



KUALI RICE

Built With Service Oriented Architecture Concepts In Mind

The Kuali Rice software provides an enterprise-class middleware suite of integrated products built with services oriented architecture in mind. All Kuali applications are built using this common infrastructure, which provides a rich set of development tools and shared services. Rice allows developers to react to end-user business requirements in an efficient and productive manner, so that they can produce high quality business applications.

rSmart Kuali Rice includes the following modules:

Kuali Nervous System	Quickly build web-based business applications in an efficient and agile fashion
Kuali Service Bus	Service bus geared towards easy service integration in an SOA architecture
Kuali Workflow	Provides common routing and approval engine that facilitates the automation of electronic processes across the enterprise
Kuali Notification	Acts as a broker for university business related communications by allowing end-users and other systems to push information messages to the campus community
Kuali Identity Management	Provides central identity and access management services

WHY rSMART KUALI RICE?

- Promote rapid development that isolates many of the complications inherent to standard J2EE development.
- Focus on solving business problems rather than wrestling with the technology used to enable those solutions.
- Provide a consistent model for development which allows for the breakdown of monolithic applications into smaller application modules and services that can easily talk to one another.



KUALI RICE

Built With Service Oriented Architecture Concepts In Mind

THE PATH TO KUALI RICE STARTS WITH rSMART

Our experience with other Rice projects has allowed us to develop the Kuali Rice Proof of Concept (POC), a methodology that effectively prepares institutions to use Kuali Rice. The rSmart Rice POC provides consulting and developer training as well as ongoing developer support and coaching throughout the engagement so that you can successfully develop applications and streamline business processes using Kuali Rice.

The Kuali Rice Proof of Concept includes the following four phases:

PHASE 1: ON-SITE KUALI RICE CONSULTING AND HANDS-ON TRAINING

(Kuali Rice training – 5 days, on-site)

- Provide hands-on training and set-up of Kuali Rice developer environment
- Provide developer training on:
 - Kuali Enterprise Workflow
 - Kuali Nervous System
 - Kuali Enterprise Service Bus
 - Kuali Identity Management
- Include and provide exercises on:
 - Data dictionary
 - Business objects
 - Simple maintenance documents
 - Business rules
 - Formatter
 - Validation and error handling
 - Exception handling
 - Transactional documents
 - Java searchable attributes
 - Authentication/authorization (KIM)



KUALI RICE

Built With Service Oriented Architecture Concepts In Mind

- Java workflow (KEW nodes, roles, permissions, groups, etc)
- Extended attributes
- Evaluate additional training needs prior to Phase 2

PHASE 2: ON-SITE WORKFLOW PROTOTYPING WORKSHOP, PART 1

(Workflow Prototyping Workshop – 5 days, on-site)

- Workflow Prototype Workshop – design strategy, set up, development
 - Review requirements for applications (completed prior to visit)
 - Design strategy/coaching
 - Plan application design project
 - Creating technical specifications
 - Architecting the system
 - Setting up version control
 - Setting up Rice project
 - Pair programming to develop initial parts of code
 - Hands-on development
- Evaluate additional training needs

PHASE 3: ON-SITE WORKFLOW PROTOTYPING WORKSHOP, PART 2

(Workflow Prototyping Workshop – 5 days, on-site)

- Workflow Prototyping Workshop – development, review, refactoring
 - Continuing hands-on development
 - Code reviews
 - Merge process for upgrading Rice code
 - Refactoring
 - Enhancements
 - Documentation
- Evaluate additional training needs



KUALI RICE

Built With Service Oriented Architecture Concepts In Mind

PHASE 4: OFF-SITE DEVELOPER SUPPORT, COACHING AND CODE REVIEW

(Developer support, code review and recommendations – 6 months, off-site)

- Ongoing support of developers as they develop applications
 - Planned weekly/bi-weekly review calls
 - Code review
 - Ad hoc developer support
 - Issue resolution
- Coaching
- Workflow analysis and recommendation (end of engagement)
 - Joint review of success criteria
 - Deployment recommendations

Learn more at: www.rsmart.com/kuali