CSC 301: INTRODUCTION TO SOFTWARE ENGINEERING

SPRINT 0 (FINAL SUBMISSION) GRADING RUBRIC

The assignment is graded out of 100. Final scores are rounded to the nearest whole point.

Method of score computation:

For each element a rating is assigned based on the rubric. Each rating has an associated point value: Excellent 100, Good 75, Adequate 65, Marginal 50, and Inadequate 0.

The scores for the elements are combined according to their respective weights to reach a score for that assignment part (out of 100).

The scores for the assignment parts are combined according to their respective weights to reach an overall score for the assignment (out of 100). The assignment grade is that overall score rounded to the nearest point.

PART 1: PROCESS [70% OF TOTAL]

PHOTOGRAPH [5% OF PART]

Excellent	Marginal	Inadequate
-Photograph is clear -Group members' and product champion's faces are clearly visible* -Group members have been labeled with names -All group members and the product champion are present	-Photograph may have some issues with visibility of some persons -Labels are missing or not readable -Some group members or the product champion may be missing	-Photograph not available, or, if available, is in a non-standard format without instructions for viewing on CDF machine, or is not viewable on CDF machines -Photograph is a composite of multiple images

*This requirement will be interpreted in accordance with The University of Toronto *Statement on Human Rights*

PROJECT PAGES [5% OF PART]

Excellent	Good	Adequate	Marginal	Inadequate
-Sprint 0 page created on wiki -All required elements (info about team, the project, subpage for meeting minutes, and anything else implicit/explicit in the assignment) are on the wiki -All links are accessible to a logged-in grader without any need to follow-up with group -One hyperlink per deliverable; in the case multiple deliverables were combined in a single document, links are appropriately titled to make this clear or multiple redundant links are employed	-Sprint 0 page created on wiki -All required elements are on the wiki -All links are accessible to a logged-in grader without any need to follow-up with group -Some elements may be missing hyperlinks / not clearly labelled, however the grader could find those elements elsewhere in the wiki relatively easily	-Sprint 0 page created on wiki -All required elements are on the wiki -All links are accessible to a logged-in grader without any need to follow-up with group -Some elements may be missing hyperlinks / not clearly labelled, however the grader could find those elements elsewhere in the wiki with some effort	-No Sprint 0 page on Wiki, however most required elements are on the wiki -Grader may have had to follow-up with group about permission problems -Some elements may be missing hyperlinks / not clearly labelled	-Wiki not completed or seriously deficient -Grader has to look through other resources (e.g. depot) to find necessary deliverables

TEAM CHARTER [5% OF TOTAL]

Excellent	Good	Adequate	Marginal	Inadequate
-Charter includes all required elements (regular meeting times, values, and commitments) to a comprehensive level of detail -Team charter is of appropriate length and is well-written	-Charter includes all required elements (regular meeting times, values, and commitments) to a reasonable level of detail -Team charter is of appropriate length -Any writing issues are minor and largely mechanical	-Charter includes all required elements (regular meeting times, values, and commitments) -Team charter may be somewhat brief / overly long -May have some issues with writing	-Charter includes most/all required elements (regular meeting times, values, and commitments), but some may be overly brief, missing, and some may require further detail -Charter is poorly explained and may be confusing to a reviewer -Writing may suffer from issues that impair comprehensibility	-Charter is inappropriate or severely lacking in required elements -May have significant writing issues the severely impair comprehensibility

PERSONAS [30% OF TOTAL]

Excellent	Good	Adequate	Marginal	Inadequate
-Two or more high quality personas provided -All required elements included: • Gender • Age • Personality • Family • Appearance • Skills • Environment • Attitude towards technology, domain, etc. • Goals when using the system -Personas are highly relevant to the system -Description is of appropriate length and is well-written	-At least two personas provided - Most required elements included -Personas are relevant to the system -Description is of appropriate length -Any writing issues are minor and largely mechanical	-At least two personas provided - Some, but not all, required elements may not be included -Personas are largely relevant to system -Description may be somewhat brief / overly long -May have some issues with writing	-May be fewer than two personas provided -May be significant deficiencies with respect to required elements -Writing may suffer from issues that impair comprehensibility	-No personas provided, only one persona, or personas provided are highly deficient -May have significant writing issues the severely impair comprehensibility

USER STORIES [35% OF TOTAL]

Excellent	Good	Adequate	Marginal	Inadequate
-At least 20 stories were generated -Key features are included is thorough and well thought out -Relevant persona clearly identified for each user story -All user stories identify goal/desire and all identify why/benefit -Any writing issues are minor and largely mechanical -Conversations with product champion recorded -Easy to derive test cases	-At least 20 stories were generated -Most key features are included and well thought out -Relevant persona clearly identified for most user stories -All user stories identify goal/desire and most identify why/benefit -Any writing issues are minor and largely mechanical -Most conversations with product champion recorded -Easy to derive most test cases	-May be fewer than 20 stories generated -One or two key features may be missing -Details not well thought out -Relevant persona clearly identified for some, but not all user stories -All user stories goal/desire but may not have why/benefit -May have some issues with writing beyond mechanical; but meaning can be inferred -Many conversations with product champion are missing -Easy to derive test cases	- May be fewer than 20 stories generated -Multiple key features are missing -Details not thought out and may conflict -Relevant personas often not identified for user stories -All user stories goal/desire, but some may be unclear without context -Writing may suffer from issues that impair comprehensibility -Conversations with product champion are missing -Difficult to derive test cases	-Unclear evidence of use of user stories or user stories significantly deficient -No record of contact with product champion -Insufficient detail to derive test cases

PROJECT CHARTER [5% OF PART]

Excellent	Good	Adequate	Marginal	Inadequate
-Goals for project are clearly stated and well explained -Success criteria for project identified; goals are specific, measurable, achievable and realistic -Project charter is of appropriate length and is well-written	-Goals for project are clearly stated and explained -Success criteria for project identified; most goals are largely all specific, measurable, achievable and realistic -Project charter is of appropriate length -Any writing issues are minor and largely mechanical	-Goals for project are stated and explained -User stories for next sprint identification; justification may be somewhat lacking -Success criteria for project identified; however those goals may not be specific, measurable, achievable and realistic -Project charter may be somewhat brief / overly long -May have some issues with writing	-Goals for project are stated but may require further detail -User stories allocated to next sprint, but there is no/deficient justification -Charter is poorly explained and may be confusing to a reviewer -Writing may suffer from issues that impair comprehensibility	-Charter is inappropriate or severely lacking in required elements -May have significant writing issues the severely impair comprehensibility

TECHNOLOGIES TO BE USED [5% OF PART]

Excellent	Good	Adequate	Marginal	Inadequate
-Technologies (software tools, frameworks, APIs) to be used are clearly stated and well justified -Technologies are suitable to the problem to be solved	-Technologies to be used are clearly stated and justified -Technologies are suitable to the problem to be solved	-Technologies to be used are stated and justified -Technologies are adaptable to the problem to be solved, however readily available superior alternatives could be discovered with a minimum of research	-Technologies to be used are stated but may require further detail -Technologies selected are in theory adaptable to the problem to be solved, but are much less-suitable than readily available alternatives	-Implementation technologies are not identified or would clearly inappropriate for the problem to a person of ordinary skill

MEETING MINUTES AND ATTENDANCE SPREADSHEET [10% OF PART]

Excellent	Good	Adequate	Marginal	Inadequate
-Meeting minutes evidence appropriate frequency of meetings -Meeting activities recorded in detail -Attendees listed and date and time of meeting noted -Action items clearly recorded and have deadlines -Evidence that all meetings recorded (including tutorial) -Separate attendance log spreadsheet is included and complete	-Meeting minutes show evidence of regular meetings -Meeting activities recorded with some detail -Date and time of meeting noted -Action items recorded but some may lack deadlines -Evidence that all meetings recorded (including tutorial) -Separate attendance log spreadsheet is included and complete, but may have minor issues	-Meeting minutes evidence various meetings through the sprint -Meeting activities recorded -Date of meeting recorded -Action items recorded, but may not be in sufficient detail for follow-up/lack deadlines -Evidence that most meetings recorded -Separate attendance log spreadsheet may be missing but attendance is in the meeting minutes	-Evidence of insufficient meeting frequency -Meeting minutes highly incomplete -Meeting record may lack date/time information -Action items difficult to comprehend without additional context -Various meetings omitted or poorly recorded -Separate attendance log spreadsheet not included or is deficient	-No meeting minutes or other evidence of meetings

PART 2: PRODUCT TASKS [30% OF TOTAL]

SELECTION OF USER STORY [5% OF PART]

Excellent	Good	Adequate	Marginal	Inadequate
-User story for implementation identified; selection is clearly justified -User story is estimated and estimate is reasonable	-User story for implementation identified; selection is largely justified -User story is estimated and estimate is reasonable	-User story for implementation identified; justification may be somewhat lacking -User story is estimated and estimate is somewhat reasonable	-User stories selected for implementation, but there is no/deficient justification -User story is estimated, however estimate appears unreasonable	-User story for implementation not identified or not estimated

TESTING [40% OF PART]

The appropriate table will be used for automation test and manual test. In the case of a mix both, an overall testing rating is assigned by considering both tables.

Automation Test

Excellent	Good	Adequate	Marginal	Inadequate
-Demonstrates a mastery of unit and integration testing -Uses a test suite -All methods and classes covered -Integration tests are included for all related components -Complete positive and negative tests cases for all methods present -Boundary conditions considered and checked -Tests include all input conditions and return values -Tests include those for errors and exceptions	-Demonstrates skill with unit and integration testing -Uses a test suite -All methods and classes are covered with rare exceptions -Integration tests are included for most related components -Positive and negative tests cases for all methods present -Tests case sets or boundary condition testing be inconsistent -Tests include most input conditions and return values -Some error conditions may be untested	-Demonstrates an understanding of unit and integration testing concepts -Uses a test suite -Most methods and classes are covered -Integration tests are included for some related components -Some positive and negative tests case sets may be lacking -Boundary conditions often remain untested -Tests include some input conditions and return values -Error testing is lacking	-Demonstrates some familiarity with unit and integration testing concepts -Uses a test suite -Numerous methods may remain and classes untested -Little or no integration testing -Tests fail to address many scenarios and boundary conditions -Only basic input conditions and return values tested -No testing for errors	-Unit and integration tests added are inadequate; numerous expected tests are omitted -No test suite -Tests are very sparse -Expected results may be invalid or incorrect -No demonstration of a clear strategy for testing

Manual Test

Excellent	Good	Adequate	Marginal	Inadequate
-Demonstrates a mastery of thorough manual testing -Uses a written test plan that thoroughly explains all steps and expected results at each stage -Complete positive and negative tests cases for all user interface -Tests include all input conditions and expected results -Tests include those for error conditions	-Demonstrates skill with thorough manual testing -Uses a written test plan that includes all steps and most expected results -Positive and negative tests cases for all user interface -Tests include most input conditions and expected results -Some error conditions may be untested	-Demonstrates an understanding of thorough manual testing concepts -Uses a written test plan that includes all steps; may omit some expected results at interim stages -Most methods and classes are covered -Some positive and negative tests case sets may be lacking -Tests include some input conditions and expected results -Error testing is lacking	-Demonstrates some familiarity with thorough manual testing concepts -Uses a written test plan -Numerous methods may remain and classes untested -Only basic input conditions and expected results tested -No testing for errors	-Thorough manual tests added are inadequate; numerous expected tests are omitted -No written test plan -Tests are very sparse -Expected results may be invalid or incorrect -No demonstration of a clear strategy for testing

DESIGN [10% OF PART]

Excellent	Good	Adequate	Marginal	Inadequate
-Evidences a mastery software design -Design is highly flexible / adaptable -Excellent application of appropriate design patterns throughout added code -Design is easily understood from code and external documentation and is explained to a high standard	-Evidences facility with software design -Design is clear and appropriate to the problem -Design patterns are applied variously throughout the code -Design is comprehensible from code and external documentation	-Evidences an understanding of software design -Design is serviceable for the problem given -Would benefit for greater application of standard design patterns or other indicia of a flexible design -Design is understandable with some effort; external documentation may be lacking	-Evidences some familiarity with software design practice -Design is inflexible -May be some evidence of "anti-patterns" within the code -Design is poorly explained and may be confusing to a reviewer	-Little evidence of even superficial understanding of software design -Design is inappropriate or not evident; haphazard implementation

IMPLEMENTATION [40% OF PART]

Excellent	Good	Adequate	Marginal	Inadequate
-Evidences a mastery of "best practices" for software implementation -All required functionality included -Code changes highly targeted -Highly adaptable coding techniques; use of implementation techniques that facilitate future changes and maintenance -Variable and method names are clear and descriptive -Comments are frequent and clear and relate to code structure / function (e.g. "populate the look-up table") rather than paraphrasing the code (e.g. "increment the counter") -Indenting is consistent across the code base -Code is clear and highly readable	-Evidences facility with general standards of software implementation practices -All required functionality included -Code changes largely isolated to appropriate components -Code does not present significant barriers to future changes or maintenance; code allows for easy replacement of more specialized portions of implementation -Variable and method names are clear and descriptive -Comments are clear -Indenting is consistent within each source file, but there may be some variation from file-to-file -Code does not exhibit any significant readability issues	-Evidences an understanding of good software implementation -Most required functionality included, however some edge or error cases unhandled -May be some code changes that evidence unnecessary reengineering -Code may present some barriers to future changes or maintenance -Some variable or method names may be unclear but their meaning can be inferred from context -Commenting is sparse; may be limited to general description of method function -Indenting is consistent within each source file, but there may be some variation from file-to-file -May employ "tricky" coding techniques that serve to limit readability	-Evidences some familiarity with good software implementation practice -Core required functionality implemented -Code changes are made across the codebase, not limited to components that required modification to implement the user story -Examples of hard-coded functionality that should have been parameterized based on input or values stored in secondary storage -Variable or method names are unclear, but meaning may be inferred from context -Comments may be misleading or serve only to paraphrase the code -Indenting is missing or highly inconsistent even with the same source file -May employ highly obtuse coding style such as idiomatic use of side effects	-Little evidence of even superficial understanding of software implementation best practice -Significant required functionality omitted -Variable or method names are poorly chosen and may be misleading -Comments, when present, may be misleading or unhelpful -Source code may show evidence of a lack of understanding of proper code formatting

RUN [5% OF PART]

Excellent	Good	Adequate	Marginal	Inadequate
-Product runs either by downloading an executable or running on Windows Azure -Product runs as expected	-Product runs either by downloading an executable or running on Windows Azure -Product runs as expected, product may have minor issues	-Product runs either by downloading an executable or running on Windows Azure -Product largely runs as expected, but many minor issues encountered	-Product runs either by downloading an executable or running on Windows Azure -Product does not run as expected	-Product does not runs either by downloading an executable or by running on Windows Azure