1. Project

Upgrade and/or re-design infrastructure and methodology for performing acceptance and "regression" testing for upcoming and future releases of SEASABS and TSUpdate. More specifically, develop a tool to test for backwards compatibility of existing generic functionality and performance between "test" and "production" versions.

Task doclink in TSA projects database:
 (Subject: Project information; Database: Time Series Analysis Projects)

Specific goals are:

(i). assemble input-test database(s) containing series cases designed to test/check for compatibility in existing generic functionality in SEASABS/TSUpdate test and production versions.

(ii). assemble output-test database(s) for storage and propagation of test results.

(iii). implement software (preferably via a flexible perl script) that compares outputs from test and production versions of SEASABS/TSUpdate and that generates archivable diagnostics and test summaries.

1.1 Project Overview

A broad overview of the main testpack implementation steps are a follows. This is a significantly simplified design over previous methods (but still retaining much of the functionality). The goal would be to further automate as many of these steps as possible. Note that the following are not necessarily in any logical order. Steps (c) and (d) can be combined in a single framework since SEASABS and TSUpdate functions are coupled in the production environment.

(a). identify (benchmark) series and groups to test for existing generic functionality in SEASABS/TSUpdate. There should be four broad categories (or groups) of test series according to periodicity and adjustment method: monthly/concurrent, monthly/forward-factor, quarterly/concurrent and quarterly/forward-factor. Every effort will be made to include at least some or all of the following properties across these groups: large extremes, additive outliers, trend breaks, seasonal breaks, Easter, Father's day and trading day effects, seasonality and non periodic seasonality, indirect vs. direct adjustment and multiplicative vs. additive models.

(b). load test series/groups into appropriate "ABS testing database" on ABSDB.

(c). acceptance testing for SEASABS:
- update/create series knowledge with new series from ABSDB using production version.
- generate and store series products from production version.

- download production version products in ascii/text format: originals, seasonally adjusted, trends, seasonal/combined factors, pure seasonal factors, prior blocks, trading day factors using "Download" tool [production version].
- generate and store series products from same series from test version.
- download test version products in ascii/text format from ABSDB: originals, seasonally adjusted, trends, seasonal/combined factors, pure seasonal factors, prior blocks, trading day factors using "Download" tool [test version if available].
- compare production and test version products using perl script, generate summary of differences and store.
- analyse differences between production and test version and work with team to decide between "expected" (ie. service-request related) and "unexpected" differences.

(d). acceptance testing for TSUpdate:
- assumes seasonal and combined factors for concurrent and forward-factor series got stored in ABSDB from a previous run of SEASABS production version.
- execute TSUpdate production version to create and/or update products in ABSDB for concurrent/forward-factor series.
- download production version products for concurrent/forward-factor series in ascii/text format from ABSDB: originals, seasonally adjusted, trends, seasonal/combined factors, pure seasonal factors, prior blocks, trading day factors.
- repeat previous three steps using test version of SEASABS/TSUpdate.
- compare production and test version products from TSUpdate using perl script, generate summary of differences and store. This is done separately for concurrent and forward-factor series.
- analyse differences between production and test version and work with team to decide between "expected" (ie. service-request related) and "unexpected" differences.

A more formal representation of the testing process is outlined in: (Subject: SEASABS/TSUpdate acceptance test procedure..)

1.2 Background

An evolution of previous work and methods is contained in the following:

General: base data set-up/inputs:
(Subject: ALL SEASABS manager needs to know; Database: Time Series Analysis WDB; Author: Lujuan Chen; Created: 09/12/2003)
(Subject: Overview of SEASABS Suite Release; Database: Time Series Analysis WDB; Author: Lujuan Chen; Created: 09/12/2003)
(Subject: 2.2 TSUpdate Test Guidelines - Testing the existing functionality; Database: Time Series Analysis WDB; Author: Lujuan Chen; Created: 09/12/2003)
(Subject: 2.3 TSUpdate Test Guidelines - Testing the new functionality; Database: Time Series Analysis WDB; Author: Lujuan Chen; Created: 09/12/2003)
(Subject: SEASABS Test Guidelines; Database: Time Series Analysis WDB; Author: Lujuan Chen; Created: 09/12/2003)

Testing of TSupdate/SEASABS using perl tool:
(Subject: Perl for Acceptance test; Database: Time Series Analysis WDB; Author: Mark Zhang; Created: 09/12/2003)

(Subject: zipped package; Database: Time Series Analysis WDB; Author: Chris Wei; Created: 09/12/2003)
(Subject: Acceptance Test (improved version with Perl); Database: Time Series Analysis WDB; Author: Chris Wei; Created: 09/12/2003)

2. Description of Project

2.1 Goals

The main goals were outlined in section 1.

2.2 Outputs and Deliverables

- new input/output database tables containing benchmark test series.
- "testpack" checking tool that reports inconsistencies of outputs between test and production versions of SEASABS and TSUpdate.
- a report giving an assessment of "expected" upgrades from new service requests versus "unexpected" (unexplainable) differences between test and production versions.

2.3 Risk assessment

Risks:

- proposed design not capturing all differences in functionality between different SEASABS/TSUpdate versions, giving misleading results (unintented revisions) for ABS clients.
- acceptance test results between software releases not comprehensively examined by members in TSA to decide if differences are service request related and expected.
- unexplainable bug-related differences not fixed or appropriately found by TSD.

Any remaining unexplainable differences on repeated iterations of these steps will render the new SEASABS and/or TSUpdate versions unreleasable.

2.4 Doclinks

For previous work, see section 1.2

3. Stakeholders

Time Series Analysis Section, clients, TSD.

4. Timing and Resources

Task	Time (planned)	Start Date	End Date	Status	Doclink (if relevant)	Time (actual)
Goals and overall design	3 hours					

Learn ABSDB data management with Phil Carruthers and take online course.	4 hours					
Assemble base testpack series with all possible known priors, load into appropriate ABSDB.	5 days/ongoing					
Design flowchart to illustrate acceptance testing methodology.	1 day				(Subject: SEASABS/TSUpdate acceptance test procedure..)	
Perl script to compare production and test version products.	2 days					
Start of actual regression testing of version 2.6 of SEASABS/TSUpdate	2 weeks (ongoing)	3/10/2006	~20/10/2006	waiting on new test release		

5. Project Journal

This section is for any more detailed comments on the project's progress.

Dates	Event/Task	Time spent	Doclink
3/3/2006	Meeting between Mark, Craig and Anna, Frank to discuss goals.	1 hour	
14/3/2006	Frank met with Craig McLaren to clarify design in section 1.1 above.	20 min.	
20/3/2006	Met with Phil Carruthers to outline dataset, TS Link and Map creation in ABSDB	45 min.	
28/3/2006	Met with Phil Carruthers on how to make existing dataset copies and presevation of original filenames.	25 min.	

6. Lessons Learnt

- None yet.