10.0 Software Development

Introduction

The technologies addressed in New World Vistas are primarily focused on ensuring that on any given day in the future, say July 4, 2025, the Air Force is better prepared to prevail in combat than its adversaries.

However, suppose that on July 5, 2025, the Air Force detects that an enemy has discovered and is ready to exploit a critical weak link in Air Force combat capability. What technology can best enable the Air Force to repair the weak link and disseminate the fix to its full complement of forces?

An electronically disseminated combat system fix would best fit this need. The most powerful and flexible electronic fixes by far are changes to the combat system's software. Enabling the Air Force to design, develop, test, and deploy these fixes is the province of software development technology.

As with other information technologies, software development technology is increasingly paced by commercial investments. From an information warfare standpoint, this must be considered a technology leveler with respect to the Air Force and its adversaries. Not using commercial software development technology would be a competitive disadvantage for the Air Force, but so would a purely reactive use of commercial technology.

This chapter addresses Air Force investment choices, and their likelihood to provide competitive advantage to the Air Force through a pro-active approach to software development technology. It begins with a summary of the major leverage areas and elements of software development technology, along with likely future trends and Air Force implications. It then elaborates on the current Air Force software development status: a mix of complex challenges and significant initiatives. In this context, it then discusses primary opportunity areas for pro-active Air Force investments in software development technology. Finally, it discusses some key research and technology strategy elements which need rethinking for effective Air Force investment in a commercially intensive technology field such as software development.

10.1 Technology Definitions and Leverage Areas

Software development technology consists of methods, processes, tools, and assets enabling faster, cheaper, and better development of computer software. Methods, such as object-oriented design, configuration management, and Cleanroom methods, enable improvements in software design, verification, version control, and quality assurance. Processes, such as incremental, evolutionary, spiral, design-to-cost/schedule, and product line management processes, enable more efficient orchestration of the methods. Tools, such as design, code generation, test, product and process management tools, can completely or partially automate software development functions. Reusable assets, such as specifications, plans, components, and test cases, enable more of a software system to be composed of already-developed program modules.

Faster software development will be particularly important to the Air Force in an information warfare context; it will enable the Air Force and other US forces to "turn their information

systems within their adversaries' turn radius." Cheaper software development will enable the Air Force to contain its escalating software costs; for example, 30% of F-22 EMD-phase costs were software costs. Better software development will, in particular, enable USAF software to achieve greater-than-commercial-grade quality attributes where necessary (e.g., for performance, security, survivability, interoperability, and scalability).

By 2005, powerful user-programming capabilities will enable over 50% of the Air Force workforce to harness packages with millions of lines of code in a few hours to support a special mission need. Thus, for straightforward applications, the Air Force can get the software it wants right away. Getting all of these special applications to interoperate will require new layers of interoperability conventions. The resulting distribution of information processing power will combine with other trends to flatten Air Force organizational structures.

By 2005-2015, advances in software architecture technology will enable concurrent engineering of hardware devices and their software. This enables systems to have better allocation of their functions to hardware, software, and people. It also helps to get software off the critical path in the product cycle. By 2005-2015, semiautomated capabilities for determining the requirements and architecture of complex information-intensive systems will provide even more dramatic improvements.

By 2005, adaptive software systems will be emerging which automatically improve themselves based on observation of usage and data patterns. This will provide the Air Force with rapid, cheaper, more assured adaptation to changing situations (e.g., to adversaries' mobilization patterns). As with other adaptive systems used in information warfare situations, this requires defenses to avoid spoofing the adaptive system. (Techniques for automatic adaption are discussed in Chapter 7, Artificial Intelligence.)

Much of software development technology will be developed and supplied commercially, but some Air Force needs involve Air Force investments to achieve faster, cheaper, and better Air Force software. Examples are capabilities which scale to very large, complex, real-time systems; tools with limited commercial markets, such as weapon system software test simulator/stimulators; and reusable software asset generation for Air Force domains. The bottom-line result of this investment can be a continuous, closed-loop software upgrade capability, enabling the Air Force to adapt to dynamic changes in threats, environments, and technology at a faster pace than its adversaries.

10.2 Current Air Force Software Development Status

Future Air Force roles, missions, and environmental trends involve major challenges for software development technology. Some of these challenges will be well covered by the pace of commercial technology. Others will require particular Air Force attention and investment to ensure that the Air Force has a competitive edge.

The primary challenge areas for Air Force software development are affordability, combat performance, interoperability, information warfare, high assurance, and legacy software. Besides their individual challenges, these areas interact in complex and even more challenging ways. Each area is discussed below, along with its cross-area interactions.

10.2.1 Affordability

Decreasing DoD budgets, broadening threat spectra, and increasing commercial-off-the-shelf (COTS) software capabilities imply that the Air Force and DoD can rarely afford their previous luxuries of full-custom software. For the future, Air Force software development must treat COTS software as the primary driver of its information processing capabilities.

In Air Force-specific applications areas (combat aircraft, missiles, sensor processing, etc.), the Air Force can no longer afford its previous luxuries of stovepipe software systems with redundant capabilities. Instead, the Air Force needs to develop software product line architectures for its families of applications, and to reuse software assets across each product line.

10.2.2 Combat Performance

A totally COTS-driven approach to Air Force software would leave Air Force warriors with no competitive advantage with respect to adversaries, who will have access to the same COTS capabilities. Instead, the Air Force needs to identify areas in which better-than-COTS performance will provide significant combat advantages, and to invest proactively both in developing Air Force-unique software capabilities and in influencing the COTS marketplace toward compatibility with and support of Air Force software assets.

Particularly important software areas with respect to combat performance are embedded real-time software, high-performance sensor processing, intelligent software, and distributed information management enabling anywhere-anytime access to decision-critical information. To ensure the best "defense against tomorrow," this software also needs to be rapidly and reliably modifiable, including warrior-tailorability wherever feasible.

10.2.3 Interoperability

In developing product line architectures, the Air Force needs to ensure that these do not become macro-stovepipes. Thus, Air Force software development and product line management needs to operate in the context of an Air Force enterprise architecture ensuring interoperability of combat systems. This enterprise architecture especially needs to evolve proactively with respect to the evolution of COTS software.

10.2.4 Information Warfare

The need to prevail in information warfare situations places particular stress on software development for the capabilities above. Not only does the Air Force need rapidly-modifiable adaptive software providing anywhere-anytime information to its own warriors, but also it needs to develop this software in ways which ensure its integrity and non-compromisability, and which selectively deny its availability to adversaries.

10.2.5 High Assurance

Implicit in the challenges above is the further challenge to provide high levels of assurance that Air Force software is secure, survivable, safe, reliable, interoperable, and scalable to high performance for large systems in crisis situations. These levels of assurance frequently exceed those considered adequate for commercial marketplace competitiveness.

10.2.6 Legacy Software

Further, the Air Force does not have the luxury of developing all of this software from a clean slate. The Air Force owns one of the world's largest inventories of antique legacy software running its current operations. Thus, attractive software development strategies such as incremental and evolutionary development are seriously constrained by Air Force needs to ensure continuing combat capability.

10.3 Current Air Force Initiatives

Several current Air Force and DoD initiatives are addressing portions of these challenges in effective ways. Recent acquisition initiatives are focused on replacing government bureaucracy and micromanagement by best commercial practices, enabling improvements in affordability and cycle time. Several Air Force and ARPA initiatives in domain specific software architectures (DSSA's) have created success stories with demonstrated returns on investment, particularly the AF/ESC PRISM initiative and the AF/ARPA STARS initiative for Cheyenne Mountain software.

The AF/SC Horizon initiative to develop and sustain a living Air Force software enterprise architecture has made significant progress, in concert with such DoD initiatives as the Technical Architecture Framework for Information Management (TAFIM). The DoD MIL-STD-498 effort, now culminating in IEEE standard 1498, will successfully harmonize, modernize, and replace two outdated standards for developing mission critical software (DoD-STD-2167A) and corporate information management software (DoD-STD-7935A). Air Force use of the Software Engineering Institute's Capability Maturity Model and related software capability models in source selection have significantly reduced risks of selecting immature software contractors.

These initiatives need to be strongly supported and extended to other Air Force domains; otherwise, Air Force software development will be increasingly cumbersome and constrained by its legacy software and culture. However, to be pre-eminent in the aerospace-cyberspace world of the future, the Air Force needs to develop a proactive investment and experimentation strategy to help shape and to capitalize on upcoming software trends. The main software development technology opportunity areas are discussed next.

10.4 Software Development Technology: Opportunities for the Future

Here are the primary opportunity areas in which Air Force investments at the margin in software development technology can provide significant Air Force competitive advantages.

- User Programming: the ultimate in rapid software development to meet user needs.
- Software Architectures: a key to user programming, as well as broader forms of rapid prototyping and rapid application development.
- Assurance Technology: analysis, development, and verification capabilities for the assurance of Air Force-critical software attributes, particularly security, safety, survivability, combat performance, interoperability, and crisis assurance.

- Software/System Concurrent Engineering: enabling integrated product teams to explore many software/system tradeoffs collaboratively, and to rapidly develop integrated software/system capabilities.
- Adaptive Software: software which can automatically improve itself based on observation of its usage and data patterns.

10.4.1 User Programming

For some applications domains, commercial capabilities are available which enable users to directly compose software applications in a few hours by specifying a number of options, parameters, and simple rules. Examples include spreadsheets, business fourth-generation languages, and tailorable packages for such applications as financial analysis, accounting, and inventory control.

Some similar capabilities are emerging in such Air Force areas as transportation planning, signal processing, and simulation. Air Force investments in further formalizing these and other Air Force-critical applications domains to create user programming capabilities will have significant payoffs in rapid user-responsive software development. However, the prospect of having many users developing special applications, and then discovering that many of them need to interoperate, also implies a need for counterpart investments in higher-level applications interoperability conventions.

10.4.2 Software Architectures

The primary key to user programming is the domain specific software architecture (DSSA), which provides the software structure for a family of software products in an application domain such as avionics, command centers, or signal processing. The DSSA determines a set of common application interface specifications, around which reusable software components can be composed. It also determines a set of common application concepts around which special purpose languages for user applications generation can be developed. It is based on a domain ontology, which organizes knowledge about the domain and its information processing aspects.

Besides application generators, which necessarily cover only a subset of applications in the domain, a software architecture provides support for rapid prototyping and composition of intermediate-scale applications, and for efficient development of large applications via reusable components.

Software architecture technology also provides complementary artifacts such as architecture definition languages and architectural styles. These capabilities are just emerging, but they strongly enhance the analysis of alternative software architectural options for such properties as security, safety, reliability, interoperability, and performance (The nature of a successful architecture is generally driven by such nonfunctional requirements; it is difficult to retrofit attributes such as security and fault tolerance into an application which did not consider them in its architecture). Strengthening these architectural techniques will help the Air Force to deal with its needs for critical nonfunctional requirements.

10.4.3 Assurance Technology

The ability to specify and analyze architectures for Air Force-critical nonfunctional requirements is one avenue of assuring systems with these attributes. Other complementary assurance technologies involve specification of desired attribute levels, including specification of the system's operating environment; and capabilities for developing, verifying, and evaluating high-assurance systems.

10.4.4 Software/System Concurrent Engineering

The Air Force is deriving considerable value from concurrent engineering of systems by integrated product teams (IPT's) involving multiple classes of system specialists and stakeholders (users, customers, developers, maintainers, interoperators). However, technology support for such concurrent engineering is still at a low level, particularly for integrated hardware/software engineering, large system development, and groupware capabilities for collaborative work.

Particular technologies that could enable more rapid and cost-effective Air Force software/system development are model-driven concurrent engineering (based on models of the domain, the system objectives, and the collaboration approach); hypercode (hyperlinking of software and system artifacts such as plans, specifications, code, and test procedures); and advanced processes and metrics for concurrent engineering (enabling both rapid execution and disciplined management of software/system development).

10.4.5 Adaptive Software

The artificial intelligence and agent technologies discussed in Chapters 7 and 5 respectively can also be focused on problems of adapting software to changing situations. For example, decision support systems could analyze patterns of decision information requests (exception reports, graphic displays, correlations) and restructure the software to better provide the most frequently requested functions. Such capabilities could both reduce software maintenance costs via automation, and effect software improvements which otherwise might not have been identified.

As with other adaptive systems used in information warfare situations, such software could be spoofed by patterns of requests for irrelevant or misleading information. This implies the additional need for defenses to avoid spoofing the adaptive system.

10.5 Rethinking Air Force Software Research and Technology Strategies

The software research and technology associated with a very large and increasing software marketplace requires the Air Force to rethink its own traditional software research and technology strategies. Focusing on Air Force software needs (e.g., for distributed processing) independent of commercial technology trends runs major risks of ineffective or commercially incompatible results. On the other hand, a purely reactive Air Force strategy would leave the Air Force with no competitive advantage with respect to other COTS consumers (and sometimes a disadvantage due to legacy software): a weak position from the standpoint of information warfare.

Thus, a new research and technology strategy is needed for Air Force software technology (and increasingly for other areas with growing commercial capabilities). For software, the major elements of this strategy include:

- Domain pre-eminence
- Living enterprise architecture
- Focus on Air Force-critical niches
- Pro-active commercial influence
- Close research-technology coupling
- Continuing closed-loop exercise and experimentation

10.5.1 Domain Pre-Eminence

Investments in software product line architectures, tools, components, and techniques for combat-critical domains such as sensor processing, data fusion, C3I, and combat platforms will give the Air Force a competitive edge with respect to adversaries possessed only with commercial capabilities.

10.5.2 Living Enterprise Architecture

Air Force combat interoperability requires that Air Force software product lines avoid becoming incompatible stovepipes. This requires a living enterprise architecture such as the Horizon initiative is addressing. Research and technology investments are necessary to keep the enterprise architecture in step with commercial technology—and further, that Air Force enterprise architecture needs are pro-actively inserted into commercial technology development and standards activities.

10.5.3 Focus on Air Force-Critical Niches

Besides combat-critical applications domains, the Air Force has needs which go considerably beyond those of mainstream commercial customers. The primary example area involves the high-assurance capabilities discussed above. Research and technology investments in these areas can also enable the Air Force and DoD to play an influential role in guiding commercial standards and technology.

10.5.4 ProActive Commercial Influence

Commercial products and standards will tend to focus on mainstream market needs, in which intermediate assurance levels will meet most customers' requirements. Often, however, commercial vendors will see a downstream advantage for incorporating more advanced Air Force and DoD needs into their next-generation products. A good past example is in software security, in which commercial operating systems, database management systems, and network software have been developed to DoD "Orange Book" security standards with the aid of DoD-developed security tools. To ensure this commercial relevance and influence, Air Force laboratories need to devote considerable effort to experimentation with and influence of advanced

commercial software technology and emerging standards. A significant current example is the Common Object Request Broker Architecture (CORBA).

10.5.5 Close Research-Technology Coupling

The rapid pace of commercial software technology means that Air Force software research and technology investments need to be well coupled with complementary software research across DoD, and also well coupled with non-DoD research and technology. Particularly important are experimental applications of new software technology to evolving Air Force software-intensive product lines such as those developed by PRISM at ESC and STARS at Space Command.

10.5.6 Continuing Closed-Loop Exercise and Experimentation

Besides development and deployment of new technologies, the Air Force needs a much stronger program of continuing exercise of its software-intensive systems, with feedback of weak and strong points to both the operational community and the technical community. This will be especially important for Air Force systems highly exposed to information warfare threats, in which feedback from continuing penetration teams can focus investments in software security, for example. This kind of closed-loop operation is another reason for the Air Force and DoD to evolve their contracting approaches toward long term customer-supplier relationships with strongly shared context and objectives.

10.5.7 Bottom Line

Changing Air Force software research and technology strategies will not be easy. But some of the initial Air Force success stories indicate that it is achievable. And with the revised strategies, when the tomorrows come, and the Air Force needs software quick-reaction capabilities, Air Force software development technology will be ready for them.