

ITC515 - Professional Programming Practice

Table of Contents

Subject Summary	2
Subject Coordinator	2
Subject Coordinator	2
Email	2
Phone	2
Consultation procedures	2
Subject Overview	2
Abstract	2
Learning outcomes	3
Subject content	3
Key subjects	3
Assumed knowledge	3
Subject Schedule & Delivery	3
Prescribed text	3
Class/tutorial times and location	4
Schedule	4
Learning materials	5
Learning, teaching and support strategies	5
Recommended student time commitment	9
Assessment Items	10
Essential requirements to pass this subject	10
Items	10
Teamwork - Version Control	11
Teamwork - Version Control - Static Review	17
Dynamic Testing	31
Debugging	41
Assessment Information	44
Academic integrity	44
Referencing	44
How to submit your assessment items	45
Online submission process	45
Postal submission process	45
Hand delivered submission process	45
Alternative submission process	45
Extensions	45
How to apply for special consideration	46
Penalties for late submission	46
Resubmission	47
Feedback processes	47
Assessment return	47
Student Feedback & Learning Analytics	47

Evaluation of subjects	47
Changes and actions based on student feedback	47
Learning analytics	47
Services & Support	48
Develop your study skills	48
Library Services	48
Policies & Regulations	48
Subject Outline as a reference document	48

Subject Summary

ITC515 - Professional Programming Practice
 Session 2 2020
 Faculty of Business, Justice and Behavioural Sciences
 School of Computing and Mathematics
 Internal Mode
 Credit Points 8

Welcome to a new session of study at Charles Sturt University. Please refer to the University's [Acknowledgement of Country \(http://student.csu.edu.au/study/acknowledgement-of-country\)](http://student.csu.edu.au/study/acknowledgement-of-country).

Subject Coordinator

Subject Coordinator Dr Sudath Heiyanthuduwa Heiyanthuduwa
Email SHeiyanthuduwa@studygroup.com
Phone 02 9291 9325

Consultation procedures

Your subject coordinator will be available for consultation. You will be informed of the consultation procedures via your Interact2 subject site; as well as the names, contact details and consultation procedures for any other staff teaching the subject.

Subject Overview

Abstract

This subject explores and analyses the four main aspects of professional programming practice, namely: teamwork, version control, testing, and debugging. The subject establishes key concepts and techniques related to the production of quality software.

This subject explores teamwork and the issues surrounding online teams; the application of version control and testing techniques and best practices; and it analyses the principles and

application of debugging.

Learning outcomes

Upon successful completion of this subject, students should:

- be able to demonstrate autonomy, expert judgement, adaptability, and responsibility in applying specialised and technical knowledge and skills in the use of version control systems as part of professional programming practice;
- be able to synthesise complex information regarding the purpose and limitations of software testing in order to generate, design and implement an appropriate suite of software tests to support the complete system development life cycle;
- be able to critically analyse software programs and apply theories of debugging together with specialised and technical skills to generate solutions to complex debugging problems;
- be able to work effectively in a team environment.

Subject content

This subject covers the general topic areas of teamwork, version control, testing, and debugging.

Key subjects

Passing a key subject is one of the indicators of satisfactory academic progress through your course. You must pass the key subjects in your course at no more than two attempts. The first time you fail a key subject you will be at risk of exclusion; if you fail a second time you will be excluded from the course.

The [Academic Progress Policy \(https://policy.csu.edu.au/view.current.php?id=00250\)](https://policy.csu.edu.au/view.current.php?id=00250) sets out the requirements and procedures for satisfactory academic progress, for the exclusion of students who fail to progress satisfactorily and for the termination of enrolment for students who fail to complete in the maximum allowed time.

Assumed knowledge

ITC538 or ITC412

Subject Schedule & Delivery

Prescribed text

The textbooks required for each of your enrolled subjects can also be found via the Student Portal [Textbooks \(http://student.csu.edu.au/study/study-essentials/textbooks\)](http://student.csu.edu.au/study/study-essentials/textbooks) page. Because of the breadth of material covered by the subject, there is no prescribed text.

However, there are a range of online resources available, which are referenced in the subject modules.

Class/tutorial times and location

Your class times can be found at [Timetable @ CSU \(http://timetable.csu.edu.au/\)](http://timetable.csu.edu.au/). Find out how to use Timetable @ CSU via the Student Portal [Class Timetable \(http://student.csu.edu.au/study/study-essentials/timetable\)](http://student.csu.edu.au/study/study-essentials/timetable) page.

Schedule

Session Week	Week Commencing	Topics	Learning activities	Assessment
1	20 July 2020	Introduction Teamwork	Team Formation, Team Charter, Team Wiki	
2	27 July 2020	Version Control - Introduction	Individual Version Control Practice	Assignment 1 - Due: 31 July 2020
3	03 August 2020	Version Control - Merging	Collaborative Version Control Practice	Census date: 9 August 2020
4	10 August 2020	Principles of Testing	Master Test Plan	
5	17 August 2020	Static Testing	Static Code Review	
6	24 August 2020	Dynamic Testing	Unit Testing	Assignment 2 - Due: 28 August 2020
	31 August 2020 - 04 September 2020	Mid-Session Break		
7	07 September 2020	Integration Testing	Integration Testing	
8	14 September 2020	User Acceptance Testing	User Acceptance Test Specification	
9	21 September 2020	Testing Review		Assignment 3 - Due: 25 September 2020
10	28 September 2020	Principles of Debugging	Debugging Techniques	

Session Week	Week Commencing	Topics	Learning activities	Assessment
11	05 October 2020	Debugging Practice	Debugging Techniques	
12	12 October 2020	Review/Contingency	Overall Subject Review	Assignment 4 - Due: 16 October 2020
	19 October 2020 - 30 October 2020	Final Examination Period - No Exam in the subject		

Please see below Additional/Replacement Class Requirements for this session: Classes running on the following Public Holidays should have a replacement class:

- Monday 5th October 2020 (Labour Day)

This study schedule has been devised to assist in your planning during the session and is intended as a guide only.

Attendance is required at all scheduled lectures and tutorials.

Learning materials

Details of learning materials that support your success in this subject can be found in the Interact2 Subject Site.

Learning, teaching and support strategies

WARNING - This is a 'learn by doing' subject. There is no exam, and the bulk of the subject workload is associated with the assignments. The university expects a workload of around 140 hours per subject, and in this subject you can expect most of that to be spent engaging with the assignments. The most important point to stress here is that because of the teamwork component, **the subject workload is regular and continuous. It is not acceptable to adopt an episodic work style** during the team based assessment.

While the lectures and topic material are important, it is more important to engage with the weekly exercises which are generally supported with a video 'HowTo'. The exercises are directly relevant to the assignments and scaffold important skills necessary to complete them.

If you haven't dealt with version control, and/or are unfamiliar with code testing frameworks, and/or have only had to debug on a casual or ad-hoc basis, you should not underestimate the time it takes to become familiar with these areas. They all have 'steep learning curves' and even though none of the areas covered are particularly complicated once you are familiar with them, there are plenty of opportunities to make time consuming errors as you gain that familiarity.

In this subject there are also lots of opportunities for you to engage with me, with your peers and with the subject. I will be holding 12 regular sessions where we can discuss content issues and assessment items. As part of your assessment in this subject you are required to work

alone for all the assessment items.

It is helpful to have a small task to complete early in the session, so that you have a focus in the subject from the very first day, and so I have set a small assessment item in week 4, as part of your assessment. If you don't complete the assessment, I may contact you by email to have a chat about study and if there are any issues that I can help with.

The study guide and/or subject materials for this subject have been written specifically to guide you through the sections (and questions) of the prescribed textbook relevant to each topic.

You should check the Interact Site at least weekly for postings, announcements, lecture information and other resources that will assist your studies or additional information and resources vital to your success in the subject.

Studying at university does not mean studying alone. Take advantage of collective wisdom and post your questions to the subject forum.

Use the subject schedule to plan your studies over the session.

Information on effective time management is available on the CSU Learning Support website via the following link: <http://student.csu.edu.au>

You can also contact an adviser through Student Central on the following number: 1800 275 278 (or +61 2 6933 7507 from outside Australia).

Academic learning support

Visit the learning support website for advice about assignment preparation, academic reading, and note-taking, referencing, and preparing for exams at <https://www.csu.edu.au/current-students/learning-resources/build-your-skills/academic-skills-help>

The Study Centres also offers a range of workshops specifically targeting your needs as an international student. These workshops run multiple times per week and build into the comprehensive Academic Skills Development Program that you should participate in. Additionally, a number of student volunteers are available to assist you in a program known as Discipline Support Sessions. Please see the timetables for these programs on the noticeboards on campus and also via the iLearn Interact2 Organisation site.

You may also contact:

- Elaine Rodrigues
Study Support Manager
Phone: 02 8055 3413
E-mail: ERodrigues@studygroup.com (<mailto:ERodrigues@studygroup.com>)
- Mazin Yousif

Senior Study Support Coordinator

Phone: 02 9291 9361

E-mail: MYousif@studygroup.com (<mailto:MYousif@studygroup.com>)

- Miranda Alagich
Study Support Coordinator
Phone: 02 9291 9360
E-mail: MAlagich@studygroup.com (<mailto:MAlagich@studygroup.com>)

For appointments, please see Reception.

Queries regarding the content of this subject should be directed to your Subject Lecturer.

Library services

The Library supports your learning journey by providing the following:

- Library orientation, database searching and information literacy workshops run during the semester
- Qualified Information professionals to assist with all your information needs.

For 24/7 access, go to **CSU Library online**

(<https://library.csu.edu.au/>) <http://student.csu.edu.au/library> (<http://student.csu.edu.au/library>)

The CSU Library provides access to online resources. These are:

- Peer-reviewed journal articles
- eBooks
- Company and government reports
- eJournals
- Dissertations & theses
- Newspapers including Business & Financial newspapers in [Factiva](#) (see Business & IT Journal databases)
- Other Reference resources (eg. Australian Bureau of Statistics, Australian standards, online encyclopaedias & dictionaries to be read on the computer)
- Online assistance via free call on 1800 808 369, or 'Ask a Librarian' - Live Chat or Web Form.

Other CSU Library services and resources:

<https://student.csu.edu.au/library/study-research> (<https://student.csu.edu.au/library/study-research>)

- Video tutorials in research skills, finding journal articles for assignments, topic analysis
- Endnote referencing software
- Other online library services to help you successfully complete your assignments

Online Tutorials

<http://student.csu.edu.au/library/study-research/training-tutorials-videos>
(<http://student.csu.edu.au/library/study-research/training-tutorials-videos>)

Learn how to:

- Use Primo Search to find eReserve study materials and journal articles
- Search journal databases and web resources for information for your assessment tasks
- Identify appropriate sources of information and peer reviewed material, to evaluate resources

Subject Library Resource Guides

<http://libguides.csu.edu.au/> (<http://libguides.csu.edu.au/>)

Subject Library Guides are a great way to get started with research. Each online guide is tailored to a specific area of study, including Accounting, Business and Information Technology outlining how to research in your area and where to look for information.

Library Help

<http://student.csu.edu.au/library/help-contacts> (<http://student.csu.edu.au/library/help-contacts>)

Friendly and quick assistance is available. Ask for help finding information and navigating the library's extensive eResources.

National Library of Australia - Trove database

<http://trove.nla.gov.au/> (<http://trove.nla.gov.au/>)

The Trove database from the National Library of Australia provides access to many different online resources on any subject.

University of Technology Sydney - Blake Library

<http://find.lib.uts.edu.au/> (<http://find.lib.uts.edu.au/>)

All students with a CSU student card can access University of Technology, Sydney (UTS) library for free 10 times per year or paying \$50 per year to join CAUL scheme to borrow books and use the UTS library within the guidelines set down by UTS. Access restrictions may be in place during exam periods. The UTS Blake library is located near Chinatown: corner Quay Street & Ultimo Road, Haymarket. Phone: 02 9514 3666.

Contact Details below for any student enquiries:

CSU Study Centre Darlinghurst Library

<http://tinyurl.com/sgprimo> (<http://tinyurl.com/sgprimo>)

Ms Mireille Eid

Library Manager

Phone: 02 9291 9326

Email: Meid@studygroup.com (<mailto:Meid@studygroup.com>)

Ms Angie Baho
Library Officer
Phone: 02 9291 9315
Email: darlibrary@studygroup.com (<mailto:darlibrary@studygroup.com>)

Recommended student time commitment

CSU Academic Senate policy states that a standard 8 point subject should require you to spend a total of 140-160 hours engaged in the learning and teaching activities (10-12 hours per week for 14 weeks). This subject complies with this policy. The actual amount of work is not excessive in any one week, **unless** you don't work regularly.

An example weekly study schedule during an 'active teaching' week might look like this:

- 3 hours attending the weekly lecture or watching the CSUReplay of the internal lecture
- 2 hours attending the weekly practical or completing the weekly exercise
- 1-2 hours engaging with the material presented on the subject website, reading the recommended articles or otherwise engaging in self-directed study.
- 1 hour interacting with teammates in a team meeting and at other times
- 3-4 hours completing the relevant assignment work for the week

WARNING - You should expect each major assignment to take around 30 hours of work to complete. Because of the team based nature of these assignments that workload should be spread out over several weeks. The important points here are that:

1. In practice it is **physically impossible** to complete the assignments in the week they are due.
2. In a team situation it is **unacceptable** not to complete assigned work items on a consistent regular basis.
3. If you don't do the weekly exercise, you should add at least the same amount of time on to the time you plan to spend doing the assignment.

Assessment Items

Essential requirements to pass this subject

You must obtain at least 50% in the total mark in order to pass this subject.

If you marginally fail these pass criteria you are entitled to a second chance in the following circumstances:

You are eligible for an Additional Assignment (AA) if you submitted all assessment items, but received an overall mark between 45 and 49;

Items

Item No.	Title	Value	Due Date*	Return Date**
1	Teamwork - Version Control	SY/US	31-Jul-2020	24-Aug-2020
2	Teamwork - Version Control - Static Review	40%	28-Aug-2020	18-Sep-2020
3	Dynamic Testing	40%	25-Sep-2020	20-Oct-2020
4	Debugging	20%	16-Oct-2020	09-Nov-2020

* Due date is the last date for assessment items to be received at the University

** Applies only to assessment items submitted by the due date

Assessment item 1 - Teamwork - Version Control

Value: Satisfactory/Unsatisfactory

Due Date: 31-Jul-2020

Return Date: 24-Aug-2020

Submission method options: Alternative submission method

TASK

The engagement task consists of the initial tasks required to support Assignment 2. The engagement task uses a Satisfactory/Unsatisfactory rating system because Assignment 2 depends critically on students collaborating online. It is essential that all students engage promptly with the subject.

A rating of Unsatisfactory can cause a student to fail the subject. Any student who receives an 'Unsatisfactory' rating in the engagement task will be required to immediately **show cause** why they should be allowed to continue, because allowing them to do so may affect other student's outcomes. If cause to continue cannot be established within one or two days of the due date for this assessment item, then the Unsatisfactory rating will stand and the student will **fail the subject**. This will occur before the HECS census date so that students may withdraw without penalty.

It is the **student's responsibility** to contact the lecturer and show cause why they should continue in the subject if they receive an Unsatisfactory rating for the engagement task.

Specifically, the steps that must be completed for the Engagement Task consist of:

1. **Organize a meeting** with your assigned team members. You will be notified of your team members and allocated an online meeting room URL by your subject coordinator. You will also be provided access to your team's wiki via the subject Interact site.
2. **Get together online.** Establish a 'team charter' setting out the team's expectations (rights and responsibilities) of each team member.
3. **Organise a schedule of weekly meetings** with your team members, and post the schedule in your team wiki.
4. **Establish a 'push' communication mechanism** for rapid communication between team members and post a link to that mechanism in your team wiki.
5. **Allocate tasks for the first iteration.** You must use the provided breakdown of tasks for Assignment 2. However, you must decide on and record the task completion criteria. Record this task assignment and completion criteria in an iteration plan (template provided), and post that iteration plan to your team wiki. **Please Note:** there are **two** iterations required to complete Assignment 2. Do not present an 'Iteration Plan' that covers the whole period for Assignment 2. Present an iteration plan that specifies what you are going to achieve in the first iteration.
6. **Record meeting minutes**, and post them in your team wiki.
7. **Establish a team version control repository.** Download the provided code and establish a code baseline under version control. Grant the lecturer read access to that repository.
8. **Create an individual 'feature branch' for each team member.**

Deliverables:

The deliverables for Assignment 1 consist of the following:

1. **A team wiki.** This should contain:
 1. Your **team charter**.
 2. The **minutes** for the first weekly team meeting.
 3. The **iteration plan** with completion criteria for work items specified.
 4. The **link/URL** for the combined team version control repository
 5. The **link/URL** to the push communication mechanism you select.
2. The **combined team version control repository**. This should initially contain the provided code baseline, and eventually contain the results of merging each team members separate development.
3. Your **individual feature branch**. This should initially contain the code baseline as branched from the team repository. Eventually it should contain a record of the successive commits you make during development of your 'feature', and finally the results of updating your branch from the central repository prior to code review and merge. **Do not perform a 'squash merge', and do not delete your feature branches following a merge back to master.** Create new feature branches as required for later work, and retain all branches and commits in the repository.

In essence - get together with your team, establish and document some ground rules, plan the first iteration's work and set up a communication mechanism, set up version control with a code baseline, and create your own feature branch. IE - get ready to implement Assignment 2. See Assignment 2 for specific details of these steps.

Guidance:

Distributed teamwork is **difficult and inconvenient**, and the essence of teamwork is **commitment and compromise**. You **must** make yourself available to participate in team meetings at a time that is generally acceptable to a majority of team members, **even if it is somewhat inconvenient to yourself**. Negotiation is acceptable. Rigid insistence on personal preference is not. Make it happen. Use weekends if necessary. If you can't make it to a generally favoured time be prepared to explain your objections with substantive reasons. Reasons such as a watching favourite television shows, spectating sporting events, or preferring to go out at that time are not acceptable.

Note: Because Assignment 2 is based on group work, there will be **no latitude or leniency shown** to students who hold this process up. Your initial meeting should be held within the second week of semester, with a hard deadline for the whole engagement task of the end of Week 3.

Important Note: It is **highly desirable** that you **complete the engagement task within Week 2**, because that will allow you to run fortnight long iterations for Assignment 2. If you don't complete the engagement task until Week 3, your Assignment 2 iterations will need to be truncated to 10 days each, which makes scheduling meetings and collaborative tasks much more difficult.

If, after a reasonable attempt has been made to reach consensus on a mutually convenient meeting time, no agreement can be reached, then it is acceptable for the majority of the team to agree to a meeting time and proceed without the dissenting team member. In this case the

lecturer must be notified by both the majority and the dissenting team member through email before the end of Week 3. Both sides must supply substantive reasons why they cannot reach agreement. Depending on the reasons supplied, it is possible that a grade of Unsatisfactory will be recorded for the Engagement Task for either or both sides.

RATIONALE

SUBJECT LEARNING OUTCOMES

This assessment task will assess the following learning outcome/s:

- be able to demonstrate autonomy, expert judgement, adaptability, and responsibility in applying specialised and technical knowledge and skills in the use of version control systems as part of professional programming practice.
- be able to work effectively in a team environment.

The intent of the assignment is to ensure that students engage immediately with the subject, and to initiate the group assignment in a timely manner.

GRADUATE LEARNING OUTCOMES

This task also contributes to the assessment of the following [CSU Graduate Learning Outcome/s](https://student.csu.edu.au/study/glo) (<https://student.csu.edu.au/study/glo>):

- Professional Practice (Skill) - CSU Graduates demonstrate discipline-specific technical capabilities and self-appraisal required for a beginning practitioner or professional.

MARKING CRITERIA AND STANDARDS

This assignment is rated satisfactory/unsatisfactory on the basis of whether students have successfully completed the first few steps of Assignment 2.

Criteria	Satisfactory	Unsatisfactory
How well does the submission demonstrate the ability to work effectively in a team environment, and to effectively utilize version control?	<p>A team meeting has been held and minutes for that meeting have been published in the team wiki</p> <p>The individual team member attended the team meeting and is recorded as doing so in the meeting minutes.</p> <p>A team charter specifying workload work quality and communication expectations has been created and published in the team wiki.</p> <p>A schedule of iteration meetings has been established and published in the team wiki.</p> <p>A 'push' communication mechanism has been established.</p> <p>An initial iteration plan identifying tasks for each team member together with objective outcomes for each task has been created and published in the team wiki.</p> <p>A version control repository has been established, and the code baseline has been</p>	<p>A team meeting has been not been held or minutes for that meeting have not been published in the team wiki.</p> <p>The individual team member did not attend the team meeting or is not recorded as doing so in the meeting minutes.</p> <p>A team charter specifying workload work quality and communication expectations has not been created or not been published in the team wiki.</p> <p>A schedule of iteration meetings has not been established or not been published in the team wiki.</p> <p>A 'push' communication mechanism has not been established.</p> <p>An initial iteration plan specifying tasks for each team member, together with objective completion criteria for each task has not been created or not been published in the team wiki.</p>

Criteria	Satisfactory	Unsatisfactory
	<p>committed to that repository.</p> <p>Each team member has created a feature branch.</p> <p>The lecturer has been granted access to both the version control repository.</p>	<p>A version control repository has not been established, or the code baseline has not been committed to that repository.</p> <p>A team member has not created a feature branch.</p> <p>The lecturer has not been granted access to both the version control repository.</p>

All students whose performance in the engagement task is rated 'unsatisfactory' will be required to **show cause why they should be allowed to continue** in the subject. If this **cannot be established** within a week of the due date for this assessment item, then the **student should withdraw immediately** since they will be deemed to have **already failed the subject**.

It is the student's responsibility to contact the lecturer and show cause why they should be allowed to continue if they receive an unsatisfactory rating for this assessment item.

PRESENTATION

Assignment 1 should be presented as a summary document posted in TURNITIN*. The document should be prepared as a team, and then each team member should post the document separately in TURNITIN. I.E. **Everyone** should post the completed summary document in TURNITIN. The document should contain:

1. A link to your team wiki.
2. Either a copy of your team charter or a link to the online version of your team charter in your team wiki.
3. A link to your team's version control repository.
4. A link to the teams push communication mechanism.
5. Either a copy of the minutes for the first team meeting or a link to an online version of those minutes in your team wiki.
6. Either a copy of the project plan or a link to an online version of that plan in your team wiki.

As outlined above, all the deliverables (i.e. team charter, meeting minutes, and project plan) should also be accessible from your team wiki.

The master branch of the team version control repository should initially contain the provided code baseline.

*If you are required to submit through TURNITIN, insert all links, and concatenate all documents into a single Word or PDF file and submit that.

Assessment item 2 - Teamwork - Version Control - Static Review

Value: 40%

Due Date: 28-Aug-2020

Return Date: 18-Sep-2020

Submission method options: Alternative submission method

TASK

Conceptually, Assignment 2 is straightforward:

1. as a team, carry out a 'development' task using best practice version control, merging different developer's contributions using a formal static code review process for each merge.
2. Reflect on the process you have just carried out.

In practice, there are a lot of processes to be carried out which require close cooperation, effective communication and coordination, and considerable attention to detail.

Distributed teamwork is **difficult and inconvenient**, and the essence of teamwork is **commitment and compromise**. You **must** make yourself available to participate in team meetings once a week at a time that is generally acceptable to a majority of team members, **even if it is somewhat inconvenient to yourself**. Negotiation is acceptable. Rigid insistence on personal preference is not. Make it happen. Use weekends if necessary. If you can't make it to a generally favoured time be prepared to explain your objections with substantive reasons.

It is critical for this part of the assignment that iteration tasks and objective completion criteria are set at the planning and evaluation meeting that marks the end of each iteration. It is also critical that evidence to support any claim of work-item completion is placed under version control. If work items and completion criteria are not recorded, or evidence to support claims of work item completion is not placed under version control, then those work items will be considered not have been undertaken or completed.

It is your responsibility to undertake an equitable amount of work, make sure your assigned tasks are recorded in the iteration planning documents, and to record objective outcomes for your assigned tasks. It is also your responsibility to provide explicit, concrete evidence of work item completion in the form of commits to version control. No verbal assurances should be accepted.

Work also needs to be completed in a timely manner to prevent hold-ups and bottle necks during development. It is your responsibility to complete your tasks by the target dates, and to **communicate and collaborate effectively before the target date** if you encounter any issues.

To be explicit: it is **not sufficient** to just **do** the work, it must be **seen** to be done. The work must be planned, criteria for its completion set, the work evaluated against those criteria, and all of this must be recorded and evidenced in the iteration evaluation and in version control.

Note: This is a team-based assignment. However, it is **not the case** that more team members mean less work or that fewer team members mean more work. The coding task in this assignment is relatively trivial. The bulk of the work in the assignment is in coordinating and conducting the merge reviews. Fewer team members might mean more coding but the

coordination and review tasks are greatly simplified. More team members might mean less code work, but the coordination and review tasks are significantly more complicated. In the end, it comes out about even.

Further Note: This assignment uses the principle of ‘**evidence-based reporting**’. What this means is that is **your** responsibility to present evidence for the assignment marking criteria to the assessor, and not the assessor’s responsibility to find it. “That wasn’t asked for” is not an acceptable reason for not presenting evidence to support a claim to meeting the marking criteria. Don’t delete your branches, don’t squash your merges, record defects as issues, commit defect rework individually and identified with the relevant defect, keep and present communication logs as necessary, and take and submit screenshots and other artefacts to support your claims to meeting the marking criteria. Make sure the assessor has the appropriate access to check what they need to.

Task 1 – Collaborative Development (90 marks)

This task is intended to develop and assess your understanding and skills in regard to teamwork processes, version control practice, and static testing in the form of formal code reviews.

Assessment for Task 1 has 3 aspects: your performance as a member of a team, your performance in regard your individual and collaborative version control practice, and your performance as author, reviewer, and mediator during the static code reviews.

The actual code development aspect of Task 1 is divided into two iterations. Each iteration consists of a ‘scatter-gather’ cycle using the git ‘feature branch’ workflow as follows:

1. Tasks are assigned to developers during an iteration planning meeting. ‘Completion criteria’ are decided and recorded. Target completion dates are also decided and recorded with regard for any dependencies between tasks.
2. Each developer creates a ‘feature branch’ in which to carry out their development.
3. Each developer completes their development task in their feature branch.
4. Then, on completion of their tasks, and **in turn** after any previous merge process is complete, each developer issues a merge request, and their work is **formally reviewed** and incorporated into the master branch.
5. On completion of the ‘merge cycle’, the next iteration’s tasks are allocated, and the cycle is repeated.

Complete the collaborative development task using best practice team processes and behaviour. This involves:

1. Turning up to all meetings
2. Behaving appropriately
3. Taking on an equitable proportion of workload
4. Setting objective completion criteria for all work items. Ensuring work items are SMART.
5. Working continuously and consistently, demonstrating that through version control.
6. Communicating promptly on issues, with no surprises on target delivery dates
7. Completing work items to a satisfactory standard on time
8. Providing objective evidence of work item completion or progress in version control
9. Providing evidence of work item completion or progress in iteration reports
10. Providing thorough and objective explanations for non-completion of work items in iteration reports.

Complete your development tasks using individual best practice version control. This involves:

1. Frequent small 'atomic' commits – “commit early, commit often”, “commit related changes”
2. Meaningful commit messages
3. Commit working code
4. Don't commit personal files (i.e. your individual project configuration files)
5. Don't commit generated files such as binaries
6. Follow agreed team conventions (i.e. development directory structure etc)

Complete merges using best practice merge protocols. This involves:

1. Coordinating the merge cycle with team members before issuing the merge request.
2. Updating your feature branch from master (and deconflicting and debugging it) before issuing the merge request.
3. Submitting the updated, deconflicted and debugged final code for **formal review** with a merge request.
4. Waiting for the results of the review and then carrying out any rework promptly
5. Addressing each defect with an individual commit identifying the defect in the commit message
6. Ensuring that the master branch executes correctly both before and after the completed merge.
7. Use a new feature branch following the merge for any further work.

In practice, the 'merge cycle' is a highly coordinated process which requires close cooperation, timely communication and response, and rapid execution of each stage of the review process.

Since merges must be carried out sequentially, later merges are highly dependent on the prompt and efficient execution of previous merges, and the whole cycle takes a long time to complete. Therefore, it is highly desirable that 'development' tasks are completed in the early part of the iteration, leaving plenty of time to conduct an orderly and effective set of merge reviews.

The formal static code review is conducted as part of the merge process. The code to be reviewed consists of all the files assigned to the developer for updating. The files to be reviewed should be stated in the merge request. The objective of the review is to ensure that the submitted code meets all coding standards as provided in the assignment resource section of the subject website. Assessment for the code review aspect of the assignment depends on the final quality of the code you **review**, not only on the quality of the code you produce.

Complete the code review aspect of each merge using best practice static code review protocols. This involves:

As Author:

1. Wait your turn before issuing the merge request
2. Update your feature branch, and perform any required deconflicting and debugging before issuing your merge request
3. Respond promptly to any rework requests
4. Address each defect with an individual commit identifying the defect addressed
5. Ensure the master branch executes correctly following your merge.

As Reviewer:

1. Carry out the review task promptly and thoroughly
2. Restrict your review to the files identified for review, and the purpose of the review
3. Identify defects individually and provide an informative comment stating the reason for identifying the defect
4. Respond promptly to any notifications of rework completed

As Mediator:

1. Coordinate effectively with the mediator of any previous and following reviews
2. Ensure code meets entry criteria before initiating the review
3. Notify the author when it is their turn to perform their merge (and chase up if necessary)
4. Notify the reviewer when it is time to perform the review (and chase up if necessary)
5. Notify the author when the review is complete and it is time to perform rework or complete the merge
6. Verify that the master branch still operates correctly following completion of the merge.

The critical quality criteria for the merge and code review aspects of collaborative development is that the master branch should remain bug free and operational at all times – both before and after any merges. This is expressed most succinctly in the maxim:

“don’t break master”

Do not underestimate the difficulty and time required to review and integrate three or more separate developer’s code contributions. The submit/review/rework/approve cycle can take several days, so the entire integration process can potentially take well over a week. Aim to complete individual development well before the end of each iteration in order to leave sufficient time to complete the review process properly. Remember, the focus in this assignment is on teamwork, correct use of version control, and correct execution of the static code review process – not on the actual code development, which is quite trivial.

Code Review Roles

Author

The author’s role is to produce code to the required standards and to carry out any rework required.

It is the author’s responsibility to provide the required code at the required standard by the time the review is scheduled.

It is also the author’s responsibility to carry out any follow up work needed to address every recorded issue and submit the revised code to the mediator’s satisfaction in a timely manner. (Timely as defined by the team, not the author).

Most importantly, it is the author’s responsibility to merge approved code with the team repository and ensure that the main line of development works correctly following the merge.

Reviewer

The reviewer’s role is to review submitted code and ensure that it meets code quality guidelines.

It is the reviewer’s responsibility to review the submitted code and ensure it meets the

required standards, within the scheduled period. The reviewer must record every instance of non-compliance with standards and all detected bugs. A copy of relevant coding standards will be available on the subject web site.

It is also the reviewer's responsibility to promptly review any rework submitted in response to one of their comments, and to notify the mediator when rework has been approved

Mediator

The mediator's role is to organise and facilitate the code review.

It is the mediator's responsibility to ensure that the author delivers code for review by the time the review is scheduled, and that the reviewer carries out the review within the scheduled period. It is also the mediator's role to ensure that any follow up work required is carried out in a timely manner.

It is the mediator's role to clarify any misunderstandings and mediate any disagreements between reviewers and author, or between reviewers.

It is the mediator's responsibility to ensure that the main line of development is working as expected before notifying the author to update their branch with the results of the latest merge prior to the static code review process. It is also the mediator's responsibility to **verify** that the main line of code is working properly after the author has carried out the final merge.

Task 2 – Critical Review (10 marks)

This task is intended to develop and assess your ability to critically reflect and evaluate what you have learned regarding each of those learning outcomes, and your ability to integrate and synthesise that knowledge into plans for future actions. It is also intended to assess your ability to clearly and concisely communicate those thoughts.

On completion of the overall development task, write a review reflecting critically on the following aspects of the assignment:

1. How well did the team achieve the development task set for the assignment? What issues did the team encounter in achieving the outcome they did?
2. Team processes. What went well, what didn't? What issues regarding team processes did you encounter?
3. Version control. What did you learn about individual and collaborative version control practice? What issues regarding version control did you encounter?
4. Static code review. What did you learn about static testing? What issues regarding static code review did you encounter?

RATIONALE

SUBJECT LEARNING OUTCOMES

This assessment task will assess the following learning outcome/s:

- be able to demonstrate autonomy, expert judgement, adaptability, and responsibility in applying specialised and technical knowledge and skills in the use of version

- control systems as part of professional programming practice.
- be able to synthesise complex information regarding the purpose and limitations of software testing in order to generate, design and implement an appropriate suite of software tests to support the complete system development life cycle.
- be able to work effectively in a team environment.

This collaborative development task is intended to develop and assess your understanding and skills in regard to teamwork processes, version control practice, and static testing in the form of formal code reviews.

The reflective practice task is intended to develop and assess your ability to critically reflect and evaluate what you have learned regarding each of the learning outcomes, and your ability to integrate and synthesise that knowledge into plans for future actions. It is also intended to assess your ability to clearly and concisely communicate those thoughts.

GRADUATE LEARNING OUTCOMES

This task also contributes to the assessment of the following [CSU Graduate Learning Outcome/s \(https://student.csu.edu.au/study/glo\)](https://student.csu.edu.au/study/glo):

- Professional Practice (Skill) - CSU Graduates demonstrate discipline-specific technical capabilities and self-appraisal required for a beginning practitioner or professional.

MARKING CRITERIA AND STANDARDS

Collaborative Development Task (90 marks)

Teamwork (30 marks)

This aspect of the assessment measures your individual performance as a team member during the collaborative development task.

Criteria for effective teamwork:

1. Turned up to all meetings
2. Behaved appropriately
3. Took on an equitable proportion of workload
4. Set objective completion criteria for all work items. Work items SMART.
5. Worked continuously and consistently as evidenced in version control
6. Communicated promptly on issues, no surprises on target delivery dates, as evidenced in push communication mechanism
7. Completed work items to satisfactory standard on time
8. Objective evidence of work item completion or progress in version control
9. Recorded evidence of work item completion in iteration report
10. Any explanations for non-completion of work items thorough and objective as evidenced in iteration report

Teamwork	HD	DI	CR	PS
	25.5 - 30.0	22.5 - 25.4	19.5 - 22.4	15.0 - 19.4
How well did the student demonstrate the ability to work effectively in a team environment?	Completely achieved all criteria.	Completely achieved criteria 1, 2, 3, 5, 6, and 8 Achieved criteria 4, 7, 9, and 10 to a large extent.	Completely achieved criteria 2, 3, 5, and 8. Achieved criteria 1, 4, 6 and 7 to a large extent. Achieved criteria 9 and 10 to a partial extent.	Completely achieved criteria 2, 3, 5, and 8. Achieved criteria 1 and 7 to a large extent Achieved criteria 4, 6, 9

				and 10 to a partial extent.
--	--	--	--	-----------------------------

Version Control (30 marks)

This aspect of the assessment monitors and measures your individual performance using version control during the collaborative development task.

Criteria for best individual practice version control (15 marks):

1. Evidence for regular small commits. ("Commit early, commit often")
2. Commits are 'atomic'. ("Commit related changes", single intent commits)
3. Meaningful commit messages
4. Personal project setup/configuration files excluded
5. Generated files (binaries) excluded.

Version Control - Individual	HD	DI	CR	PS
	12.8 - 15.0	11.3 - 12.7	9.8 - 11.2	7.5 - 9.7
How well did the student apply knowledge and skills to effectively utilise version control in regard to independent practice?	Completely achieved all criteria.	<p>Completely achieved criteria 1</p> <p>Achieved criteria 2 and 3 to a large extent</p> <p>Achieved criteria 4 and 5 to a partial extent</p>	<p>Achieved criteria 1 to a large extent</p> <p>Achieved criteria 2 and 3 to a partial extent</p> <p>May not achieve criteria 4 or 5</p>	<p>Achieved criteria 1, 2, and 3 to a partial extent</p> <p>May not achieve criteria 4 or 5</p>

Criteria for best collaborative practice version control (15 marks):

1. Followed team conventions. (No unauthorised commits to master, worked in feature branch)

2. Coordinated merge request with team.
3. Updated feature branch with latest version of master before code review
4. Debugged/deconflicted in feature branch, changes incorporated in merge request
5. Ensured master operated correctly before and after merge.

Note: the key skills that you MUST achieve in order to pass the collaborative version control section are the skills to create a branch, work in that branch, and to use correct merge protocol.

Version Control - Collaborative	HD	DI	CR	PS
	12.8 - 15.0	11.3 - 12.7	9.8 - 11.2	7.5 - 9.7
How well did the student apply knowledge and skills to effectively utilise version control in regard to collaborative practice?	Completely achieved all criteria.	Completely achieved criteria 1, 3, and 5 Achieved criteria 2 and 4 to a large extent	Completely achieved criteria 1, and 3. Achieved criteria 2 to a large extent Achieved criteria 4 to a partial extent May not achieve criteria 5	Completely achieved criteria 3. Achieved criteria 1 to a large extent Achieved criteria 2 to a partial extent May not achieve criteria 4 or 5

Static Code Review (30 marks)

Author Criteria:

1. Code ready for review on time
2. Code meets entry criteria when submitted for review
3. Preparation for review coordinated with team (code updated from latest version of master)
4. Code rework carried out promptly and to an acceptable standard (no further rework required)

5. Defect resolutions committed independently and identified with defect.
6. Master branch operates correctly following merge.

Static Code Review - Author	HD	DI	CR	PS
	8.5 - 10	7.5 - 8.4	6.5 - 7.4	5.0 - 6.4
As author, how well did the student demonstrate understanding and skills related to teamwork and software testing in the context of a static code review?	Completely achieved all criteria	Completely achieved criteria 1, 2, 3, and 6 Achieved criteria 4 and 5 to a large extent	Completely achieved criteria 2, 3, and 6 Achieved criteria 1 to a large extent Achieved criteria 4 and 5 to a partial extent	Completely achieved criteria 2 and 3 Achieved criteria 6 after intervention Achieved criteria 1, 4, and 5 to a partial extent

Reviewer Criteria:

1. Code meets prescribed coding standards following review.
2. Review carried out promptly and thoroughly.
3. Review stayed on task. (Correct files reviewed, reviewed against objectives of review, no extraneous comments)
4. Defects identified individually and informatively. (Defect ID assigned, informative reason, defect externally recorded as issue.)
5. Rework reviewed promptly.

Static Code Review - Reviewer	HD	DI	CR	PS
	8.5 - 10	7.5 - 8.4	6.5 - 7.4	5.0 - 6.4
As reviewer, how well did the student demonstrate understanding and skills	Completely achieved all criteria	Completely achieved criteria 2, 3, 4, and 5	Completely achieved criteria 2, and 3	Achieved criteria 2 and 3 to a large extent

related to teamwork and software testing in the context of a static code review?		Achieved criteria 1 to a large extent	Achieved criteria 4 and 5 to a large extent Achieved criteria 1 to a partial extent	Achieved criteria 1, 4, and 5 to a partial extent
--	--	---------------------------------------	--	---

Mediator Criteria:

1. Review coordinated with previous and following reviews
2. Code submitted for review meets entry standards. (A cursory inspection does not reveal numerous obvious defects, and executes correctly)
3. Review effectively monitored and supervised. (Author and reviewer notified when actions required, lack of action chased up)
4. Master branch operates correctly following merge.

Static Code Review - Mediator	HD	DI	CR	PS
	8.5 - 10	7.5 - 8.4	6.5 - 7.4	5.0 - 6.4
As mediator, how well did the student demonstrate understanding and skills related to teamwork and software testing in the context of a static code review?	Completely achieved all criteria	Completely achieved criteria 1 and 4 Achieved criteria 2 and 3 to a large extent	Completely achieved criteria 4 Achieved criteria 1, 2, and 3 to a large extent.	Completely achieved criteria 4 Achieved criteria 1 to a large extent Achieved criteria 2 and 3 to a partial extent

Reflective Practice (10 marks)

Criteria:

1. Responses address all questions

2. Responses are comprehensive, objective, and insightful
3. Responses are consistent with an external view, and supported by available evidence
4. Responses go beyond description and display critical analysis.
5. Logical conclusions and implications for future practice are stated.
6. Analysis and conclusions drawn are logically justified and/or correct in terms of subject material
7. All content relates directly to critical reflection, analysis, and argument for conclusions reached
8. Responses are concise, relevant, easy to read, and easy to understand.

Reflective Practice	HD	DI	CR	PS
	8.5 - 10	7.5 - 8.4	6.5 - 7.4	5.0 - 6.4
How well does the reflective report demonstrate an ability to reflect on and critically evaluate issues encountered involving teamwork, version control, and static code review?	Completely achieved all criteria.	Completely achieved criteria 1, 2, 3 Achieved all other criteria to a large extent	Completely achieved criteria 1 and 3 Achieved criteria 2 to a large extent Achieved all other criteria to a partial extent	Achieved criteria 1, and 3 to a large extent Achieved criteria 2 and 4 to a partial extent May not achieve other criteria

PRESENTATION

There are four aspects of presentation for this assignment:

1. Evidence of teamwork.
2. Evidence of best practice version control
3. Evidence of best practice static code review.
4. Evidence of reflective practice

You should submit a summary document through TURNITIN* containing the following:

* - if you are required to submit through TURNITIN, concatenate all documents together in a single Word or PDF file and submit that.

Teamwork

Copies of and/or links to the iteration reports identifying and reporting on the work items associated with your contribution.

A summary of your contribution to the code base. I.E. identify which classes you developed and/or what coding tasks you performed.

Note: make sure the iteration reports contain links to the commits associated with work item completion.

Version control

Either:

An accessible link to your team's version control repository which allows the history of all merge reviews to be inspected, including all defects identified, and the exchanges involved in resolving those defects.

Or:

Copies of all files you were responsible for updating.

A copy of the complete commit history and branch structure of your version control repository (i.e. the network model of branches and commits).

A log of your commits to version control.

Static Code Review

Either:

An accessible link to your team's version control repository which allows the history of all merge reviews to be inspected, including all defects identified, and the exchanges involved in resolving those defects.

A summary list identifying the merges in which you participated and the role you

performed in those merges.

Or:

A log of the merge/pull requests in which you participated as author, including a list of defects identified, and all exchanges between reviewers and authors.

A log of the merge/pull requests in which you participated as reviewer, including a list of defects identified, and all exchanges between reviewers and authors.

A log of the merge/pull requests in which you participated as mediator, including all exchanges between mediator and other roles.

Reflective Practice

A document in which you individually reflect on the questions put in that section of the assignment.

Important Note: It is **your responsibility** to ensure that all evidence necessary to assess your assignment submission against the specified marking criteria is presented in or accessible from your summary document.

Assessment item 3 - Dynamic Testing

Value: 40%

Due Date: 25-Sep-2020

Return Date: 20-Oct-2020

Submission method options: Alternative submission method

TASK

Assignment 3 walks you through the full range of developer based dynamic software testing.

In Part A, you need to critically analyse the provided case study and design documents to determine the areas of highest product and technical risk. You then need to create a minimal 'Master Test Plan' in which you specify the amounts and types of testing you will apply to different aspects of the application.

In Part B, you are required to implement a full range of dynamic tests for a particular use case, ranging from low level unit tests, through at least two levels of integration testing, to system testing.

Part C asks you to engage in some critical reflection on the effectiveness and utility of different aspects of dynamic testing.

Please note that it is recognised that standard practice is to specify user acceptance tests before development activities occur, and to conduct unit testing as classes are initially coded. Unfortunately, the semester timetable makes this impractical.

Also note that the focus in this assignment is on the process and techniques applied, not the complexity of the code. It is recognised that the process applied is 'way over the top' for the size of problem addressed.

Finally, please note that the code supplied is deliberately NOT bug free. The purpose of your testing is to expose those bugs. You are specifically instructed NOT to modify the code to eliminate those bugs. You are ONLY to implement and report on the results of your testing.

Tasks

Part A – Master Test Plan (10 marks)

You are provided with a case study and a set of design documents and diagrams which support a use case. You are required to analyse the product and technical risks implied by these artefacts and develop a minimal 'Master Test Plan' which addresses these risks.

Quality criteria for a minimal master test plan include:

1. Provides an overview of the project objectives
2. Identifies testing objectives in terms of problem domain outcomes
3. Identifies areas of product and technical risk
4. Identifies a test strategy for each risk (test level at which risk is addressed, extent and coverage of testing, including static testing approaches)
5. For each test level, identifies the test approach (test environment, test goals, entry

criteria, exit criteria, defect response)

Part B – Dynamic Testing (80 marks)

You are provided with an implementation and a ‘master test plan’ which specifies the areas of concern, types of testing and coverage required for the application. You are required to implement a suite of dynamic tests to enact that plan using best practice testing techniques and procedures.

Unit Testing - 30 marks

Implement a suite of unit tests according to the master test plan using best practice unit test techniques and processes.

These include:

1. Tests are well organised and well named
2. Tests are well structured (Arrange, act, assert internal structure)
3. Tests are isolated (Tests only one code unit)
4. Tests are independent (Can be run in any order, don't depend on external services or state)
5. Tests for error conditions included
6. Tests achieve appropriate level of coverage (as specified in the master test plan)
7. Each test tests only one thing (No logic in test, in general one assert per test)
8. Tests are automated (Do not require visual inspection to differentiate passing or failing)
9. Makes effective use of a test framework and test substitute library as appropriate (such as Junit, and Mockito)
10. Tests separated from production code

Integration Testing - 30 marks

Implement a suite of integration tests according to the master test plan using best practice integration test techniques and processes.

Use a ‘bottom up’ or ‘top down’ approach to check interaction between entities and between control and entity classes. Using your ‘master test plan’ as a guide, make sure you test the most critical and/or risky system operation. You are not required to test boundary (user interaction) classes.

Guidelines for best practice integration testing include:

1. Tests are well organised and well named
2. Tests are well structured (Arrange, act, assert internal structure)
3. Tests are isolated to only those code units included in the intended level of integration.
4. Tests are independent (Can be run in any order, don't depend on external services or state)
5. Tests for error conditions included
6. Tests achieve appropriate level of coverage (as specified in the master test plan)
7. Each test tests only one thing (no logic in test, in general one assert per test)
8. Tests separated from production code
9. All levels of integration are addressed
10. Tests are automated (Do not require visual inspection to differentiate passing or

failing)

11. Makes effective use of a test framework and test substitute library as appropriate (such as Junit, and Mockito)

System Tests - 20 marks

On the basis of the requirement and design documentation identify some system test cases which can be used to check correct functionality in the implemented use case. Specify system test procedures, test data, and expected results. A template for a system test will be available in the Resources section of the subject's Interact site.

Carry out the system tests you specified. Use the test data and procedures specified and check that it meets stated acceptance criteria. Report on the success or otherwise of the test. Include screenshots as evidence of your results in your test reports.

System test quality criteria include:

1. Tests are well organised (test cases are organised into test scenarios addressing specific functional areas.
2. Test are well named
3. Preconditions are completely and concretely specified.
4. Test data are completely and concretely specified
5. Postconditions are completely and concretely specified
6. Test steps and expected results for each step are completely and concretely specified
7. Tests are objectively evaluated and supporting evidence for test results is provided

Part C – Reflective Practice (10 marks)

Write a reflective report in which you critically evaluate the following questions:

1. Which aspect of testing was the most effective? i.e. Which aspect of testing discovered the most defects for the effort expended?
2. How relevant and useful are the guidelines for good unit testing practices?
3. How easy is it to achieve comprehensive test coverage? What proportion of programming effort should be allocated to testing?
4. How effective is dynamic testing compared to static code review in terms of discovering defects?

RATIONALE

SUBJECT LEARNING OUTCOMES

This assessment task will assess the following learning outcome/s:

- be able to demonstrate autonomy, expert judgement, adaptability, and responsibility in applying specialised and technical knowledge and skills in the use of version control systems as part of professional programming practice.
- be able to synthesise complex information regarding the purpose and limitations of software testing in order to generate, design and implement an appropriate suite of software tests to support the complete system development life cycle.
- be able to work effectively in a team environment.

The intent of the assignment is to articulate taught concepts and skills in the context of

developing a suite of unit, integration, and user acceptance tests for a small application.

GRADUATE LEARNING OUTCOMES

This task also contributes to the assessment of the following [CSU Graduate Learning Outcome/s \(https://student.csu.edu.au/study/glo\)](https://student.csu.edu.au/study/glo):

- Professional Practice (Skill) - CSU Graduates demonstrate discipline-specific technical capabilities and self-appraisal required for a beginning practitioner or professional.

MARKING CRITERIA AND STANDARDS

Master Test Plan (10 marks)

Quality criteria:

1. Provides an overview of the project objectives
2. Identifies test objectives in terms of problem domain outcomes
3. Identifies areas of product and technical risk
4. Identifies a test strategy for each risk (test level at which risk is addressed, extent and coverage of testing, including static testing approaches))
5. For each test level, identifies the test approach (test environment, test goals, entry criteria, exit criteria, defect response)

Master Test Plan	HD	DI	CR	PS
	8.5 – 10.0	7.5 – 8.4	6.5 – 7.4	5.0 – 6.4
How well did the student demonstrate understanding and skills relating to software testing in the context of specifying a master test plan?	All criteria completely satisfied.	Completely achieved criteria 1, 2, and 3 Achieved criteria 4 and 5 to a large extent	Completely achieved criteria 1 Achieved criteria 2 and 3 to a large extent Achieved criteria 4 and 5 to a partial extent	Achieved criteria 1 and 2 to a large extent Achieved criteria 3, 4, and 5 to a partial extent

Unit Testing (20 marks)

Quality criteria:

1. Tests are well organised and well named
2. Tests are well structured (Arrange, act, assert internal structure)

3. Tests are isolated (Tests only one code unit)
4. Tests are independent (Can be run in any order, don't depend on external services or state)
5. Tests for error conditions included
6. Tests achieve appropriate level of coverage (as specified in the master test plan)
7. Each test tests only one thing (no logic in test, in general one assert per test)
8. Tests are automated (Do not require visual inspection to differentiate passing or failing)
9. Makes effective use of a test framework and test substitute library as appropriate (such as Junit, and Mockito)
10. Tests separated from production code

Unit Testing	HD	DI	CR	PS
	17.0 – 20.0	15.0 – 16.9	13.0 – 14.9	10.0 – 12.9
How well did the student demonstrate understanding and skills relating to software testing in the context of implementing a set of unit tests?	All criteria completely satisfied.	<p>Completely satisfied criteria 1, 2, 3, 4, and 5</p> <p>Satisfied criteria 6, 7, 8, 9, and 10 to a large extent</p>	<p>Completely satisfied criteria 1, 2, 3, and 4</p> <p>Satisfied criteria 5, 6, and 7 to a large extent</p> <p>Satisfied criteria 8 and 9 to a partial extent</p> <p>May not have satisfied criteria 10</p>	<p>Satisfied criteria 1, 2, 3, and 4 to a large extent</p> <p>Satisfied criteria 5, 6, and 7 to a partial extent</p> <p>May not have satisfied criteria 8, 9, and 10</p>

Integration Testing (20 marks)

Quality criteria:

1. Tests are well organised and well named
2. Tests are well structured (Arrange, act, assert internal structure)
3. Tests are isolated to only those code units included in the intended level of integration.

4. Tests are independent (Can be run in any order, don't depend on external services or state)
5. Tests for error conditions included
6. All levels of integration are addressed
7. Tests achieve appropriate level of coverage (as specified in the master test plan)
8. Each test tests only one thing (no logic in test, in general one assert per test)
9. Tests are automated (Do not require visual inspection to differentiate passing or failing)
10. Makes effective use of a test framework and test substitute library as appropriate (such as Junit, and Mockito)
11. Tests separated from production code

Integration Testing	HD	DI	CR	PS
	17.0 – 20.0	15.0 – 16.9	13.0 – 14.9	10.0 – 12.9
How well did the student demonstrate understanding and skills relating to software testing in the context of implementing a set of integration tests?	All criteria completely satisfied.	<p>Completely satisfied criteria 1, 2, 3, 4, 5, and 6</p> <p>Satisfied criteria 7, 8, 9, 10, and 11 to a large extent</p>	<p>Completely satisfied criteria 1, 2, 3, and 4</p> <p>Satisfied criteria 5, 6, 7, and 8 to a large extent</p> <p>Satisfied criteria 9 and 10 to a partial extent</p> <p>May not have satisfied criteria 11</p>	<p>Satisfied criteria 1, 2, 3, and 4 to a large extent</p> <p>Satisfied criteria 5, 6, 7, and 8 to a partial extent</p> <p>May not have satisfied criteria 9, 10, and 11</p>

System Tests - 20 marks

Quality criteria:

1. Tests are well organised (test cases are organised into test scenarios addressing specific functional areas.
2. Test are well named
3. Preconditions are completely and concretely specified.
4. Test data are completely and concretely specified

5. Postconditions are completely and concretely specified
6. Test steps and expected results for each step are completely and concretely specified
7. Tests are objectively evaluated and supporting evidence for test results is provided

System Tests	HD	DI	CR	PS
	17.0 – 20.0	15.0 – 16.9	13.0 – 14.9	10.0 – 12.9
How well did the student demonstrate understanding and skills relating to software testing in the context of implementing a set of system tests?	All criteria completely satisfied.	Completely satisfied criteria 1 and 2 Satisfied criteria 3, 4, 5, 6, and 7 to a large extent	Satisfied criteria 1 and 2 to a large extent Satisfied criteria 3, 4, 5, 6, and 7 to a partial extent	Satisfied all criteria to a partial extent

Reflective Practice (10 marks)

Quality Criteria:

1. Responses address all questions
2. Responses are comprehensive, objective, and insightful
3. Responses are consistent with an external view, and supported by available evidence
4. Responses go beyond description and display critical analysis.
5. Logical conclusions and implications for future practice are stated.
6. Analysis and conclusions drawn are logically justified and/or correct in terms of subject material
7. All content relates directly to critical reflection, analysis, and argument for conclusions reached
8. Responses are concise, relevant, easy to read, and easy to understand.

Reflective Practice	HD	DI	CR	PS
	8.5 - 10	7.5 - 8.4	6.5 - 7.4	5.0 - 6.4

How well does the reflective report demonstrate an ability to reflect on and critically evaluate issues encountered involving teamwork, version control, and static code review?	Completely achieved all criteria.	Completely achieved criteria 1, 2, 3 Achieved all other criteria to a large extent	Completely achieved criteria 1 and 3 Achieved criteria 2 to a large extent Achieved all other criteria to a partial extent	Achieved criteria 1, and 3 to a large extent Achieved criteria 2 and 4 to a partial extent May not achieve other criteria
--	-----------------------------------	---	--	---

PRESENTATION

You should submit a zip file through TURNITIN containing the following:

1. A document containing your Master Test Plan
2. Your Development Directory containing all source and test code
3. A summary document explaining where test files for each section of the assignment can be found
4. A document containing your responses to the reflective practice task.

If you are required to submit using TURNITIN, submit the following:

A Word or PDF document containing the concatenated contents of :

1. Your master plan
2. A screen shot showing the directory structure and contained files of your test development environment
3. The source code for all tests you developed, with each test file separately identified
4. Your responses to the reflective practice task.

Assessment item 4 - Debugging

Value: 20%

Due Date: 16-Oct-2020

Return Date: 09-Nov-2020

Submission method options: Alternative submission method

TASK

Assignment 4 revolves around debugging a small application.

You will be provided with some code which contains several known bugs. The source code and initial bug reports will be made available in the Resources section of the subject's Interact site.

Your task is to replicate, simplify, trace, and eliminate these bugs using best practice debugging techniques. Your debugging effort should take place within a version control system using appropriately sized, commented, and timely commits.

Please note that the focus in this assignment is on the process and techniques applied, not the complexity of the code. It is recognised that the process applied is 'way over the top' for the size of problem addressed.

Tasks

1.Replication.

1. For each of the reported bugs, generate a test case in UAT format that will reliably reproduce the buggy behaviour.
2. Commit the test case documentation to version control.

2.Simplification.

1. For each of the reported bugs, produce a simplified and automated test that reliably reproduces the bug.
2. Commit the test code demonstrating the bug to version control.
3. Record output from your test, demonstrating the buggy behaviour.

3.Tracing.

1. For each of the reported bugs, trace back through the program from the point of failure to find the origin of the bug. Use a sequence of hypotheses and tests to support this search, and document this sequence in a debugging log.
2. For each of the reported bugs, demonstrate that before the identified origin, variable values are 'sane' and that after the origin, at least one value is 'infected'.
3. Record screen shots demonstrating the origin of the bug.

4.Resolution.

1. For each of the reported bugs, correct the bug.
2. Commit the corrected code to version control.

3. Record output listings from your automated test demonstrating that the buggy behaviour has been eliminated.
4. Carry out the user test case with the specified data and show that the buggy behaviour has been eliminated.

RATIONALE

SUBJECT LEARNING OUTCOMES

This assessment task will assess the following learning outcome/s:

- be able to critically analyse software programs and apply theories of debugging together with specialised and technical skills to generate solutions to complex debugging problems.

GRADUATE LEARNING OUTCOMES

This task also contributes to the assessment of the following [CSU Graduate Learning Outcome/s \(https://student.csu.edu.au/study/glo\)](https://student.csu.edu.au/study/glo):

- Professional Practice (Skill) - CSU Graduates demonstrate discipline-specific technical capabilities and self-appraisal required for a beginning practitioner or professional.

MARKING CRITERIA AND STANDARDS

Debugging - 100 marks

Criteria:

1. Detailed and comprehensive description of the circumstances leading to the expression of the bug
2. Reliable automated test that demonstrates the bug and simplifies the circumstances in which it is expressed is created
3. Detailed and comprehensive log of the sequence of hypotheses and tests used to trace the origin of the bug is presented
4. Insightful and correct resolution of the bug with no side effects or other bugs created.

Debugging	HD	DI	CR	PS
	85.0 – 100.0	75.0 – 84.9	65.0 – 74.9	50.0 – 64.9
How well does the student demonstrate understanding and skills in relation to debugging?	Completely achieved all criteria.	Completely achieved criteria 1 and 3	Completely achieved criteria 1	Achieved criteria 1 and 3 to a large extent
		Achieved criteria 2 and 4 to a large extent	Achieved criteria 3 to a large extent Achieved criteria 2 and 4 to a partial extent	Achieved criteria 4 to a partial extent May not have achieved criteria 2

PRESENTATION

Submit a summary file through TURNITIN containing the following:

1. For each bug:
 1. The user tests (in system test format) that demonstrate the bug. (specifying expected buggy output)
 2. Output demonstrating the buggy behaviour.
 3. A **debugging log** recording the succession of hypotheses, tests, and results of those tests that you used to trace the origin of each bug.
 4. 'Before' and 'after' screen shots identifying the origin of the bug. (show variable values)
 5. Output demonstrating correct operation after resolution of each bug.
 6. Test results from the user test you specified for each bug demonstrating correct operation.
2. Copies of the files and/or a link to the version control repository where all code development and tests for the debugging assignment can be found. Include a README file at the top of the directory indicating where in your repository the files related to each debugging effort can be found. Make sure you commit both at the start of the debugging process, once you have developed an automated test demonstrating each bug, and after you have resolved each bug.

Assessment Information

Academic integrity

Play your part in maintaining yours and everyone's academic integrity at our university.

Complete your compulsory [Academic Integrity at Charles Sturt University \(https://www.csu.edu.au/current-students/learning-resources/build-your-skills/academic-integrity\)](https://www.csu.edu.au/current-students/learning-resources/build-your-skills/academic-integrity) course. This course will help you understand our Academic Integrity Policy, our expectations of you and the support services available to meet your academic obligations. You'll learn how to avoid plagiarism, cheating and contract cheating and collusion.

Charles Sturt University treats plagiarism seriously. We may use Turnitin to check your submitted work for plagiarism.

You can use also [Turnitin to check for plagiarism \(http://student.csu.edu.au/library/integrity/referencing-at-csu/checking\)](http://student.csu.edu.au/library/integrity/referencing-at-csu/checking) in your assessments before submission.

Referencing

Referencing is an important component of academic work. All assessment tasks should be appropriately referenced. The specific details of the referencing requirements are included in each assessment task description. [Get referencing style guides and help \(http://student.csu.edu.au/library/integrity/referencing-at-csu\)](http://student.csu.edu.au/library/integrity/referencing-at-csu) to use for your assessments.

How to submit your assessment items

ONLINE SUBMISSION PROCESS

Assessment tasks that are not completed through the Subject site need to be submitted electronically via Turnitin site by the due date. Turnitin password details for this subject will be given by your lecturer.

Unless advised otherwise, all Turnitin submissions are due by midnight (AEST) of the date specified. Please note that the time and the date of your Turnitin submission will be used to determine your official submission time.

All textual elements within an assessment must be submitted in a format that is readable by Turnitin. Specific exceptions, where an assessment requires the insertion of image based evidence of workings will be outlined in the context of the assessment. Students that deliberately attempt to insert content of assessments in a format that is not readable by Turnitin may be subject to Academic misconduct investigations.

Additional Submission Information:

It is recommended that **your name, student ID** and **page number** are included in the header or footer of every page of any assignment. You are also required to rename your assignment file before you submit via Turnitin as per below protocol:

SUBJECT CODE, SI, SURNAME, STUDENT ID, ASSESSMENT NUMBER, SESSION.

Example – ITC515 SI PATEL 11554466 A3 202060.doc

POSTAL SUBMISSION PROCESS

Under normal circumstances postal submissions will not be accepted for any of the assessments required.

HAND DELIVERED SUBMISSION PROCESS

Under normal circumstances hand delivered submissions will not be accepted for any of the assessments required.

ALTERNATIVE SUBMISSION PROCESS

See online submission above.

Extensions

It is best to complete assessment items by the due date. However, when something unavoidable comes up an extension may be possible. The following principles are used when processing extensions

1. For in-session assessment items, an extension request for up to three (3) calendar days can be made by emailing your subject coordinator directly before the due date. In your email please state the reason why you need more time as well as what precisely you are requesting. Supporting documentation is not required. If an extension is requested in the above format

with a valid reason and your request does not disadvantage other students, the extension will be approved.

2. For in-session assessment items, extension requests of more than three (3) calendar days must be made via the special consideration form: <https://apps.csu.edu.au/specialcons/>. The request must be made before the due date and must include supporting documentation. Acceptable reasons are given in the Special Consideration Policy <<https://policy.csu.edu.au/view.current.php?id=00298>>. Each request will be considered on a case by case basis. The request may not be granted. The maximum extension possible will be seven (7) calendar days.

3. If you receive an extension, then you should expect the assessment item and its feedback to be returned later. If you submit later than the extended due date you will receive late penalties as per guidelines below.

4. Unless your extension permits otherwise, submissions received 10 days after the original due date will receive zero.

5. For end of session exams, you can request a supplementary exam via

<https://apps.csu.edu.au/specialcons/>

(<http://antispam.csu.edu.au:32224/?dmVyPTEuMDAxJiY4MDQwZTgzYTVlYzNjZDdjNz01RUM3NTBCOV82Nz>

. This request must be made within 3 working days of the date of exam and must include supporting documentation. Acceptable reasons are given in the Special Consideration Policy <https://policy.csu.edu.au/view.current.php?id=00298>

(<http://antispam.csu.edu.au:32224/?dmVyPTEuMDAxJiY4MDQwZTgzYTVlYzNjZDdjNz01RUM3NTBCOV82Nz>

. For medical issues, a CSU medical certificate is required. If the supplementary exam (SX) is awarded then your exam is moved to the next examination period. In order to preserve exam integrity and manage the logistics of exams, the timing of a supplementary exam is heavily restricted.

How to apply for special consideration

Academic regulations provide for special consideration to be given if you suffer misadventure or extenuating circumstances during the session (including the examination period) which prevents you from meeting acceptable standards or deadlines. Find the form on the Student Portal [Special Consideration, Misadventure, Advice and Appeals](http://student.csu.edu.au/study/academic-advice) (<http://student.csu.edu.au/study/academic-advice>) page.

Penalties for late submission

The penalty for late submission of an assessment task (without obtaining the Subject Coordinator's approval for an extension) will be:

10% deduction per day, including weekends, of the maximum marks allocated for the assessment task, i.e. 1 day late 10% deduction, or 2 days late 20% deduction.

An example of the calculation would be:

Maximum marks allocated = 20

Penalty for one day late = 2 marks (so, a score of 18/20 becomes 16/20 and a score of 12/20 becomes 10/20).

If an assignment is due on a Friday but is not submitted until the following Tuesday, then the penalty will be four days (40% deduction or 8 marks in the example above).

Submissions more than 10 days late will be acknowledged as received but will not be marked.

Resubmission

Normally, resubmission of assessment items is not allowed.

Feedback processes

Feedback for assessment items will be provided by subject lecturer/s.

Assessment return

You should normally expect your marked assignment to be returned to you **within 15 working days** of the due date. If you submitted your assignment on time but have not returned by the return date, you should make enquiries in the first instance to the subject lecturer. If the subject lecturer is not available then contact your Course Coordinator on Level 4, 30 Church Lane.

Student Feedback & Learning Analytics

Evaluation of subjects

Charles Sturt University values constructive feedback and relies on high response rates to Subject Experience Surveys (SES) to enhance teaching. Responses are fed back anonymously to Subject Coordinators and Heads of Schools to form the basis for subject enhancement and recognition of excellence in teaching. Schools report on their evaluation data; highlighting good practice and documenting how problems have been addressed. You can view a summary of survey results via the Student Portal [SES Results \(https://student.csu.edu.au/study/subject-experience-survey-results\)](https://student.csu.edu.au/study/subject-experience-survey-results) page.

We strongly encourage you to complete your online Subject Experience Surveys. You will be provided with links to your surveys via email when they open three [3] weeks before the end of session.

Changes and actions based on student feedback

Changes made to the subject in response to previous student feedback include changing the first major assignment (Asg2) to a more development oriented task rather than a code quality exercise, and extending the teamwork aspect of the subject to include the second major assignment (Asg3) in order to reduce the individual volume of testing required in that assignment.

Learning analytics

Learning Analytics refers to the collection and analysis of student data for the purpose of

improving learning and teaching. It enables the University to personalise the support we provide our students. All Learning Analytics activities will take place in accordance with the Charles Sturt University Learning Analytics Code of Practice. For more information, please visit the University's [Learning Analytics \(http://www.csu.edu.au/division/student-learning/home/analytics-and-evaluations/learning-analytics\)](http://www.csu.edu.au/division/student-learning/home/analytics-and-evaluations/learning-analytics) website.

Data about your activity in the Interact2 site and other learning technologies for this subject will be recorded and can be reviewed by teaching staff to inform their communication, support and teaching practices.

Based on past analytics, changes made to the subject included <ABC> and <XYZ>.

Services & Support

Your [Student Portal \(http://student.csu.edu.au/\)](http://student.csu.edu.au/) tells you can how you can seek services and support. These include study, admin, residential, library, careers, financial, and personal support.

Develop your study skills

[Develop your study skills \(https://student.csu.edu.au/study/skills\)](https://student.csu.edu.au/study/skills) with our free study services. We have services online, on campus and near you. These services can help you develop your English language, literacy, and numeracy.

Library Services

[CSU Library \(https://student.csu.edu.au/library\)](https://student.csu.edu.au/library) provides access to the eBooks, journal articles, books, and multimedia resources needed for your studies and assessments. Get the most out of these resources by contacting Library staff either online or in person, or make use of the many Library Resource Guides, videos and online workshops available.

Policies & Regulations

This subject outline should be read in conjunction with all academic policies and regulations. Please refer to the collated list of [policies and regulations relevant to studying your subject\(s\) \(http://student.csu.edu.au/administration/policies-regulations-subjects\)](http://student.csu.edu.au/administration/policies-regulations-subjects) which includes links to Charles Sturt University's [Policy Library \(http://www.csu.edu.au/about/policy\)](http://www.csu.edu.au/about/policy) - the sole authoritative source of official academic and administrative policies, procedures, guidelines, rules and regulations of the University.

Subject Outline as a reference document

This Subject Outline is an accurate and historical record of the curriculum and scope of your subject. Charles Sturt University's [Subject Outlines Policy \(https://policy.csu.edu.au/view.current.php?id=00267\)](https://policy.csu.edu.au/view.current.php?id=00267) requires that you retain a copy of the Subject Outline for future use such as for accreditation purposes.