

Enterprise Architectures

A Just-in-Time Approach for Decision-Making

Carlos E. Martinez

**The MITRE Corporation
7515 Colshire Drive
McLean, Virginia 22102**

March 24, 2014

Approved for Public Release; Distribution Unlimited. 14-1210

Abstract

This paper presents a summary of the range of potential uses of enterprise architectures (EAs), some of the challenges facing the users of EAs, and practical approaches for developing them incrementally over time to provide “just in time” utility to decision makers.

Background

There were two events over the last couple of decades that have significantly influenced my thinking about enterprise architectures and their use.

The first occurred back in the mid-1990s, in the midst of the war in Bosnia-Herzegovina. I had been working for a company who supported the Department of Defense in building information system architectures for the command, control, communications, and intelligence (C3I) community. In particular, we were charged with building the intelligence architectures for all of the overseas combatant commands such as European Command, Pacific Command, etc. We had been at this job for quite some time and over the years had developed a large number of architecture documents that described the as-is and to-be intelligence architectures for the commands. The effort involved many dozens of contractor and government staff across the globe who gathered on an annual basis to compare notes and share experiences. One annual conference (c. 1994) was in the Washington, DC area at which we were honored to have Dr. Barry Horton, the DoD’s Principal Deputy Assistant Secretary of Defense for Command, Control, Communications and Intelligence (ASD C3I), in attendance. He listened for hours to the various presentations describing all the architectures developed at the various commands. In response, Dr. Horton asked a very pointed question which was something like, “Having spent tremendous amounts of time and money building these architectures, are they of any use to you?” Realizing that our collective future employment hung on the answer, we all listened quietly as the Navy Captain representing U.S. European Command responded. He started out by saying that when the war in Bosnia-Herzegovina broke out, they turned to their architecture documents to figure out how to configure their C3I systems to meet the situation. Our hearts sank as he said that no architecture document developed to date provided the answer they needed. He went on to say, however, that the fact that they were immediately available on the shelf provided the Command with a ready source of “building blocks” of information (e.g., organizations, missions, information needs, systems, etc.) from which they could pick and choose and quickly rearrange to reconfigure their C3I systems to meet the new need, and that they would not have been able to respond as readily as they did without them. We breathed a collective sigh of relief.

The second event took place in the Pentagon in the early 2000s. I had been involved in the planning, design, and development of a variety of intelligence and command and control system architectures for the Department of Defense for several years. Consequently, following 9/11 I was recalled to active duty to help forge a closer relationship between the Air Force’s command and control community and its intelligence community. My duty assignment was at the Pentagon in the Air Force’s newly formed Directorate of Warfighting Integration. As a relatively new one-star general, my duties often included filling in for senior officers when they were called away on pressing business. One day I was asked to receive a previously scheduled briefing from an out-of-town team that had been working on a particular command’s mission architecture and had come to Washington to brief our directorate on their on-going

work. The team explained their overall objective and scope of their work, described their methodology, and presented some examples of operational and systems view products that they had begun to develop in accordance with the then governing document for developing and publishing architectures, *The CAISR Architecture Framework* [2], which was subsequently replaced by *The DoD Architecture Framework (DODAF)* in 2003. They explained that they had segmented their command's architecture into portions principally based on different operational missions, and finished up by presenting their proposed schedule for completing the command's comprehensive architecture. Their schedule, presented as a Gantt chart, showed the sequential build of the architecture over a period that spanned approximately five years. They had just finished building the products for the first segment and were getting ready to proceed to the next. Not seeing any obvious rationale for the proposed schedule, I asked them on what basis they had selected the order for building the various segments. They responded that they had estimated how much work would be required to build each segment and then ordered them such that they could start with the easiest one first, then proceed to more complicated ones, leaving the most challenging one for last. I pointed out that the Air Force might need to make acquisition decisions in the coming year regarding mission segments that would not even be started till four years out and asked them if it might not make more sense to prioritize the development of the segments in such a way that they could better inform upcoming decisions. They looked at me with blank expressions and said they had not considered that possibility at all.

Introduction

These two incidents form the basis for the propositions presented herein—that architectures may serve many (sometimes even unexpected) needs, and that building a comprehensive set of architecture artifacts to meet all possible needs is not practical or affordable. So, what one needs is a structure against which one can build an architecture repository that is populated with artifacts that are built “just in time” to meet specific decision making needs. By “just in time,” however, I do not mean to imply that the artifacts are necessarily dynamic or subject to frequent change. What I mean is that while these artifacts may be fairly static and serve long-term purposes, you should not spend the time or money to develop them until you actually need them. The discussion that follows provides an overview of the decision-making cycle that architectures support, describes a selection of the types of information needed to support decision makers, identifies challenges that decision-makers often have in making use of architecture information, and presents a strategy for building a repository incrementally to meet evolving needs.

Discussion

Most organizations, regardless of what line of business they are engaged in, if they expect to have any longevity must evolve over time to adapt to changes in their business environment. To do so, they generally follow a logical process that consists of determining what changes they need to make to meet future needs, planning how and when to implement those changes, implementing the planned changes, and then operating with the changes in place.

Inspired by other commonly accepted concepts like the Observe-Orient-Decide-Act (OODA) loop developed by USAF Colonel John Boyd [1], I like to call this cyclic process, as depicted in Figure 1, the Analyze-Plan-Implement-Operate (APIO) Decision Cycle, which forms a continuum for the life of the

organization, and often consists of concurrent activities within each portion of the cycle. This is particularly true for large enterprises like major corporations or government organizations, which consist of many subparts each of which must evolve at a different pace to meet future needs.

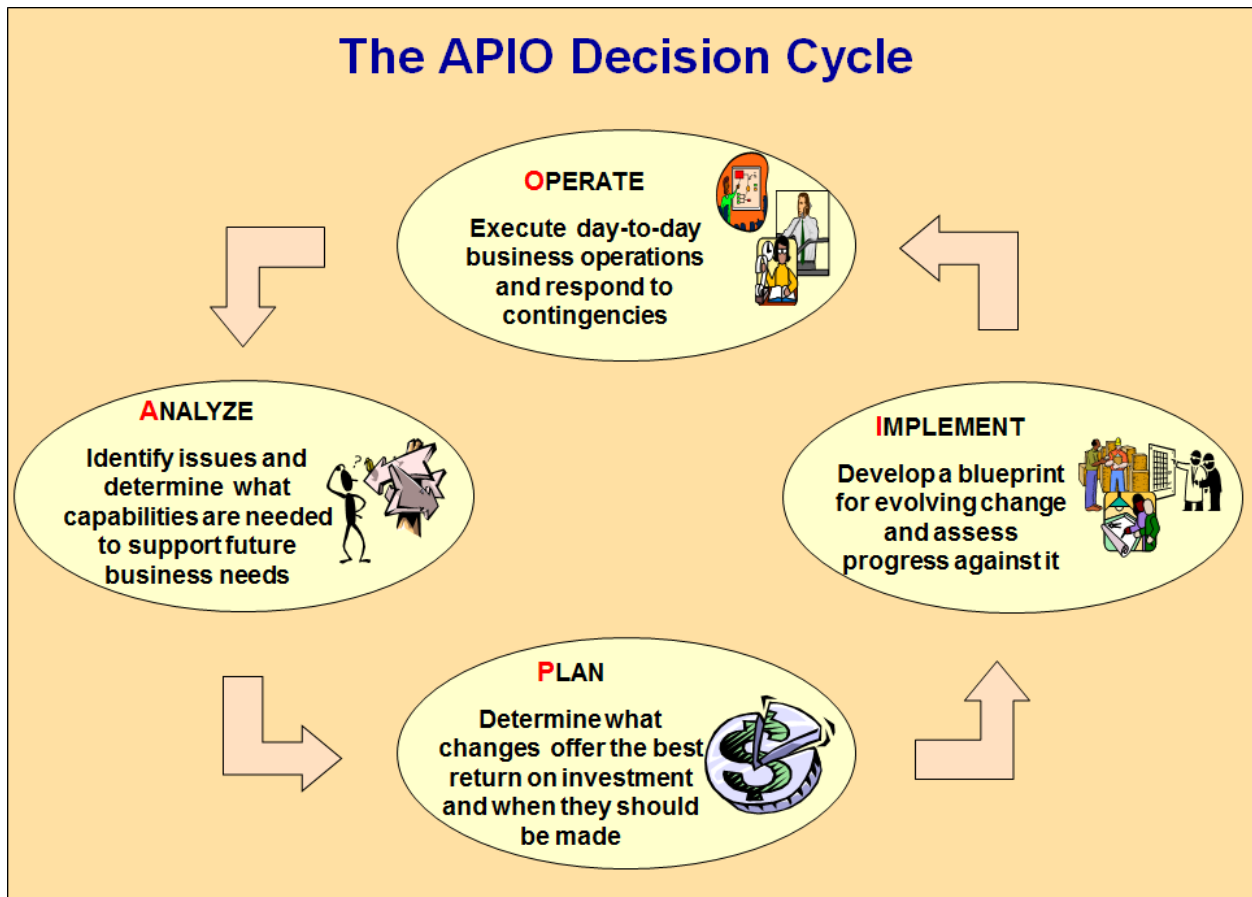


Figure 1. The Enterprise’s APIO Decision Cycle

At each step, the organization must make informed decisions aimed at minimizing risks to its long-term viability. To support these decisions, the organization needs access to a variety of types of information such as:

- Customer desires and needs
- Environmental, cultural, and legal constraints
- Business objectives and strategies
- Operational concepts
- Business processes and procedures
- Organizational structures
- Human resources
- Facilities and equipment
- Financial resources
- Investment plans and schedules
- Performance measures: rework, volumes, cycle times, etc.

Much of this information is traditionally captured as artifacts within the enterprise architecture of the organization, and as depicted in Figure 2 should be made available to support the APIO decision cycle.

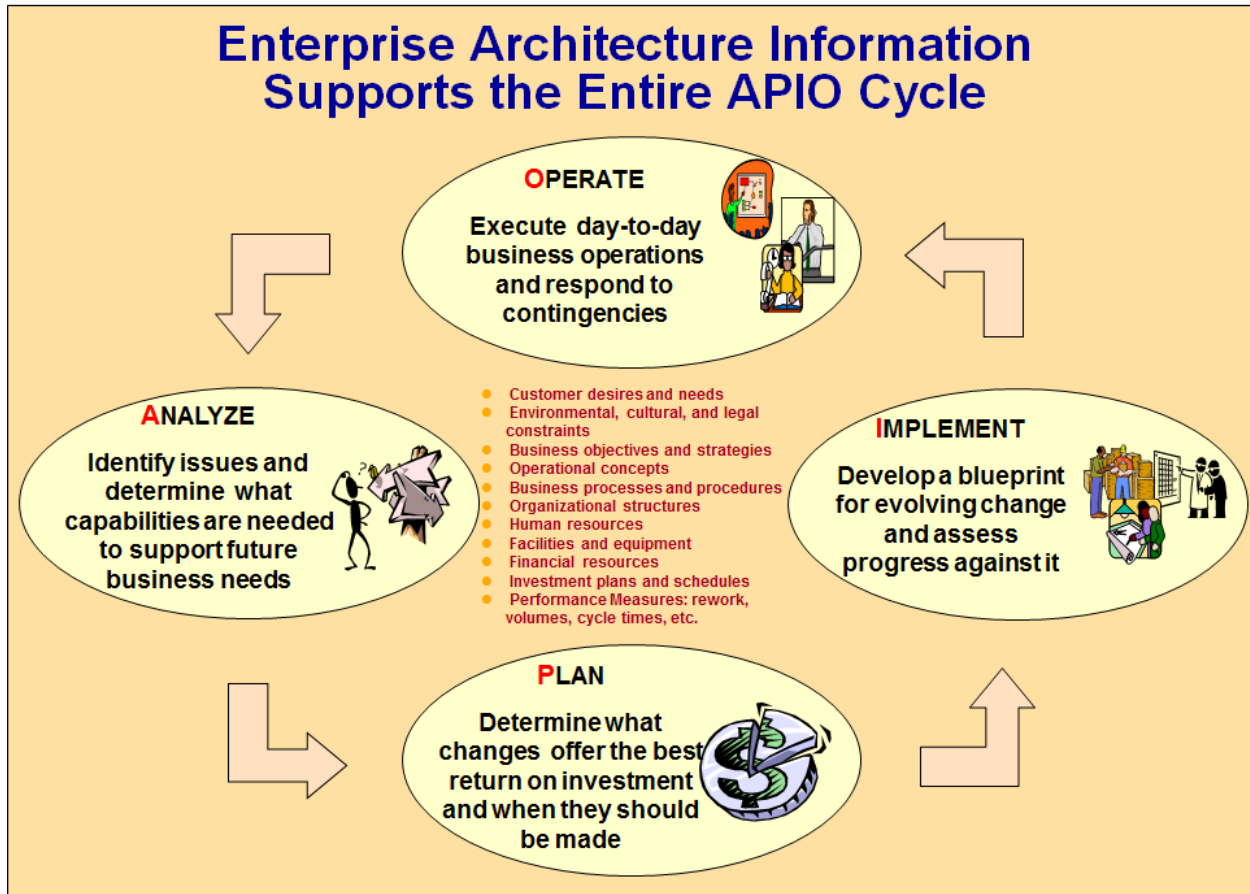


Figure 2. Enterprise Architecture Information Support to APIO Cycle

For example, as part of conducting Analysis, decision makers must assess the ability to meet today’s and tomorrow’s mission needs to identify gaps, shortfalls, and redundancies. In support of Analysis, they need EA information such as:

- Future mission concepts and performance objectives
- As-Is architecture (business processes and resources)
- To-Be architectures (business processes and resources)
- Approved and funded time-phased planned improvements to As-Is architecture
- Performance metrics associated with As-Is, funded improvements, and To-Be architectures
- Executable performance models

As part of Planning, decision makers need to conduct analysis of alternatives (AOAs), develop investment justifications for modernization initiatives, and develop transition strategies and roadmaps. In support of Planning, they need access to EA information such as:

- Technology forecasts and assessments

- Initial and recurring costs, to include labor and facilities for modernization alternatives
- To-Be architectures (business processes and resources)
- Expected benefits of alternatives
- Linkages to other organizations' related architectures and their transition strategies and roadmaps

During Implementation, decision makers need to design, develop, test, and field planned modernization improvements, develop training plans and schedules, and conduct initial training. In support of such Implementation activities, they need access to EA information such as:

- As-Is architectures (business processes and resources)
- To-Be architectures (business processes and resources)
- Transition Strategies and Implementation Roadmaps and Schedules
- Expected performance objectives and acceptable thresholds

Finally, in the Operations phase, decision makers must train new personnel, monitor performance of day-to-day operations, develop contingency plans, and respond to contingencies. In support of Operations, they require EA information such as:

- As-Is architectures (business processes and resources)
- Expected performance objectives and acceptable thresholds
- Executable performance models

Where does this EA information normally reside? It resides in many places such as:

- Document libraries (e.g., policy memos, master plans, operating manuals, etc.)
- Databases (e.g., financial, human resource, supply and logistics, etc.)
- Architecture artifacts developed in accordance with various frameworks (e.g., Zachman, FEA, TOGAF, DoDAF, etc).

Furthermore, different “communities” own various pieces of EA information. For example,

- Personnel data is owned by Human Resources
- Cost information is owned by Finance
- Concepts of Operation are owned by Operations
- Infrastructure information is owned by the Chief Information Officer (CIO)

Consequently, the decision-makers who need the EA information are faced with three typical problems. The first is that the decision maker is not aware the information even exists, and is forced to make decisions without it. The second is that the decision maker knows that the information exists but has difficulty in obtaining it, or in obtaining it in time to support critical decisions. The third is that the information is available to the decision maker but it is in such a form that it is difficult to comprehend or use.

These problems exist principally because each community compiles its data for different purposes, using different timeframes, at different levels of detail, and in different forms and formats. The bottom line is that it's hard to find and make sense of all the information that exists.

Ideally, as depicted in Figure 3, decision makers would like to have an EA repository at their fingertips that contain all the necessary information to support the entire Analyze-Plan-Implement-Operate cycle.

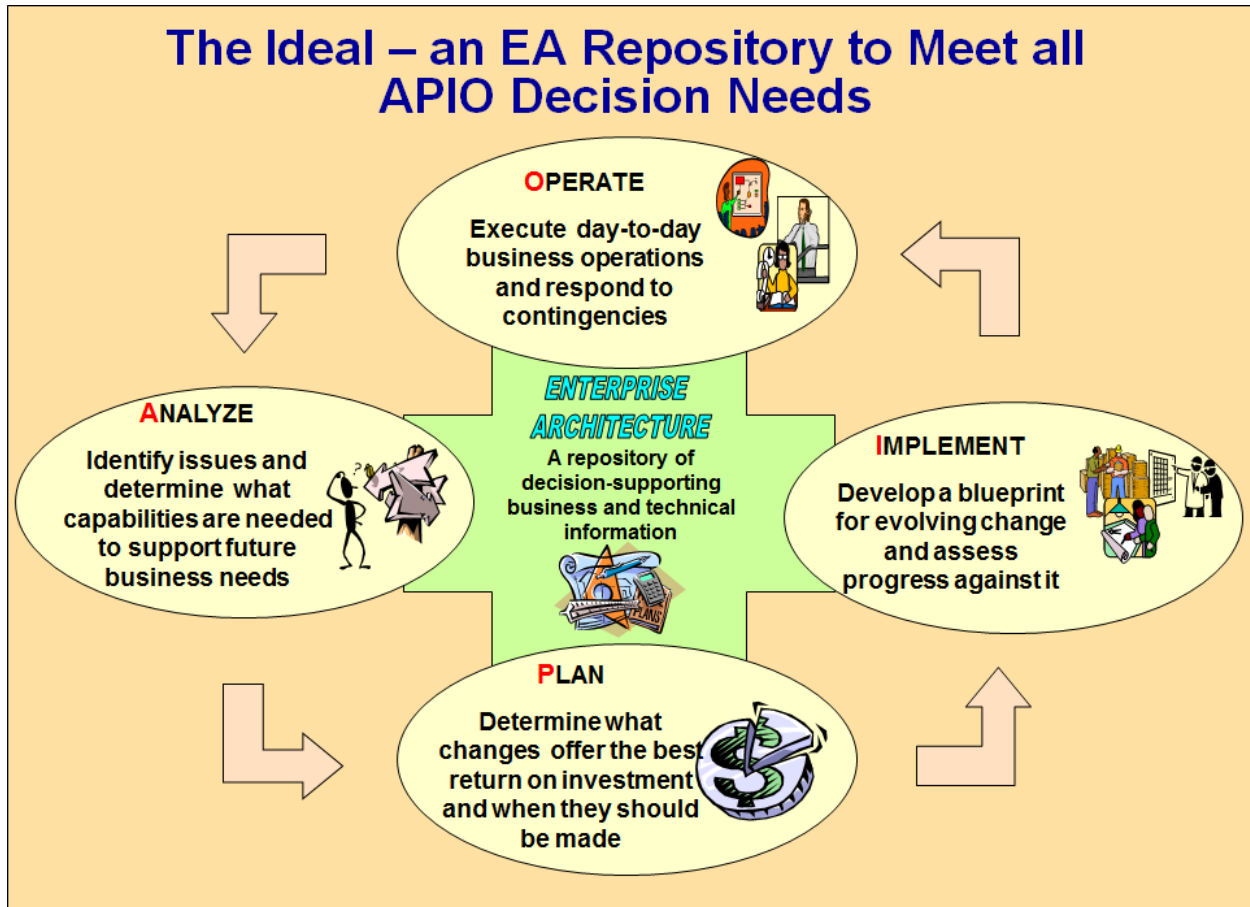
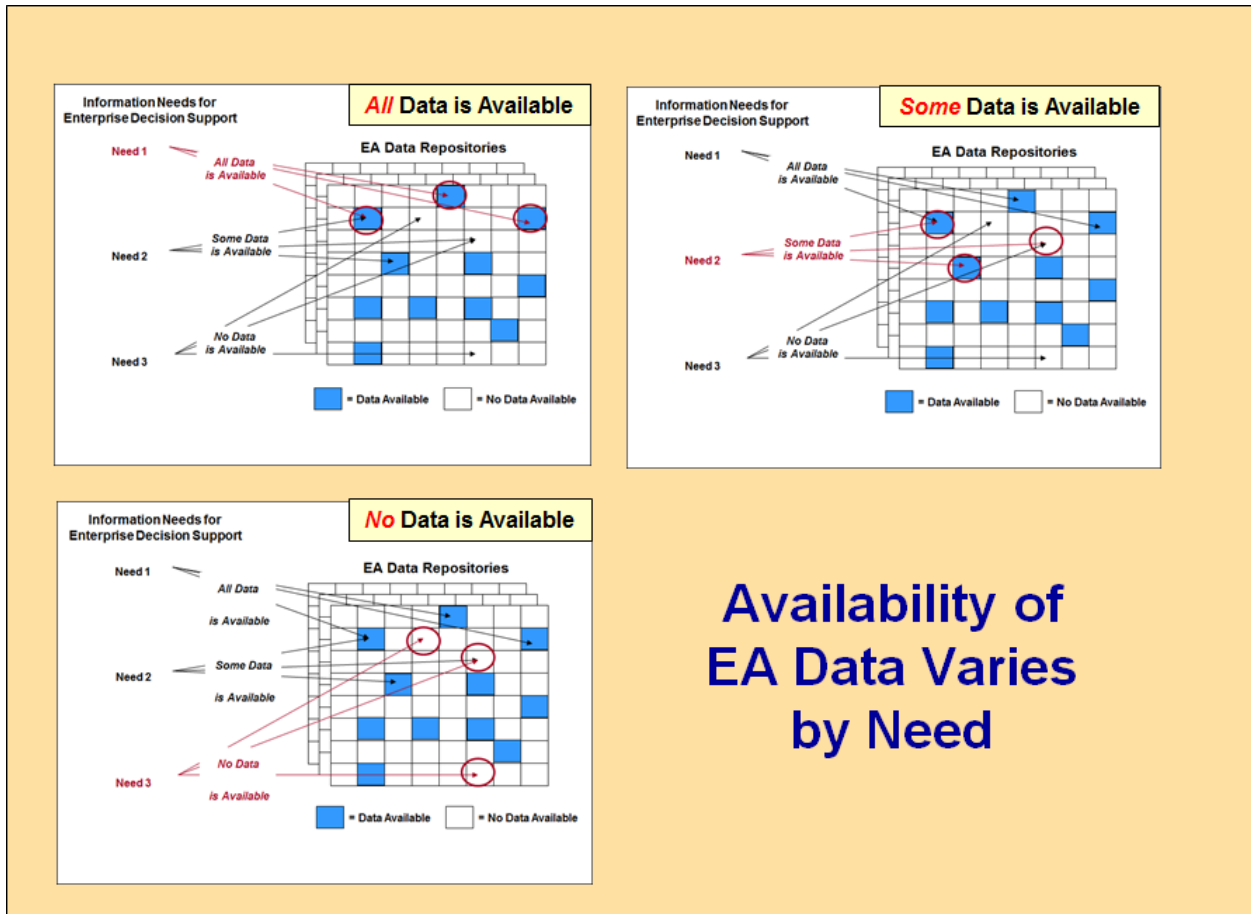


Figure 3. A Comprehensive EA Repository to Support Decision Making

The various EA Frameworks have provided guidance for capturing and documenting EA information. The DoDAF version 2.0 provides a considerable expansion over earlier versions regarding the types of information that should be included in an EA and specifically recognizes the many uses to which EA information can be put. However, this Framework stops short of describing what types of information can help satisfy which user needs.

Even if we had a mapping of user needs to the appropriate EA information to support them, one would still be faced with the problem, as suggested by the mappings in Figure 4, that not all information that might possibly be needed would ever be available in an EA repository because there is simply not time or money in the world to fill that repository and keep it current.



Availability of EA Data Varies by Need

Figure 4. Uneven Availability of EA Information

Consequently, what is needed is a complementary strategy for developing EA information in direct response to decision making needs as depicted in Figure 5, and not on developing comprehensive sets of EA views and artifacts.

EA Development Should Be Driven by Decision Support Needs

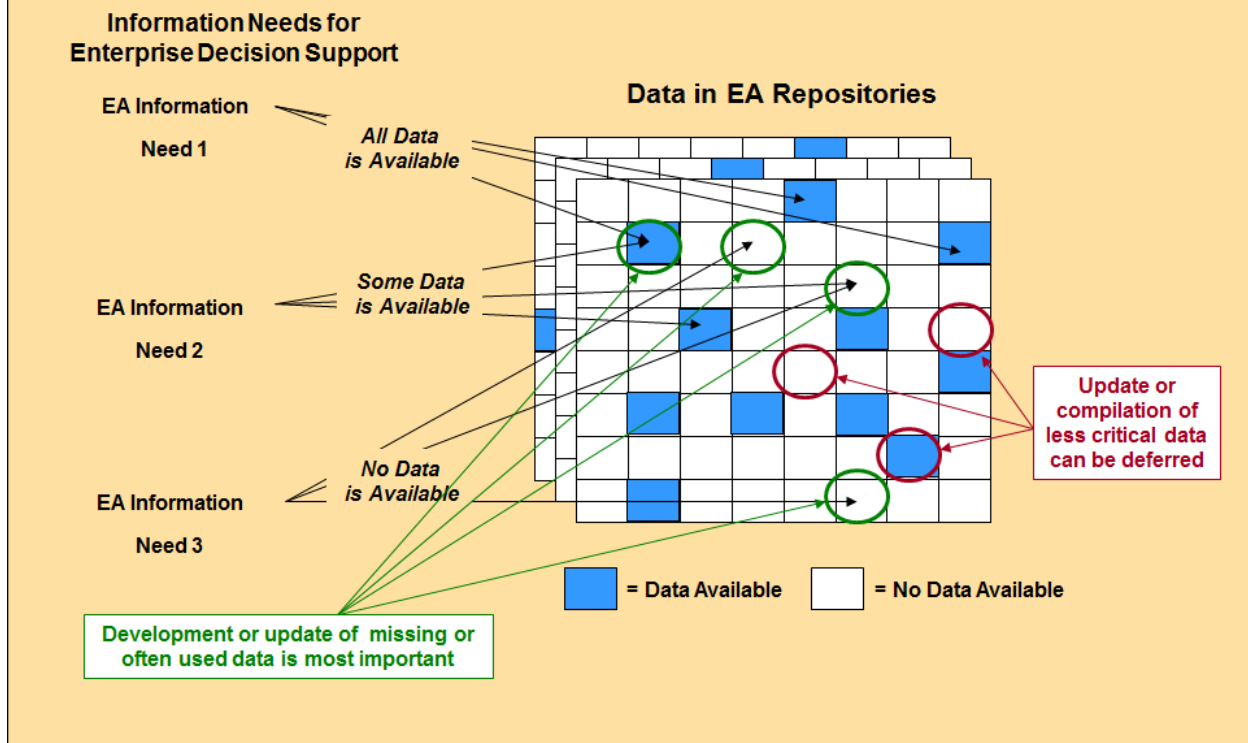


Figure 5. Decision Need Driven Development of EA Information

This strategy should focus development of EA artifacts on forthcoming issues and decisions to be made. This is what I refer to as “just in time” architecting. This should be done by:

- Identifying the types of EA data needed to support particular decision making needs
- Determining the level of detail needed to provide usable answers
- Finding out if the data is already available
- If not, determining timelines and costs to obtain the necessary data
- Letting decision makers know what data is available and when
- Developing or updating the data needed to answer the most pressing issues first

Furthermore the strategy should embrace the principle of providing information to decision-makers at the earliest opportunity during its development. This provides the opportunity for adjusting EA development in response to what is usually a constantly changing decision environment.

Conclusion

By recognizing that EA supports the full organizational decision cycle of Analyze-Plan-Implement-Operate, EA developers should endeavor to identify the key decisions that may benefit from EA data then time EA development to coincide with the most critical decision needs. Furthermore, EA developers

should make the EA data available to decision makers at the earliest possible time to enable quick feedback and enable “re-vectoring” of EA efforts. Doing so should result in more cost-effective use of scarce architecture resources and reduce development of the “shelfware” that has characterized so many EA efforts over the last few years.

References

1. Department of Defense. *Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) Architecture Framework, Version 2.0*, December 1997.
2. Wikipedia. *OODA Loop*, http://en.wikipedia.org/wiki/OODA_loop.