Subject: Development Diary (markcarranza)
Posted by markcarranza on Tue, 18 Dec 2012 10:08:53 GMT
View Forum Message <> Reply to Message

Hi, I'm a framework, RAD guy.

UI2 with DBaseIII in 85, Clipper in 87.
TRO (Tom Rettig's Office) in FoxPro in the 90's
MM.NET in .Net C#
Ruby on Rails
I've written a few of my own, including a lightweight PHP framework.

I'm likely to get deep into Radicore for a few months, and to give to and get from the Radicore community, I'd like to record the overall list of suggestions, changes, tips, tricks, pitfalls, successes and get appropriate feedback.

I'm redoing an entire legacy custom MIS intranet/extranet system. I'm recommending Radicore to my client in a way they understand:
[pluses]
+ since 2006, 4 version releases in 2012, well supported
+ highly opinionated, lots of writing, more and better documentation than any other (I know of)
+ free, open source
+ good working code
+ workflow, auditing, data dictionary
+ 3-tier architecture: data access, business logic, presentation logic
[minuses]
- user interface not good for client services, more for programmer types
- likely need to edit core files, Radicore upgrades would need programmer interaction
- special security needs requires audit of source and upgrade code
- possible intimidating learning curve for future maintenance PHP developers
- single person company/support
- no unit tests, test suite, automated tests
- no needed HTML reporting, only PDF (CTO: user network user temp area traffic/space issue)

The special situations of this client are:
- (very) low computer-skill employees
- high security
I appreciate all the work that has going into the UI standards of Radicore:
- classes of controls always in the same places
- handling of multiple records, and Parent records in hierarchy
- paging
But the UI is tremendously too busy, not the simplest possible for the user community I have to support. I have to radically minimize the choices, degrees of freedom for the user. Make it perfectly clear, with the minimum of choices on screen at any one time.

For example, the Radicore UI has the idiom:
1) display all tasks for context
2) select record (or records) from list

3) choose task
Where I need to have an idiom:
1) select one record from list (click)
2) display screen of (most important to see, dashboard) record data
3) display tasks applicable to record (edit, delete, ...)
4) choose task

I'm not sure if this above explanation is clear enough, but I hope future application screen shots will help explain. It will take rethinking of UI side of transaction patterns, if I use them.

For employee-facing screens, I'd prefer using a rich-client JavaScript MVC framework using AJAX and JSON data from the Business Logic layer, but that will be decided later.

The first step is a data element inventory of the legacy system. I've created several Radicore Subsystems matching major legacy components and imported those legacy tables and columns into the Radicore Data Dictionary.

The Description field holds the new column name, existing data examples and problems and a '(?)' where there are discussions needed.

The Comments field holds the extended data element description and sample data, business rules, presentation rules, valid ranges, etc.

The Required field is used, but most other Data Dictionary fields are unused because this is a documentation task preceding database redesign. The 'legacy' Data Dictionary will never be used for an application. I may add fields (new_column_name, show_historical_change_alert) for new application metadata. I'll certainly create new reports and upload them here.

The next steps before the database redesign are:
- Report Inventory (legacy paper reports in binder)
- Forms Inventory (paper forms (not in software) in binder)
- Screen Inventory (legacy screen shots in binder)
These will be much scribbled on and probably written up in .doc files, notes printed up and included with the examples.

Suggestions, observations:
List Subsystem screen
- two buttons named 'search'
- screen/example search should be named 'Find' or 'Query'
- in column search, when column dropdown is blank, search fails and typed-in search term unexpectedly disappears. Blank column dropdown is useless and broken. Show no blank: first or best column in dropdown. Or change blank column dropdown search to query all text fields.
- if 10 or less records, hide paging top (show...) and bottom
- list column heading 'Select' could be link/toggle select all/select None. If all selected, then 'None' otherwise 'All'. JavaScript required.
- not sure what 'locked' check box does, not discoverable or documented in http://www.tonymarston.net/php-mysql/menuguide/mnu_subsystem(list1).html
- list column heading 'Count' clearer as 'Tasks'

Thanks! --Mark