Technical Development; Unit Testing

The requirements should be in the ACE task/ATHR request. If they are not there, the developer will work with the Functional Lead to get a clear picture of all the requirements, documenting them in the task/request.

All development projects are associated with a unique project number in the format HR_NNN_NNA. The middle 3 numbers are associated with a particular functional area of the application, the last 2 numbers are sequential within that area. The alpha character at the end is used if there are numerous changes to that customization.

For example, HR_100_01 would designate a customization within Administer Workforce. Once that is in production, if a further change is required for that customization, project HR_100_01A would be used, the next would be HR_100_01B, and so on. If we should reach HR_100_01Z, the next project number used would be HR_100_AA, then HR_100_AB, and so on.

The exception to the naming standard would be the case where we are implementing a temporary fix while awaiting a PeopleSoft patch. In this scenario, the project number would be an HR_FIX_NN where NN is a sequential number from 01-99.

The Project Lead/Manager or the Production Support Backup will supply the project number for a development task. It will be put in the heading of the ACE task/ATHR request.

UW Request

a) New Customization

If the request is for a new customization, the developer will create a Customization Requirement and a Customization Design document. Developers, please refer to Guidelines for Completing Customization Specifications for further information.

b) Modification to Existing Customization

If the request is for a change to an existing customization, a modification to the Requirements document will be required for that customization. For example, if the request is a change to HR_100_01 customization, then:

1. the Requirements document for HR_100_01 will be modified to incorporate the new requirements
2. the Requirements document for HR_100_01 will be updated in any relevant areas to do with this new change (navigation, security, related project numbers)
3. a link and description of the new project number will be added under "Links to Project Designs"

4. the Design document will be created. Please refer to Guidelines for Completing Customization Specifications for further information.

These documents will be posted to the Specifications Index where functional team members can refer to them for specification sign off before development begins. The specification sign off will be done with a status update in ACE (Please refer to the ACE process flow of a typical development project for more information on how to use ACE through the evolution of a task).

At this point, development should begin. Please refer to Overview of our Development Environment for more details on our typical development environments.

The University has a number of standards that must be followed for all development. This allows us to easily identify our customizations, separate from what was delivered by PeopleSoft. Please refer to PeopleTools Standards & Procedures for a standard associated with different PeopleTools objects, using secure shell, password controls, use of Change Management for development project migration, naming standards, etc.

Development for all PeopleTools objects will be in the DV instance, where developers have full tools and sysadm access.

There are times development may include objects that are external to PeopleTools (SQR's, SQL scripts, shell scripts, etc). Please refer to Unix Development Environment for further details.

There may be points through the development and test cycle where clarification is required on a requirement, and the document(s) may have to be updated.

Once a developer has completed development for a customization in the development environment (DV), that developer is responsible for unit testing the customization and ensuring that it meets all specified requirements. Please see Developer Testing Guidelines for more information.

After development testing is complete, the change should be migrated to the test environment (TS instance), update the status to Functional Testing & group to Functional in ACE, and the functional user can proceed with testing. Please refer to Migrating Development Work for more information.

PeopleSoft Update

Development for PeopleSoft updates is a little different than a UW customization. Please refer to Update Procedures for more information.

i) Tax Update

We do not execute a freeze when applying tax updates. Development, testing and project migration continues in our regular environment, at the same time as it is happening for our tax update environment. For this reason, the tax update development and testing must reside on a separate technical software silo than our regular development/test instances. In other words, we don't want the tax update changes affecting our regular changes that are being tested.

Our QA and TX instances reside on a separate software silo from our other development and test instances.

The Project Lead/Manager will work with Functional Leads on an appropriate date for applying the tax update for testing and to production. Once the dates are decided, the Project Lead/Manager will submit a request to the App Admins in Request Tracker. The request typically:

- requests a refresh of QA instance so that it is up to date with production for a data compare
- requests application of the tax update
- requests date that it is to be applied to production (production down time typically scheduled for half day)
- includes any correspondence of problems encountered once testing begins
- includes development project number that must be migrated with project, if one exists
- any special instructions that may be required for migration to production

The Project Lead/Mananger will also set up a task in ACE for the tax update. This task may or may not include a development project number. All correspondence for the tax update between Functional Leads and the developer assigned to the tax update should take place within ACE. Please refer to the ACE process flow of a typical development project for more information on how to use ACE through the evolution of a task.

App Admins apply the tax update to demo and QA. The QA instance is then backwashed into TX.

A tax update includes a large number of changed objects redelivered by PeopleSoft. These objects are either tools related(internal) for sqr's/cobol(external). This means that we have to compare these objects with any we have customized, and reapply our customizations, if they are still applicable. Developers have to manually compare the SQRs that have been redelivered with the tax update, and determine which ones we have customized, then reapply the customizations. The compare is typcially done by doing a file compare in UltraEdit.

To determine PeopleTools objects that we have customized and are changed by the tax update, the App Admins will run compare reports and give the developers the results.

To reapply our customizations to objects that are changed in the tax update, development of PeopleTools objects are done in the TX instance. Development for SQRs will be done in package hrms1d on boss in the sqr002 directory (the developer can work on the changes in sqr001 and then copy them to sqr002). Development unit testing can be done in the TX instance, which looks at sqr002 in hrms1d.

Once development is complete, the developer must set the development project up in Change Management, under the x.x Tax Update application (where x.x is the current application version). This Tax update application in Change Management points to the TX instance for project object references.

Once the project is set up in Change Management, the developer will ask the App Admin to migrate the changes to QA.

All functional testing for the tax update is done in QA. Please refer to Testing Procedures on how to document tax update testing.

ii) Patches/Fixes

A PeopleSoft patch/fix would follow the same type of procedure as above. It would be applied to demo and QA. QA would then be backwashed to TX.

All functional testing for the patch/fix would happen in QA.

iii) Bundles/Maintenance Packs

We would execute a development freeze while testing a bundle or maintenance pack. This is necessary to reduce the risk of:

1. overwriting a re-delivered PeopleSoft object in the bundle/mtce pack while migrating a customized object outside of the scope of the bundle testing
2. a customization outside of the scope of bundle/mtce pack testing that could be affected by a change in the bundle/mtce pack.

Other than the freeze on any further development, the application, development and testing procedures would be pretty much the same for a bundle as they would be for the tax update. In fact, most bundles will be applied concurrently with a tax update to take advantage of the testing that would have to occur for a tax update. Please refer to the above Tax Update section for further information on procedures.

Maintenance packs would be more extensive and would be more like a mini upgrade. New development and test instances would likely be created to develop and test for the maintenance packs, with a separate silo from the existing ones.

Any development that absolutely must proceed during bundle/maintenance pack testing (and that can be migrated to production with or after the bundles) should be done in TX where the bundles reside. Special arrangements will have to be made with the App Admins to migrate the development project to QA for functional testing, where the bundles/maintenance packs reside. It can be signed off and migrated to production with the bundles/maintenance packs. If it is going to be migrated at a later date, it will have to be exported from TX and imported into DV after DV is refreshed from production with the bundles/maintenance packs.

If the production support development will be migrated to production before the bundles/maintenance packs, it will be done through the normal route (DV, TS). After it is in production, and before the bundles/maintenance packs are signed off, it will also have to be tested in QA to ensure that the bundles/maintenance packs will not affect it.

Refreshes of all instances MUST be done after a bundle or maintenance pack is applied to production. If there is an instance that is at a critical state where a refresh is not possible, the bundle or maintenance pack must be manually applied to that instance.