gravity and airflow properties of the jet. For the A-10, the entire wing — nearly the entire airplane — had to be digitally rendered because drawings from the original manufacturer (i.e., the authoritative source of truth) had been lost. For myriad programs trailblazing Conditions-Based Maintenance with the Rapid Sustainment Office, the starting point was digital maintenance data, vice its originating airplanes, in order to model and predict when parts would fail and retrain our maintenance enterprise. Each one of these is a form of authoritative virtualization because certified predictive models removed or truncated time-consuming real-world activities.

No matter your program, there will be some case for digital metamorphosis. Just understand your starting point and projected return on investment to know what spoon-bending efforts make sense.

"This is a sparring program…. It has the same basic rules, rules like gravity." **_Morpheus*, The Matrix* (1999)**

Second, as noted in above, the environment gets a vote. **Whether internal or external, if the operating environment affects outcomes or performance, you must either model its physics-based rules (like gravity) or account for it via empirical data**. Just like inputs, environmental impacts must be understood analytically or your digital threads and twins won't live in a realistic digital world.

"The Matrix is everywhere. …You can see it when you look out your window or when you turn on your television."
_Morpheus*, The Matrix* (1999)

Third, **whatever you are virtualizing, if it contains internal elements that affect outputs, they inherit the same analytic burden, requiring authoritative modeling or verified anchoring data.** Successively applying this principle until outputs match reality will finally produce a true Matrix-like VR and not some simulacrum. Your digital twins and threads cannot be digital veneers overlaying analog interiors.

This ***does not*** mean everything in a system must be modeled. Nor does it mean all sub-models must be of equal fidelity. As long as the final errors bars of the virtualization enable it to substitute for reality, you've succeeded.

Don't let imprecise or unknown error bars imprison your intentions. When building or testing physical systems, not every component or test is a perfect clone, creating a statistical distribution. Quantifying risks posed by one-sigma, two-sigma, and higher-order effects is how we ultimately certify systems for operational use. Your models must accurately reflect these distributions (if applicable) or real-world replication will still be needed. *This is why formal standards and methodologies for virtualizations are needed: replacing physical activities with virtual ones is a new acquisition risk, potentially a big one.* Having a definitive digital "building code" to follow to develop authoritative virtualizations ensures this risk can be understood and managed by the Air Force and Space Force.

And by the way, once you have such a virtualization, you can also manipulate many of the root causes of those statistical distributions altogether. (Just think how determinant assembly eliminated scrap, rework, and repair stats for *e*T-7A.) Our real goal isn't merely recreating yesterday's reality. Just like Formula 1 racing, it is creating a more-winning one tomorrow.

3c. DIGITAL BUILDING CODE

"You believe that you are special, that somehow the rules do not apply to you. Obviously you are mistaken."
_Mr. Rhineheart*, The Matrix* (1999)

🥄 Fourth, just as construction architects and engineers obey building codes certified by local governments, **our digital models and infrastructure must obey a similar "digital building code" that is certified by the Air Force and Space Force.** This is the pith of "owning the tech stack" and paramount because safety and mission success may depend on your models, in addition to cost.

Just as physical building codes ensure walking into new constructions is safe, our digital building codes must ensure Air Force and Space Force models result in physical systems trusted at first use.

Presently, our digital building code is a de facto case-by-case assessment of individual programs. 🥄 But as you read in *There is No* Spoon, **owning the tech stack — at a minimum the government reference architecture by which underlying models and software are built — and even furnishing the digital environment itself, especially for software and automation, is the best way to enforce our digital building code, especially at scale.** (A lot more on automation coming up.)

We have a long way to go on our digital building code. I expect trailblazing programs to continue helping us experiment to find the right tools and techniques for quite some time, especially as underlying technologies evolve. We must learn by doing – even learning the hard way at times – so that enterprise practices can emerge. To all digital trailblazers ready to hack through this virtual jungle, you have my personal thanks!

Our “digital building code,” as it stands today, will be maintained online at the website below. This building codes applies to all new acquisition programs as well as major modifications of existing systems. Exceptions will require a waiver by the Milestone Decision Authority (MDA) – else we’re broadcasting pirate signals to hack into our own Matrix, not unlike Morpheus and the Nebuchadnezzar crew.

The same website contains the current *e*-Series Designation Criteria Scorecard, which MDAs will use in conjunction with the digital building code for *e*-Series designations. (Think of this as the equivalent of a building inspector’s checklist with the MDA as the inspector. Programs should receive validation of their digital engineering approach during their acquisition strategy reviews, including whether they are in *e*-Series contention.)

All digital engineering materials, and additional resources, will also be continuously updated at https://ww3.safaq.hq.af.mil/.

3d. AUTOMATION & FURNISHING THE TECH STACK TO INDUSTRY

***“It means buckle your seat belt, Dorothy, ‘cause Kansas is going bye-bye.”* _Cypher, *The Matrix* (1999)**

Fifth**, automation is a special case of virtualization, but one with special powers to accelerate digital acquisition.** Although nearly all functions within a program’s lifecycle may be virtualized, not all may be automated *yet*. Robotics may have revolutionized the car industry – and even crossed into Defense like digital engineering – but we are far from factories where stealth fighters and satellites may be “grown” like *The Matrix.* J

However, many things may be automated – and are today – especially in our software factories like Kessel Run and Kobayashi Maru and CloudONE/PlatformONE tech stack. Cybersecurity checks and authority to operate, normally a paper checklist ticked off by people, is now a digital checklist being ticked off as-a-Service (aaS) by the tech stack itself — not just once — but continuously.

And while DevSecOps software development has been a major acquisition energizer, we can go much, much further in radically accelerating other lifecycle functions via automation.

The right question to start off is, “What can we *not* automate?” Design reviews, contract writing and definitization, select tests, document writing, you name it: most things are amenable to fully or partially scripting aaS. In fact, it’s such a potential time-saver that the Survivable Airborne Operations Center is spearheading an “automation phase” of its program. Companies will propose measures to increase tech stack automation as a means to accelerate delivery while lowering cost and risk. (You didn’t think acquisition phases would stay the same after our digital revolution?)

So to answer the question, **absent required human judgment or critical thinking, activities we can turn into computerized checklists may be automated.** Though tech stack abstraction layers that create automated checklists require more upfront effort than just manually completing those checklists for individual programs, that extra upfront effort *continues working for us all the time*. Imagine the net result of crowdsourcing broadly-applicable automation across our programs and functions!

And even should human oversight still be required for automated outputs, the net personnel savings should still be game-changing. Automation is enabling increasingly smaller commercial development teams to manage larger, more-complex product lines today. We should expect an equivalent trend for defense programs. ***Our present Cold War personnel system — decomposing our massive mission into submissions, supporting functions, subfunctions, and so on, until we ultimately end on jobs done by an army of people at human speeds — equally decomposes our long-term competitiveness. Human speed loses.*** Authoritative automation is our chance to change that.

Which leads to a second question: “How do we make our automations *authoritative*?” This is where a government-furnished tech stack is required. Furnishing “Government-aaS” layers gives industry more control of their destiny, especially for time-certain delivery. These authoritative automations *are* the government in every legal sense, substituting for what people and paperwork do today. Just like *The Matrix*, this means strict configuration control and verification testing for government acceptance of automation risks — a huge, if not impossible, challenge if riding on industry information technology (IT).

This **does not** mean industry won't require their own tech stack, one hopefully built according to our digital building code for seamless interoperability. (Internal research and development — and a handy back-up should our network go down — will necessitate it.) But industry IT cannot be given the same legal standing as the government's in the foreseeable future.

So if we want to virtualize authoritatively, we must own the tech stack design; but if we want to automate "Government-as-a-Service" functions authoritatively, we must furnish our tech stack — at least its automation layers — to industry to overcome legal hurdles.

This challenges historical notions about IT procurement and government IT roles and responsibilities. We truly stand at a crossroads, where historical lowest-price technically-acceptable approaches as well as outsourcing to industry are primrose paths to future irrelevance. **IT, now and in future, is a warfighting system — the infrastructure by which code, data, and AI will bring war-winning, machine-speed automation to the battlefield — so it must be bought for overmatch, not as a "technically-acceptable" business commodity.**

Using this same war-winning infrastructure for competition-winning acquisition automation is a win-win for government and industry. And with Fourth Industrial Revolution technologies already powering impressive automation in commercial industries, keeping up with this tech trend is paramount, even existential, for national defense. Like AI, automation technology appears to accelerate itself.

3e. TESTING, THE ART OF E-SERIES, & DIGITAL METAMORPHOSIS

"I disagree, Trinity. I think the Matrix can be more real than this world. _**Cypher*, *The Matrix*** **(1999)**

Sixth**, once a model is built to digital code, whether or not it is predictive is a matter of test data, which can vary widely between programs and functions.** Virtualized business or contracting processes are trivially predictive after verification tests. (But don't underestimate how powerful they could be!) Derivatives of previously-tested designs likely require some additional testing to anchor performance envelopes. New physics-defying warfighting systems — like hypersonic weapons — will doubtless requires ground and flight test campaigns to certify predictive models.

However much testing is required, **a second principle here is the polarity of testing changes in digital acquisition. In the past, systems were physically tested to graduate *from* models. Today they should be tested to graduate *to* digital models.** The digital environment becomes the operational one — as real as the physical, arguably more.

"I'd ask you to sit down, but you're not going to anyway.
_The Oracle*, The Matrix* (1999)

Seventh, once oracular predictive powers have been certifiably established, you're ready to implement the second part of the primary principle of digital engineering and *e*-Series: replacing, automating, or significantly truncating real-world activities. This is where the art really kicks in and its impacts flourish.

You might virtualize designs to truncate integration activities, assemblies to replace real-world learning curves, training to truncate required hours, automatic software updates to avoid lengthy regression tests, contract deliverables to reduce paper-driven lag time, or….

You get the picture.

Only your imagination, digital tools, and training limit how far you might go with virtualization. (Just don't construct a dystopian Matrix!) **Once you have replaced, truncated, or automated real-world activities via virtualization — *the art of digital engineering* — being declared an *e*-Series is simply a final assessment by the Milestone Decision Authority.** Art, after all, is in the eye of some beholder. But this art is based on objective, measurable criteria. So if your program is changing the acquisition game because of digitization, you're in the right ballpark for an *e*-Series designation.

But why have the designation at all? Aside from safety and mission reliance on virtualizations, **certified *e*-Series should help propel the Air Force and Space Force's analog-to-digital metamorphosis.** Much like the dial-up internet transformed into the Internet of Things over the course of a decade in which phones, homes, cars – the world of things – picked up the preface *smart* as they explored a new, connected paradigm that invited both designers and users to think differently about their uses. Today nearly every device consuming electricity is *smart*. So much so, we've nearly dropped the honorific, which successfully played its role in transforming myriad commercial tech markets.

The Air Force and Space Force are now in the larval stage of our analog-to-digital metamorphosis. **Each new *e*-System invites us to reimagine both its acquisition and operationalization.** Years from now, when a majority of programs have taken digital flight and are seamlessly interconnected, we'll doubtless drop the term *e-Series*. But just like *smart*, the term will have played its part in transforming us into a more competitive acquisition system that accelerates the changes our service Chiefs – and all warfighters – need.

And speaking of that *smart* preface, the same trend *is* happening for Joint All-Domain Command and Control and the Advanced Battle Management Systems (ABMS) as we speak. ABMS Release 1 is inviting us to imagine what a smart-tanker – an internet-enabled data-processing and -relaying node – could bring to future battlefields other than just gas. Smart-fighters, -bombers, -satellites, and -weapons will almost certainly follow suit until *smart* is our military norm, just as it is a commercial one.

With the ABMS "IoT.mil" being birthed, and AI now demonstrated on a military platform for the first time, an era of algorithmic warfare is beginning. Building a truly digital Force – from the early digital foundation, to authoritative virtualizations, to acquisition *e*-Series, to the smart operational systems they must ultimately become – is paramount to feed future war-winning algorithms at digital speeds needed for relevance and, hopefully, dominance. We must all blaze the path together — electron by electron.

IV. THE NEXT *e*-SERIES & CONCLUSIONS

"… there's a difference between knowing the path and walking the path." **_Morpheus*, The Matrix* (1999)**

Based on the criteria outlined here in *Bending the Spoon,* the following programs have also met the *e*-Series bar, in addition to *e*T-7a: Next Generation Air Dominance (NGAD), A-10 Rewing Program, B-52 Commercial Engine Replacement Program (CERP), and Ground Based Strategic Deterrent. Authoritative virtualizations, significantly replacing or truncating real-word activities, and paradigm-shifting performance all attained — congratulations!

Several space programs, like ones at the Space Rapid Capabilities Office, are also off to a great start towards *e*-satellites. I am excited to see how many lessons from aviation convey and how many new ones must be learned for space.

Additionally, Conditions-Based Maintenance Plus has a great chance of becoming an *e*-Series as it expands across our aircraft fleets. If its authoritative virtualizations flip a preponderance of formerly unscheduled maintenance activities to predictive preemptive ones, it will check every box. You do not have to be a platform to be an *e*-Series.

Which takes us to Air Force and Space Force agile DevSecOps software programs: these are *e*-Series by default. But "*e*-software" is a duplicative and unnecessary naming convention.[4] Instead we'll track their progress towards software built to win future algorithmic wars: (i) agile DevSecOps in terms of process, (ii) containerized in terms of technology and technique, and finally, (iii) machine-learning-enabled in terms of training and interaction with data.

[4] With regard to *e-Series naming conventions, w*hether or not we apply the "*e*-" prefix to the name of platform or program will be based on degree of digitization and common sense. *e*T-7a and NGAD merit the formal name change with the "*e*-" in the platform name. Whereas, "*e*CERP" is more appropriate than "*e*B-52". Programs should feel free to add and drop prefixes as convenient based on context. The achievement is more important than the prefix or name.

While none of these initial *e*-Series programs was digitally perfect — nor will there ever be — they all have had real-world, spoon-bending impacts. Start by following their trails before branching off to blaze your own.

Now that we've come to the end, let's circle back to the original questions that likely drove you to this document:

- ***What is digital engineering and an e-Series, really?*** Digital engineering achieves a measure of *authoritative virtualization* that replaces, automates, or truncates real-world activities. *e*-Series, remarkably so. They could be found in any system or function!
- ***Do we need them?*** Yes. Replacing physical activities with virtual ones is a risk, especially for certifying safety and mission success. Virtualization, which culminates in *e*-Series, requires a standard methodology for managing this risk while benefiting from the digital acquisition paradigm shift.
- ***What are their criteria?*** The 14 principles for authoritative virtualizations — and the associated digital building code — are the rules you should follow. Additional e-Series criteria may be found at https://ww3.safaq.hq.af.mil/.
- ***How far must we go to effect a digital transformation for the Air Force and Space Force?*** Practically speaking, you must ensure return on investment for your digital labors, but strategically, you are looking to flip the current acquisition paradigm — exchanging real-world activities with faster, agiler digital ones. The world is increasing unpredictable. Speed and agility are greater weapons — and more to be feared in future militaries — than any individual system we could build.

"Where we go from here is a choice I leave to you."
_Neo*, The Matrix* (1999)

Only one driving question remains, one only you can answer, so I leave the final spoon in your hands:

Who and what is next?

APPENDIX: Summary of Definitions and Principles

Digital Foundation is the infrastructure, policy, training, and culture that enables digital acquisition, digital engineering, and *e-Series*.

Authoritative Virtualization is a digital model of a system that renders its inputs, operational environments, and internal functions and behaviors — along with all subsystems necessary to capture them — such that outputs can be certified as *predictive*.

Digital Engineering must achieve a measure of *authoritative virtualization* that replaces, automates, or truncates formerly real-world activities. *e*-Series, remarkably so.

1. Every system virtualization has a starting point — its basic building blocks — and those building blocks must be quantitatively understood
2. Whether internal or external, if the operating environment affects outcomes or performance, you must either model its physics-based rules or account for it via empirical data.
3. Whatever you are virtualizing, if it contains internal elements that affect outputs, they inherit the same analytic burden, requiring authoritative modeling or verified anchoring data.
4. Just as construction architects and engineers obey building codes certified by local governments, our digital models and infrastructure must obey a similar “digital building code” that is certified by the Air Force and Space Force. This is the pith of “owning the tech stack” and paramount because safety and mission success may depend on our models, in addition to cost.
5. Owning the tech stack — at a minimum, the government reference architecture by which underlying models and software are built — and even furnishing the digital environment itself, especially for software and automation, is the best way to enforce our digital building code, especially at scale.

6. ↘ Automation is a special case of virtualization, but one with special powers to accelerate digital acquisition.

7. ↘ Absent required human judgment or critical thinking, activities we can turn into computerized checklists may be automated.

8. ↘ Even should human oversight still be required for automated outputs, the net personnel savings should still be game-changing.

9. ↘ Authoritatively automating “Government-as-a-Service” functions requires furnishing our tech stack — or at least automation layers — to industry to overcome legal hurdles.

10. ↘ Information Technology, now and in future, is a warfighting system — so it must be bought for overmatch, not as a “technically-acceptable” business commodity.

11. ↘ Once a model is built to digital code, whether or not it is predictive is a matter of test data, which can vary widely between programs and functions.

12. ↘ Polarity of testing changes in digital acquisition. In the past, systems were physically tested to graduate *from* models. Today they should be tested to graduate *to* digital models.

13. ↘ Once you have replaced, truncated, or automated real-world activities via virtualization — *the art of digital engineering* — being declared an *e*-Series is simply a final assessment by the Milestone Decision Authority.

14. ↘ Certified *e*-Series should help propel the Air Force and Space Force’s analog-to-digital metamorphosis.