Systems Engineering and Enterprise IT

Systems Engineering and Enterprise IT

Lead Authors: Chuck Walrad and Rich Hilliard

SE is integral to the effective functioning of Enterprise Information Technology (EIT) organizations, which are devoted to the successful realization, use, and retirement of engineered systems that are largely based on information technology. Effective means to develop, manage, and support EIT organizations are described in the Enterprise Information Technology Body of Knowledge (EITBOK), which was jointly developed by the IEEE Computer Society and ACM, and is maintained as a public wiki by the IEEE (IEEE/ACM n.d.). This article largely explains EIT organizations through the lens of the EITBOK and how it relates to SE principles, practices, and activities as described elsewhere in the SEBoK.

☐

Contents

Enterprise IT Introduction

Systems Engineering is integral to the effective functioning of EIT organizations, whose work is devoted to the successful realization, use, and retirement of engineered systems that are largely based on information technology. Examples of EIT systems include enterprise resource planning, supply chain management and customer relationship management systems, accounting, finance, inventory and order management, marketing, planning, purchasing, and sales. In the late 1980s and throughout the 1990s, EIT organizations became increasingly aware that the success rates of IT projects were very low, and that their enterprise customers were increasingly unhappy with their EIT investments. The Standish Group sought to understand what was going on. Their findings were presented in Standish (1994). The Standish Group research shows a staggering 31.1% of projects were canceled before they ever got completed. Further results indicated 52.7% of projects cost 189% of their original estimates. The cost of these failures and overruns were just the tip of the proverbial iceberg. The lost opportunity costs were not measurable but could easily be in the trillions of dollars.

As a result, commercial IT organizations through the auspicious parallel emergence of the Total Quality Management movement (ASQ n.d.) and the increasingly widespread development of application development methodologies by IT management consulting firms, recognized the need for the principles of SE and cross-functional teams (sometimes called Integrated Product Teams (IPTs)). It should be noted that there is considerable overlap today between SE and Project Management (PM) (see Systems Engineering and Project Management). Note that the following business management activities can be supported by Enterprise Systems Engineering (ESE) activities:

- mission and strategic planning
- business processes and information management
- performance management
- portfolio management
- resource allocation and budgeting
- program and project management

These same areas are often supported by Project Management Offices. However, since a key difference between PM and SE is that the latter includes technical expertise in the product/system of concern, focus should be on SE and IPTs.

With IPTs, each product is shepherded from concept through delivery and support to retirement by a cross-functional team that includes representatives from each area of responsibility, such as customer requirements development, design and development, test, configuration management, release engineering, delivery, and support.

As enterprises have come to depend more and more on technology to perform effectively, EIT organizations have matured, often due to the influence of IT consulting firms that make it their business to keep abreast of best practices.

EIT Concepts

Much of this section builds on concepts found in the EITBOK, the *Guide to the Enterprise Information Technology Body of Knowledge* (IEEE/ACM n.d.), and on general SE concepts elsewhere in the SEBoK.

An enterprise may be viewed from various perspectives: such as a system or system of systems (SoS):

- system-like enterprise: enterprises, like other systems, may have a tight coupling, single-executive function;
- system of systems-like enterprise: may be a loosely coupled group of organizations with blurred boundaries and (yet) with (some) shared outcomes.

The choice of perspective may vary with many factors: the enterprise itself, the mission or purpose, or the particular engineering task at hand. The enterprise can be viewed as a system with inputs and outputs in relation to the outside world, or as a system of systems interacting through its individual component inputs and outputs.

An enterprise usually involves one or more organizations but need not be identified with the organization. As the scope of human and organizational participants of the enterprise widen, there may be no single point of control. For example, an extended enterprise could include a company, its suppliers and its customers.

Whatever perspective is taken (system-like, SoS-like, other), there are common key ingredients of any enterprise: capabilities, individual competencies, and the organizational design. Capabilities can be further categorized into organizational, system, and operational. These capabilities determine what the enterprise is able to produce, which may be external offerings or internal mechanisms. Operational capabilities create services and products which in turn produce value as determined by stakeholders. The capabilities of an enterprise are realized through various enterprise elements including hardware, software, personnel, facilities, data, materials, techniques, and even services which include soft items such as processes, principles, policies, practices, organizations, doctrine, theories, and beliefs.

EIT capabilities exist through lifecycles spanning from creation through eventual replacement and disposal. System Life Cycle Models aid in understanding, designing, planning and managing the capability of interest. As stated in the EITBOK, "A life cycle for a system generally consists of a series of stages regulated by a set of management decisions which confirm that the system is mature enough to leave one stage and enter another. Basic life cycle terms and concepts are discussed in System Life Cycle Approaches. Some typical life cycles are discussed in System Life Cycle Models.

A life cycle-based approach can be deployed across the

several levels and used to manage evolution. These levels are:

- the enterprise itself
- the processes and organizations constituting the enterprise
- the systems enabling the capabilities, services or products provided.

It is important to distinguish between *product* and *project* lifecycle models. A product's life cycle extends from its initial concept approval, through development and distribution, to its retirement. This is the cycle that is managed within an Enterprise IT organization, since EIT must manage the hardware or software product from its introduction (in the case of acquired capabilities) or from its concept approval and development (in the case of capabilities it builds), through its deployment, maintenance and eventual retirement or withdrawal. Each of these stages is usually managed as a project or a series of projects. However, most discussions of life cycles within software and systems discussions focus on *project* life cycles. See, for example, the *Software Extension to PMBOK Guide* (PMI 2013).

System of Systems Context in EIT

The phrase "Enterprise IT" (EIT) is often used both to refer to the technology organization/business unit within the enterprise and to the enterprise's information technology systems. This can be confusing. Here EIT refers to technology systems. When referring to the organization, "Enterprise IT organization" or EIT organization is used instead.

Enterprise IT organizations are responsible for complex socio-technical systems. Socio-technical systems in a business context (thus, EIT) contain technical elements (software, hardware, networks, storage, databases, etc.) but they also contain essential human components and necessary business processes. As covered in Introduction to Systems Engineering Fundamentals, they exist in an engineered System Context that is a set of system component interrelationships within a real-world environment:

This includes where problems come from and how they are defined, how candidate solutions are identified and selected, how to balance technology and human elements in the wider solution context, how to manage

the complex organizational systems needed to develop new solutions, and how developed solutions are used, sustained and disposed of. SE should consider the full engineered system context so that the necessary understanding can be reached and the right systems engineering decisions can be made.

Socio-technical systems also have the following key characteristics: *Openness* in that they strongly interact with their environments, and *unfinished* in that they evolve over time. A complex of EIT systems is essential to the successful functioning of the people in the enterprise. They rely on EIT's technology for communication, paying employees, people management, business measurement, and often for managing supply chains. Thus, "EIT needs to be viewed as a strategic partner to all the other functions and business units within an enterprise. This means that the results produced by EIT along with its effectiveness and efficiency are all critically important to the success of the enterprise." (IEEE/ACM n.d. (a))

To accomplish this mission, the EIT organization builds and acquires new systems or system components, integrates them into the mega SoS, and maintains the integrity and interoperability of the individual components as well as the integrity of the entire complex of systems. In addition, security and performance levels must be achieved, benchmarked, and maintained or improved.

Major concerns of the EIT Organization

The EITBOK provides a good deal of information about major organizational concerns. Five of those concerns are summarized here:

- Enterprise Architecture
- Strategy and Governance
- Change
- Interoperability of Component Systems
- Security of Systems and Data

Enterprise Architecture

The EITBOK defines Enterprise Archirecture (EA) as the practice of conducting enterprise analysis, design, planning, and implementation using a holistic approach

for the successful development and execution of strategy. There are, in fact, many alternative definitions common in the community. See enterprise architecture for several of them, including the one from the EITBOK. EA applies architecture principles and practices to guide organizations through the business, information, process, and technology changes necessary to execute their organization's strategies. EA builds upon many of the concepts and practices of Systems Architecture and also software architecture (Bourque & Fairley 2014).

This section will discuss EA from the perspective of key Principles of Systems Thinking, designated in this article with ***bold italics***. There are many commonalities in architecture whether one is architecting in systems, software or the enterprise. This article highlights both commonalities and differences between Enterprise Architecture and Systems Architecture. Architecture takes place at the juncture of design problems and their solutions. Design problems rarely arrive fully formulated or "hatched" for the architect; much of the architecture effort is to analyze or formulate the problems to be solved. Often this involves negotiation both within the enterprise and with outside stakeholders. Possible solutions are limited by resources and schedule and therefore often must also be negotiated.

A dominant principle of EA is ***separation of concerns*** (SoC). This is due to the complexity of enterprises, with many forces acting upon them; the many technologies and disciplines involved in their creation and day-to-day operations; and due to the diversity of stakeholders involved in the enterprise (see *EIT Roles* below).

An application of SoC is the ***boundary*** principle which is crucial to separate what is "inside" the enterprise from what is "outside". While the former is often under the enterprise's control, the latter is often beyond its control.

For some systems, determining the boundary is a key decision, subject to many possibilities and consequences. This is perhaps less the case for enterprises whose bounds are determined by the extent of organizations, regulations, supply chains or other criteria. Within an established boundary, the ***holism*** principle applies: to treat the enterprise as a unified whole, and, in particular, with respect to the ***interaction*** of the enterprise with its *environment*—including stakeholders, other enterprises and systems. From an EIT perspective, an enterprise is a system of systems. SoC offers a way to focus on the constituent systems in terms of their

interconnections to one another to form the enterprise.

Another useful application of SoC is to apply the principle of **views** to depict the architecture of the enterprise in different ways to address the diverse interests (i.e., *concerns*) of an enterprise's various stakeholders. Each *architecture view* presents a specific perspective on the enterprise. To be understandable to its audiences (those various stakeholders), each view should follow the conventions of its *architecture viewpoint* (ISO/IEC/IEEE 2011). Just as a map should have a legend, each view should have a documented viewpoint providing a prescribed set of conventions for looking at the enterprise through the resulting view. A viewpoint definition serves as a contract between the architect and stakeholders on *how* a selected set of concerns are to be addressed. Each viewpoint allows the architect to consider the enterprise in terms of various kinds of elements and relations among them and to present that view to those "concerned" stakeholders. Dominating concerns in enterprise architecture include interoperability, quality of systems and services, operations, managing change, and alignment with the enterprise's mission, market, and technology. Via the principle of **dualism**, it is recognized that some viewpoints are paired, such as with process/data duality. Another pervasive duality in the enterprise is of people/technology in terms of what capabilities are automated and which are performed by people.

In support of EA, there are numerous *enterprise architecture frameworks* each providing a pre-defined, integrated set of viewpoints to frame typical, recurring concerns (ISO/IEC/IEEE n.d.). The earliest recorded framework was Hammer et al. (1986), developed in the mid-1980s in response to growing availability of distributed computing to the enterprise (Rivera 2013). In the late 1980s, the US National Institute of Standards and Technology (NIST) created its Enterprise Architecture Model as a guide for US federal departments to organize their IT offerings (Fong and Goldfine 1989). NIST's 5-layer model remains influential today. Despite never using the term "enterprise," Zachman's information systems architecture has had a major influence on how subsequent frameworks have been organized and depicted using grids or cubes. Later frameworks have evolved in response to emerging concepts, such as the move from client-server to service-oriented applications, as in the OASIS SOA Reference Model and its associated framework. The 1990s saw a proliferation of frameworks for defense enterprises including US DODAF, UK MODAF, NATO's NAF and

their successors such as OMG's UAF.

Yet another application of **separation of concerns** is distinguishing today's needs versus facilitating future needs; balancing **stability/change** and highlighting the temporal planning aspect of architecture of the enterprise. In concert with applying SoC, the architect uses the principle of **abstraction** to "forget" some details of the subject and thus focus on essentials. Just as SoC is a way of taking things apart, **abstraction** is a way of unifying or putting like things together through generalization. These two principles are perhaps the most central to architecture. SoC and abstraction work together: SoC picks out subjects and abstraction helps architects to focus upon the essentials of each subject.

Returning to the **boundary** and **holism** principles, the enterprise architect applies these recursively. As architecting proceeds, the internal details of the enterprise become known in terms of **interaction** among its parts and other **relations**. Again, **multiple views** are used to factor specific **relations** into distinct views. Typical relations include Causality, Dependency, Client-Server, and Input-Output.

There are a variety of ways to structure the enterprise based upon stakeholder needs, mission or charter, technologies or capabilities already in place. Among these are structuring principles including:

- **Modularity** (separate and group capabilities into modules by commonalities)
- **Network** (modules are linked together)
- **Encapsulation** (separating architecture modules use from how they are implemented)
- **Layer Hierarchy** (separation through gradual refinement)
- **Regularity** (uniformity of interactions, of structure within layer, within network)
- **Similarity/Difference** (recognizing the limits of Regularity)

Strategy and Governance

EIT governance is the established process to define the strategy for the EIT organization and oversee its execution to achieve enterprise goals. Strategic planning translates the larger enterprise's goals into EIT goals and defines what EIT must do to achieve them. The EIT

governance effort should communicate EIT goals, how they support the larger organization's goals, and drive change to achieve them while maintaining agreed upon levels of operation.

The systems thinking principle of **parsimony** applies here, too. For example, the simplest possible measurement and reporting systems are most likely to be used and not corrupted. Similarly, the EIT organizational structure should be straightforward and easy to traverse so that staff can communicate and problem-solve in teams. In governing the EIT organization to meet its dual goals of satisfying its customers and meeting its budget and schedule commitments, EIT management encounters the difficulty of managing in the face of **dualism**. The most obvious instance of handling **dualism** in EIT is the trade-off between budget, project prioritization, and features within projects.

Change

There are two levels of change management discussed in the EITBOK: organizational change management and changes to the IT organization's assets, whether they be software, hardware, or telecommunications systems. In the larger sense, the introduction of new systems and services often also requires changes in people's jobs and in the processes used to accomplish those jobs. Such efforts are intended to increase business value. Such changes require change initiatives because they involve guiding a mixture of stakeholders through the change process. This is one aspect of change management. Change also occurs when existing systems or services must be updated to meet new needs. Depending on the scope of the proposed changes, a change initiative may be required in this situation, too. At any rate, a change request process must be initiated and tracked to disposition (deferral, rejection, or implementation). Changes also come about as a result of defect reports. In such cases, the defect report must be tracked throughout the report's life cycle, to disposition (deferral, rejection, or implementation). These latter types of change management are discussed in detail in the EITBOK.

Interoperability of Component Systems

As explained in the EITBOK, *interoperability* means "the ability of systems (including organizations) to exchange

and use exchanged information without knowledge of the characteristics or inner workings of the collaborating systems (or organizations)." This definition is based on an ISO definition: "degree to which two or more systems, products, or components can exchange information and use the information that has been exchanged." (ISO/IEC 2011) Note that this definition includes hardware components as well as software systems and the organizations in which they are used.

Because technology changes so rapidly, and organizations themselves are frequently changing, achieving and maintaining interoperability is crucial. It is one of the main reasons that providing IT services is difficult and how it differs from engineering and delivering finished products, whether off-the-shelf products or bespoke systems.

Security of Systems and Data

According to former FBI Director Robert Mueller in 2012, "There are only two types of companies: those that have been hacked, and those that will be." Two years later, his successor, James Comey said "There are two kinds of big companies in the United States. There are those who've been hacked by the Chinese and those who don't know that they've been hacked by the Chinese." Achieving and maintaining acceptable security is daunting but necessary.

While there are many definitions for security, most include the three dimensions of confidentiality, integrity, and availability. As such, the primary goal of EIT security is to preserve the confidentiality, integrity, and availability of information and information systems.

The principles behind an organization's information security management system should be to design, implement, and maintain a coherent set of policies, processes, and systems that keep the risks associated with its information assets at a tolerable level, and yet, manage the cost and inconvenience of said risk management. As such, according to the EITBOK, EIT security practices should strive to enable the organization to:

- Always understand the current risk tolerance of the enterprise with respect to information and device security.
- Understand the security threats and potential damages to information, devices, and individuals.

- Create and follow policies and procedures that keep cyberattack risk and damages at or below a tolerable level.
- Effectively and efficiently detect and deal with cyberattack incidents.

Ensuring security in EIT has been made even more pressing and more difficult by the proliferation of mobile devices in enterprises and their ever-increasing use to conduct business. Increasingly, personal devices are used, accessing both corporate networks and public networks. The cost of mobile security incidents very easily can outweigh the cost of the mobile device itself.

Quality of Systems and Services

Typically, one of the primary responsibilities of the EIT organization is the timely provision of accurate and relevant information to the larger organization. While the EIT organization itself may not be responsible for creating the data, it is usually responsible for the data's physical and logical validity and integrity central to constructing information. It is for this reason that defining and using appropriate processes is essential to ensure high-quality operation, including the development or acquisition, implementation and integration of new or changed capabilities.

The EITBOK details the quality concepts of:

- How to approach measuring quality
- The importance of quality in requirements management
- Understanding an organization's quality capability
- The differences between quality management systems, quality control, and quality assurance
- Cost of quality in technical debt and the cost of rework

Disaster Preparedness

As explained in the EITBOK, disaster preparedness and disaster recovery support business-continuity planning and include planning for EIT resiliency, as well as recovery from adversity, so that critical business services affected are restored to a satisfactory working state within an acceptable timeframe after an event. The EITBoK devotes a chapter to how the EIT organization and its parent must plan in advance for how it will

provide sufficient organizational resilience to recover from disasters and return to normal operation as rapidly as possible. It provides the following examples of disasters and service interruptions that could occur and should be considered in prevention and recovery planning:

- Natural disaster affecting datacenters or EIT service operations (flood, fire, earthquake, wind)
- Security breach resulting in a disaster (destruction of data, admin password changes, virus/malware installation, sabotage)
- Usage error (accidental deletion, unplug/turn off system resulting in corruption)
- Utility failure affecting datacenters (loss of power even after UPS)
- Vendor failure (cloud provider security failure, oil spill)
- Staffing issue (employment dispute/walkout, epidemic)

These are examples of unpreparedness:

- Requiring use of computers or printers when power is out
- Requiring use of Internet when power or connectivity is out
- Single point of knowledge/control for administration access
- Lack of offsite backup storage
- Lack of working restoration from backups
- Lack of failover datacenters in separate locations
- Undocumented or out-of-date documentation for system interfaces
- Requiring use of phones that are out of power
- Lack of designation of leaders in restoration efforts (who is in charge of restoring service and they know they are in charge)
- In general, no cohesive, comprehensive EIT service restoration plan

Operations and Support of Delivered Systems and Services

Operations engineering is responsible for the operation of the system. SE is responsible for bringing together the

needed experts to insure proper evaluation of proposed changes/upgrades to the system capabilities. It is difficult in some organizational structures to understand how to access requisite information quickly, and what information to share and how to share it. Decisions often must be made quickly, such as when a system is down. Decision effectiveness depends on involving the right decision-makers with a sufficiently complete understanding of the decision context and of criticality of the decision need.

In Operations and Support of deployed technologies, SE is still essential as the overall system of systems goes through maintenance upgrades, new application additions, obsolescence driven re-designs, etc. In addition, during decommissioning and disposal, SE is essential to deal with the proper decoupling of the system and ensuring conformance with policy and laws affecting the system disposal.

Operations and support activities focus on maintaining a normal state of operations in EIT environments, where systems and processes execute according to expectations, with no surprises to users. Normal states can be disrupted either by the expected and planned transition of an asset, or by an unexpected event, be it as large as a disaster or by the discovery of a system defect. In the former case, if there is a disaster recovery plan, it is executed; in the latter, incident management kicks in when a system halts, or problem management is exercised when an operational system is not functioning as expected.

ISO/IEC/IEEE 20000, the International Standard for Service Management (ISO/IEC 2018) may be used by an organization to develop and manage EIT operations, along with complementary information for the operations and infrastructure side of IT, such as the Information Technology Infrastructure Library (ITIL) (Axelos n.d.).

Ethics

Both the IEEE and ACM require their members to adhere to codes of ethics. (IEEE 2020, ACM 2018) In addition, the IEEE Computer Society and the ACM worked together on the Software Engineering Code of Ethics. (ACM/IEEE 1997)) Systems engineers in INCOSE have their own code as well. (INCOSE n.d.) All of these codes of ethics are somewhat different in style and length, but serve the same basic purpose.

Per the EITBOK, an EIT organization should create an easily accessible Code of Ethics:

- Ethical assertions should be documented and socialized in EIT in the form of a Code of Ethics, tied to the overall business code of ethics, and signed off by all staff so that violations can be cause for reprimand or dismissal.
- Vendors should be held to similar ethical standards.
- Lists of employees who have signed off on the Code of Ethics should be maintained through EIT governance practices.

The EITBOK has identified EIT ethical topics that include:

- Those actions and decisions that may be illegal
- Consideration of the social impact of the work at hand—if it will cause harm
- Alignment of business and EIT strategies and policies to an ethical standard
- Incorporation of the ideals of professionalism
- Adherence to applicable regulatory intent
- Protection for whistle blowers

Areas to specifically highlight:

- Activities that interfere with or corrupt the proper function of computers, applications and systems
- Activities that interfere with digital privacy or intellectual rights of others
- Respect for confidentiality, privacy, permissions, and access rights
- Inappropriate bias (skewing) of analysis and reporting
- Inaction in the face of likely ethics violations

EIT Roles

Major EIT systems have been disrupted and taken far too long to remedy due to the lack of information flow through the organization. Particularly for large systems, support activities are highly diverse in the roles people play, the technical knowledge required to build and install systems, to deal with downed systems, and in the organizational structure and culture in which people operate.

The roles range across the business functions requesting and supported by Enterprise technology, the evaluators, designers and implementers of those technologies, those who manage the successful transition of new systems into operation and manage and run the technology systems, to those who manage the quality of systems and services via defect reporting and tracking and who interface directly with end users.

Later sections will provide information about some major international models for defining EIT skills and capabilities.Accomplishing the EIT organization's mission requires people filling many roles ranging from enterprise architect, configuration manager, product designer, financial manager, and product manager. SE activities orchestrate the work of these roles so that they function together, rather than in silos or at cross purposes. Just as a systems engineer can play several roles in a given project, so might several people in an EIT project combine to fulfill the SE function. Larger EIT organizations may have many people performing SE functions and SE may be integrated into the EIT organization in many different ways depending on size, culture, organizational mission, and any number of other factors. Part 5 of the SEBoK on Enabling Systems Engineering explores how systems engineering is woven into various organizations including EIT organizations.

Examples of Systems Engineering in EIT

SE provides a holistic discipline for integrating and managing the contributions of all roles needed to establish and provide EIT products and services. For her doctoral dissertation, Diane Aloisio from Purdue University performed a survey and causal analysis of project failures and of accidents. Aspects of the dissertation were published as a conference paper in Aloisio et al. (2018). The results were similar to those found in reviews of IBM consulting projects in the 1990s.

The following are examples of Systems Engineering interventions that prevented project failures in EIT systems. Almost all point to the lack of the System Engineering balancing act needed for product delivery.

- A widely used document processing product was undergoing a major upgrade. The new VP of Engineering wanted a project review before committing to a schedule. The review showed that the

proposed schedule only considered the development effort and did not give sufficient weight to the voices of the testing and documentation organizations, while the needs of the tech support group had been completely overlooked. By establishing an integrated product management team that included technical expertise, as well as expertise in work breakdown and estimating, a realistic effort was defined and the resulting 9-month schedule was met within 3 weeks.

- A Silicon Valley start-up was depending entirely on an offshore development effort to produce its first product. All senior management was in the U.S. The VP of engineering made monthly visits to the development team. The CEO began to feel uneasy about the team's ability to deliver on schedule sensing that a lot of finger pointing within the development team itself signaled problems. There was no SE or IPT guiding the project. The VP of Engineering seemed unable to sort things out. A project review showed that the probable delivery date was at least 6 months beyond what the CEO was being told. The project was shut down.

- A major player in Human Resource Management had made commitments to deliver a new product. The project involved more than 100 people across the U.S. The delivery date kept slipping. Project roll-out would involve both installation of software and hardware at client sites, as well as installation and staff training at the large, centrally located Client Services organization. Executive Management needed to decide whether to reorganize this very expensive effort or cancel it. Again, the successful solution depended on establishing an effective cross-functional team (IPT) that necessarily included representatives from the technical teams. The IPT was supported by a newly established Project Management Office (PMO). The PMO integrated the geographically disparate efforts by establishing communication channels, as well as common estimating and scheduling practices. The PMO also instituted monthly critical metrics reports to show the IPT bottlenecks and inefficiencies across the project, from development efforts to Client Services set-up to Payroll technical support.

Systems Engineering provides a holistic discipline for integrating and managing the contributions of all of the roles needed to provide EIT products and services. It

ensures that a single perspective does not dominate outcomes.

Key External Models of EIT Organizational Capability and Maturity

Because EIT is critical to businesses globally, considerable effort has gone into establishing universal frameworks for EIT skills, along with definitions of levels of capability of each skill. In addition, at least 2 models have been developed for evaluating the organizational performance of EIT. The effectiveness of systems engineering is key to the effectiveness of the EIT organization.

IT-CMF: IT-Capability Maturity Framework

The IT-CMF had its origin in a quality improvement effort at Intel Labs Europe in Ireland. Its intent, as described at the CIO Index website was to quantify and demonstrate the true value impact of IT. (CIO Index n.d.) IT-CMF is used to measure, develop, and monitor an organization's EIT capability maturity. It consists of 35 EIT management capabilities organized into four macro capabilities: managing EIT like a business; managing the EIT budget; managing the EIT capability; and managing EIT for business value. Its 35 critical capabilities are described in the EITBOK as "A defined EIT management domain that helps mobilize and deploy EIT-based resources to effect a desired end, often in combination with other resources and capabilities".

CMMI: Capability Maturity Model Integration (US)

The CMMI had its origin at the Software Engineering Institute in 1987 in a model requested by the U.S. Department of Defense for determining the capability maturity of contractor software development organizations. This became the Capability Maturity Model (CMM). Since then, it has been expanded into a set of Capability Maturity Models, all based on five levels of demonstrably increasing maturity as measured by organization performance. That set was then integrated and became the CMMI. The CMMI set addresses development, services, supplier management, people management, and data management. (CMMI Institute.

n.d.)

SFIA: Skills Framework for the information Age (Europe, Canada and US)

The SFIA Framework has been downloaded and used by organizations and individuals in nearly 180 countries and is overseen by the SFIA Foundation (SFIA n.d.) It identifies 97 professional skills across EIT and supporting areas with seven levels of responsibility, which provide generic levels of responsibility, taking into account reflect experience and competency. These describe the behaviors, values, knowledge, and characteristics that an individual should have in order to be considered competent at a particular level.

E-CF: European Competency Framework (EU)

The ECF was the first sector-specific implementation of the European Qualifications Framework. The European Norm (EN) 16234-1 European e-Competence Framework (e-CF) provides a reference of 41 competences as applied at the Information and Communication Technology workplace, using a common European reference for competences, skills, knowledge and proficiency levels. The e-CF Explorer web tool is hosted by IT Professionalism Europe (ITPE n.d.).

I Competency Dictionary: EIT skills as Defined by METI/ IPA (Japan)

The Japanese government's Information-Technology Promotion Agency (IPA), established by the Ministry of Economy, Trade and Industry (METI) developed the "I Competency Dictionary" (iCD) to support the IT Engineers Examination. It underlies the METI/IPA Skills Standards for IT Professionals, abbreviated as ITSS, established in 2002. ITSS organizes skills within the information services industry into 11 job categories and 35 specialty fields. In each field, there are seven levels of competence based on individual experience and ability to perform. One appealing feature of ITSS is that this standard allows engineers to draw roadmaps for their own futures and career advancement. Approximately 90% of large enterprises and over 60% of SMEs in Japan have introduced or are considering using ITSS. Information in English is available at

International Standards Guiding EIT

The following standards guide the EIT community:

- ANSI/AIAA. 2012. *ANSI/AIAA Guide to the Preparation of Operational Concept Documents.* ANSI/AIAA G-043A-2012e
- IEEE 2012 Std 828™-2012, *IEEE Standard for Configuration Management in Systems and Software Engineering*
- ISO 10004:2012, Quality management—Customer satisfaction—Guidelines for monitoring and measuring
- ISO 10007:2003, Quality management systems—Guidelines for configuration management
- ISO 18238:2015, Space systems—Closed loop problem solving management
- ISO/IEC 25010:2011 Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—System and software quality models
- ISO/IEC 16350:2015, Information technology—Systems and software engineering—Application management
- ISO/IEC 19770-1:2012, Information technology—Software asset management—Part 1: Processes and tiered assessment of conformance
- ISO/IEC 20000-1:2011, (IEEE Std 20000-1:2013) Information technology—Service management—Part 1: Service management system requirements
- ISO/IEC 20000–2:2012, Information technology—Service management—Part 2: Guidance on the application of service management systems
- ISO/IEC TR 20000–10:2015, Information technology—Service management—Part 10: Concepts and terminology
- ISO/IEC TR 20000–11:2015, Information technology—Service management—Part 11: Guidance on the relationship between ISO/IEC 20000-1:2011 and related frameworks: ITIL®
- ISO/IEC TR 20000-12—Information technology—IT Service management—Part 12: Guidance on the

relationship between ISO/IEC 20000-1:2011 and service management frameworks: CMMI-SVC®

- ISO/IEC/IEEE 14764-2006, Software Engineering—Software Lifecycle Processes—Maintenance
- ISO/IEC 15939:2007, Systems and software engineering—Measurement process
- ISO/IEC/IEEE 42010:2011, Systems and software engineering — Architecture description

References

Works Cited

ACM. 2018. *ACM Code of Ethics and Professional Conduct.* Accessed on May 7, 2023. Available at https://www.acm.org/code-of-ethics.

ACM/IEEE. 1997. ACM/IEEE Computer Society. *Software Engineering Code.* Accessed May 7, 2023. Available at https://ethics.acm.org/code-of-ethics/software-engineering-code/.

Alioso, D.C., Marais, K., Sun, Hanxi. 2019. "2018 Best PIC II Paper: Systems Engineering Division: Development of a Survey Instrument to Evaluate Student Systems Engineering Ability", in 126th ASEE Annual Conference and Exposition. Tampa, Florida.

ASQ. n.d. "What is Total Quality Management". American Society for Quality. Accessed May 7, 2023. Available at: https://asq.org/quality-resources/total-quality-management.

Axelos. n.d. "Axelos products and services for professionals". Accessed May 7, 2023. Available at https://www.axelos.com/for-professionals.

Bourke, P. and Fairley, R.E. eds. 2014. *Guide to the Software Engineering Body of Knowledge*, Version 3.0, IEEE Computer Society. Accessed May 7, 2023. Available at : www.swebok.org.

CIO Index. n.d. "IT Capability Maturity Framework". CIO Index. Accessed May 7, 2023. Available at https://cio-wiki.org/wiki/IT_Capability_Maturity_Framework_(IT-CMF).

CMMI Institute. n.d. "CMMI". CMMI Institute. Accessed May 7, 2023. Available at https://cmmiinstitute.com/cmmi.

Fong, E.N. and A.H. Goldfine. 1989. *Information Management Directions: The Integration Challenge.* National Institute of Standards and Technology (NIST) Special Publication 500-167, September 1989. Accessed May 7, 2023. Available at https://csrc.nist.rip/library/NIST%20SP%20500-167.pdf.

Hammer, M., T.H. Davenport and J. Champy. 1986. *Dispersion and Interconnection: Approaches to Distributed Systems Architecture*, Cambridge, MA: CSC Index Inc. and Hammer and Company, Inc. Accessed May 7, 2023. Available at https://www.globalaea.org/general/custom.asp?page=Re sources.

Hilliard, R. 2015. "Lessons from the fundamental unity of architecting". *Software Engineering in the Systems Context*, editors: Ivar Jacobson and Harold 'Bud' Lawson.

Hofmeister, C. et al. 2007. "A general model of software architecture design derived from five industrial approaches". *The Journal of Systems and Software,* 80(1).

IEEE. 2020. *IEEE Code of Ethics.* Accessed May 7, 2023. Available at https://www.ieee.org/about/corporate/governance/p7-8.h tml.

IEEE/ACM n.d. *Guide to the Enterprise IT Body of Knowledge*. IEEE and ACM. Accessed May 7, 2023. Available at: http://eitbokwiki.org.

INCOSE. n.d. INCOSE Code of Ethics. Accessed May 7, 2023. Available at https://www.incose.org/about-incose/Leadership-Organiz ation/code-of-ethics.

ISO/IEC. 2018. *ISO 20000: The International Standard for Service Management.* Accessed May 7, 2023. Available at: https://www.iso.org/obp/ui/#iso:std:iso-iec:20000:-1:ed-3 :v1:en.

ISO/IEC. 2011. *ISO/IEC 25010:2011 Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—System and software quality models.* Accessed May 7, 2023. Available at https://www.iso.org/standard/35733.html.

ISO/IEC/IEEE. n.d. *Survey of Architecture Frameworks* . ISO/IEC/IEEE 42010 Users Group. Accessed May 7, 2023. Available at http://www.iso-architecture.org/42010/afs/.

ISO/IEC/IEEE. 2011. ISO/IEC/IEEE 42010:2011, *Systems and software engineering — Architecture description.* Accessed May 7, 2023. Available at https://www.iso.org/standard/50508.html.

ITPE. n.d. "e-CF Explorer". IT Professional Europe. Accessed May 7, 2023. Available at https://ecfexplorer.itprofessionalism.org/.

PMI. 2013. *Software Extension to PMBOK® Guide – Fifth Edition.* Newtown Square, PA, USA: Project Management Institute (PMI). Accessed May 7, 2023. Available at https://www.pmi.org/pmbok-guide-standards/foundational/pmbok/software-extension-5th-edition.

Rivera, R. 2013. "The PRISM Architecture Framework – Was it the Very First Enterprise Architecture Framework?" *Journal of Enterprise Architecture*, November 2013.

SFIA. n.d. "SFIA: The Global Skills and Competency Framework for the Digital World". Accessed May 7, 2023. Available at https://sfia-online.org/en.

Standish Group. 1994. *The Chaos Report.* Accessed May 7, 2023. Available at https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf.

Zachman, J. 1987. "A Framework for Information Systems Architecture". *IBM Systems Journal,* pp454-470, 1987. Accessed May 7, 2023. Available at https://www.zachman.com/images/ZI_PIcs/ibmsj2603e.pdf.

Primary Reference

IEEE Computer Society. *Enterprise IT Body of Knowledge*. Accessed May 7, 2023. Available at http://eitbokwiki.org.

Additional References

Aloisio, D.C. 2018. *Lessons from Systems Engineering Failures: Determining Why Systems Fail, The State of*

Systems Engineering Education, and Building an Evidence-Based Network to Help Systems Engineers Identify and Fix Problems on Complex Projects. PhD Dissertation, Purdue University. Accessed on May 7, 2023. Available at https://hammer.purdue.edu/articles/thesis/Lessons_from_Systems_Engineering_Failures_Determining_Why_Systems_Fail_the_State_of_Systems_Engineering_Education_and_Building_an_Evidence-Based_Network_to_Help_Systems_Engineers_Identify_and_Fix_Problems_on_Complex_Projects/7488569.

Curley, M., Kenneally, J., Carry, M. eds. 2015. IT Capability Maturity Framework (IT-CMF): The Body of Knowledge Guide. 2015. Van Haren Publishing.

< Previous Article | Parent Article | Next Article >
SEBoK v. 2.10, released 06 May 2024