The University of North Florida seeks to purchase comprehensive web-based strategic planning and assessment management software that will streamline and organize the collection and management of institutional, college, departmental, program, course, and student level planning and outcomes data for purposes of strategic planning, outcomes assessment, accreditation, program review, annual reporting, and budgeting.

Evaluation Criteria

- Vendor’s solution “fit” to the University’s General Requirements
- Degree to which software integrates/interfaces with Sungard’s Banner ERP
- Ease of use
- Training and support provided
- Local control (independent of hosting location)
- Total cost for software solution

The assessment management software should meet the following requirements:

6.1 Content capabilities:
1. Strategic planning: Mission, goals, actions/strategies, outcomes [results], evaluation [reflection], continuous improvement plan, and ASWOT (achievements, strengths, weaknesses, opportunities, and threats) should be linked horizontally and vertically within and among units and the institution.
   a. Longitudinal tracking: The software should accommodate longitudinal presentation and tracking, regardless of the tracking period. Tracking periods could be static, or vary according to some defined cycle (e.g., every year, every 5 years, etc.).
   b. Budgeting: The software should be able to support various elements of the University’s budgeting process. Specifically, the University wishes to maintain, track, and link (where applicable) the following items for each element and sub-element (e.g., goals, actions) of the strategic plan:
      i. Initial cost(s)
      ii. Continuing cost(s)
      iii. Additional FTE required
      iv. Continuing costs associated with additional FTE
      There should be an ability to assign priorities to items i. – iv. listed above at the department, college, division, and university levels. As a reporting element, there must be a capability to extract and “roll up” budget items at all levels.
2. Program review: For purposes of periodic program review, the software should provide the ability to define programs in various ways according to the needs of individual units, such as by major, concentration, degree, college, department, center, or activity.
3. Assessment of learning outcomes: The software should accommodate the entry and linkage of outcomes, curriculum maps, measures, results, evaluations, and continuous improvement, and provide the ability to create and attach rubrics and other supporting artifacts.
   a. Multiple levels: The software should accommodate assessment of learning outcomes at various levels, e.g., course, major, program, department, college, division, and university.
   b. Longitudinal tracking: The software should accommodate longitudinal presentation and tracking, regardless of the tracking period. Tracking periods could be static, or vary according to some defined cycle (e.g., every year, every 5 years, etc.).
   c. Syllabus repository: The software should provide a syllabus repository, with linkage of course outcomes to department level outcomes.
4. **Breadth**: The software should accommodate outcomes assessment and evaluation of programs and activities that cut across academic and nonacademic units, e.g., interdisciplinary programs, general education, and community-based transformational learning (QEP).

5. **External accreditors**: Regional (SACS) and a variety of professional accreditation standards should be preloaded.
   a. **Edit existing standards**: The software should provide the ability to add or modify standards in preloaded programs.
   b. **Add new standards**: The software should provide the ability to add new sets of program standards that are not preloaded (e.g. created locally or provided by additional external accreditors).
   c. **Standards updates**: When regional and professional accreditation standards change, standards should be promptly updated by the vendor, or should be modifiable by the University.

6. **Student work**:
   a. **Portfolios**: The software should provide a student portfolio tool that allows a student to export their portfolio to another medium, or to permit external access to specific individuals.
   b. **Repository**: Repositories of student work should be accommodated separately in both portfolio and course tools.
   c. **Faculty review**: Faculty should have the capability to provide feedback and evaluation of student work using rubrics.
   d. **Rubrics**: The software should provide a rubric creation tool and should also allow the loading of externally created rubrics. Data from rubric ratings should be rolled into higher level reporting of student learning outcomes. Individual rubric traits should be linkable to specific goals, outcomes, or standards.
   e. **Approval queues**: Faculty should be provided with easy-to-manage approval queues that prompt them to review relevant student work.
   f. **Subscription pricing**: The option should exist to subscribe some but not all students to the portfolio option, with price reflecting the actual number of portfolio subscribers.

7. **Tests and surveys**: The software should support the creation and administration of tests and surveys, with automatic collection and reporting of results.

8. **Faculty annual reports**: The software should contain a faculty annual reporting tool.

6.2 **Reporting features**:

1. **Ease of extraction**: The software should provide the ability to easily extract information and data for both regular reporting and ad-hoc requests.
2. **End-user queries**: The software should accommodate access via end-user query tools, for example, Crystal Reports.
3. **Web display**: The software should provide the ability to easily extract information for public website display (e.g. Academic Learning Compacts, see links from [http://www.unf.edu/acadaffairs/IE/acl/Alpha_index_new.htm](http://www.unf.edu/acadaffairs/IE/acl/Alpha_index_new.htm)).
4. **Historical reporting**: The software should allow us to compare data extracted from strategic plan, program review, outcomes assessment, and other elements at any two or more points in time, regardless of reporting cycles. The length of time for which data will be stored should be at the discretion of the University.
5. **Longitudinal views**: The software should provide the ability to perform longitudinal comparisons for reporting and analysis purposes.
6. **Reporting cycles**: The software should provide the ability to define and manage differing reporting cycles, for example, strategic plans on a five-year cycle, program review on a seven-year cycle.

6.3 **Usability**:

1. **Ease of use**: The software should be easy to use by diverse user groups.
   a. **Email prompts**: Occasional users should receive email prompts with a direct link into appropriate data entry fields.
   b. **Level linkage**: Goals and learning outcomes should be easily linked to a higher or lower level in the hierarchy.
c. **Navigation:** Navigation should be intuitive and require a minimum number of clicks or keystrokes.

d. **Reports:** Reports should be easy to assemble and read.
e. **Document uploads:** Users should be able to upload or link to a variety of objects, such as documents, video, and audio.

2. **Keyword search:** The software should provide keyword search functionality.

3. **Workflow management:** Software should enable feedback loops and queues for tasks and approvals, with email prompts.

4. **Feedback and quality control:** At all levels, administrators should be able to approve and provide feedback on entries.

5. **Help:** The software should provide a context-sensitive help function.

6. **Other users:** The software should be demonstrated (through documented references with specific contacts provided) to be effectively functioning in other SACS institutions and preferably in other State University System institutions in Florida.

7. **Support:** The vendor should provide documentation (with references and with specifics) of excellent customer service, training, and support.

6.4 **Functional System Administration:**

1. **Banner interface:** The software must integrate or interface cleanly with Banner. The vendor must demonstrate how this integration is done at other institutions.

2. **Blackboard interface:** The software must integrate or interface cleanly with Blackboard’s Course Management System. The vendor must demonstrate how this integration is done at other institutions.

3. **Administrator interface:** The software should provide a dedicated administrator interface.

4. **Delegation of authority:** System administrator should be able to delegate selected authorities (permissions) to other unit administrators, who in turn may delegate their own authorities downward. Software should provide the capability to add, delete, and move users into roles (including multiple roles for a given user), and should accommodate the configuring of hierarchal approval queues.

5. **User access cloning:** The software should provide the ability to clone user access.

6. **Single sign-on:** The software should provide Active Directory linkage, enabling single sign-on capability.

7. **Update logging:** The software should log entries with a date, time, and user stamp.

8. **Security:** The software should accommodate hierarchical role-based application security with the ability to apply data-level security. For example, a dean can manipulate any data for his/her college and a department chair can only manipulate data related his/her department. Within forms, different fields should be assigned different levels of security.

9. **Department and Program changes:** The system administrator should be able to rename/split units and alter academic programs without compromising data integrity. Examples include:
   a. Programs moving to another college/department;
   b. Departmental splits;
   c. Changing a concentration within a major to a stand-alone major.

10. **Lock down:** System administrator should be able to lock down data and plans (e.g. strategic plans, Academic Learning Compacts) in order to prevent individuals from modifying them after a point in time. This feature should allow for an administrative override from the system administrator when an exception to the rule is approved.

11. **Database access:** The software should provide general inquiry and update capabilities against the underlying database management system using (preferably) Oracle or SQL Server. The software should provide the capability to easily import and export data.