

Preliminary Recommendations for Application Platforms to Foster Interoperability, Sharing, and Reuse

DRAFT - Preliminary Recommendations for Application Platforms to Foster Interoperability, Sharing, and Reuse - DRAFT

Table of Contents

- 1 Purpose
- 2 Audience
- 3 Definitions
- 4 Scope
- 5 Summary of Recommendations
- 6 Analysis and Discussion
 - 6.1 Overview
 - 6.2 When Does Platform Matter?
 - 6.3 Common Platforms at UCD
 - 6.4 Platform Context
 - 6.5 Additional Considerations
- 7 Future Work

Purpose

This document contains initial recommendations from the [Software Standards Work Group \(SSWG\)](#) for applications acquired, developed and operated at UC Davis.

The purpose of these preliminary recommendations is to:

- Foster *interoperability* among UC Davis's software applications to enable integration and realignment of business processes and information exchange, and
- *Sharing* and *reuse* of software components. Sharing and reuse eliminate redundancy of effort, saving resources and allowing development groups to focus on their unique needs. Sharing also can also enforce common implementations of business rules.

Audience

This document addresses two audiences:

1. IT administrators and decision makers who are bringing new applications to the campus, whether purchasing or building.
2. Software developers and systems administrators who are responsible for developing, deploying, maintaining and supporting applications for the campus.

Definitions

- **Platform.** "Platform" refers to the stack of hardware, operating system, database servers, middleware, programming languages, *etc.* that is required to run an application. Some platforms are flexible at some layers of the stack, allowing alternative selections for operating system, hardware, database, *etc.*, and others are less flexible.

For the purpose of this document, we have adopted UC's [Information Technology Architecture Group's \(ITAG\)](#) definitions of interoperability, sharing, and reuse:

- **Interoperability.** A software component is *interoperable* with another component if it is able to use the other's services as part of its own, particularly over a network. The prerequisite for interoperability is common semantics, but interoperability is facilitated when network protocols and data formats are also common. When two different organizations provide the two components, common organizational trust is also required.
- **Sharing.** A software component is *shared* by two or more organizations when a single instance of that component is operated to provide

- services to those organizations.
- **Reuse.** A software component is *reused* by two or more organizations when those organizations each install a copy of that component to provide services to themselves.

Scope

There are many factors to consider in selecting a platform, including existing infrastructure and expertise, scalability, security, *etc.* However, the recommendations in this document focus on the factors most salient for interoperability, sharing and reuse to the exclusion of other factors. These recommendations are grounded in UCD's current IT environment. Future work for the SSWG will include extensive discussion of what changes to that environment might be desirable in the future.

Summary of Recommendations

The following table summarizes the preliminary recommendations of the SSWG. Please see the subsequent discussion section for a more detailed and nuanced analysis and explanation.

Scope of Application	Platform Recommendations
Core Administrative Systems of Record	Java
Shared among Multiple Colleges and Departments	Java, .NET, PHP, or ColdFusion
Reused among Multiple Colleges and Departments	Consider the platforms in use or supported by recipient departments
Local to a Single College or Department	Any platform acceptable, but UCD-supported platforms still recommended.

Analysis and Discussion

Overview

The selection of a platform can affect many aspects of the design, implementation, operation, and maintenance of software applications. This document focuses on those selection criteria most salient to interoperability, sharing and reuse. There are, of course, many other reasons for choosing a platform, such as existing infrastructure and expertise, scalability, security, *etc.*

These recommendations are grounded in UCD's current IT environment. Future work for the SSWG on [Platform Strategy](#), to be completed early in 2010, will include extensive discussion of what changes to that environment might be desirable in the future.

As mentioned above, these recommendations are preliminary. They are intended to provide quick guidance to IT administrators and application developers in the short term. See the [Future Work](#) section of this document for continuing work the SSWG will do to foster interoperability, sharing, and reuse.

When Does Platform Matter?

For interoperability and sharing, platform is often not a major factor.

Reuse of software, however, has a heavy dependence on platform. In order for a group to operate an application that has been built elsewhere, a number of conditions must be met:

- The platform's technical infrastructure must be in place, either locally, or from a service provider.
- The group must have expertise in the management of the platform.
- If the group will be performing development or modification of the application, it must have expertise in the development languages and tools.
- The group must have sufficient resources (human and otherwise) to do all of this.

Acquiring any or all of these can be difficult and costly. If an application is being designed, and there is an expectation of someone else operating it in the future, care should be taken in platform selection.

Interoperability and sharing are more factors of the strict adoption of interface protocols and formats, rather than platform. Some platforms may have features to make the construction of those interfaces easier, but the construction of interfaces is not generally a dominating factor in the overall effort required for an application.

It should be noted that platforms may be assembled from components supported by multiple groups. For example, the College of Letters & Science's Time Sheets system is moving to a three-way partnership where the hardware is managed by Information and Educational Technology, the other software components are managed by the College of Letters & Science, and Administration and Resource Management provides end-user support.

Common Platforms at UCD

The SSWG considered the following generic platform types. They represent the principle platforms currently in use at UC Davis.

- **Java** - Java is probably the most-used platform for large-scale, enterprise administrative applications in the IT industry and within higher education particularly. It has been adopted by numerous software development communities, including Quali, Jasig, Sakai, and Internet2 middleware. Java applications can be executed on many hardware / operating system configurations.
- **.NET** - Like Java, .NET is used widely for large-scale, enterprise applications in the industry, but less so in higher education. At UC Davis, it has been used to build many applications in colleges and administrative units, and its development tools are well-suited for development of applications ranging from medium to high complexity. .NET applications must be executed on system configurations that use the Windows operating system.
- **ColdFusion** - ColdFusion is a platform that is well-suited for quick development of relatively simple applications, such as database reporting. It also tends to require fewer support personnel than Java and .NET. Historically, it has been popular for use at UC Davis. The advantage of ColdFusion is that its simple to use language can be learned in a relatively short time. The current version of ColdFusion core is built on a on Java foundation, so it can be executed on many hardware / operating system configurations.
- **PHP** - Like ColdFusion, PHP is a platform that is well-suited for quick development of relatively simple applications, and it tends to require relatively fewer support personnel. It also is a simple to use language can be learned in a relatively short time.

It should be noted that, at this point in time, the SSWG did not consider specific configurations of the components that comprise specific instances of these platforms, although that is work that the SSWG will undertake.

Platform Context

Not all platforms are appropriate for all application contexts. The SSWG considered the following contexts:

- Core administrative systems of record, such as Banner, DaFIS, or PPS
- Applications that are shared among multiple colleges and departments
- Applications that are reused among multiple colleges and departments
- Applications that are local to a single college or department

The following sections recommend platforms for each of these contexts.

Core Administrative Systems of Record

Because of the crucial need for business continuity associated with these systems, the ability to share services and expertise among core administrative systems is critical. Cost effectiveness is also crucial. In order to have sharable services and expertise at a minimum cost, Java is the recommended platform.

Most of UC Davis's core systems have a planned migration to Java. It is interesting to note, however, that few of those systems are based on Java at this time. This is due to historical reasons and is likely to last for a long time, as the high cost of conversion often dominates other considerations. PPS (the Payroll Personnel System) is operated outside of UC Davis, so it is in many ways more similar to shared applications than the other core applications.

Applications that Are Shared among Multiple Colleges and Departments

When the availability of infrastructure and expertise is less of an issue, other platforms are used for shared applications. Java, .NET, PHP, and ColdFusion are recommended for these applications, with ColdFusion and PHP more appropriate for less complex applications. Infrastructure and expertise for all of these platforms is available within UC Davis for these platforms, although that availability may be subject to local priorities.

From an IT perspective, sharing is the preferred method to avoid duplication of non-core applications across UC Davis. However, care should be taken to consider the commitment of resources made when offering to share the operation of an application. There should be clear statements of what the service is, including its expected lifetime and the financial support model.

Applications that Are Reused among Multiple Colleges and Departments

When it is planned or likely that an application will developed by one group will be passed to a different group, there should be consideration of the recipient group's ability to operate the platform (or to have someone operate it for them). For example, if the intent is to expand locally-developed applications for campus-wide deployment supported by IET, the recommendation is Java. In general, locally developed or acquired applications that may be reused at the campus, division, or college levels should verify that the platform can be supported at that level prior to development or acquisition.

Applications that Are Local to a Single College or Department

Applications that are not shared or reused may use any platform, although use of UCD-supported platforms is still recommended. Also, the chance that it will be desirable to share or reuse an application at some time in the future should be considered.

Additional Considerations

A Comment on Oracle Forms

Historically, Oracle Forms has been used for many applications at UC Davis. The platform, however, is no longer mainstream and is not recommended for new development unless that development is an extension to existing Oracle Forms applications, and the work is being done by a group that already has the requisite infrastructure and expertise.

A Comment on Application Complexity

Many of the recommendations mention that some platforms are more appropriate for either simpler or more complex applications. This is not to say that any platform cannot be used for applications of any complexity. For example, a department that has invested in Java may very well use Java for simple applications as well as complex applications, because they already have the expertise and infrastructure. Conversely, using an inappropriate platform for a complex application may negatively impact the long-term management and maintenance of such applications.

Future Work

The SSWG plans to undertake the following activities to further our goals of interoperability, sharing, and reuse of software:

- Non-technical advice for colleges and departments that are developing applications they would like to offer for sharing or reuse. This would include roles and responsibilities, cost sharing models, service expectations, end-user support, *etc.*
- Interoperability standards, particularly for interactions with core applications and middleware.
- Definitive descriptions of the supported platforms, including configuration, developer support and documentation, *etc.*