

ACID

MTM Program Product Software Requirements Specification

Version 1.0
April 10, 2020

Standard Version Number: 3.5
Standard Version Date: March 10, 2018

Version History

<i>Version</i>	<i>Date</i>	<i>Authors</i>	<i>Comment</i>
0.01	2/4/2020	Celia	Celia's edits from 1 st client meeting (Jan. 24 th)
0.02	2/6/2020	Combined version	Class edits from 1 st client meeting (Jan. 24 th)
0.03	2/25/2020	Combined version	Class edits from 2 nd client meeting (Feb. 7 th)
0.04	3/12/2020	Combined version	Class edits from 3 rd client meeting (Feb. 21 st)
0.05	4/10/2020	Combined version	Class edits from 4 th client meeting (March 6 th)
1.0	4/21/2020	Combined version	Class edits from 5 th client meeting (April 3) and the final presentation (April 17)

Template Version History

<i>Version</i>	<i>Date</i>	<i>Authors</i>	<i>Comment</i>
3.0	7/21/2012	Frank Ackerman	Initiating standards versions
3.1	8/2/2012	Frank Ackerman	Some non-functional requirements definitions .Added adaptability, enhanceability, and portability
3.2	1/17/2013	Frank Ackerman	Added usability comment
3.3	3/6/2013	Frank Ackerman	Added a bit more explanatory text and final section 8.
3.5	3/10/2018	Celia Schahczenski	Changed format of dates, rearranged, renamed items, removed Illustrative Use Cases, and increased some explanations, added appendices including data and report sections.

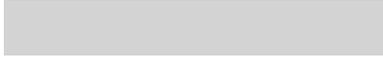
Montana Tech Software Engineering Students:

These Montana Tech Method software engineering standards encapsulate Dr. Ackerman's decades of experience in the software industry, the IEEE software engineering standards, and many suggestions from various texts. They have gone through many revisions and additions over the last several years. They are part of your software engineering studies so that (1) you may have the experience of developing software to a standard (which you may find you need to do if you take a job that requires high reliability software), and so that (2) you will have the experience of developing high quality software. You are also invited to participate in the continuing evolution of these standards by studying them critically and making suggestions for their improvement and correction.

TABLE CONTENTS

INTRODUCTION.....	5
1.1 SOFTWARE PURPOSE AND SCOPE	5
1.2 DOCUMENT PURPOSE AND CONTENTS	5
1.3 REFERENCES	6
2 GENERAL FACTORS.....	6
2.1 PRODUCT PERSPECTIVE	6
2.2 PRODUCT FUNCTIONS	6
2.3 ENVIRONMENTAL CONDITIONS.....	7
2.4 USER CHARACTERISTIC	8
2.5 DEPENDENCIES	8
2.6 ASSUMPTIONS.....	8
3 USE CASES.....	9
3.1 ROLES	9
3.2 USE CASES.....	10
3.2.1 Associate and describe metric	10
3.2.2 Create metric	12
3.2.3 Create student outcome.....	14
3.2.4 Score metric association.....	15
3.3 USE CASE PRIORITIES	17
4 SPECIFIC REQUIREMENTS.....	18
4.1 FUNCTIONAL REQUIREMENTS	18
4.1.1 Login.....	18
4.1.2 Create student outcome.....	18
4.1.3 Metric scoring.....	18
4.1.4 Importing grades.....	19
4.1.5 CI tracking	19
4.1.6 Update ABET mapping	19
4.2 QUALITY ATTRIBUTES	19
4.2.1 Privacy.....	19
4.2.2 Security	20
4.2.3 Usability.....	20
UR 6: Users will be informed of all changes made to the system.....	20
4.3 NON-FUNCTIONAL REQUIREMENTS WHICH ARE NOT QUALITY ATTRIBUTES ...	22
4.3.1 External Interface Requirements	22
5 FUTURE ENHANCEMENTS.....	22
APPENDICES.....	24
APPENDIX A: DEFINITIONS, ACRONYMS, AND ABBREVIATIONS	24
A1. DEFINITIONS.....	24
A2. ACRONYMS AND ABBREVIATIONS.....	25
APPENDIX B: ANALYSIS MODELS	26

APPENDIX C: DATA DICTIONARY 28
APPENDIX D: REPORT SPECIFICATION 32
APPENDIX E: BUSINESS RULES 33
APPENDIX F: ASSESSMENT DATA EXAMPLES 35
APPENDIX F: SAMPLE USER INTERFACE 36
APPENDIX G: ISSUES..... 38



Introduction

This section provides an overview of the ACID software. It includes the business objectives and vision of ACID along with the purpose and contents of this document. It ends with references that the reader may find useful.

1.1 Software Purpose and Scope

The purpose of the ACID software application is to help faculty in the School of Mines and Engineering perform continuous improvement of programs, particularly with ABET Criteria 3.

ACID's primary goals are to:

- Produce reports to facilitate continuous improvement of engineering programs and make it easy for faculty, accreditors and others to see the extent to which ABET Criteria 3 is being met.
- Save faculty time by allowing faculty and staff to easily and flexibly input, store, and retrieve assessment information.

ACID's vision is as follows:

“For faculty in the School of Mines and Engineering who need to assess student outcomes for ABET, ACID is a software tool that captures, tracks and compiles information related to student outcomes and reports it in a meaningful format for continuous improvement of programs. Unlike the AbOut system that does this but only for the Computer Science and Software Engineering programs, our product does it for everyone.”

Facilitating evaluation of ABET's other criteria is not currently supported by ACID. Nor does ACID directly facilitate evaluation for other accrediting agents such as Northwest Accreditation Commission, except to the extent that Northwest utilizes ABET's Criteria 3. Other areas could be supported in future enhancements of ACID.

1.2 Document Purpose and Contents

The purpose of writing this Software Requirements Specification (SRS) is to allow future developers of ACID to better understand the customer goals and desires. These developers will be able to use details of this document to implement features and behaviors into a final product. This document can also be used to design tests to ensure the application behaves as intended.

Engineering faculty at Montana Technological University, the eventual users of the software, may also find this document helpful to learn about ACID.

This document does not attempt to tell how ACID should be implemented except in those cases where the customers want the application to be developed in a particular way. Deciding exactly what a system should do, before deciding how it will do it, significantly reduces development time.

This SRS was developed by the software engineering (SE) students in the course Requirements and Specification (ESOF 328) during the spring semester of 2020.

1.3 References

ABET, Accreditation Board of Engineering and Technology, <http://www.abet.org>

ABET assessment planning tools, <https://www.abet.org/events-and-workshops/assessment-planning-resources/assessment-reading-list/>

Software requirements specification of AbOut, an earlier assessment tool developed by students of the Computer Science Department at Montana Tech, https://katie.mtech.edu/classes/esof328/Resources/AbOut_SRS_v3.6.pdf

AbOut, the assessment tool developed earlier by students of the Computer Science Department at Montana Tech, <https://katie.mtech.edu/AbOut/login/LoginController.php>

2 General Factors

This section provides a high-level view of the ACID system, its major functions, environment, users and dependencies.

2.1 Product Perspective

This product could be similar to AbOut, software created by and used in the Computer Science (CS) Department at Montana Tech. However, other engineering departments at Montana Tech map performance indicators to student outcomes, while the CS Department maps course outcomes to student outcomes. The intent for this product is to make ACID flexible enough to handle both methods.

2.2 Product Functions

The following lists the functions and features ACID may include. The details of many of these product functions will need to be

1. Report generator: One of the core purposes of this product is to generate reports. ACID shall be able to produce reports to help with continuous improvement and measuring the extent which ABET Criteria 3 is being met.
2. Personalized front-end/dashboard: ACID should have a personalized front-end/dashboard that lists programs and courses for a given semester so the user is only shown information (programs, courses, etc.) that is relevant to them. What is relevant can be determined from the login credentials.

3. Tracking improvements due to remediation: One of the core principles of the product vision statement is continuous improvement. To facilitate this, the system will be able to track improvements and record what efforts have been made in the remediation of the weaker aspects of courses.
4. Simple and Intuitive data input: this software will be designed such that using it for continuous improvement will save faculty time, compared with not using it for continuous improvement.
5. Map old ABET criteria to new criteria: ACID shall be able to store and utilize the mappings ABET provides between old and new criteria. ABET provides these mappings when they update their outcomes
6. Historical reports: ACID shall be able to generate reports from previous semesters. When information is changed, those changes must not permeate back to previous reports.
7. Audit trails: ACID shall create audit trails of modifications to the system data, detailing who made the changes and exactly what was changed.
8. Backup Data: ACID shall provide a mechanism in which large backups of the database can be created.

2.3 Environmental Conditions

ACID will interface with two existing applications in use by the Montana Tech's engineering faculty, Moodle and the Montana Tech Central Authentication Service (CAS). Interfacing with Moodle for assignment grades will make grade entry more convenient for faculty. The system will also work with CAS so faculty members can use their usual Montana Tech credentials to use ACID. A context diagram showing how ACID interacts with its environment is given in Figure 1. In this figure faculty can input metric information and have grades inputted from Moodle. ACID interfaces with CAS to authenticate the users of the system. The diagram suggests that ACID may interface with an unspecified system, via its Application Programming Interface (API) in the future.

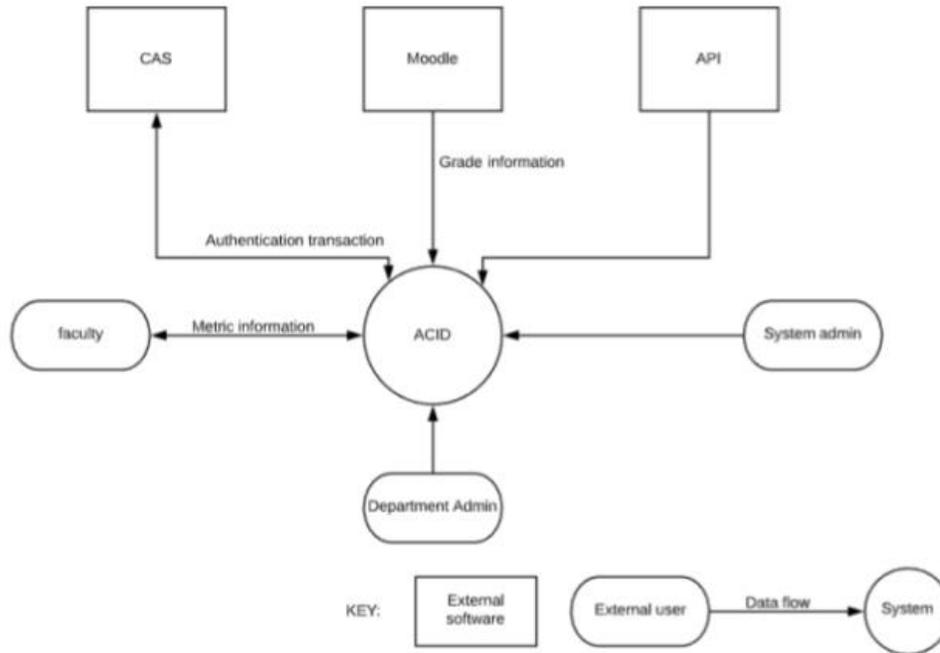


Figure 1. Context Diagram

2.4 User Characteristic

The primary users of this system are the faculty and staff of the School of Mines and Engineering of Montana Technological University in Butte, Montana. An understanding of the assessment process, a familiarity with web browsers, and proficiency completing forms on a computer, is assumed. Between research, teaching, and service, engineering faculty are busy and are looking for a quick, easy-to-use system that will help them continually assess their programs.

2.5 Dependencies

ACID will be dependent on Montana Tech’s Central Authentication System, CAS. This enables users to use their existing Montana Tech passwords to access the software.

ACID will allow grades to be downloaded from Moodle. However, the system is not dependent on Moodle for grade entry.

2.6 Assumptions

Faculty have been encouraged to use the assessment methodology which the Electrical Engineering Department is using for tracking ABET criterion 3. Thus, one motivation for creating this software is to standardize current methods for tracking ABET criterion 3.

3 Use Cases

Use cases describe how users will interact with ACID. They outline, from a user’s point of view, ACID’s behavior as it responds to user interactions. Each use case is represented as a sequence of steps, beginning with a user’s goal and ending when that goal is fulfilled, or the user has exited the use case.

3.1 Roles

Users can be performing one or more of the following roles:

1. Department ABET Coordinator: This is a faculty member in a department who is tasked with overseeing the continuous improvement and ABET assessment of the program. The person with the role of ABET Coordinator may change over time and multiple people can play this role for each department.
2. Department Chair: A faculty member in a given academic department, assigned by the dean to manage that department.
3. Faculty Member: This is someone who inputs metric data for the classes they teach into the system.
4. Department Admin: This person may enter student names into course offerings and other tasks. This role is typically more involved with data input.
5. System Admin: This person has responsibility for maintaining the system after it has gone live. This person(s) will probably have read/write access to all data.
6. API: This actor isn’t human. An API (Application Program Interface) allows a system to programmatically access the data in the application and re-use it in the future.
7. Guest: This user class is for the ABET accreditors that visit the school. This user class is optional as it has not yet been determined if it is wanted.

The use cases which each role can perform are summarized in the following table.

Role	Use Cases
Department ABET Coordinator	<ol style="list-style-type: none">1. Associate and describe metric2. Create metric3. Create student outcome4. Score metric association
Department Chair	<ol style="list-style-type: none">1. Associate and describe metric2. Create metric3. Create student outcome4. Score metric association
Faculty Member	<ol style="list-style-type: none">1. Create metric2. Create student outcome3. Score metric association

Department Admin	1. Associate and describe metric 2. Create metric 3. Create student outcome 4. Score metric association
System Admin	Can perform all use cases
API	1.
Guest	1.

A use case diagram is another way to summarize the use cases that an ABET Coordinator can perform. The ‘Create outcome’ use case has been renamed ‘Create student outcome’ and the ‘Select item’ use case has been renamed ‘Choose PI and/or CO’.

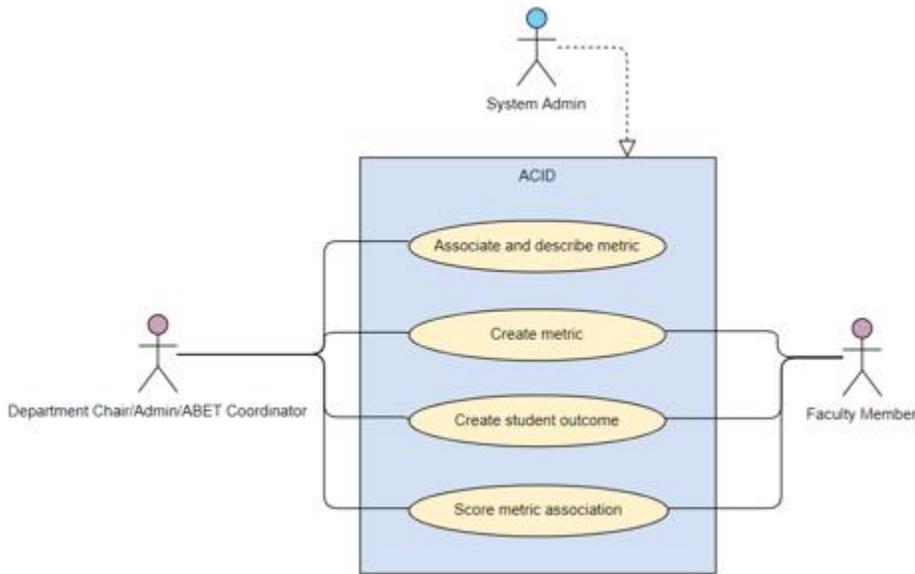


Figure 2. Use Case Diagram Showing Some Roles Use Cases

3.2 Use Cases

The following use cases are a sample of those needed for ACID.

3.2.1 Associate and describe metric

Created By:	Celia Schahczenski	Last Updated By:	Carson Fiechtner
Date Created:	April 2, 2020	Date Last Updated:	April 14, 2020
Actors:	Department ABET Coordinator, Department Admin, Faculty Member		

Description:	User associates a metric with one or more PIs and/or COs and describes the metric characteristics for this association.
Preconditions:	<ol style="list-style-type: none"> 1. User is logged in and has permission to do this action. 2. A metric has been selected.
Postconditions:	Unless the user exits this use case early, the metric has been associated with one or more items and the audit log is updated.
Normal Flow:	<p>1.0 Associate and describe metric</p> <ol style="list-style-type: none"> 1. User indicates desire to create an association for the selected metric 2. An 'associate and describe metric' interface appears that allows the user to do the following: <ol style="list-style-type: none"> i. select one or more existing PIs and/or COs which are associated with the department of the user ii. determine the type of association this will be iii. enter the information for the association type iv. to submit the data 3. The user is informed that the metric association has been created 4. The audit trail has been updated.
Alternative Flows:	<p>1.1 User doesn't submit and no changes were made (branch during step 2)</p> <ol style="list-style-type: none"> 1. User navigates away from the interface, or indicates a desire to exit the interface, before any changes were made 2. Use case exits <p>1.2 User doesn't submit after changes were made (branch during step 2)</p> <ol style="list-style-type: none"> 1. User navigates away from the interface, or indicates a desire to exit after changes have been made. 2. User is warned that they have unsaved changes and asked if they wish to proceed 3. User indicates preference 4. If 'yes' use case exits; if 'no' the user remains in the interface (i.e. they return to step 2). <p>1.3 Association already exists (branch after step 2)</p> <ol style="list-style-type: none"> 1. User tries to associate a metric with a PI/CO where an association already exists. 2. User is informed that an association already exists between the given metric and PI/CO. 3. Use case exits.
Exceptions:	<p>1.0.E.1 Internal error (branch after step 2)</p> <ol style="list-style-type: none"> 1. The system is not able to connect to the database or some other internal error. 2. User is informed that an error occurred and that no changes were made to the system.

Includes/Extends:	none
Priority:	High (metrics can't be scored until they are associated)
Frequency of Use:	Differs by department. For the CS departments, many times per semester. For the other departments, more rarely.
Business Rules:	<p>A metric cannot have multiple associations to the same PI/CO.</p> <p>Associations are one directional in ACID. A is associated with B is not the same as B is associated with A.</p> <p>The following is the hierarchy of items in ACID: SO PI CO Metric</p> <p>Items lower in list may only be associated with items higher on the list. For example, a metric may be associated with a PI. However, a CO may not be associated with a metric, but a metric can be associated with a CO.</p>
Special Requirements:	none
Assumptions:	1. Existing metrics associations may be updated through an "Update metric association" use case.
Notes and Issues:	<ol style="list-style-type: none"> 1. Descriptions 'exemplary', 'satisfactory' and 'unsatisfactory' can be customized for the metric association, or may be standardized for the department. 2. Scores for 'exemplary', 'satisfactory' and 'unsatisfactory' can be directly entered, or thresholds of the scores (out of 100%) can be entered. 3. Allowing departments to use labels other than 'exemplary', 'satisfactory' and 'unsatisfactory' may be included in a future enhancement. 4. Allowing departments to use more than 3 labels may be included in a future enhancement. 5. Currently CS only uses 'satisfactory' and 'unsatisfactory'. 6. Currently CS associates metrics with course offerings, rather than courses.

3.2.2 Create metric

Created By:	Celia Schahczenski	Last Updated By:	Justin Bak
Date Created:	Feb 17, 2020	Date Last Updated:	April 21, 2020
Actors:	Department ABET Coordinator, Department Admin, Faculty Member		
Description:	User creates a metric.		
Preconditions:	User is logged in and has permission to do this action.		

Postconditions:	Unless the user exits this use case early, the new metric has been created and the audit log is updated
Normal Flow:	<p>1.0 Create metric</p> <ol style="list-style-type: none"> 1. User indicates desire to create a metric 2. An ‘enter metric’ interface appears that allows the user to enter a phrase that describes the metric and to submit the data 3. The user is informed that the metric has been created 4. The audit trail has been updated
Alternative Flows:	<p>1.1 User doesn’t submit and no changes were made (branch during step 2)</p> <ol style="list-style-type: none"> 1. User navigates away from the interface, or indicates a desire to exit the interface, before any changes were made 2. Use case exits (no additions are made to the audit log) <p>1.2 User doesn’t submit after changes were made (branch during step 2)</p> <ol style="list-style-type: none"> 1. User navigates away from the interface, or indicates a desire to exit after changes have been made. 2. User is warned that they have unsaved changes and asked if they wish to proceed 3. User indicates preference 4. If ‘yes’ use case exits; if ‘no’ the user remains in the interface (i.e. they return to step 2).
Exceptions:	<p>1.0.E.1 Internal error (branch after step 2)</p> <ol style="list-style-type: none"> 1. The system is not able to connect to the database or some other internal error. 2. User is informed that an error occurred and that no changes were made to the system.
Includes/Extends:	none
Priority:	High (tracking metrics and generating reports about them is the main purpose of ACID)
Frequency of Use:	Differs by department. For the CS departments, many times per semester. For the other departments, more rarely.
Business Rules:	<p>ABET requires mapping between courses and student outcomes. Therefore, most metrics will be associated with courses but they can also be associated with other activities, such as participation in clubs, interviews, standardized exams and internships.</p> <p>Metrics need to define what student performance constitutes as exemplary, satisfactory, or unsatisfactory.</p>
Special Requirements:	None
Assumptions:	<ol style="list-style-type: none"> 1. Existing metrics may be updated through an “Update metric” use case.

	<ol style="list-style-type: none"> 2. Metrics will be associated with PIs and/or Cos via an ‘Associate and describe metric’ use case. 3. Scores will be associated with metrics via a ‘Score metric’ use case.
Notes and Issues:	<ol style="list-style-type: none"> 1. At some point users should not be allowed to update a metric. For example, once an ABET report has been made, changes to past system data should not be allowed. 2. Not all metrics are associated with courses. Exit interviews and surveys can be sources of metrics. 3. Metrics themselves are not scored, but rather the association between a metric and a PI/CO is scored.

3.2.3 Create student outcome

Created By:	ESOF 328 Class of 2020	Last Updated By:	ESOF 328 Class of 2020
Date Created:	Feb. 15, 2020	Date Last Updated:	March 12, 2020
Actors:	Department ABET Coordinator, Department Admin		
Description:	User creates a student outcome		
Preconditions:	<ol style="list-style-type: none"> 1. User is logged in. 2. User has permission to create a student outcome 		
Postconditions:	<ol style="list-style-type: none"> 1. Unless the outcome prefix and identifier are duplicated, or the user exits the use case early, the new student outcome has been created. 2. The audit log has been updated. 		
Normal Flow:	<p>1.0 Create a student outcome</p> <ol style="list-style-type: none"> 1. User indicates a desire to create a student outcome 2. A ‘enter student outcome’ interface appears that allows the user to select a prefix, enter a prefix identifier, enter the text of the student outcome and to submit the data 3. The user is informed that the student outcome has been created 4. The audit trail has been updated 		
Alternative Flows:	<p>1.1 User doesn’t submit student outcome and no changes were made (branch during step 2)</p> <ol style="list-style-type: none"> 1. User navigates away from entering the student outcome, or indicates a desire to exit ‘input student outcome’ interface, before any changes were made 2. Use case exits <p>1.2 User doesn’t submit student outcome after changes were made (branch during step 2)</p>		

	<ol style="list-style-type: none"> 1. User navigates away from entering the student outcome, or indicates a desire to exit ‘input student outcome’ interface, after changes had been made 2. User is warned that they have unsaved changes and asked if they wish to proceed 3. User indicates preference 4. If ‘yes’ use case exits; if ‘no’ the user remains (or returns) to step 2 <p>1.3 Duplicate student outcome (branch after step 2)</p> <ol style="list-style-type: none"> 1. The user is informed that a student outcome with the given prefix and identifier is already in the system and the prefix, identifier and text of the student outcome are displayed. 2. The user is informed that no changes were made to system data.
Exceptions:	<p>1.0.E.1 Internal error (branch after step 2)</p> <ol style="list-style-type: none"> 1. The system is not able to connect to the database or some other internal error 2. User is informed that an error occurred, that the developers are sorry, and the nature of the error
Includes/Extends:	None
Priority:	Low (the system can be prepopulated with necessary student outcomes for each program)
Frequency of Use:	Frequently when setting up the system, or when ABET changes outcome. ABET rarely changes student outcomes.
Business Rules:	Some outcomes are used by multiple programs
Special Requirements:	none
Assumptions:	<ol style="list-style-type: none"> 1. The software won’t allow a user to create a student outcome when that user doesn’t wield permission to do so. 2. The software won’t allow the user to submit student outcome information unless valid information is entered. 3. Existing outcomes should be updated through an “Update student outcome” use case.
Notes and Issues:	

3.2.4 Score metric association

Created By:	Celia Schahczenski	Last Updated By:	Diedrich Brush
-------------	--------------------	------------------	----------------

Date Created:	April 2, 2020	Date Last Updated:	April 2, 2020
Actors:	Department ABET Coordinator, Department Admin, Faculty Member		
Description:	User enters scores for a metric association.		
Preconditions:	<ol style="list-style-type: none"> 1. User is logged in and has permission to do this action. 2. A metric association has been selected. 		
Postconditions:	Unless the user exits this use case early, the metric association has been scored and the audit log is updated.		
Normal Flow:	<p>1.0 Score metric association</p> <ol style="list-style-type: none"> 1. User indicates desire to score a metric association. 2. The appropriate ‘score metric association’ interface appears that allows the user to do the following: <ol style="list-style-type: none"> i. enter the number of students who were than ‘exemplary’, ‘satisfactory’ and ‘unsatisfactory’, enter student scores, or upload student scores ii. to submit the data iii. reset the data 3. User is shown a preview of the changes and asked if they wish to proceed 4. User indicates preference 5. If ‘yes’ the changes are committed; if ‘no’ the user remains in the interface (i.e. they return to Step 2.) 		
Alternative Flows:	<p>1.1 User doesn’t submit and no changes were made (branch during step 2)</p> <ol style="list-style-type: none"> 1. User navigates away from the interface, or indicates a desire to exit the interface, before any changes were made 2. Use case exits <p>1.2 User doesn’t submit after changes were inputted (branch during step 2)</p> <ol style="list-style-type: none"> 1. User navigates away from the interface, or indicates a desire to exit after changes have been made. 2. User is warned that they have unsaved changes and asked if they wish to proceed 3. User indicates preference 4. If ‘yes’ use case exits; if ‘no’ the user remains in the interface (i.e. they return to step 2). <p>1.3 User indicates a desire to reset after changes were previewed (branch during step 4)</p> <ol style="list-style-type: none"> 1. User indicates that they do not want the changes that are previewed. 2. The user is reset to step 2. 		
Exceptions:	<p>1.0.E.1 Internal error (branch after step 2)</p> <ol style="list-style-type: none"> 1. The system is not able to connect to the database or some other internal error. 		

	2. User is informed that an error occurred and that no changes were made to the system.
Includes/Extends:	None
Priority:	High (metric association scores are needed to calculate values in reports)
Frequency of Use:	Differs by department. For the CS departments, many times per semester. For the other departments, more rarely.
Business Rules:	The way a metric is scored must be based on the association. Allowing one metric to be scored different ways depending on if it is scoring a PI or CO. A metric may be associated to both a PI and a CO. So the scoring must be a function of the association
Special Requirements:	None
Assumptions:	<ol style="list-style-type: none"> Existing metrics association scores may be updated through an “Update metric association score” use case. For associations using student scores, the student names will be entered via a use case. For associations using student scores, the scores can be updated from Moodle or a spreadsheet, via a use case.
Notes and Issues:	<ol style="list-style-type: none"> More information is needed on course offerings, student lists and uploading student grades.

3.3 Use Case Priorities

Here is a list containing expected use cases. The prioritization shown is based on the assumption that items can be pre-populated in the system.

High Priority	Medium Priority	Low Priority
1. Create metric	1. Delete metric	1. Export report to csv
2. Update metric	2. Create PI	2. Create SO
3. List metrics	3. Update PI	3. Update SO
4. Associate metric	4. Delete PI	4. Delete SO
5. Score metric association	5. Create CO	5. Create course
6. Delete metric association	6. Update CO	6. Update course
7. Generate report	7. Delete CO	7. Delete course
8. List course offerings	8. Score SO	8. Create report template
9. View SO	9. Update course offering	9. Edit report template
10. List SOs	10. Delete course offering	10. Import report template
11. View PI	11. View course	11. Select source data
12. List PIs	12. List courses	12. Print report
13. View CO	13. Associate PI with SO	13. Select report
	14. Score PI	

14. List Cos 15. Create course offering 16. View course offering	15. Associate course with SO 16. Associate CO with SO 17. Associate CO with course 18. Score CO 19. Associate course offering with course	
--	---	--

4 Specific Requirements

This section is to contain all of the requirements for ACID. Currently, however, it only shows a few example requirements. These could be alphabetized by name. Instead, they are presented in the order they are likely to occur.

4.1 Functional Requirements

Functional requirements describe the behavior that a software system will exhibit under specific conditions.

4.1.1 Login

When a user attempts to login to ACID the system shall use Montana Tech login names and passwords to verify users via CAS. If the credentials are accepted by CAS and if the user is registered in the system, the user is authorized to use the system.

Rationale: Login exists for security reasons, to protect sensitive data and data change.

Priority: Imperative

4.1.2 Create student outcome

After the user indicates a desire to create a student outcome, the system shall generate an interface that allows them to select the prefix and enter the prefix id and text of the student outcome.

Rationale: The user needs an easy way to enter all the data required to create a student outcome.

Priority: Important

4.1.3 Metric scoring

When a user is scoring a metric, which will be for a course or activity offering, the system shall facilitate the scoring by presenting a list of names of the students for which scores will be entered.

Rationale: Metrics will be scored for a different cohort of students.

Priority: Imperative

4.1.4 Importing grades

When a user wants to add metric scores to the system, the system shall allow grades to be imported via a standard API or a CSV file by generating an interface that allows the user to select a file from which to import scores.

Rationale: All grading and accreditation information is currently stored on other platforms. The ability to import the data will make the transition to using ACID much simpler for the users.

Priority: Medium, while not critical for system functionality, this feature has been requested multiple times by clients.

4.1.5 CI tracking

When a user is tracking continuous improvement, the system shall provide a space for annotations connected with a low score where faculty can record how this was addressed, and later, see the result of that remediation intervention.

Rationale: Users need an easy way to track continuous improvement, and attaching information to low scores allows them to track what they did and if their change worked.

Priority: Medium

4.1.6 Update ABET mapping

Whenever ABET updates their criteria, ACID shall be able to store and utilize this mapping.

Rationale: ACID needs to be able to keep up to date with the latest criteria and mappings provided by ABET.

Priority: Medium

4.2 Quality Attributes

Quality attributes specify criteria that applies to the system as a whole, rather than specific behaviors of the system. Quality attributes should be measurable and testable, so that the extent to which the system satisfies the quality attribute can be determined.

4.2.1 Privacy

This section includes quality attributes related to the privacy of data in the system.

PR 1: The system must comply with the current FERPA privacy regulations at the time it is released. The system must be updated to remain current with these regulations for as long as it is in use.

4.2.2 Security

This section includes quality attributes related to the security of the system.

S1: The system will only be accessible to faculty and staff members who are registered within the system, unless a guest login is implemented.

S2: Only registered users who are logged in and have permission to view student and class information will be allowed to view student and class information.

SR 3: The system must sanitize any input received from a user to prevent XSS and SQL injection attacks.

4.2.3 Usability

This section includes quality attributes related to the usability of the system.

U1: Data annotations and tool tips, telling what belongs in each field and how data calculations are performed, will be used to simplify data input.

U2: Users will be able to exit the application at any time.

U3: Users will be able to preview their changes to the system and will be given the option to confirm or undo those changes before the changes are committed to the system.

U4: Users will be informed if changes were attempted in the system but no changes were made.

UR 6: Users will be informed of all changes made to the system.

UR 7: Users will be informed if internal errors occur in the system and an audit log will be updated.

U8: No faculty or staff member should spend longer than 10 minutes figuring out how to complete a task of the application. Such tasks include:

1. Generate Reports
2. Create an Item
3. Update an Item
4. Score an Item
5. Associate an Item
6. Delete an Item
7. View an Item
8. List Items

U9: Users expect ACID to be flexible. Different departments track ABET criteria 3 differently and should be allowed to use whichever method they like in ACID.

For example:

- Petroleum counts their sample size by metric/scores. That is, if there is a class of 10 students and a class of 15 students, where the students overlap, the count the sample size as 25. Petroleum also collects data twice every six years, which was recommended by the Dean of the School of Engineering to allow for an early indication if an outcome isn't being met.
- Safety, Health & Industrial Hygiene collect data every semester and this is done by individual classes.
- Data could also be collected in cycles, so each year some outcomes are being assessed, but not all. For instance, focus on 2-3 outcomes each year, so that during years, all 7 outcomes are assessed twice.

U9: The following is the hierarchy of items in ACID:

SO
PI
CO
Metric

That is, metric measurements can contribute to create measures for one or more CO and/or PI. Similarly, CO measures can contribute to create measures for one or more PI and/or SO. Similarly, PI measures can contribute to create measures for one or more SOs. Having a hierarchy disallows cycles in the “contributes” network. “Contribute” links only go up the hierarchy, not down.

The following points are associated with metrics:

A metric can measure an item (PI or CO) in one of 3 ways:

- What it means to be exemplary, satisfactory and unsatisfactory can be described. (Example: “Exemplary means that the student understood the underlying concepts, completed the lab and obtained the desired results.”)
- Threshold scores for exemplary, satisfactory and unsatisfactory can be given. (Example: “Exemplary is at least 95%”. Geological Engineering uses this method of turning quantitative data to qualitative measures. OSH has also set threshold scores for exemplary, satisfactory and unsatisfactory.)
- A single threshold score for passing can be given, such as 70% (CS uses this method.)

Scoring will be different for each situation:

- The number of students in each category (exemplary, satisfactory and unsatisfactory) are directly given.
- Based on the metric scores, the number of students in each category are given.
- Based on the metric scores, the number of students passing are given.

Metrics can be associated with courses, or with other activities, such as participation in clubs, interviews, standardized exams and internships. It was suggested that these be called “extracurricular activities” or just “activities”.

Facilities for preventing changes made to metric scores after certain events should be present. For instance, once data has been placed into an ABET self-study report, it should no longer be possible to change relevant metric scores

4.3 Non-Functional Requirements Which Are Not Quality Attributes

Non-functional criteria such as platform, deployment, interface, design and document requirements are described in this section.

4.3.1 External Interface Requirements

4.3.1.1 Software

Support for bringing metric scores in from Moodle is highly desired specifically using csv files.

4.3.1.2 Communications

ACID should support an API to automatically assessment data from Moodle.

5 Future Enhancements

This software may eventually be enhanced to support non-engineering departments at Montana Tech, Northwestern accreditation and facilitating assessment of ABET criteria, other than criterion 3.

The system should allow users to associate other activities like internships, club memberships, and standardized exams as metrics in order to provide more continuous improvement data.

ACID should eventually have a custom report builder, where a user could define their own reports. ACID should save these definitions and allow them to be used over and over.

Appendices

Appendices contain detailed information, expanding on topics in the main document.

Appendix A: Definitions, Acronyms, and Abbreviations

This appendix provides definitions for terms used in this document or in the software.

A1. Definitions

ABET Coordinator	See “Department ABET Coordinator”
Activity	Possible source for metrics, such as participation in clubs, interviews, standardized exams and internships. These can be called “extracurricular activities” or just “activities”.
Assessment	One or more processes that identify, collect, and prepare data to evaluate the attainment of student outcomes. Effective assessment uses relevant direct, indirect, quantitative and qualitative measures as appropriate to the outcome being measured. Appropriate sampling methods may be used as part of an assessment process.
Course offering	A course given during a particular semester. There may be multiple sections of a course offered in a semester.
Department ABET Coordinator	A user role. A Person in a department tasked with overseeing continuous improvement and ABET accreditation for the department.
Metric	Items used to determine if students have met student outcomes, these are typically mapped to PIs or COs.
Moodle	A web application where instructors can post student grades on assignments and exams. Moodle is currently used by all departments in the School of Mines and Engineering.
Performance indicator (PI)	<p>Concrete, measurable statement of action the student should be able to perform to demonstrate attainment of student outcomes.</p> <p>Typically, 2-3 PIs per outcome and 3-4 “measures” per PI. It was speculated that AbOut “assessments” are what clients call “measures”. Thus, whereas AbOut maps outcomes to measures (via courses and offerings of courses), clients map outcomes to PIs to measures.</p> <p>PIs differ from one department to the next, but are standardized within a department</p>
Student outcome	Describes what students are expected to know and be able to do by the time of graduation. These relate to the skills, knowledge and behaviors that students acquire as they progress through the program.

Evaluation	Evaluation is one or more processes for interpreting the data and evidence accumulated through assessment processes. Evaluation determines the extent to which student outcomes are being attained. Evaluation results in decisions and actions regarding program improvement.
------------	--

A2. Acronyms and Abbreviations

AbOut	ABET Outcomes – a software system developed and used by the CS department
ABET	Accreditation Board of Engineering and Technology
ACID	Assessment and Continuous Improvement Database
CAC	Computing Accreditation Commission
CAS	Central Authentication Service
CO	Course Outcome
CS	Computer Science
EAC	Engineering Accreditation Commission
EE	Electrical Engineering
PI	Performance Indicator
SO	Student Outcome
SRS	Software Requirements Specification
SW	Software

Appendix B: Analysis Models

The following diagram gives a high-level view of acquiring the data to generate a report. Most reports summarize data on how students performed on metrics. The diagram shows the process of scoring a metric. The metric cannot be scored until it is associated with a performance indicator or course outcome and offering.

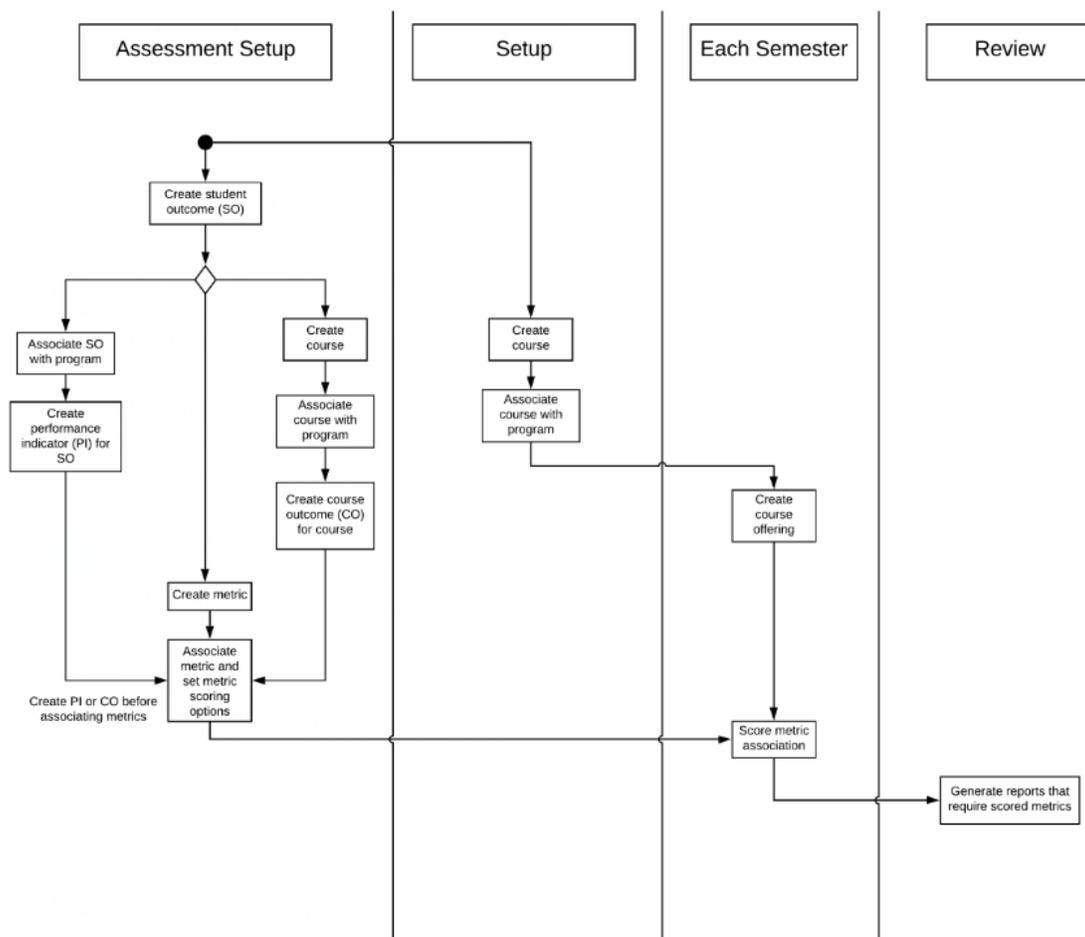


Figure 3. Generate Report Activity Diagram

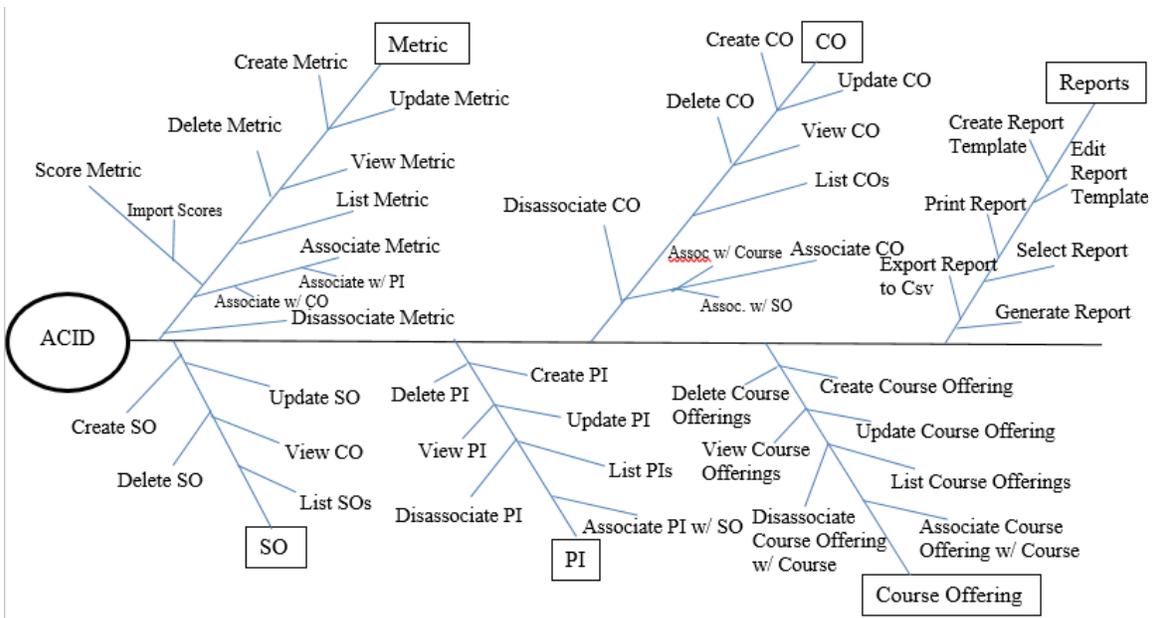


Figure ?? Feature Tree

Appendix C: Data Dictionary

The data dictionary defines the composition of data structures and the meaning, data type, length, format, and allowed values for the data elements that make up those structures.

Data Element	Description	Composition or datatype	Length	Values
student outcome	Describes what students are expected to know and be able to do by the time of graduation.	prefix, identifier, text	NA	EAC 1 an ability to identify, formulate, and solve complex ...
student outcome prefix	Initials ABET uses to group student outcomes	CHAR	10	EAC, CAC, SO, EAC-old
student outcome identifier	How ABET distinguishes outcomes in same group	CHAR	5	1, a
student outcome text	Working of student outcome	VARCHAR	300	The students will demonstrate an ability to ...
performance indicator	Concrete, measurable statement of action the student should be able to perform to demonstrate attainment of student outcomes.	program abbrev, student outcome prefix, student outcome identifier, PI identifier, text	NA	Pet EAC 1 1 Is able to write clear and technically correct reports
performance indicator text	Working description of performance indicator	VARCHAR	300	Is able to write clear and technically correct reports
program	Montana Tech Engineering program	program abbrev, name	NA	GEOE Geological Engineering
program abbrev	Montana Tech acronym for a program	CHAR	5	GEOE

program name	Montana Tech name for a program	VARCHAR	50	Geological Engineering
course	A course taught at Montana Tech	Prefix, identifier, name	NA	ESOF 328 Requirements and Specifications
course prefix	Initials denoting department that teaches a course	CHAR	5	ESOF, PET
course identifier	Number associated with the course	CHAR	5	328, 406W
course name	Name of the course	VARCHAR	100	Requirements and Specifications
course offering	Course offering in a particular term	Course prefix, course identifier, term	NA	ESOF 328, Spring 2018
term prefix	Sessions available at Tech	CHAR	10	Fall, Spring or Summer
semester	Term in which courses can be offered	Term prefix and year	NA	Spring 2018
course outcome	Describes what students are expected to know and be able to do by the end of a course.	Course, identifier, text	NA	ESOF 328 - Requirements and Specifications R1 Students can enumerate the various types and purposes of requirements.
course outcome identifier	How a department distinguishes outcomes for the same course	CHAR	5	R1
course outcome text	Working description of course outcome	VARCHAR	300	Students can enumerate the various types

				and purposes of requirements.
metric	Item used to measure student knowledge or skills	VARCHAR	300	selected exam questions and lab exercises in GEOE 410W
Metric association	Item used to associate a metric to a PI or SO	Identifier, metric, Measure (CO PI), qualData, quantData, sampleSize, score, qual, exemplary threshold, satisfactory threshold	NA	selected exam questions and lab exercises in GEOE 410W, Pet EAC 1 1 Is able to write clear and technically correct reports, null,
Measure	The identifier of either a CO or a PI	CHAR	5	R1
qualData	Qualitative data from a metric	Category, numStudents	NA	Exemplary: 3, Unsatisfactory: 2
qualData category	Qualitative category for scoring a metric	CHAR	14	Unsatisfactory
qualData numStudents	Number of students that fit into a category	SMALLINT	4	4, 27
quantData	A comma-separated list of scores	VARCHAR(500)	500	88,75,99
qual	Determines whether the association should be measured qualitatively	BOOL	NA	TRUE
Metric association identifier	Uniquely identifies a metric association	CHAR	10	12
Exemplary threshold	Threshold for a score to be exemplary	SMALLINT	3	90

Satisfactory threshold	Threshold for a score to be satisfactory	SMALLINT		70
score	The score of a measure	DOUBLE(4, 2)	4	95.24
sampleSize	The number of students contributing to the associated score	SMALLINT	4	12

The following diagrams show the relationship of major data items to each other. These are not meant to prototype the database but give an overview of the relationships between data entries.

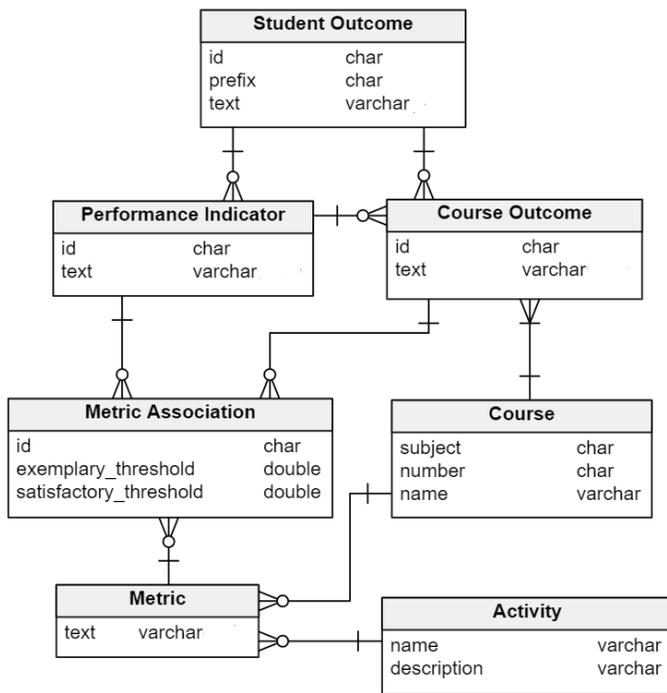


Figure 4. Data Model for a Program

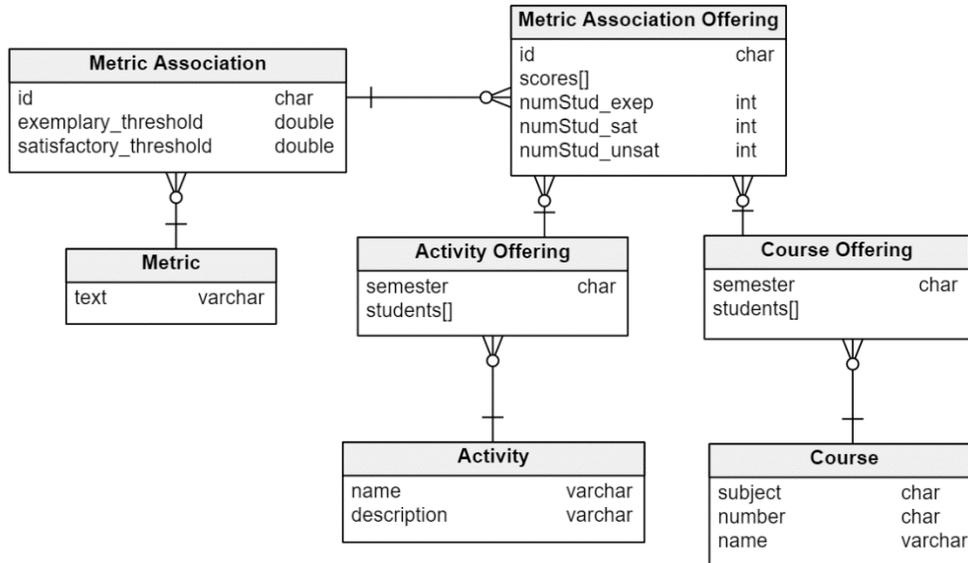


Figure 5. Data Model for a Semester

Appendix D: Report Specification

Reports must include sample sizes, along with percentage results. Flexibility is needed in generating the reports as different departments collect data at different.

Appendix E: Business Rules

The following rules have emerged from combining the Criterion 3 assessment processes of the engineering departments.

ID	Rule Definition	Type of Rule	Static or Dynamic	Source
BR 1	Metrics will never directly measure SOs.	Fact	Static	March 6th client meeting
BR 2	The “measures” hierarchy is SO, PI, CO, Metric.	Constraint	Static	March 6 th client meeting
BR 2	The following is the hierarchy of items in ACID: SO PI CO Metric That is, metric measurements can contribute to create measures for one or more CO and/or PI. Similarly, CO measures can contribute to create measures for one or more PI and/or SO. Similarly, PI measures can contribute to create measures for one or more SOs. Having a hierarchy disallows cycles in the “contributes” network. “Contribute” links only go up the hierarchy, not down.	Constraint	Static	March 6 th client meeting
BR 3	A metric can measure an item (PI or CO) in one of 3 ways: <ul style="list-style-type: none"> • What it means to be exemplary, satisfactory and unsatisfactory can be described. (Example: “Exemplary means that the student understood the underlying concepts, completed the lab and obtained the desired results.”) • Threshold scores for exemplary, satisfactory and unsatisfactory can be given. (Example: “Exemplary is at 	Fact	Static	March 6th client meeting

	<p>least 95%”. Geological Engineering uses this method of turning quantitative data to qualitative measures. OSH has also set threshold scores for exemplary, satisfactory and unsatisfactory.)</p> <ul style="list-style-type: none"> • A single threshold score for passing can be given, such as 70% (CS uses this method.) 			
BR 4	<p>Scoring metrics will be different for each situation:</p> <ul style="list-style-type: none"> • The number of students in each category (exemplary, satisfactory and unsatisfactory) are directly given. • Based on the metric scores, the number of students in each category are given. <p>Based on the metric scores, the number of students passing are given.</p>	Fact	Static	April 3, Client meeting
BR 5	<p>Metrics can be associated with courses, or with other activities, such as participation in clubs, interviews, standardized exams and internships. It was suggested that these be called “extracurricular activities” or just “activities”.</p>	Fact	Static	April 3, Client meeting
BR 6	<p>Facilities for preventing changes made to metric scores after certain events should be present. For instance, once data has been placed into an ABET self-study report, it should no longer be possible to change relevant metric scores</p>	Constraint	Static	April 3, Client meeting
BR 7	<p>ABET’s official SO’s should be pre-entered into ACID. Departments can then use the ‘Create SO’ functionality to enter program specific outcomes into ACID.</p>	Fact	Static	April 3, Client meeting
BR 8	<p>Metric scores cannot be changed after a given scoring deadline</p>	Constraint	Static	March 6 th client meeting

BR 9	A metric cannot be associated with the same PI/CO multiple times	Constraint	Static	April 3 rd client meeting
BR 10	Official ABET outcomes should be pre-entered into ACID	Constraint	Static	March 6 th client meeting
BR 11	Metrics can be associated with courses or with other activities, such as participation in clubs, interviews, standardized exams and internships.	Fact	Static	March 6 th client meeting
BR 12	Departments can enter data into ACID at different intervals, ranging from every semester to twice every six years	Fact	Dynamic	Feb. 21 st client meeting

Appendix F: Assessment Data Examples

As metrics are a core item in ACID, stakeholders have provided some examples of what their metrics look like:

Performance indicators	Courses	Method of assessment
1. Can choose a mathematical or statistical model to solve an engineering problem	PET 410 PET 372 PET 404	Performance on select questions on final exam, midterm, final project or homework
2. Understands limitations of models and checks solutions for reasonableness	PET 410 PET 499	History matching assignment, senior design modeling component
3. Can relate scientific concepts to engineering problems	PET 372 PET 404	Performance on select homework and exam questions

SO

PIs

Metrics

Figure 2. Petroleum Engineering Metrics

	an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
1	1.1 - Apply non-GEOE general engineering knowledge
	i. Metric 1: Level I, II, and III ETS Math proficiency exam scores (average of all 3)
	ii. Metric 2: selected exam questions and lab exercises in GEOE 410W.

Metrics

Figure 3. Geological Engineering Metrics

ABET SOs	MTech SOs	PEOs	OSH 4216 IH I	OSH 4226 IH II	IH 5076 Statistical Analysis	IH 5136 IH Mgmt	IH 5276 Advanced Ind Tox	IH 5286 Sampling & Eval	IH 5777 Sampling & Eval Lab
1	1								
2	2					X			
3	3a				X				
	3b				X				
4	4a						X		
	4b								

Metrics

SOs PIs

Figure 4. Industrial Hygiene Metrics

Appendix F: Sample User Interface

This section contains a dialog map sample user interfaces for scoring a metric in ACID. Please note that the user interfaces are only an example.

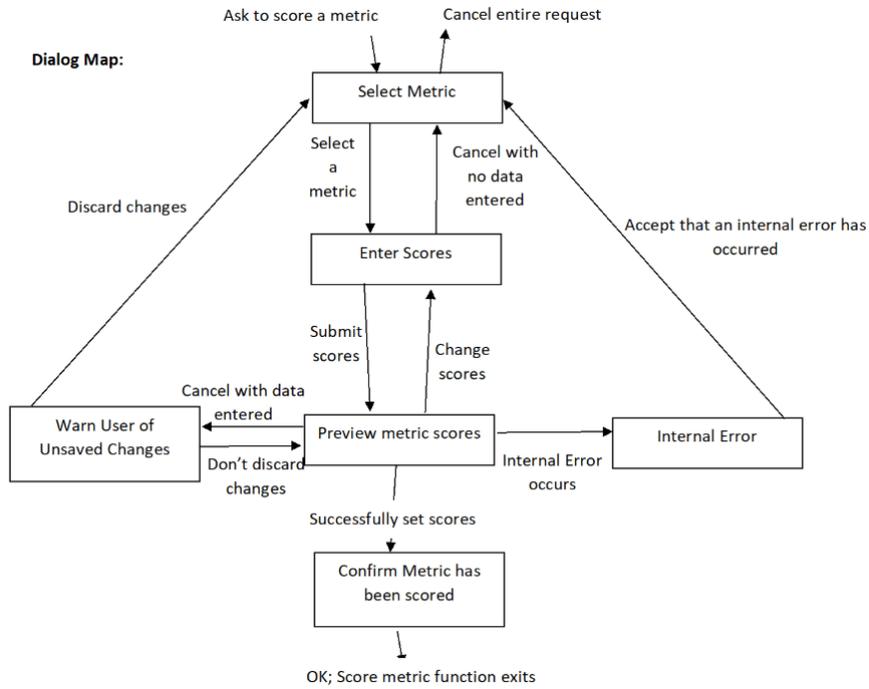


Figure 7. Dialog Map for Scoring a Metric

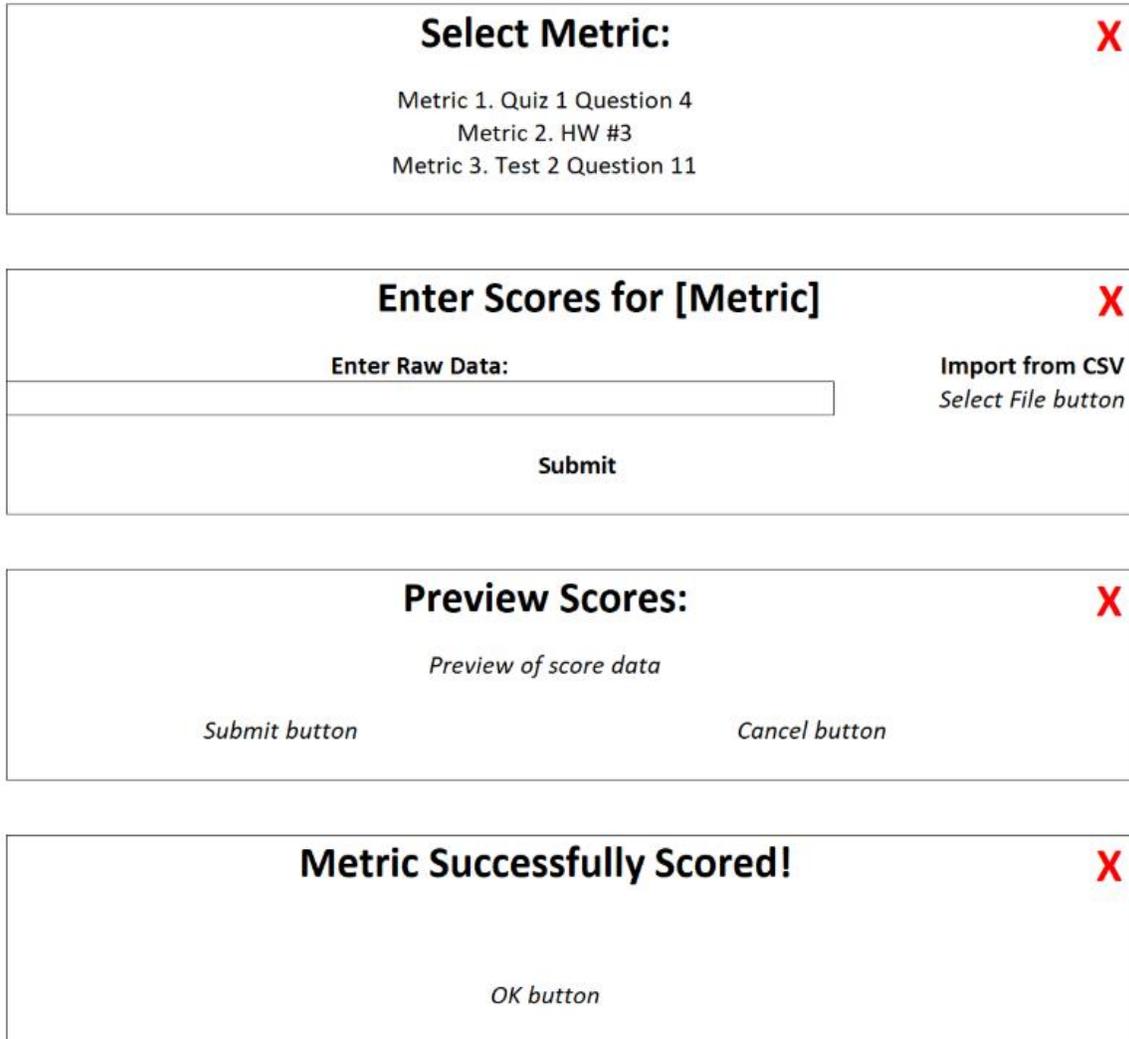


Figure 8. Sample UI's for Scoring a Metric

Appendix G: Issues

This section outlines any unresolved issues pertaining to ACID.

Outstanding issue – Some clients use rubrics, rather than scores, and use summary data, rather than student data. At least one of the clients was worried about tracking data by specific users.

The clients were asked how they name metrics. From what the clients have told us, and what data and models we have been shown, it is likely that metrics could be named simply as metric 1, metric 2, etc. If this is the way to go, then metrics should be auto named, and should have a user entered description of what that metric is

A client suggested including a “group” field with each outcome. A group field would allow for easy tracking when that group was created, what outcomes come from which group, and quickly expiring all old outcomes. A group field could hold things like an id, a start date, and a mapping

to help with historical logging and reporting. Using semesters for the start and end date for these groups might be sufficient, instead of dates.