SAINT LOUIS UNIVERSITY

# Program-Level Assessment: Annual Report

Program Name (no acronyms): Software Engineering

Department: Computer Science

Degree or Certificate Level: MS

College/School: School of Science and Engineering

Date (Month/Year):

Assessment Contact: Erin Chambers

In what year was the data upon which this report is based collected? AY 2021-2022

In what year was the program's assessment plan most recently reviewed/updated? 2018

Is this program accredited by an external program/disciplinary/specialized accrediting organization or subject to state/licensure requirements? No

If yes, please share how this affects the program's assessment process (e.g., number of learning outcomes assessed, mandated exams or other assessment methods, schedule or timing of assessment, etc.):

## 1. Student Learning Outcomes

Which of the program's student learning outcomes were assessed in this annual assessment cycle? (Please provide the complete list of the program's learning outcome statements and **bold** the SLOs assessed in this cycle.)

This year, assessment was targeted at the following outcome:

PLO 1 - Design, implement, evaluate and test a complex software system that meets a given set of computing requirements.

PLO 6 – Function effectively as a member of a team in developing computing technology and solving technical problems.

## 2. Assessment Methods: Artifacts of Student Learning

Which artifacts of student learning were used to determine if students achieved the outcome(s)? Please describe the artifacts in detail, identify the course(s) in which they were collected, and if they are from program majors/graduates and/or other students. Clarify if any such courses were offered a) online, b) at the Madrid campus, or c) at any other off-campus location.

CSCI 5300: Students were asked to participate in an in-class assessment and were rewarded with a small participation credit. Students were instructed to not use any internet resources to answer assessment questions. Since the grade was based on completeness and not correctness, most of the answers likely represent students' learning outcomes.

Assessment details:

1. Pick one data structure you are using in your team project. Briefly describe the data structure and the purpose it serves in your project.
2. Analyze the choice of this data structure for the purpose it serves in terms of program efficiency, coupling, and/or cohesion.
3. What alternative data structure could you have used? Analyze if this alternative would be a better choice for your project.
4. Explain what the term "security" means in the context of software.
5. Describe what measures you would take to ensure that the software you produce is "secure".

6. State and explain what you believe is the ideal team size for a: Small project (about the size of our class project) and a medium project
7. Given your ideal team size and project requirements, explain how you would organize your team and approach the development process to deliver the required software.
8. Describe the git workflow we have utilized in this class for the team project.
9. Explain the difference between the workflow we used in this class and the workflows you have used in other situations.
10. What considerations do you need to take into account when deciding on what workflow to use?
11. Explain how we applied various development tools to assure code quality.

Two other courses were supposed to provide assessments, but were taught by adjuncts or visitors who did not complete the data as required.

## 3. Assessment Methods: Evaluation Process

What process was used to evaluate the artifacts of student learning, and by whom? Please identify the tools(s) (e.g., a rubric) used in the process and **include them in/with this report document** (please do not just refer to the assessment plan).

The rubric used to assess is attached.

## 4. Data/Results

What were the results of the assessment of the learning outcome(s)? Please be specific. Does achievement differ by teaching modality (e.g., online vs. face-to-face) or on-ground location (e.g., STL campus, Madrid campus, other off-campus site)?

For CSCI 5300:

| Score | Data Structures | Computer systems: Security | Software development: Team and organization | Software development: Team and workflow |
|---|---|---|---|---|
| 4 | 9 | 1 | 10 | 5 |
| 3 | 3 | 4 | 3 | 11 |
| 2 | 4 | 7 | 1 | 0 |
| 1 | 0 | 4 | 2 | 0 |

## 5. Findings: Interpretations & Conclusions

What have you learned from these results? What does the data tell you? Address both a) learning gaps and possible curricular or pedagogical remedies, and b) strengths of curriculum and pedagogy.

Overall, we saw reasonable scores in terms of the application of data structures, as well as software development processes.  Security and systems was a weak point in this assessment, but we wish to evaluate the systems classes to see how students perform in core classes from this area.

As a result of the two classes which failed to gather assessment data, we had quite a bit of discussion about how to handle assessment when faculty are not full time at the university, which has led to further discussions of quality control in general on such courses.

## 6. Closing the Loop: Dissemination and Use of Current Assessment Findings

**A.** When and how did your program faculty share and discuss the results and findings from this cycle of assessment?

This discussion occurred in fall faculty meetings, as well as in administrator meetings with the associate dean for the new college.

**B.** How specifically have you decided to use these findings to improve teaching and learning in your program? For example, perhaps you've initiated one or more of the following:

Changes to the Curriculum or Pedagogies

- Course content
- Teaching techniques
- Improvements in technology
- Prerequisites
- Course sequence
- New courses
- Deletion of courses
- Changes in frequency or scheduling of course offerings

Changes to the Assessment Plan

- Student learning outcomes
- Artifacts of student learning
- Evaluation process
- Evaluation tools (e.g., rubrics)
- Data collection methods
- Frequency of data collection

Please describe the actions you are taking as a result of these findings.

As a result of both this year's discussion and the growth we are experiencing, we have resolved as a department to transition our assessment plan and discussion to a new model. Assessment will be conducted in area clusters, focused around the following broad areas: introductory classes, ethics, software engineering, AI/ML, theory, and systems/networking/security.

Each faculty area group is charged to meet over the course of the year, to reflect on current content, agree on an assessment plan, and report back to the faculty in department meetings in the next 1-2 years with a developed plan. Given the increasing size of the program, we expect the next 1-2 years to involve significant overall revision of course sequencing and content, as well as pedagogy, since many of these classes will transition to larger sizes in the next few years. It is likely this will necessitate that we include smaller statical samples of direct student assessment, supplemented by overall grades in the larger classes.

In addition, one of the more helpful pieces of assessment in discussion was in faculty reflections. The chair will collect individual reflections in the fall for every class taught, so that faculty can share these within areas and use them to develop improvements to the content both individually and in groups.

We will also avoid the use of adjuncts as much as possible for key assessment courses, as it is clear gaps can enter if the faculty choose not to report the data and/or do not come back to teach and share it.

Finally, in the coming year we will also try to incorporate exit interviews for all students, to gain a more big-picture and holistic view of the student experience.

If no changes are being made, please explain why.

## 7. Closing the Loop: Review of Previous Assessment Findings and Changes

**A.** What is at least one change your program has implemented in recent years **as a result of previous assessment data**?

This is no prior round of successful assessment, since the MS program began just as the COVID pandemic hit. As a result, this represents our first round of solid assessment data.

**B.** How has the change/have these changes identified in 7A been assessed?

N/A

**C.** What were the findings of the assessment?

N/A

**D.** How do you plan to (continue to) use this information moving forward?

N/A

**IMPORTANT: Please submit any assessment tools (e.g., artifact prompts, rubrics) with this report as separate attachments or copied and pasted/appended into this Word document. Please do not just refer to the assessment plan; the report should serve as a stand-alone document. Thank you.**

# SE PLO 2 - Project Management Processes and Tools

### Outcomes

Graduates of the program will have an ability to...

**MS-SE** Utilize project management processes and tools through the complete software life cycle.

### Project Management Processes

| Criterion | 4: Exemplary | 3: Accomplished | 2: Developing | 1: Beginning |
|---|---|---|---|---|
| **Development Methodology**[1] | Student can evaluate, select, and justify a set of development practices that support a given project and team. | Student recognizes how the use of specific development practices enables achievement of project goals and outcomes. | Student can enact common agile and waterfall development practices. | Student can compare and contrast defined process methodologies (e.g. waterfall) with emperical process methodologies (e.g. agile), and give examples of projects where one is more appropriate than the other. |
| **Refactoring** | Is able to appropriately balance the competing concerns of cost, time, project velocity, and technical debt to determine how refactoring as a practice fits into a larger software development effort. | Student uses refactoring tools[2] to automate refactoring processes. Uses regression testing to verify code behavior is constant. | Student can rework code and can apply suitable transformations to improve code quality while holding behavior constant. | Student can identify code smells and articulate possible solutions. |
| **Code Reviews** | Student can develop a code review strategy to support specific project and team objectives. | Student can create and format appropriate changesets to maximize code review effectiveness. Uses technology appropriately[3] to facilitate reviews. Employs communication strategies to balance code improvement versus team harmony and human impact. | Student makes meaningful requests for review of their own work. Can follow a code review procedure and provide meaningful feedback. | Student knows code review concepts and practices. Student is haphazard in asking for code review and reviewing code of others |

Note: A score of zero should be given for students that do not meet the basic standard.

Comments on rubric:

1. The rubric does not require the use of any specific agile or defined process methodology or practices, but does require that the student recognizes the division in perspective between these concepts.

2. Automatic tools such as linters, static code analyzers, dynamic code analyzers, memory leak detectors, etc.

3. Appropriate technology such as using branching and pull requests when using GitLab or GitHub.

### Tools

| Criterion | 4: Exemplary | 3: Accomplished | 2: Developing | 1: Beginning |
|---|---|---|---|---|
| **Version control** | Can articulate a suitable versioning and branching strategy for a given project and team. Can integrate version control with automatic and manual processes for refactoring, code quality, and code review. | Student successfully uses a version control system as part of a development team. Can track changes to the source tree, create and manage suitable feature branches, and merge changes appropriately. Uses tags appropriately. | Student can use a version control system to track changes to the source tree. Uses commits but may be overly large or small in scope. Uses branches to separate development code from release code. Uses commit messages to appropriately document changes. | Student uses version control system to store periodic snapshots of project. |

Note: A score of zero should be given for students that do not meet the basic standard.