Putting IT all together

Dave Berry
Enterprise Architect
What do our online services look like to students?
Inconsistent interfaces
The current ("as-is") architecture – a small subset
Timetabling: the applications architecture in more detail
An illustration of what our services could look like
Consistent appearance; Meaningful names
Let’s look at what this entails behind the scenes

(Business) Processes

Applications

Data

Technology (Hardware & Software)
Processes: Map the User Journey

Accept offer
- Reading List
- VLE
- E-mail

Prepare
- Where to stay
- Student Union

Register
- Personal Tutor
- Tuition Fees

All parties working together

Agree the communication plan
Process: Digital First

- (Re-)Design processes from scratch to be used online
  - Don’t just replicate existing processes
- What do our students & staff want to do?
- How can we make this easy?
  - Let them focus on learning, teaching & research
Processes: Avoid duplication

• One way of doing each task
  – Consistency for users
  – Consistency for support staff
  – Allows us to focus on the user experience instead of spending time implementing multiple variations
Applications: Integrated & Personalised

Accept offer • Reading List • VLE • E-mail

Prepare • Where to stay • Student Union

Register • Personal Tutor • Tuition Fees

All information in a single place

All important dates in the student’s diary
Applications: Reusable Building Blocks

Notifications

Presentation Layer

Microservices (REST, JSON, Swagger, etc)

Back-office systems

EUCLID Library Learn Office 365 Accommodation
Applications: Easy to use

- Usability workshops – with real users
- Web code libraries: responsive design
- Accessible by design
Applications: Consistent experience

• **Global Experience Language**
  – Websites instantly recognised as part of the Edinburgh family

• **Currently implemented for EdWeb**

• **We should implement EdGEL for web applications, e-mails, etc.**

• **C.f. BBC GEL** [www.bbc.co.uk/gel](http://www.bbc.co.uk/gel)
• Definitive value of core data
  – E.g. EUCLID = Student Record
• All updates made to golden copy
  – May propagate to other systems
• Example:
  – If a student updates their address in EUCLID; this should update their address in Payroll too.
• Golden Copy Data Catalogue
Data is a shared asset

• Data from golden copies must be made available
  – Other systems need it
  – APIs don’t work without it!
• Prevents discrepancies between central and local systems
• Appoint *Data Stewards* to manage access and quality
Data: Common vocabulary and definitions

- Present consistent information
  - Using terms that users understand
  - Shared across org units

- University data model
  - Shared across IT systems
  - If necessary, translate vendor terminology to ours
Data Life Cycle

- Ensure that data is correct and current
  - Processes must keep data up to date
- Meet legal requirements
  - Data protection
  - Data retention
- Meet Business Intelligence requirements
  - Data warehouse
Technology: Standards

- Standards allow building blocks to fit together easily
- Formal (ISO) standards and industry standards
- Building software
  - REST, HTML5, Swagger, ...
- Accessibility
  - WCAG
- Data integration
  - JSON, LTI, XCRI-CAP, ...
- Security
  - OAuth2, Kerberos, ...
- Messaging...
- Infrastructure...
Technology: Monitoring and Analytics

- Usage, performance, availability
- Optimise user experience
- Analyse infrastructure requirements
- Timeously respond to problems
Technology: Automated and Scalable Provisioning

- Consistent results every time
- Automated testing for assured reliability
- Quicker turn-around time
- Deploy on-premises, in the cloud, or a hybrid
Process: This is for everyone

These principles apply to all organisational units within the University
## Recap: Key Architecture Principles

### (Business) Processes
- Integrated and Personalised Experience
- Reusable Building Blocks
- Consistent User Experience
- Easy to Use
- Design for Failures
- Innovation

### Applications
- This is for Everyone
- Digital First
- Map the User Journey
- Avoid Duplication
- Common Use
- Federation

### Data
- Golden Copies and Data Stewards
- Data is a Shared Asset
- Common Vocabulary and Definitions
- Extract Value from Data
- Data Security
- Data Life Cycle

### Technology (Hardware & Software)
- Manage Technical Diversity
- Standards
- Monitoring and Analytics
- Automated and Scalable Provisioning
- IT Security
- Technology Life Cycle

### Technology
- Manage Technical Diversity
- Standards
- Monitoring and Analytics
- Automated and Scalable Provisioning
- IT Security
- Technology Life Cycle

<table>
<thead>
<tr>
<th>Applications</th>
<th>Data</th>
<th>Technology (Hardware &amp; Software)</th>
<th>(Business) Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>This is for Everyone</td>
<td>Golden Copies and Data Stewards</td>
<td>Manage Technical Diversity</td>
<td>Integrated and Personalised Experience</td>
</tr>
<tr>
<td>Digital First</td>
<td>Data is a Shared Asset</td>
<td>Standards</td>
<td>Reusable Building Blocks</td>
</tr>
<tr>
<td>Map the User Journey</td>
<td>Common Vocabulary and Definitions</td>
<td>Monitoring and Analytics</td>
<td>Consistent User Experience</td>
</tr>
<tr>
<td>Avoid Duplication</td>
<td>Extract Value from Data</td>
<td>Automated and Scalable Provisioning</td>
<td>Easy to Use</td>
</tr>
<tr>
<td>Common Use</td>
<td>Data Security</td>
<td>IT Security</td>
<td>Design for Failures</td>
</tr>
<tr>
<td>Federation</td>
<td>Data Life Cycle</td>
<td>Technology Life Cycle</td>
<td>Innovation</td>
</tr>
</tbody>
</table>
What has all this to do with architecture?

- In IT, “Architecture” refers to the relationships between components
  - Infrastructure, applications, data, business functions & processes
- ... and the principles and guidelines governing their design and implementation over time
- Target blueprints, that describe where we’re heading
- Co-ordinating across organisational units
The question is not whether you have architecture...

- You always have some sort of architecture
- Even if you don’t know what it is
- And even if it isn’t pretty
The question is how you manage it to enable change

Enterprise Architecture
- Know the current state
- Plan the desired state

Solution Architecture
- Agree a roadmap
- Implement incrementally
Architecture can work at different scales

• Co-ordinating for the entire University
  – E.g. Contact management
• Large-scale review of an entire area
  – E.g. HR, or Student Admin
• Review of a particular service
  – E.g. to reduce support calls
  – E.g. to exploit new capabilities
Coda: What do our support services look like to users?

- A query can end up like a game of “pass the parcel”
  - We should treat users as members of the University, not as parcels
Support: Integrated & Personalised

**Academic**
- Change course selection
- Special circumstances

**Finance & Accommodation**
- Student loans
- Outstanding invoices
- Where to stay

**Information Systems**
- Printing
- Wifi access
- Library books

All queries via any channel

The person you ask ensures the question is resolved