

# Important Unfunded Mandate List

I d e n t i f i e r	Problem	Thoughts on Solution	Notes /comments
	Confusing build/deploy process that I don't believe anyone in this community fully understands.		
	Wildly outdated support libraries (thanks Noah for tackling this one).		
	Failure to support basic browser features such as tabs and the back button		
	Non-normalized, unsearchable data in core tools such as Content.		
	Lack of support for database failover except through driver indirection.	First step: undo our weird binding to a very old DBCP and allow a real DataSource to be plugged in -- whatever pool or config you need. Second step: pick the Tomcat pool by default and document how to switch if you need something else (hint: should be very standard and basically link to JDK/Tomcat docs). Third step: add some better connection/query/transaction/request metrics that work OOTB so implementors can better see what's going on and decide what to do about it.	
	Poor or non-existent logging of basic processes.		
	Users are bound to a single app server, making zero-downtime maintenance and high-availability very difficult to achieve.		
	An authz system that scales poorly and is near impossible to maintain.		
	An API tightly bound to the particulars of the portal /portlet presentation.		
	Outstanding security issues in both unmaintained and brand new core tools.		
	Runaway database activity in most hibernate-based tools.		
	Serious lack of anything beyond the most basic administrative tools.		
	No support for a departmental tech support or junior administrator role.	Put DAC into core! <a href="#">Delegated Access Tool</a>	
	Complete lack of inter-tool database relationships, sometimes even a lack of anything that could even be *used* as a pseudo-foreign key.		
	Lack of an asynchronous request system for long-running processes	First step: retrace the steps done to integrate JMS / ActiveMQ (MessageService?) and decide if that's the right way to go. Second step: implement something basic and very standard. Third step: move some nasty bits like site duplication to the work queue and set up some good docs about how to queue (standard), query for status (standard), and update the user/browser (specific to our portal/notification services).	
	Incomplete and inconsistent archive/restore support.		
	RSF (do we mean rewrite to eliminate use of RSF)?		

No clear separation between core functionality and toolset.		
No registration of allowed property values by tools.		
Two configuration systems munged together: properties and Spring.		
A site management tool that's confusing and overwhelming, both to end-users and to developers who might want to clean it up.		
A source code base that's full of redundant and inoperative code not to mention transition steps from half a dozen versions ago that still run at every startup.		
A widely used display tech (JSF 1) that is both unmaintained and known to kill app servers		Do we mean eliminate the use of JSF or switch to a newer version of JSF?
Find a 90% Solution to making browser back button working in most tools		