Enterprise Architecture
Workgroup Kickoff

December 11, 2014  |  Thursday  |  3:00-4:00 p.m.  |  8 Story Street #6016
Agenda

• The EA Vision
• Milestones for EA at Harvard
• Governance
• EA Program Approach
• Definition of Terms
• Architecture Maturity
• EA Focus Areas
## The EA Vision

### Our Vision for Enterprise Architecture

Provide a technology framework and a set of standards to enable acquisition, development, and deployment of IT services that maximize interoperation, minimize duplication, and simplify the IT environment across all of Harvard.

### Strategic Objectives

- Deliver an enterprise architecture framework that drives technology and development standards across Harvard
- Provide common approaches for integration across enterprise applications, processes, and data
- Align and rationalize technology decisions and investments
- Identify redundant or conflicting processes and data across organizations

### Guiding Principles

- Ensure that EA provides active direction and delivers value to the organization
- Counter complexity with common solutions
- Enable sharing of data across organizations
- Preference for open-source, COTS, and programmatic interfaces — both in what we obtain and what is produced
- Encourage, define, and ultimately provide best-practice solutions
- Evolve framework and solutions with advances in technology

### Key Performance Indicators

- Decrease in project delivery timeframes to production
- Increase in the number of integrated applications using programmatic interfaces
- Increase in the number of funded projects that conform to an EA Checklist
- Decrease in ad-hoc data sharing
- Increase in automated data exchange
- Increase in the number of known authoritative data sources
- Decrease in the number of copies of data
## Milestones for EA at Harvard

<table>
<thead>
<tr>
<th>HUIT Top 40 Goal</th>
<th>EA Milestones</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>20. Establish an IT enterprise architecture</strong></td>
<td>Oct 2014: Launch EA strategic initiative, including vision and strategic plan</td>
</tr>
<tr>
<td></td>
<td>Dec 2014: Define a Harvard EA framework to incorporate key elements in principles, data, integration, and technology architecture</td>
</tr>
<tr>
<td></td>
<td>March 2015: Conduct a current state analysis on integration to identify data passed between enterprise applications and the means of exchange</td>
</tr>
<tr>
<td><strong>21. Implement an architecture review process</strong></td>
<td>Sept 2014: Identify a set of technical architects who can undertake architectural reviews</td>
</tr>
<tr>
<td></td>
<td>Oct 2014: Review and refresh PRC technical review process</td>
</tr>
<tr>
<td></td>
<td>Dec 2014: Review and refresh ITCRB technical review process</td>
</tr>
</tbody>
</table>
## Governance

### Enterprise Architecture Executive Committee

IT executives who ensure that the vision and plan are addressed by the working group. Also provides consistent direction and problem-solving approaches for the working group and the EA program at large. **Meets monthly.**

**Co-Chairs: Anne Margulies and Stephen Gallagher**  
Members: Scott Bradner, Ben Gaucherin, Stephen Ervin, Gabriele Fariello, Praneeth Machettira, Pratike Patel, Jason Shaffner, Jason Snyder, Jim Waldo, Bob Wittstein

### Enterprise Architecture Working Group

- Technical members of HUIT, Harvard Schools, and other IT departments that meet on a regular basis
- Defines the Enterprise Architecture framework for review by Steering Committee
- Defines sub-groups to detail layers
- Builds and reviews other EA components as per vision
- Publishes a monthly report on enterprise architecture progress, issues, and direction for the organization

**Chair: Jason Snyder**  
Members: Scott Bradner, Bill Brickman, Dan Kaplan, Arnold Paul, Robert Piscitello, Jon Saperia, Raoul Sevier, Michael Thomas
EA Program Approach

Re-Evaluate: Identify Places Where EA Can Make an Impact

Layers

- Security
  - User Experience
  - Applications and Software Components
  - Interoperation
  - Data
  - Middleware
  - Infrastructure and IaaS
  - Networking

Advisories, Methodologies, and Principles

- Architects
- UX Consultation
- Ad-Hoc Consultation
- ITCRB and PRC Reviews
- Evaluate Skills & Organizational Needs

Communication & Education

Enterprise Architecture Implementation Plan

Requirements and Needs

Enterprise Technology Assessment

Technology Trends and Best Practices

Outreach and Training
# Definition of Terms: EA Layers

<table>
<thead>
<tr>
<th>Layers</th>
<th>Definition</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Xperience</strong></td>
<td>End-user look-and-feel and navigation style of an application or service</td>
<td>The appearance of the Harvard brand, color schemes, use of breadcrumbs, position and appearance of navigation bars.</td>
</tr>
<tr>
<td><strong>Applications, services, SaaS</strong></td>
<td>Algorithms and code that provide technical or business value</td>
<td>Large-scale applications such as the Student Information System (SIS), small applications such as Electronic Submission Tracking and Reporting (ESTR), services such as Informatica for data transfers, and Software-as-a-Service solutions such as Office365.</td>
</tr>
<tr>
<td><strong>Interoperation</strong></td>
<td>Exchanges of information and provisioning of business transactions between different applications and services</td>
<td>Examples of information exchanges include transfers of student registration from SIS systems to central directories and transfer of account balance values from financial systems to CRM systems. An example of a remote service is the Identity and Access Management service for Authentication.</td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td>Information represented in formats that are managed by applications and services</td>
<td>Data includes structured information such as student records and general ledger financial data. Examples of unstructured data include electronic books, the content of wikis, and most of the information available from the Internet.</td>
</tr>
<tr>
<td><strong>Middleware</strong></td>
<td>Common business or technical services that are implemented separately from applications and services</td>
<td>Database technologies are the most common example of middleware, but this layer can also include reporting ‘engines’, rules ‘engines’, application servers, data transfer applications, and other common shared services.</td>
</tr>
<tr>
<td><strong>Infrastructure</strong></td>
<td>Hardware and virtualized platforms that operate applications, services, and their components</td>
<td>Servers, associated storage components, operating systems, and other computing devices are the common examples of infrastructure, more recently joined by cloud-based infrastructures of Platform-as-a-Service and Infrastructure-as-a-Service.</td>
</tr>
<tr>
<td><strong>Networks</strong></td>
<td>Communications technologies that join infrastructures in disparate locations</td>
<td>Technologies that allow computing devices to communicate with each other include wired and wireless communications supported by devices such as routers, switches, and naming services.</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>Use of resources by authorized individuals and computing services to information, business functions, and computing services</td>
<td>Examples of security mechanisms include door locks, user IDs and passwords, and intrusion detection/prevention tools. These mechanisms are supported by applications and services that manage user and systemic authentication, authorization, access to functionality, and access to data.</td>
</tr>
</tbody>
</table>
## Definition of Terms: EA Deliverables

<table>
<thead>
<tr>
<th>Deliverables</th>
<th>Definition</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Principles</td>
<td>Foundational elements to drive decision-making and alignment</td>
<td>Principles can be applied at many levels, from guiding principles that characterize strategic, enterprise-wide systemic behavior to principles that help explain detailed technical behaviors of applications and services.</td>
</tr>
<tr>
<td>Methodologies</td>
<td>Methodologies divide IT work into phases containing activities with the intent of better planning and management; they help determine which methods or “best practices” should be applied to specific cases, and may include specific deliverables and artifacts</td>
<td>Examples of IT methodologies include waterfall, prototyping, iterative and incremental development, spiral development, rapid application development, extreme programming, and Agile.</td>
</tr>
<tr>
<td>Advisories</td>
<td>Recommendations offered as a guide to specific actions or practices</td>
<td>Common examples of advisories include security notifications of newly discovered vulnerabilities with recommendations for patching systems or changing passwords, and announcements of changes to the features, forms, or functions of applications.</td>
</tr>
<tr>
<td>Patterns</td>
<td>Generic models or descriptions from which specific implementations can be based or derived</td>
<td>In the IT context, patterns include reusable approaches for connecting applications to databases, establishing user security within an application, and implementing user experience in a solution.</td>
</tr>
<tr>
<td>Reference Architectures</td>
<td>A template solution, using multiple patterns and a vocabulary that promotes commonality, defining an architecture for a particular domain</td>
<td>Examples of business reference architectures include Insurance Application Architecture for the insurance domain, and ‘HL7 V2.5’ for the electronic health record domain. An example of a technical reference architecture is the Java Enterprise Edition for IT systems construction.</td>
</tr>
<tr>
<td>Outreach</td>
<td>Elevating awareness of programs and initiatives to affected populations</td>
<td>Examples of broadly-focused outreach include ABCD meetings on many IT topics, while more narrowly focused outreach include Big Group meetings regarding IT skills upgrades.</td>
</tr>
<tr>
<td>Training</td>
<td>The acquisition of knowledge and skills as a result of teaching that relates to specific competencies, with the goals of improving an individual’s productivity and performance</td>
<td>IT training of techniques could include database design, software coding in node.js, and process modeling with BPMN. Examples of vendor tool training include Oracle Financials, PeopleSoft, and Informatica ETL.</td>
</tr>
</tbody>
</table>
Architecture Maturity

Focus on Architecture as a Process

Establish EA outcomes on a maturity scale in order to deliver value at all stages of the program.

EA Maturity

Continuous-Improved EA Program

- EA maturity is defined.
- Architecture vision is consistent.
- Architecture activities and budgeting processes are well defined.
- Rigorous governance processes are in place.
- EA compliance processes are formal.
- Teams across the organization use consistent and up-to-date data.
- Enterprise data is well defined and organized.
- Cross-organizational and cross-domain standards and practices are in place.
- Teams across the organization collaborate on cross-organizational initiatives.
- EA stakeholders contribute to cross-organizational and cross-domain standards and practices.
- EA governance committees are well established.
- Architecture board reviews work.
- Well-defined reference models and libraries are in place.
- Architecture patterns, based on consistent processes, are in place.
- EA compliance process is followed consistently, with exception and change processes.
- Interoperability requirements are considered in strategic planning and budgeting processes.

Solution Architecture Reference Models

User Experience

- UX standards are in place.
- UX is visible across composite applications.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
- Implementations use external definitions for easy and rapid evolution.
## EA Focus Areas

### Enterprise Architecture: Interoperability

Deliver a plan outlining how we will deliver the following for interoperability:

<table>
<thead>
<tr>
<th>Current State</th>
<th>Requirements &amp; Frameworks</th>
<th>Future Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Current-state integration analysis</td>
<td>• Principles, methodologies, and advisories</td>
<td>• Building a place to publish the outcomes</td>
</tr>
<tr>
<td>• Identify data passed between enterprise applications</td>
<td>• Patterns, reference architectures, and standards</td>
<td>• Next steps</td>
</tr>
<tr>
<td>• Identify the means of exchange</td>
<td>• Resource requirements, including software and services</td>
<td>• Deliverables</td>
</tr>
<tr>
<td></td>
<td>• Identification of a repository and process for keeping the information gathered up to date</td>
<td>• Timings</td>
</tr>
</tbody>
</table>
Questions or comments?
Thank you!