UVU™ DIGITAL TRANSFORMATION

UTAH VALLEY UNIVERSITY

Enterprise Architecture Guide (EAG)

for

Digital Transformation

Office of Digital Transformation (Dx)

Table of Contents

UVU’s Digital Transformation Enterprise Architecture Guide

Version 1.2, 11 November 2021

Introduction

UVU’s Digital Transformation Enterprise Architecture Guide (EAG) documents the University’s architectural principles for Information Technology (IT) within a UVU Enterprise Architecture framework[1]. With the help of members of the UVU community, we expect the guide to evolve over time to include an Architecture Review Process; a view of the current state of UVU’s enterprise architecture and its future architectural vision; a description of IT standards; and a list of services available to development teams across campus.

Applicability

The EAG applies specifically to IT systems and solutions that are enterprise-class and mission critical, which typically have the following attributes:

- University - or enterprise-wide
  - Non-departmental
  - Expand beyond the initial departmental scope
- Use critical university data
  - Sensitive data (classified as Sensitive or Restricted according to the University’s approved information classification taxonomy)
  - IDs and passwords are created and stored
  - Data used would affect other university data entities
- Integrated with existing “enterprise-wide, mission critical” applications/systems
- Supports essential operations required for the day-to-day operations of the university; it is indispensable.
  - Failure or disruption would cause a failure in the university’s core operations.

Audience

The intended audiences for the EAG are enterprise system developers and project teams developing new systems or making enhancements to existing systems; sponsors of IT initiatives; and any university leader planning to acquire, implement, or build IT-enabled solutions – especially those that are expected to be enterprise-class and mission critical.

Each group can benefit in a different way as described below:

- **Enterprise System Developers and Project Teams:** Enterprise system developers and project teams can use the EAG to gain an understanding of the current architectural landscape, the future vision of the enterprise architecture, and the services available to development teams. By understanding the recommended technical standards and available UVU services, project teams can re-use existing services and create applications that fit into the long-term architectural vision. Teams can also leverage the information to develop new enterprise-wide services. Finally, the EAG will assist

[1] UVU’s framework is based on The Open Group Architectural Framework (TOGAF) model and draws extensively from NIH’s Enterprise Architecture model.

project team members in identifying whom to contact to mitigate risks in different aspects of their project.

- **Sponsors:** Sponsors can benefit from the EAG by gaining an understanding of the technical direction of the University as well as the Architectural Governance Process. This knowledge can then be used to shape their decisions regarding IT investments.
- **IT Architects:** IT Architects can use EAG to gain a common understanding of Enterprise Architecture at UVU. Additionally, the EAG will be used during the project review process to provide a consistent representation of the context and principles of both the current and future state. Both will assist IT Architects in making informed architectural decisions and in identifying gaps in Enterprise Architecture.
- **Information Technology Policy and Planning Committee (ITPC):** ITPC should be established and can use the EAG to gain a common understanding of the Enterprise Architecture at UVU, to proactively identify potential risks in projects, and to assist in identifying individuals to help mitigate those risks.

Value

By establishing and publishing principles and standards in this guide, the institution can reduce the risk and inefficiencies associated with building a “patchwork” of applications and solutions using different tools or different integrations, and built by temporary or outside teams, that may have given scant regard to security, legal compliance, scalability, efficiency, and maintainability. The EAG can provide guidance and requirements in the following critical areas of competence:

- Application management and policy application
- Coding and maintenance standards
- Data security
- Development tools
- Enterprise architecture
- Governance
- Integration
- Policy compliance
- Platform support
- Source code control to protect university sensitive data and intellectual property and ensure ongoing maintenance and support
- Technical support

Digital Transformation Building Blocks

This describes technology building blocks to enable digital transformation at Utah Valley University, but the principles and concepts are universal.

To facilitate digital transformation, the Office of Information Technology (OIT) and Academic and Student Digital Services (ASDS) must also transform.

- OIT and ASDS must provide reliable and easy-to-use technology solutions that faculty and staff can use to enhance their interactions with others and improve the products and services they deliver.
- OIT and ASDS must adapt, modernize, provide existing products and services at a reduced cost, and provide new products and services with exceptional customer service.

- In general, products and services should be available via self-service, 24 by 7, with ample support to make technology consumers successful, satisfied, and even delighted.
- All services and products in production environments are required to be electronically monitored and integrated with tools and systems maintained by the Operations team. Additionally, all production services must be well documented with robust knowledge base articles in place as well as "start, stop, and restore" instructions that enable Operations personnel to restore service.

This building blocks section describes architecture, principles, and philosophies intended to make these necessary changes and the dream described above possible.

Application Programming Interfaces (APIs)

Application programming interfaces (APIs) make it easier for others to interact with applications, create and use alternative user interfaces, and use the available services for alternative and even unexpected purposes.

- When acquiring or developing an application, it must have an API, and preferably a RESTful one.
- To make them most valuable, APIs must be exposed and consumed through API management tools.
- APIs should be more than simple JSON-based CRUD interfaces; APIs should expose appropriate business logic so that API consumers cannot violate required business processes.
- Enforcing this principle allows others to build delightful user experiences without institutional concern about policy, practice, or process compliance.

Domain-Driven Design

A quick summary of the book Implementing Domain-Driven Design by Vaughn Vernon.

- Bring domain experts and developers together to create a ubiquitous language embedded in the application code itself.
- Define or determine bounded contexts wherein this language is valid.
- This helps software developers genuinely understand the business processes they are being asked to automate.
- It also helps the business participants understand the code being written and allows them to question decisions, test assumptions, and find bugs before deployment.
- This collaborative group of business leaders and developers is "the team"; success or failure is in their hands.

Microservices

Microservices are an architectural style that will be used at UVU to create larger systems.

- Systems built using microservices are loosely coupled, implement a single business capability, have well-defined interfaces, and communicate using only these interfaces.
- The size of a microservice is governed by the associated bounded context.
- At UVU, an essential part of a microservices' interface is its ability to raise events.

Event-Driven Architecture (EDA)

Build systems that raise events so other systems don't have to waste time and resources.

Application Acquisition

UVU acquires applications to get something done.

- A strong preference to cloud based applications.
- When we build services or applications, they should use the most abstract service offerings that make sense.
- We should avoid instantiating servers and consume storage and then build queues, notification services, &etc.
- We should instead use services such as queues, notification systems, serverless functions, &etc.

DevOps

DevOps is a culture and a practice.

- DevOps is rapid development, testing, and software deployment.
- Increase accountability by allowing those who develop an application to be responsible for running and supporting it.
- Design Driven Development teams are in charge and responsible for the functionality, performance, and reliability of "their" products.
- Network hardware (switches, routers, firewalls, servers, appliances, AV equipment &etc.) will be software configurable.
- Control hardware using DevOps principles to configure, test, and deploy hardware platforms as rapidly as "other" developers.

Where to Compute

There was a paradigm shift from computing on physical servers to virtual servers. The paradigm is changing again to compute in the cloud.

- Our compute and storage will be a combination of our data center and cloud.

- Acquired applications will also run in the cloud.

Networks

According to Metcalfe’s Law, the value of the network is proportional to the square of the connected users.

- Unlike server and storage, we believe we will have a wired and wireless network on campus for the foreseeable future.
- However, the way we deploy, configure, and maintain these networks will change drastically. Remember, software is eating the world, and networking is not an exception to the rule.
- Network components will be physically installed in some generic way and then configured remotely via software.
- In a DevOps fashion, when a problem occurs, you’ll figure out what went wrong in the configuration script, you’ll repair the script, you’ll test the script, and you’ll redeploy.

Contact

Suggestions/Updates to the EAG If you have any suggestions or updates to make to this EAG, please contact the Enterprise Architecture group at (ea@uvu.edu).

Whom to Contact with Questions. If you have specific questions regarding the items presented in this guide, you can contact the Enterprise Architecture group at ea@uvu.edu.

What is Enterprise Architecture?

Enterprise architecture defines how information and technology support university operations and provides benefit for the university. In other words, enterprise architecture is a comprehensive framework used to manage and align the university's Information Technology (IT) assets, people, operations, and projects with its operational goals and characteristics.

It illustrates the university's core mission, each component critical to performing that mission, and how each of these components is interrelated. These components include:

- Guiding principles
- Organization structure
- Business processes
- People or stakeholders
- Applications, data, and infrastructure
- Technologies upon which networks, applications and systems are built

Guiding principles, organization structure, business processes, and people does not sound very technical. That's because enterprise architecture is about more than technology. It is about the entire university (or enterprise) and identifying all the bits and pieces that make the university work more efficiently

An Analogy to City Planning

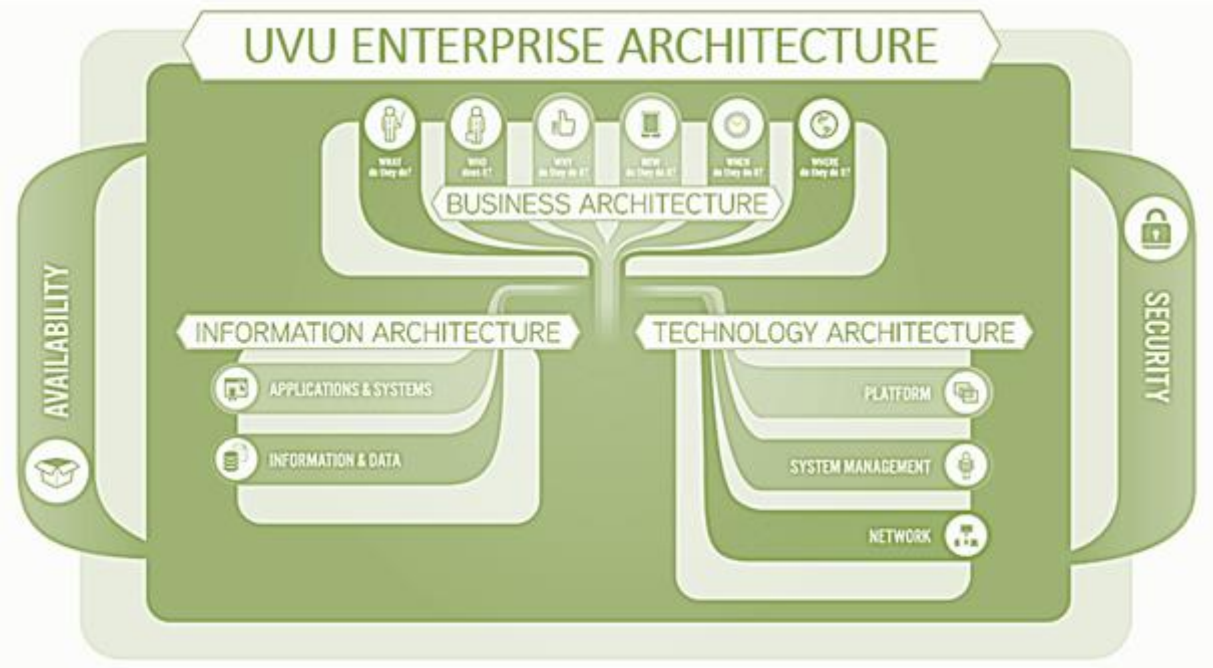
Enterprise architecture can be compared to the more widely understood concept of city planning. In city planning, zones are established for very specific purposes. The buildings that are built in these zones are constructed to specifications to meet those purposes.

For example, a hospital is built to different specifications than a house or office building. Additionally, to ensure uniformity of the city and to ensure roads link to each other and pipes attach to each other without a problem, city planners establish specific guidelines on building materials and interface specifications.

In the case of enterprise architecture, the enterprise is analogous to the city. The organization structure represents the zones established to execute the enterprise's core mission. Buildings are analogous to applications and systems. Likewise, technical elements, such as infrastructure hardware, design specifications, and development languages, are analogous to building materials and interface specifications and are used to implement the applications and systems.

City Plan is to . . .	**as enterprise architecture is to . . .**
1. zones	1. organization structure
2. buildings	2. applications and systems
3. building materials and interface specification	3. infrastructure hardware, design specifications, and development languages

UVU’s Enterprise Architecture Framework



UVU’s Enterprise Architecture framework[2] consists of four distinct architecture types:

- Business Architecture
- Information Architecture
- Application Architecture
- Technology Architecture

All four of the architecture types, Business, Information, Application, and Technology architectures, lie within the context of Information Availability and Security (as illustrated in the diagram). This is intended to emphasize that enterprise business priorities and risk are considered and evaluated from the beginning of potential IT endeavors rather than as an afterthought.

Business Architecture

Business Architecture documents and models an organization’s policies, processes, work activities, artifacts, and assets. Specifically, Business Architecture answers the following questions concerning UVU's organizations and processes:

- What do they do?
- Who does it?
- Why do they do it?
- How do they do it?
- When do they do it?
- Where do they do it?

[2] UVU’s framework is based on the TOGAF model and draws extensively from NIH’s Enterprise Architecture model.

Information Architecture

Information Architecture documents and models key information assets, the applications that use them to enable business processes, and how applications and information together support the enterprise's functions. The information architecture also specifies which parts of the business process are supported by each application and where each type of data is stored and managed.

- **Information and Data Architecture**[3] – through data models, the Information and Data Architecture identifies the information and data UVU manages to perform its mission. For example, access and distribution models identify UVU enterprise data stores and information flows.
- **Application Architecture** – Establishes the principles and models defining how applications are designed, built, and integrated. This architecture enables and supports the execution of UVU's business processes.

Technology Architecture

Technology Architecture represents UVU's technical infrastructure and the specific hardware and software technologies that support UVU's information systems. Technology Architecture consists of the following domains:

- **Platform**– consists of the combination of software, middleware, hardware infrastructure and development frameworks that enable the development, deployment, operation, integration, and management of applications.
- **Systems Management**– consists of the technical tools used to collect and analyze data that measure UVU system performance to improve system availability, performance, and reliability.
- **Networks** – consist of the technical elements required to provide data and Internet connectivity and communication within and outside of UVU.

[3] For an explanation of the difference between information architecture and data architecture see Data architecture vs. information architecture

What are Principles?[4]

Principles are simple statements of values that inform and support the way an organization fulfills its mission. Enterprise architecture principles are intended to guide IT decision-making processes, serving as a base for IT architectures, development policies, and standards.

A common misunderstanding concerns the difference between **principles** and **standards**. While they are similar in nature, principles and standards differ significantly in intention and application.

Essentially, principles provide high-level guidance on ***what*** should happen within architecture, while standards define ***how*** they should happen. For example, a standard may state something like: *"All web pages must be WCAG 2.0 compliant."* While a principle would give higher level guidance such as: *"All data, information, applications and processes must be easy to use and must be accessible to all relevant parties."*

The remainder of this document describes the overarching Enterprise Architecture principles followed by principles for each of the architecture types (Business, Information, Application, and Technology) and the domains within those architecture types.

[4] A rule or code of conduct. —Merriam-Webster

Enterprise Architecture Principles

Enterprise Architecture Principle 1: Broadly Applicable

Statement:

UVU’s Enterprise Architecture applies to any IT systems and applications regardless of who pays for or builds it.

Rationale:

A consistent framework for information technology promotes better results.

Implications:

- This applies to all UVU systems, data, and infrastructure. University or enterprise-wide, mission critical solutions will be subject to enterprise architecture review and, where deemed appropriate, may require ITPC approval.

NOTE: The enterprise architecture review is NOT an audit. Rather, the purpose of the review is to help the project team leverage the existing architecture and common services, proactively identify the risks to a project, understand the university-wide context in which the proposed solution is to operate, and provide input on how UVU’s architecture may need to be modified.

Enterprise Architecture Principle 2: Business Needs Drive Technology

Statement:

UVU’s IT systems and applications must support the university’s mission, vision, strategies, and plans. It is critical IT delivers business value to the University.

Rationale:

Information Technology has the most value when closely aligned with UVU’s strategic plans, and other university-level direction, concepts, and objectives.

Implications:

- IT is not merely a contractor delivering whatever the customer asks for, but a partner in creating value for the business and meeting strategic goals.
- Business and IT professionals work together to determine how to solve business problems with IT.
- Architecture will be generated with a specific purpose and for a specific audience to ensure they meet the expectations and needs of its intended stakeholders.

Enterprise Architecture Principle 3: Maximize Benefit for UVU long-term success

Statement:

Architectural decisions will maximize the overall long-term benefit to UVU.

Rationale:

Architecture is designed to provide long-term benefits to the enterprise. Decisions must balance multiple criteria based on business needs.

Implication:

- Criteria used to prioritize may include available funding, governance, and the knowledge, skills, and abilities of UVU's faculty and staff and may receive different emphasis in different situations.

Enterprise Architecture Principle 4: Reusability of Components

Statement:

UVU's Enterprise Architecture will be built on reusable, loosely coupled modular components.

Rationale:

Reusable components provide opportunities to reduce IT development costs and time by leveraging investments in existing systems. Modular components improve system ability to adapt to changing requirements.

Implications:

- Loosely coupled modular components must be archived, documented, and searchable for ongoing support and future use.
- APIs will be used wherever possible to ensure multiple applications can take advantage of business workflow.
- IT organizations across the university will make use of open-source code whenever available.
- IT organizations across the university will provide their code to others through sharable code repositories.

Enterprise Architecture Principle 5: Leverage Investments

Statement:

All systems will leverage existing and planned components, enterprise software, management systems, infrastructure, and standards.

Rationale:

UVU has invested heavily in several processes, technologies, infrastructures, and standards. Therefore, to maximize UVU's return on investment, all proposed systems should leverage existing systems as much as possible.

Implication:

- Considering how to leverage or reuse investments is applicable to all technology domains.
- Advocates of proposed solutions should first determine how they might be completed using existing services, modules, or systems.
- Searchable repositories and a commitment to populate them are needed for this to be possible.

Enterprise Architecture Principle 6: Balance Access and Security

Statement:

System designs, standards, and practices should make university information available to appropriate audiences while protecting information the university controls in accordance with legal, contractual, and ethical requirements for information security and use.

Rationale:

Information is a vital institutional resource whose value is enhanced when used appropriately and diminished when misused, misinterpreted, or when access is unnecessarily restricted.

Implication:

- While providing access to, and safeguarding the integrity of, university information is a shared university responsibility, information stewards[5] are responsible for classifying the information in their units and for the processes that generate, distribute, share, and/or receive university information.
- Decisions about providing access to information are based on the classification and the value of the assets, which is determined, by its use, sensitivity, and importance to the university and in compliance with university policy, state and federal regulations, and other obligations regarding privacy and confidentiality of information.

[5] Need to define – Information Governance procedures for descriptions of the roles and responsibilities of information stewards.

Business Architecture (BA) Principles

Business architecture is a part of enterprise architecture that creates “a blueprint of the enterprise that provides a common understanding of the organization and is used to align strategic objectives and tactical demands.”

To create the blueprint, BA uses models to represent an organization to help executives answer commonly asked questions: who, what, where, when, why, and how. The answers are then used to develop plans and implement business decisions.

Business architecture may include the following:

- Organizational strategy
- List of customers, suppliers, and competitors
- Units and groups
- Capabilities
- Products and Services
- Assets
- Current initiatives and projects
- Information
- Processes
- Policies, rules, and regulations
- Common Vocabulary

The principles of business architecture will guide those developing technology solutions.

Business Architecture Principle 1: BA is about business – not IT.

Statement:

Business architecture is not constrained by technology.

Rationale:

Business architecture describes the business model independently of its supporting technology and provides the foundation for the analysis of opportunities for automation.

Implication:

- Eliminate technology constraints when defining business architecture and ensure automated processes are described at the business process level for analysis and design.
- University departments and IT organizations must have a common vision of both a unit’s business functions and the role of technology in them.
- The university departments and IT have joint responsibility for defining the IT needs and ensuring that the solutions delivered by the development teams meet expectations and provide the projected benefits.

Business Architecture Principle 2: BA is about Examining Processes

Statement:

Business processes must be documented, analyzed, simplified, or otherwise redesigned for optimization and efficiency before automating them.

Rationale:

Opportunities for increasing efficiency, effectiveness, and quality can be identified and realized through simple and flexible business processes.

Implication:

- Analyze business processes to simplify, integrate, eliminate redundancy, and increase efficiency.
- Identify common business processes for reuse and design business processes to enable business agility.
- Models are domain specific. There may be sub domains with different models.

Business Architecture Principle 3: Enterprise Business Design is Balanced with Architectural Realities

Statement:

The business models of the enterprise and the architectures that serve them are balanced to recognize the realities of IT systems and the associated costs of the system(s) to the University.

Rationale:

Solutions that do not align with the processes of the enterprise inevitably result in inefficiencies, confusion, and poor performance—of both systems and processes. Models and architecture are aligned to serve the business and ensure that solutions can be efficiently implemented and maintained.

Implication:

- The architecting, designing, and building of solutions must be considered within the context of the enterprise's business processes, policies, and desirable practices.
- Solving business problems often involves buying rather than building the solution, the realities of the purchased systems must be considered, and domain models modified where such modification makes implementation and operation less expensive without undue changes to the needs of the business. This may require business process adjustments to maximize application utilization.

Business Architecture Principle 4: Business Architecture is reusable

Statement:

BA provides a foundation for future analysis and decision-making and can be used as a starting point for future efforts.

Rationale:

BA is *not* a one-time analysis of a business environment; it also provides a foundation for future analysis and decision-making.

Implication:

- Establish domain teams that include both business and IT experts that are responsible for creating the business architecture for their respective domains. Domain teams meet as needed but exist so long as the business domain does.
- Identify opportunities for common components and implement them in such a way that there is an opportunity for reuse by another business domain, department, program, or unit of the university.
- Provide a mechanism for units across campus to learn about and access the “reusable” components.
- Use the existing business architecture description and deliverables as a starting point for future efforts unless an organization has fundamentally changed.

Information Architecture Principles

Information and Data Architecture documents and models key information assets, the applications that use them to enable business processes, and how applications and information together support the enterprise's functions. The information architecture also specifies which parts of the business process are supported by each application and where each type of data is stored and managed.

Information Architecture Principle 1: Information is an Asset

Statement:

Information is an asset that has value to the enterprise and needs to be managed and treated much like a physical asset.

Rationale:

Information is a valuable corporate resource; it has real, measurable value. In simple terms, information is used to aid decision-making and accurate, timely information is critical to accurate, timely decisions. We must also carefully manage information to ensure that we know where it is, can rely on its accuracy, and can obtain it when and where we need it.

Implication:

- We must carefully manage information to ensure that we know where it is, can rely on its accuracy, and can obtain it when and where needed. An information asset's classification is determined by its value and risk. A fundamental understanding of the value of an information asset to the business and the business risk if it is exposed, corrupted, or lost is essential to determining the appropriate strategies and investments to make for protecting and managing it.
- Ensure that all organizations understand the relationship between the value of information, sharing of information, and accessibility to information.
- Make the cultural transition from "information ownership" thinking to "information stewardship" thinking.
- Measure and document the current value, risk, and cost of the asset as you would any physical enterprise asset, such as property or equipment.
- Recognize that the ease of moving and copying information is a hidden source of significant inefficiencies and cost in terms of information management.
- Make information accessible to its authorized end users without wasting resources

Information Architecture Principle 2: Information Management is Everybody's Business

Statement:

All organizations in the enterprise participate in information management decisions needed to accomplish business objectives.

Rationale:

Information users are the key stakeholders, or customers, in the application of technology to address a business need. To ensure that information management is aligned with the business, all organizations in the enterprise must be involved in all aspects of the information environment.

Implication:

- To operate as a team every stakeholder or customer will need to accept responsibility for developing the information environment.
- Commitment of resources will be required to implement this principle.
- The information architecture must encourage management and sharing of data to maximize value at the lowest cost to the university.

Information Architecture Principle 3: Information is Accessible and Shared

Statement:

Information is appropriately shared across enterprise functions and organizations.

Rationale:

Users have access to the information necessary to perform their duties; therefore, wide, and timely access to accurate information is essential to improving the quality and efficiency of enterprise decision-making. Simply put, it is less costly to maintain timely, accurate information in a single application, and then share it, than it is to maintain duplicate information in multiple applications.

Implication:

- Shared information will result in improved decisions since we will rely on fewer (ultimately one virtual) sources of more accurate and timely information for all of our decision-making.
- Electronically shared information will result in increased efficiency. Staff time is saved, and consistency of information is improved.
- Information needs to be classified in accordance with the security principles of the organization to determine levels of access by faculty, employees, contractors, vendors, partners, suppliers, students, general release, &etc.
- To enable information sharing we must develop and abide by a common set of policies, procedures, and standards governing information management and access for both the short and the long term.
- Access to information does not constitute understanding of the information; personnel should take care not to misinterpret information.

- Access to information does not necessarily grant the user access rights to modify or disclose the information. This will require an education process and a change in the organizational culture, which currently supports a belief in “ownership” vs. “stewardship” of information by functional units.

Information Architecture Principle 4: Value and Risk Determine Information Classification

Statement:

An information asset’s classification is determined by its value and risk.

Rationale:

A fundamental understanding of the value of an information asset to the business and the business risk if it is exposed, corrupted, stolen, misused, or lost is essential to determining the appropriate strategies and investments necessary for protecting and managing it.

Implications:

- Enterprise information/data standards should be identified when the value of commonality across UVU and interoperability with other information systems exceeds the value of uniqueness.
- Information/data should be maintained in a separate data layer decoupled from applications.

Information Architecture Principle 5: Information Stewards are Accountable

Statement:

Information stewards are formally assigned accountability for establishing and enforcing the information policies that govern the access, security, quality, and classification of information assets within their domain of responsibility.

Rationale:

One of the benefits of an architected environment is the ability to share information (e.g., text, video, sound, &etc.) across the enterprise. As the degree of information sharing grows and business units rely upon common information, it becomes essential that policies and decisions about information within their stewardship are made by the information steward.

Implications:

- Real stewardship dissolves the information "ownership" issues and allows the information to be available to meet all users' needs. This implies that a cultural change from information "ownership" to information "stewardship" may be required.
- Stewards manage the information assets on behalf of others and in the best interests of the organization.
- Stewards must have the authority and means to manage the information for which they are accountable and be responsible for meeting quality requirements levied upon the information for which the trustee is accountable.

Information Architecture Principle 6: Enterprise Information Requires Primary Information Sources

Statement:

All enterprise information/data will have an authoritative, official, primary source that is the location for all Create, Update, and Delete actions.

Rationale:

For enterprise information/data to be managed effectively, there can be only one primary source for each information/data element. Otherwise, inconsistent, and erroneous information/data may result.

Implications:

- When the value of commonality across UVU and interoperability with other information systems exceeds the value of uniqueness, enterprise information/data standards should be identified
- Data should only be collected once electronically within a single interface and then shared across systems to provide a “single version of the truth”.
- Information and data management will be looked at from an enterprise perspective.
- A master data approach needs to be adopted and implemented to manage the lifecycle of shared data across the organization.

Information Architecture Principle 7: Use Common Information Vocabulary and Definitions Across the Enterprise

Statement:

Information is defined consistently throughout the enterprise, and the definitions are understandable and available to all users; information architecture needs to be metadata driven.

Rationale:

The information that will be used in the development of applications must have a common definition throughout the enterprise to enable sharing of information. A common vocabulary will facilitate communications and enable dialogue to be effective. In addition, it is required to interface systems and exchange information.

Implications:

- A comprehensive approach for metadata management is the key to reducing complexity and promoting re-usability across the enterprise. A metadata-driven approach makes it easier for users to understand the meaning of data and to understand the lineage of data across the environment.
- Whenever a new information definition is required, the definition effort will be coordinated by an “enterprise information administrator” and reconciled with the university "glossary" of information descriptions.

Information Architecture Principle 8: Information Quality Needs to be Measured

Statement:

Information quality is neither abstract nor qualitative and should be measured in absolute terms.

Rationale:

Information quality is relative to the purpose to which it is to be applied. Decision makers need to not only have access to information; they also need to understand the timing, reconciliation, and accuracy of that information. Research shows that the quality of information is more important than the volume of information.

Implications:

- Most enterprises underestimate the extent, severity, and business implications of information quality problems. A pragmatic effort needs to be made to appreciate the types of problems that can exist and reveal them before plunging into data cleansing and integration efforts.
- Additional policies and procedures may need to be developed to implement this principle.

Information Architecture Principle 9: Use an Integrated Information Security Approach

Statement:

Security best practices are integrated throughout the entire software development lifecycle.

Rationale:

By embedding security and privacy throughout the software development lifecycle, the total cost of development is decreased, overall quality is improved, and security vulnerabilities are significantly reduced.

Implications:

- Design information security into information elements from the beginning.
- Approach security in a deliberate, well-defined, procedural manner--not as an afterthought or haphazardly.
- Design and integrate security in alignment with the organization's "business rules" and the existing laws and regulations that stipulate the safeguarding and the privacy of information.
- Identify and develop security needs at the information and data level, as well as the application level to protect information from unauthorized use and disclosure.
- Information security safeguards can be put in place to restrict access to "view only", or "never see". Sensitivity labeling for access to pre-decisional, decisional, restricted, sensitive, or proprietary information must be determined.

- Systems, information, and technologies must be protected from unauthorized access and manipulation, including inadvertent or unauthorized alteration, sabotage, disaster, or disclosure.
- Open sharing of information and the release of information must be balanced against the need to restrict the availability of classified, proprietary, and sensitive information.

Application Architecture Principles

Applications (and Systems) Architecture represents the IT application portfolio, the models for how applications are integrated, and identifies the business systems that enable and support the execution of UVU's business processes.

Application Architecture Principle 1: Applications are Aligned with Business Needs

Statement:

Applications will have an identified business owner and technical owner.

Rationale:

Technology is effective only when aligned with business needs. Both technical and business interests will be represented when making decisions.

Implications:

- Establish domain teams that include both business and IT experts that are responsible for creating the business architecture for their respective domains. Domain teams meet as needed but exist so long as the business domain does. As identified in Business Architecture Principle 4.
- Applications will have a stated business purpose. If there are multiple business purposes, they will be closely related.
- Architectural decisions will seek to maximize value to the customer.
- Architectural decisions will seek to simplify operations
- Architectural design precedes application development

Application Architecture Principle 2: Enterprise Solutions Chosen Over Point Solutions

Statement:

Enterprise applications will meet broad needs.

Rationale:

Enterprise applications that consider only a subset of needs are unsuitable for enterprise-wide use.

Implications:

- The project and deployment plan will need to consider this up-front. Adequate communication methods will be developed.
- Cross-silo solutions are preferred over duplicative silo-specific applications, systems, and tools.
- Solutions will consider the enterprise impact of architectural decisions.

Application Architecture Principle 3: Services Implementing a Bounded Business Context Communicate via APIs

Statement:

Services do not share data except through APIs

Rationale:

Each service must keep its model independent of other services. APIs provide the means of doing that.

Implications:

- APIs might be RESTful, event-driven, or message-based depending on the needs of the service.
- Services must make efforts to protect the consistency of their internal data model from other services.

Application Architecture Principle 4: Purchased Solutions are Preferred to Custom Solutions

Statement:

A decision to build custom applications should be made only after an analysis that considers other UVU sources and third-party alternatives will not meet the business requirement.

Rationale:

Decisions that are made without considering the alternatives may be expensive and difficult to support. There is no reason to re-invent something that already exists.

Implications:

- Existing technologies are to be considered before developing new ones.
- An analysis of industry-leading third-party solutions will also be considered.
- Consider "subscription" over purchasing where possible and practical.
- Total Cost of Ownership should be a part of any analysis.

Application Architecture Principle 5: Applications are Built to Support Continuous Integration/Continuous Delivery Methods

Statement:

Continuous delivery supports getting new features, configuration changes, bug fixes and experiments into production, or into the hands of users, safely and quickly and in a sustainable way.

Rationale:

Deployments should be routine and predictable to reduce risk, lower costs, and increase speed to market.

Implications:

- We will buy, develop, and use tools to support the deployment of systems, configurations, and bug fixes.
- All software development code will have integrated tests.

- Monitoring systems will detect failures and automatically roll back deployments when necessary.

Application Architecture Principle 6: Applications are Technologically Independent

Statement:

Applications are independent of specific technology choices and therefore can operate on a variety of technology platforms.

Rationale:

Applications should be independent of underlying technology to allow for the most cost-effective and timely development, upgrade, and operation.

Implications:

- Application Program Interfaces (APIs) will need to be developed so legacy applications can interoperate with applications and operating environments developed under the enterprise architecture.
- For Commercial Off-The-Shelf (COTS) applications, there may be limited current choices, as many of these applications are technology and platform dependent.
- Middleware should be used to decouple applications from specific software solutions.
- Technology independence will require standards that support portability.

Application Architecture Principle 7: Applications are Easy-to-Use and Consistent User Experience

Statement:

Applications should be easy to use with the underlying technology invisible to users, so they can concentrate on tasks at hand. Consistent user experiences insulate people from the proliferation of services implied by respecting business contexts.

Rationale:

Ease-of-use is a positive incentive application use. It encourages users to work within the integrated information environment instead of developing isolated systems to accomplish the task. A consistent user experience reduces error, increases user satisfaction and protects people from security and privacy breeches.

Implications:

- An application should be easy to learn.
- Content and navigation will be consistent.
- Guidelines for user interfaces should not be constrained by narrow assumptions about user location, language, systems training, or physical capability.
- Underlying technical implementations should be hidden from users.
- User actions should have predictable results.
- User interfaces will be as simple and intuitive as possible.

- User interfaces will be designed to maximize accessibility (to as wide an audience as possible).
- User interfaces will be designed to provide fail-safe features to protect users from unintended consequences of actions.

Application Architecture Principle 8: Applications are Easy-to-Support and Maintain

Statement:

Enterprise applications should be easy to support, maintain, and modify.

Rationale:

Enterprise applications that are easy to support, maintain, and modify lower the cost of support and improve the user experience.

Implications:

- Enterprise applications will be well documented. Documentation standards need to be developed that include architecture, design, and run book information.
- Applications have a limited life span—end of life should be anticipated and plans for replacement developed.
- Experimental or early release technologies will not be used unless they are critical to competitive advantage.
- Applications will handle errors in a controlled fashion and continue to operate normally (graceful degradation).
- Applications will have the following characteristics:
  - **Flexible:** Applications will be designed to minimize the costs of future changes.
  - **Extensible:** Applications will provide “hooks” (i.e., SOA APIs) that allow functionality to be extended in the future.
  - **Highly Available:** All applications will publish availability targets that have been agreed upon with the business.
  - **Interoperable:** Applications will be interoperable.
  - **Minimal Feature Set:** Features add complexity and should be kept to a minimum (avoid bells and whistles and systematic handling of improbable exceptions).
  - **Monitored and Measured:** Applications will be built to support monitoring and measurement.
  - **Scalable:** Applications will be designed to handle higher loads when allocated more resources.

Application Architecture Principle 9: Build for Reusability: Service Oriented Architecture (SOA)

Statement:

Applications should be assembled, in part, from reusable components or services.

Rationale:

Reusable components lower cost of subsequent application development.

Implications:

- Developers should create new components as part of the implementation of new functionality.
- Services are to be defined around business functions.
- Separation of Concerns: It will be possible to change a component with minimal impact to other components.
- Services will be loosely coupled (producers loosely coupled from consumers), self-describing, designed to maximize enterprise-wide reuse, discoverable, hide their underlying implementation details, stateless, autonomous, have significant control over the functions they provide and finally, have a defined security policy.

Application Architecture Principle 10: Plan for Integration

Statement:

Enterprise applications will plan for and include known, published mechanisms for integration.

Rationale:

Better application integration will reduce redundant data entry, will decrease the number of reconciliation problems and will make accurate data available quickly.

Implications:

- Application integration is to be planned for and is required in the early project planning process for enterprise applications and is to be included in the project plan and key deliverables such as requirements, analysis, and design.
- Interfaces are to be loosely coupled, backward compatible, self-describing, and offer a low impact to the enterprise when changes occur.[6]
- Integration points are to be published.
    - "Public" inputs and outputs of an application must be known, published, and understood to promote open data exchange and interfaces for enterprise application integration.
    - Diagrams of connections and data syntax and semantics should be published to promote re-use.
    - "Private" interfaces should not be published or used, as their stability is not

[6] When tightly coupled, interfaces are 1) less general and 2) more likely to result in undesired side effects when changed. Loose coupling means that services (e.g., enterprise APIs) are designed with no affinity to any particular service consumer. Inside the service, nothing is assumed as to the nature of the consumer. Thus, a service is fully de-coupled from a service consumer. However, the service consumer is dependent on the service (that is, it embeds literal references to service interfaces). The service is also responsible for exception handling. The result is a semi-coupled (or loosely coupled) architecture.

guaranteed.

- Open/Industry standards are preferred for enterprise application integration solutions; mechanisms should be language- and platform-independent.
- Enterprise application integration mechanisms should be as non-invasive to the applications as possible. For instance, data transformation should be done externally from the applications involved. Real-time integration is preferred over batch integration.

Application Architecture Principle 11: Applications are Standards-Based

Statement:

Applications that support open standards are preferred. Developers of community or enterprise applications will comply with all UVU standards in effect at the time. This principle applies to internal or outsourced development and to third-party software.

Rationale:

Compliance with standards will improve overall quality of applications. Therefore, when third-party software does not fully comply with UVU standards, a balanced evaluation is necessary.

Implications:

- UVU standards need to be published and made available to the developers' community.

Technology Architecture Principles

Technology Architecture Principle 1: Platforms for Mission-Critical, Enterprise-Class Solutions are CTO-Approved

Statement:

Mission-critical, enterprise-class solutions are to run on CTO-approved and supported platforms.

Rationale:

Because of the inherent risks to the University of developing, deploying, integrating, and managing applications on non-secure and non-stable platforms, it is critical that these be vetted and approved by the CTO.

Implications:

- CTO-approved platform standards and guidelines need to be developed.
- Because of the university's investment in centralizing the expertise and resources required to provide and manage mission-critical, enterprise-class applications in OIT, OIT will be the preferred provider of platforms for this purpose.
- Cloud, hybrid, or on-premises platform will be determined on a cloud-smart approach. This approach will determine the appropriate platform where the application/service will reside. The ever-evolving architecture of the cloud and need for cost control requires the ability to move in and out of the cloud environment without impact to the applications/services provided by OIT.
- Platforms can be cloud, hybrid, or on-premises supported. These will be overseen by Dx engineers.
- Platforms, whether OIT-provided or not, are to be highly scalable, reliable and deliver necessary performance
- OIT will leverage strategic relationships with other businesses and vendors to facilitate building and evolution of IT architecture.

Technology Architecture Principle 2: Platforms are Available through Self-Service

Statement:

The campus community can acquire platforms via an online order fulfillment process.

Rationale:

The process to deploy approved university systems onto these platforms should be well-documented and simple to use.

Implications:

- Members of the campus community can obtain enterprise platforms from OIT without a formal project.

- Platforms are well documented, well supported, and available to be used much like other platforms in industry by providers like Amazon.com.
- New systems, or upgrades to existing solutions that require additional customizations should request access to OIT engineers early in the discovery process to identify requirements, timelines, and key deliverables

Technology Architecture Principle 3: Platforms are Customizable

Statement:

Basic server platform configurations are pre-configured, well documented and “ready to use.” These platforms are readily available to approved members of the campus community. Users can request additional features, or “custom” (non-standard) versions of platforms.

Rationale:

Enterprise platforms are flexible and easily customized. Documentation is readily available and informs the user on how to use it, how to customize it, and how to make and save changes.

Implications:

- Approved users, or consumers of IT systems and platforms, can access, download, and customize standard configurations.
- Once installed, users can make enhancement requests, save modifications to, and even share alternate versions.

Technology Architecture Principle 4: Platforms are Scalable

Statement:

IT platforms can grow in capacity to match the dynamic demands of applications and systems. IT platform interfaces will be loosely coupled, backward compatible, self-describing, and offer a low impact to the enterprise if changed. Like applications, platforms are interoperable and have published API’s and established interfaces for integration.

Rationale:

Loosely coupled interfaces result in solutions that are more general and require less effort to change when necessary.

Implications:

- Elastic, scalable, and flexible
- Well-documented APIs are necessary.
- “Public” inputs and outputs of an application must be known, published, and understood to promote open data exchange and interfaces for enterprise application integration.
- Diagrams of connections and data syntax and semantics should be published to promote re-use.

- The IT platform architecture and related components built upon it should be viewed as a set of independent services that can be composed to provide a solution.

Systems Management Principles

Systems Management Principle 1: Enterprise Systems are Managed by Central IT (Dx)

Statement:

Systems Management solutions are expected to be centrally provided regardless of the platform and include monitoring and management of key functions.

Rationale:

System providers need a complete view of all the components that support enterprise applications, including those outside of OIT, so they can provide stable and reliable services to their users.

Implications:

- All systems and IT services should have meaningful performance measurements on which to base reviews and decisions for future direction and improvement needs.

Systems Management Principle 2: Systems Management is Part of Every Solution Design

Statement:

As a critical element to a "completely finished solution" systems management is to be designed into the solution early in the design process.

Rationale:

Without adequate systems management, service providers cannot be proactive in avoiding user problems with their solutions.

Implications:

- Solutions should be designed to be autonomic, self-operating, self-monitoring/analyzing, self-protecting, and self-optimizing.
- In the event of an error condition that cannot be self-corrected, systems should be designed to report appropriate events describing the error condition.
- Events regarding a system's availability, performance, &etc., should also be identified and reported for each system.
- Systems management must not only resolve immediate issues, but it must also facilitate the organization's ability to continuously improve its products and services

Systems Management Principle 3: Systems Management Tools are Self-Service

Statement:

Systems Management tools will provide a simple self-service interface that allows providers and users to check system, network, or problem status.

Rationale:

Users and administrators should have access to system, network, or problem status without having to call the Service Desk or NOC. This capability would assist them in their own troubleshooting and service level processes. Best practices also recommend allowing users to check the status of their own problems.

Implications:

- Systems should be designed to provide support and operational access tools/panels that allow controls for diagnostics, reset-restart-restore, performance measurement, and fulfillment—in short, functions that require technician interaction. This access provides for secure, limited human interface when needed.
- Self-reporting is preferred.
- Monitoring information should be available through web service APIs.

Network Architecture Principles

Key to the university infrastructure is the network. We believe that the network should be open, easy to access, and adequate in capacity to deliver information quickly and securely.

Network Architecture Principle 1: Networks are Centrally Provided and Managed

Statement:

The Office of IT (OIT) operates both enterprise wired and wireless networks.

Rationale:

OIT network engineers are expertly trained and uniquely qualified to develop, operate and maintain the network. Much like platform architecture, this centralized approach will reduce total cost of ownership to the university, minimize complexity, increase reliability, and improve performance.

Implications:

- OIT engineers will keep UVU networks current with hardware and software provider's updates and releases.
- The network will meet the network service level objectives agreed upon by OIT management, campus users, and external partners. The network management systems will have the capability to measure and report end-to-end network service level statistics.
- Service Level Agreements (SLAs) will be documented, and network statistics and SLAs will be published to the end-user community, as needed, to ensure network resources are aligned with business needs.

Network Architecture Principle 2: Networks Are Standards-Based

Statement:

UVU-hosted networks refer to both wired and wireless communications and adhere to industry standards for communication protocols, hardware, and security.

Rationale:

Adherence to proven and adopted industry standards for network architecture will improve the use, operation, and management of wired and wireless networks on campus. This approach ensures the best possible service for network users, will reduce unnecessary costs, and maintain security.

Implications:

- UVU networks will use open industry standards and leverage existing best practices.
- Open, rather than proprietary, systems will be used where feasible and cost effective.
- OIT engineers will minimize the number of platforms and systems supported.
- UVU will need to have experts assigned to track evolving trends and improvements in network security.

- UVU networks will leverage strategic relationships with other businesses and vendors to facilitate the building and evolution of IT architecture.
- Improvements and upgrades to infrastructure will need to be planned, resourced, and communicated to sponsors and users.
- Audits and reviews will be necessary to monitor compliance.

Network Architecture Principle 3: Networks Are Readily Accessible

Statement:

Users (visitors, students, faculty, staff, &etc.) should be able to quickly and easily access UVU networks.

Rationale:

Users can easily, if not seamlessly, attach devices. The remote access network will consist of a defined set of technical options for the delivery of reliable, cost-effective, secure, and ubiquitous remote access capabilities.

Implications:

- Network tools, like Virtual Private Network (VPN) clients, must be issued to users wishing to access secure resources remotely.
- It is not necessary to register devices, scan devices, &etc.
- UVU provides users with the ability to access the network remotely.

Network Architecture Principle 4: Networks Are Fast

Statement:

UVU networks will provide application response times acceptable to support the business need and cost-effective bandwidth to satisfy the current and future networking needs of its users. The UVU network should meet the requirements of its users and provide the ability to deliver information.

Rationale:

The network will not be overbuilt and will be efficient. The network should also be able to accommodate future networking needs. In conjunction with response time, throughput ensures that information is adequately delivered.

Implications:

- Network performance is measured in terms of “response times” which should always be monitored and optimized.
- The network should be designed to provide appropriate application response times at a reasonable cost.

Information Availability and Security Principles

Security needs to be everyone's focus. The three common goals of Information Availability and Security are:

Availability - For any information system to serve its purpose, the information must be available to authorized individuals and systems when needed.

Confidentiality - Confidentiality means that disclosure of information to unauthorized individuals or systems is prevented.

Integrity - In Information Security, integrity means that data is accurate and cannot be modified undetectably by unauthorized individuals, or systems.

Information Availability and Security Principle 1: Alignment with Security Policies

Statement:

Security policies should drive the implementation of business and technical security controls.

Rationale:

Business and technical security controls are put in place to enforce compliance with existing security policies.

Implications:

- There should be a way to monitor and measure the security compliance of each IT system.
- Information stewards review security compliance regularly.

Information Availability and Security Principle 2: Least Privilege

Statement:

The principle of least privilege requires that in a particular abstraction layer of a computing environment, every module (such as a process, a user, or a program depending on the subject) must be able to access only the information and resources that are necessary for its legitimate purpose.

Rationale:

Least privilege is an important design consideration protecting data and functionality from faults and malicious behavior. Benefits include better system stability, better system security, and ease of deployment.

Implications:

- User accounts should be given only those privileges essential to that user's work.
- Applications should only acquire information necessary for their operations.
- Applications should not store information that is readily available from online sources.

Information Availability and Security Principle 3: Adequate Security Controls

Statement:

Security controls will be selected based on a risk analysis and risk management decision; will provide only the required level of protection, alerting, and response; will consider the ability to be applied uniformly across the UVU enterprise (standardized); and will be modular so that they may be removed or changed as the system and enterprise risk profile changes

Rationale:

By providing only the required levels, we will not impose unreasonable constraints or operate in a manner that causes unreasonable response.

Achieving a standards-based environment will reduce operational costs, improve interoperability, and improve supportability.

Following the idea of trust but verifying, the interdependence of controls should be minimized.

Implications:

- The appropriate security officers and committees will need to be involved.
- Controls should default to the most secure condition.
- Controls should have the capability to be shut down gracefully and restored automatically to the conditions prior to shut down.
- The CIO, acting as primary sponsor, should be apprised of any changes, or recommendations.
- Consideration for industry-proven standards will be applied in the requirements phase of project definition.
- The implementation of controls should be aligned closely with operations' personnel and adhere to approved procedures for change management.
- Protection and response are the responsibility of the appropriate information stewards.
- The implementation of controls should be aligned closely with operations' personnel and adhere to approved procedures for change management.

Information Availability and Security Principle 4: Security Planning

Statement:

Information systems security will be built into systems from their inception rather than "bolted on" after system implementation.

Rationale:

The cost and complexity of adding security controls to a system after it is already in production is significantly greater. Controls should provide only the required level of protection, alerting, and response.

Implications:

- Preliminary architectural plans will address security consideration before systems are purchased and/or developed internally.
- Early design work will need to include the definitions of how/where controls will be implemented.
- Security officers and appropriate committees (to be identified, i.e., ITPC) will review large-scale, enterprise-wide impact to security modifications.
- Protection and response decisions should align with the appropriate information stewards.

Information Availability and Security Principle 5: Security Measurement

Statement:

All functional security requirements that define the "what" a system or product does will have associated assurance requirements to define "how well it does it."

Rationale:

To ensure that risk is being maintained at acceptable levels, security controls that can be reviewed or audited through some qualitative or quantitative means are needed.

Implications:

- Documentation will need to be developed to allow approved groups to review and measure performance of security systems.

Information Availability and Security Principle 6: Compartmentalization and Defense-in-Depth

Statement:

The architecture will embrace the concepts of compartmentalization and defense-in-depth.

Rationale:

Compartmentalization localizes vulnerabilities and defense-in-depth establishes a series of imperfect countermeasures.

Implications:

- Compartmentalization concepts will need to be included in both preliminary and detailed designs.
- Management and review of various areas will need to be assigned to the appropriate party.

Information Availability and Security Principle 7: Manual Operations

Statement:

Controls will minimize the need for manual operation.

Rationale:

Manual operation can create vulnerabilities and cause disruption of service due to misinterpretation and misconfiguration.

Implications:

- Automation will be a key requirement in the deployment of each new system.
- Standardization with automation is preferred over manual security operations/settings.
- Documentation will need to be developed on how to implement these systems.

Information Availability and Security Principle 8: Separation of Duties

Statement:

The designer and the operator of security controls will be independent of each other.

Rationale:

Separation of duties ensures that there is no conflict of interest in the design and implementation of security controls.

Implications:

- The appropriate roles and responsibilities for operation and implementation of security systems will need to be clearly defined and communicated.
- Separation of roles and accountability between security policies and security control implementations providing an appropriate check and balance.

Information Availability and Security Principle 9: Standards, Rules, and Services for Business Continuity and Disaster Recovery

Statement:

Systems in production will provide documentation standards for “start, stop and restart” that allow the Operations team to respond to outages and service interruption quickly and securely. Additionally, production systems will be enabled for monitoring purposes. These operational standards and monitoring rules will enable support personnel to better restore service to students, faculty, and employees.

Rationale:

Standards, Rules, and Services documentation are essential for successful production systems and must support the guidelines established within this document.

Implications:

- Established standards, rules, and services provide the needed requirements for business continuity and ensure IT services meet these requirements before they are production ready. The Operations team is responsible for maintaining this information.
- Standards, rules, and services documentation for business continuity and disaster recovery ensure basic parameters and requirements are established and known to developers, administrators, and other departments allowing all future systems to establish a consistent environment, methodology, expectation according to the EAG.