Technical Development

Analysis

Learn System Architecture - Become familiar with the top level aspects of the VIVO system. See VIVO Documentation and select the documentation for the version you are using.

- You can estimate how much computer power you will need by considering standard and preferred configurations for the software you are using.
- What about additional software? Most VIVO sites use only open-source software.
- Some sites modify the VIVO software stack, either because they have special needs, or simply because they have expertise with different tools. See configuration options for the version you are using.

Identify Customizations - Every site makes changes to VIVO. Some are as simple as changing the styling and the logo. Others add data types and properties to the ontology. Some add new pages to the application, or new functionality to existing pages. VIVO is made to be customized, but some changes are easier to accomplish than others, and you will need to take that into account. Many stylistic changes are easily done, as described in the configuration documentation for the version you are using.

Establish Data Feeds - For most VIVO sites, the biggest challenge of a VIVO system is not VIVO itself. Instead, the challenge comes in populating VIVO with data from your institutional systems. Many pages in the VIVO wiki are concerned with data issues. Start with VIVO Data - what and from where, and the pages linked to it. Data challenges are often particular to the individual institution, ranging from the practical (Data source specifications for implementation) to the political (Policy and planning questions for VIVO data).

Develop Prototypes - You may need to develop your own programs and scripts for ingesting data, or you may be able to configure a more general tool to your own needs. In the VIVO community, the two favorite tools are the VIVO Harvester and Karma. You can view some tutorials on YouTube to learn more about using Karma with VIVO (note that some videos are for use with the VIVO 1.5 version and some for later versions).

Implementation

Build Customized System - Add your customizations to VIVO. If you are only making small changes, mostly to appearance, you might add your changes directly to the VIVO distribution files. For larger customizations, you should review the methods recommended for preserving customizations in the version you are using. This helps to organize your installation, and will make it easier to implement system upgrades when the time comes.

Test Performance - Before announcing your VIVO system to the public, you should test to be sure that it performs acceptably. If not, you might consider Use HTTP caching to improve performance, or the methods in Troubleshooting VIVO's Performance.

Launch

Provide System Support - As with any IT project, you should expect VIVO to require ongoing support. This includes things like backups, security checks, upgrades to operating systems, etc.

Implement System Upgrades - The VIVO team issues a new release about every year, with occasional extra releases to fix problems. For the most part, upgrading is straightforward. If there is a change in the VIVO ontology, the new release will include an automatic script to translate your existing data. Again, it is not unusual to find that the largest task is making changes in your data ingest processes. Most VIVO sites try to adopt new releases as soon as possible, to take advantage of new features or improved performance.

Maintenance

Develop New Features - You should also plan to develop new customizations for your VIVO installation. As your users gain experience, they will likely request new data and new displays. In the best case, these changes may be something that you can contribute to the VIVO community, so others may benefit from your work.