CSC 301: INTRODUCTION TO SOFTWARE ENGINEERING

SPRINT 1 (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 [60% OF TOTAL]

BACKLOG: ESTIMATION OF STORY POINTS [10% OF TOTAL]

Excellent	Good	Adequate	Marginal	Inadequate
-Evidence of clear estimation strategy and methodology -All or almost all stories have appropriate estimates -Estimates were consistent with one another -Estimates took into account other user stories if necessary	-Some evidence of estimation strategy -Most stories have appropriate estimates -Estimates were mostly consistent with one another -Most estimates took into account other user stories if necessary	-Estimation strategy is unclear or not sufficiently documented -Some stories have appropriate estimates, some are unreasonable -Many estimates were inconsistent with one another- Related user stories not usually considered in estimates	-No evidence of any estimation methodology -Estimates are unreasonably low/high -Estimates were not consistent with one another- Related user stories not considered in estimates	-No user story point assignment

BACKLOG: PRIORITIZATION BY PRODUCT CHAMPION [10% OF TOTAL]

Excellent	Good	Adequate	Marginal	Inadequate
-All user stories were prioritized -Evidence of extensive Product Champion participation	-All user stories were prioritized -Evidence of some Product Champion involvement	-Most user stories were prioritized -Evidence of some Product Champion input	-Many user stories were not prioritized -Limited contact with Product Champion	-No prioritization of user stories -No evidence of contact with Product Champion

SPRINT PLANNING [15% OF PART]

Excellent	Good	Adequate	Marginal	Inadequate
-Tasks incorporate all required work (both implementation and process) -Tasks are of suitable size -Tasks estimated and tasks estimates are reasonable and consistent relative to other tasks -Task estimates in hours -Sprint planning meeting well-documented in the meeting minutes	-Tasks incorporate all required work, but may not include all process tasks -Tasks are of reasonable size, but could be further decomposed -Tasks estimated and tasks estimates are reasonable and largely consistent relative to other tasks -Task estimates in hours -Sprint planning meeting documented in the meeting minutes	-Tasks cover most required work -Tasks may span more than a few days work -Tasks would benefit from further decomposition -Tasks estimated and tasks estimates are generally reasonable -Task estimates in hours -Sprint planning meeting documented in meeting meetings	-Tasks cover only some required work -Tasks may be excessively coarse -Tasks estimates are provided but may be unreasonably or inconsistent with other tasks -Task estimated are provided but may not be in hours -Documentation of sprint planning meeting may be lacking	-Little evidence of proper task construction -Task estimates may not be provided -Tasks if present may be highly coarse

RELEASE PLANNING [10% OF PART]

Excellent	Good	Adequate	Marginal	Inadequate
-User stories allocated to both Sprint 1 and later sprints -Clear justification provided as to how stories were allocated between sprints	-User stories allocated to both Sprint 1 and later sprints -Some justification provided as to how stories were allocated between sprints	-User stories allocated to both Sprint 1 and later sprints, however later sprints allocation may only be evident from comparing user story backlog to Sprint 1 plans -Allocation of stories between sprints may not be clearly justified	-User stories allocated to only to Sprint 1	-User stories allocated to only to Sprint 1 -May be no or insufficient evidence of consideration of work to be completed in later sprints

SCRUM BOARD [15% OF PART]

Excellent	Good	Adequate	Marginal	Inadequate
-Scrum board consistently used to track and monitor tasks/stories -Scrum board assigns all tasks to group members	-Scrum board usually used to track and monitor tasks/stories-Scrum board assigns most tasks to group members	-Scrum board sometimes used to track and monitor tasks/stories -Scrum board assigns some tasks to group members	-Scrum board rarely used to track and monitor tasks/stories -Tasks often not assigned to specified group members	-Scrum board not used to track and monitor tasks/stories

BURN DOWN CHART [10% OF PART]

Excellent	Good	Adequate	Marginal	Inadequate
-Chart includes planned work and actual work, each clearly labelled -All axes labelled -Scale uses appropriate units and units are clearly indicated -Chart is professionally presented and easy to interpret -Estimated and actual velocity calculated	-Chart shows planned work and actual work, however labelling may be unclear -Axes may be missing labels -Scale uses appropriate units; units are indicated or may be inferred from context -May be some minor issues with chart readability or presentation -Estimated and actual velocity calculated	-Chart shows planned work and actual work which may be distinguished from context, but are unlabelled -Axes may be missing labels -Scale may have some issues with interpretability -May be issues with chart readability or presentation -Estimated and actual velocity, but one or both may have a computation issue	-Planned work and/or actual work series are not clearly distinguished; one or more series may be missing entirely -Axes may be missing labels -Scales may not be indicated or are marked incorrectly -Chart has significant issues with readability or presentation -One or both of estimated and actual velocity may be omitted	-Burn down chart is not produced or fails to include required elements -Chart has issues with readability or presentation that cause it to be difficult or impossible to interpret

MEETING MINUTES AND ATTENDANCE SPREADSHEET [20% 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

SPECIFICATION FOR NEXT SPRINT [10% OF PART]

Excellent	Good	Adequate	Marginal	Inadequate
-User stories for next sprint identified; selection is clearly justified -Allocation of stories takes into account velocity in the present spent and the velocity for the next sprint -An ordering is provided over the selected stories or there is a contingency plan that explains how stories taking more or less time than expected will be addressed -Success metrics for next milestone identified; goals are specific, measurable, achievable and realistic -Plan is clearly explained to a high standard -Specification is of appropriate length and is well-written	-User stories for next sprint identified; selection is largely justified -Allocation of stories takes into account velocity in the present spent and the velocity for the next sprint -At least some suggestion of a contingency plan for over/under estimation or there is a clear global ordering of user stories (including those not allocated to this sprint) -Success metrics for next milestone identified; most goals are largely all specific, measurable, achievable and realistic -Plan is comprehensible -Specification is of appropriate length -Any writing issues are minor and largely mechanical	-User stories for next sprint identified; justification may be somewhat lacking -Allocation of stories may not take into account velocity in the present spent and/or the velocity for the next sprint -Contingency plan for over/under estimation may be lacking and is not saved by a clear ordering of user stories -Plan is understandable with some effort; however some gaps may exist in the explanations -Success metrics for next milestone identified; however those goals may not be specific, measurable, achievable and realistic -Specification may be somewhat brief / overly long -May have some issues with writing	-User stories allocated to next sprint, but there is no/deficient justification -Allocation of stories may not take into account velocity in the present spent and/or the velocity for the next sprint -Contingency plan for over/under estimation may be lacking and is not saved by a clear ordering of user stories -Plan is poorly explained and may be confusing to a reviewer -Writing may suffer from issues that impair comprehensibility	-Plan is inappropriate or not evident -May have significant writing issues the severely impair comprehensibility

TESTING: TEST CONTENT [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 the automation and manual test tables. In addition, regardless of the method of testing, testing process will always be considered.

Testing process

Excellent	Good	Adequate	Marginal	Inadequate
-Tests provided for all user stories -All tests implemented and checked-in to repository before corresponding product code implemented	-Tests provided for all user stories -Tests implemented and checked-in to repository before corresponding product code implemented with very few exceptions	-Tests provided for all user stories -Most tests implemented and checked-in to repository before corresponding product code implemented	-Tests provided for most user stories -Many tests checked-in after the corresponding product code	-Tests not provided for many user stories -Many tests, if implemented, checked-in after the corresponding product code

Automation Test

Excellent	Good	Adequate	Marginal	Inadequate
-Demonstrates a mastery of unit and integration testing -Uses a test suite to run multiple test cases -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 to run multiple test cases -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 to run multiple test cases -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 to run multiple test cases -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

(element continued on next page)

5

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

IMPLEMENTATION: DESIGN [20% 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 – developer notes, diagrams, and documented as used 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 – developer notes, diagrams, and documented used as appropriate	-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: CODING [20% 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 -Evidence code was written by all group members	-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 -Evidence code was written by all group members	-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 -Evidence code was written by all group members	-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 -Code was NOT written by all group members	-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 -Code was written by only one or two group members

DESIGN AND TECHNOLOGY MANIFESTO [20% OF PART]

Excellent	Good	Adequate	Marginal	Inadequate
-Software structure, decomposition, data model and related elements of design are clearly explained to a high standard and decisions are explained and well-justified -Diagrams are employed where appropriate and effectively illustrate design -Technologies (software tools, frameworks, APIs) used are clearly stated and decisions made about how to use them are well justified and reasoning is explained -Technologies are applied in a fashion suitable to the problem to be solved -Decisions are clearly justified and reasoning is explained to a high standard -Decision making process is well documented -Manifesto is of appropriate length and is well-written	-Software structure, decomposition, data model and related elements of design are clearly explained to a high standard and decisions are explained and justified, though some justification may be lacking -Diagrams may be employed where appropriate to illustrate design -Technologies used are clearly stated and decisions regarding use of same are explained -Technologies are applied in a fashion suitable to the problem to be solved -Manifesto is of appropriate length -Decisions and the process to arrive at them are comprehensible and justified -Any writing issues are minor and largely mechanical	-Software structure, decomposition, data model and related elements of design are explained -Technologies used are stated and decisions regarding use of same are explained -Diagrams may be employed but are not effective or should have been used more extensively -Technologies are applied to address the problem to be solved, however readily available superior alternatives approaches could have been discovered with a minimum of research / effort -Decisions are understandable with some effort; however some gaps may exist in the explanations -Manifesto may be somewhat brief / overly long -May have some issues with writing	-Software structure, decomposition, data model and related elements of design are explained but the explanation is incomplete -Technologies used are stated but decisions about how to use them may require further detail regarding reasons -Diagrams are sometimes used inappropriately or are not used where they would have benefitted the manifesto -Technologies selected are applied in a way that addresses the problem to be solved only tangentially -Decisions are poorly justified and reasoning is explained and may be confusing to a reviewer -Writing may suffer from issues that impair comprehensibility	-Software structure, decomposition, data model and related elements of design are not explained to an adequate level -Implementation technologies are not identified or are applied to the problem in a way that, to a person of ordinary skill, is clearly inappropriate for the problem -Decisions made are inappropriate or not evident -May have significant writing issues the severely impair comprehensibility