

# **Module Handbook**

**MSc Information Technology  
Group Software Development  
(UFCE4J-15-M)**

**2023-2024**

**Dr Emmanuel Ogunshile**

**AND**

**Dr Sami Abuezhayeh**

**WARNING: All email communication from me to you and vice versa must be via the Outlook Live system.**

**Module Leader:**

**Dr Emmanuel Ogunshile** ([emmanuel.ogunshile@uwe.ac.uk](mailto:emmanuel.ogunshile@uwe.ac.uk)) Room: 4Q59, Tel: [+4411732 83010](tel:+441173283010)

**Module Tutor:**

**Dr Sami Abuezhayeh** ([sami.abuezhayeh@uwe.ac.uk](mailto:sami.abuezhayeh@uwe.ac.uk)) Room: 3Q42

## **Section 1: Teaching and Learning Outcomes**

On successful completion of this module students will be able to:

1. Devise and participate in strategies for the effective working of a group towards a common goal; [Comp A1, A2]
2. Demonstrate project management skills and techniques in a group-based setting while engaging in a practical software project to produce a high quality software product; [Comp A1, A2]
3. Critically understand and employ both algorithmic and non-algorithmic software cost estimation techniques, planning/tracking techniques, risk management and other project management techniques for their chosen group software development project; [Comp A1, A2]
4. Employ appropriate software development process models, software development languages, methods, tools and modeling notations for the chosen group software development project; [Comp A2]
5. Demonstrate critical understanding and consideration of legal, social, ethical and professional issues; [Comp A2, A3]
6. Employ appropriate configuration and quality management standards and procedures for both the software process and the developed software product; [Comp A2]
7. Provide critical evaluation of the experience in undertaking a group-based software development project with critical reflections on software engineering lessons learned with regards to software development process, methods, modelling techniques, quality and configuration management aspects, etc. [Comp A3]

## Syllabus Outline

- Software engineering as a coherent process.
- The role and benefits of the SEI Capability Maturity Model.
- Selection, evaluation and use of CASE tools.
- Software cost estimation both algorithmic and non-algorithmic methods.
- Critical reflection on current software engineering practices.
- Organisation and management of a software development team.
- The Software Requirements Specifications Document.
- Documentation issues and change management using traceability techniques.
- Quality and configuration management issues.
- Risk management.
- Project monitoring and control.
- Task allocation and resourcing.
- Computer supported collaborative work.
- Application of system/software modelling, design and construction techniques to a group project
- Legal, Social, Ethical and Professional issues
- Usability issues

## Contact Hours / Scheduled Hours

This will be 2 hours per week delivered over 24 weeks or 4 hours per week delivered over 12 weeks including lectures and tutorials.

## Teaching, Learning Methods and Reading Strategy

### Scheduled learning:

This module gives the students the opportunity to extend, consolidate and apply the knowledge they have gained in the prerequisite modules. The focus of the module is on the completion of a group project. Teaching and learning is largely driven by the requirements of the project.

This module will be taught by a combination of lectures and tutorials in every weekly session. Students will receive learning material in advance of the weekly classes and will be expected to use the material to prepare for class. The class based sessions will therefore allow for increased interaction during lecture/tutorials in addition to raising potential knowledge exchange between students with industrial background and tutors.

**Independent learning:**

Group work will be used to enable the students to gain practice of real software construction. The group will be student led, with the tutor acting in the role of client and user. A member of staff will act as a customer/client and will supply a minimal specification. The students will then be expected to progress this minimal specification into working a software product with interim deliverables appropriate to accepted practice. Both management and development practices will need to be utilised. Appropriate communication and coordination will take place among students and tutors, using the facilities offered by Blackboard. In addition, the students will be expected to research, evaluate and then select new technology for use in the group development environment.

**Core readings:**

Any essential reading will be indicated clearly, along with the method for accessing it, e.g. students may be required to purchase a set text, be given a print study pack or be referred to texts that are available electronically or in the library. Module guides will also reflect the range of reading to be carried out.

**Further readings:**

Further reading will be required to supplement the set text and other printed readings. Students are expected to identify all other reading relevant to their chosen topic for themselves. They will be required to read widely using the library search, a variety of bibliographic and full text databases, and Internet resources. Many resources can be accessed remotely. The purpose of this further reading is to ensure students are familiar with current research, classic works and material specific to their interests from the academic literature.

**Access and skills:**

The development of literature searching skills is supported by a Library seminar provided within the first semester. Students will be presented with further opportunities within the curriculum to develop their information retrieval and evaluation skills in order to identify such resources effectively. Additional support is available through the Library Services web pages, including interactive tutorials on finding books and journals, evaluating information and referencing. Sign up workshops are also offered by the Library.

**Module Assessment and Assessment Strategy**

The assessment strategy for this module comprises uncontrolled components as part of the group-based software development project. These components are:

(a) a project proposal to be submitted by the fourth week of the term and is graded on a pass/fail basis. This is group-based with allocation of marks dependent on the individual contribution that will be controlled by the respective resources allocated to the associated tasks set in the assignment specifications. Feedback will be given to students in four weeks' time from submission, and hence students will utilise such feedback to inform better attainment in the remaining parts of the

coursework with not much of a loss of marks given the only 15% of the proposal allocation from the coursework total assessment.

(b) Detailed group-based project deliverables including some specific project individual contributions and these comprise in total 75% of the coursework assessment.

(c) Individual report comprising 25% of the coursework assessment and reflecting on problems observed or faced, software engineering lessons learned, and suggestions to enhance the run of the project itself and also suggested functional and non-functional enhancements of the respective software application specified, designed and developed. The report should not exceed 2500 words.

**Note:** For the resit assignment, this is not group-based and it will relate to the individual component(s) that the student failed in the original submission.

## Resources

### Required textbook and Reading List

- Introduction to Java Programming by Y. Daniel Liang, 10<sup>th</sup> Edition, Pearson (9<sup>th</sup> edition/international edition/brief version/pdf eBook would be OK if there are good deals, but we will use 10<sup>th</sup> edition as reference).
- Brooks, F. (1982) The Mythical Man Month. Addison Wesley.
- Glass, R. (2006) Software Conflict 2.0. Developer Books.
- Glass, R. (1998) Software Runaways. Prentice-Hall, Yourdon Press.
- Hill, P (2010) Practical Software Project Estimation: A Toolkit for Estimating Software Development Effort and Duration. McGraw-Hill Osborne
- Jarvis, A and Hayes, L. (1999) Dare to be Excellent: Case Studies of Software Engineering that Worked. Prentice Hall.
- Levin, P (2005) Successful teamwork! For Undergraduates and Taught Postgraduates working on Group Projects. Open University Press.
- Module's lecture notes and tutorial worksheets on the Blackboard VLE.
- San San Sy and Cathleen Anderson (2004) Online Conferencing: A Guide for Virtual Group Discussion. .
- Sommerville, I (2011) Software Engineering, 9th ed., Pearson.
- Thomsett, R. (2002) Radical Project Management. Prentice Hall.
- van Lamsweerde, A. (2009) Requirements Engineering: From System Goals to UML Models to Software Specifications. John Wiley & Sons Ltd.

### Module Learning Materials

- These will be made available to you via Blackboard

## Section 2: Task Specification for Project Proposal

This assignment (**Project 1**) is graded on a pass/fail basis and is worth 0% of the overall module mark.

Broadly speaking, the assignment requires you to research the literature on a major hospital based system in order to:

- Identify the initial system requirements
- design the system (using UML diagrams)
- Create acceptance tests needed for the system
- Write a report that documents this.

The hospital scenario is described in more detail in section 2.1.

This is a group-based assignment.

Working on this assignment will help you to:

- Research the literature on the hospital based system project (see section 2.1).
- Read literature material critically, i.e. with an initially sceptical frame of mind about whatever is being asserted.
- Identify and summarise key points.
- Summarise and draw conclusions from your research.
- Create professional reports

If you have questions about this assignment, please email the teaching team or post them to the discussion board **Assignment forum** on Blackboard.

## Section 2.1: Hospital System Initial Description

Please read through the basic description of the hospital system below.

A typical hospital has several wards divided into male wards and female wards. Each ward has a fixed number of beds. When a patient is admitted they are placed onto an appropriate ward. The doctors in the hospital are organised into teams such as Orthopedics A, or Pediatrics, and each team is headed by a consultant doctor. There must be at least one grade 1 junior doctor in each team. The Administration department keeps a record of these teams and the doctors allocated to each team. A patient on a ward is under the care of a single team of doctors, with the consultant being the person who is responsible for the patient. A record is maintained of who has treated the patient, and it must be possible to list the patients on a ward and the patients treated by a particular team.

Now read thorough the transcript below of an interview between the Software Engineer (**SE**) and the Hospital Administrator (**HA**). The Software Engineer is trying to clarify the scope of the system and to determine the use cases that are needed.

**SE:** So let me just check what this system must do. You need to admit patients and discharge them, and you also need the system to be able to output various lists?

**HA:** Yes, oh and also to record who has treated a patient.

**SE:** So the system needs to know about doctors and consultants too?,

**HA:** Yes that is right.

**SE:** Let me just get this clear, does this system also need to be able to record what treatment a patient has received?

**HA:** No, that is done by a separate medical records system, not by this system.

**SE:** OK, so let's look at something specific to this system. You initially need to admit a patient to a ward, which we talked about earlier, but does a patient always stay on the same ward?

**HA:** No, not necessarily, sometimes we transfer patient between wards.

**SE:** What information needs to be provided when this happens?

**HA:** The administrator has to identify the patient, and then the ward they are being transferred to.

**SE:** Presumably the new ward has to be of the right type?

**HA:** Of course, and there must be room on it too.

**SE:** And what is recorded when a patient is transferred?

**HA:** That the patient is on the new ward.

**SE:** Is there any need to record that they used to be on a different ward?

**HA:** No, not by this system.

**SE:** You mentioned some lists; tell me about what kind of information the system needs to output?

**HA:** We need a list of all the patients on a particular ward, and a list of all the patients cared for by a particular team.

**SE:** Let's take the first one first. What does that involve?

**HA:** The administrator identifies a particular ward and the patients' names and ages should be displayed. Perhaps the ward could be identified by selecting from a list of ward names?

**SE:** We'll bear that in mind, but we'll leave the details of the user interface until later. How about the second list?

**HA:** Well the administrator identifies a team and all the patients cared for by that team should be listed.

**SE:** And what do you need on this list?

**HA:** The patient's name and the name of the ward that the patient is on.

**SE:** So let's move on to what you need to do to record treatment. Do you need to know every doctor in a team who has actually treated the patient?

**HA:** Yes, we need the system to record this.

**SE:** What exactly does this entail?

**HA:** Well the administrator has to identify the patient, and the doctor (consultant or junior doctor) who has treated the patient, and the system should record that the doctor has treated the patient.

**SE:** But only doctors in the team that the patient is allocated to can treat the patient.

**HA:** Yes, that is right.

**SE:** OK, do you ever need to list exactly who has treated the patient?

**HA:** Yes, I forgot about that. For any particular patient we need to know the name of the consultant responsible for the patient, and the code of the team that is caring for the patient, as well as the name of each doctor who has treated the patient, and their grade if they are a junior doctor.

**SE:** Do you really need all of this information on this list? Surely if you know the code of the team you could have the system look up the name of the consultant doctor?

**HA:** Yes that is true, but it would be convenient to have it on the list.

**SE:** So tell me what happens when a patient leaves the hospital?

**HA:** Well we have a discharge process, and the discharged patient should be removed from this system.

**SE:** So what does the administrator do?

**HA:** Oh they should just tell the system which patient is to be discharged, and all information relating to that patient should be removed from the system.

**SE:** So you don't need any record of a patient once they are discharged?

**HA:** Well of course we keep their medical records, but not on this system, that is a totally different system.

### Task 1

Take full ownership of the hospital based system described in section 2.1; and based on the information provided, and using related literature, write a convincing project proposal about the need for such a system.

### Task 2

Using the interview transcript presented in section 2.1, create a complete UML Use Case Diagram for the hospital based system.



### Task 3

Identify and specify the initial requirements for the system described in section 2.1

### Task 4

Imagine you are writing the requirements document for what you know of the hospital system so far. Describe the following items:

- The system domain.
- All of the use cases.
- All of the acceptance tests needed.

## **Section 2.2: Deliverables**

You must submit: A report as defined above in the Task Specification section (Tasks 1 to 4).

For Tasks 1 to 4 in section 2.1, cite relevant work of others in the body of your report and, at the end of your report, include a list of the cited references (expressed using the Harvard standard). The page limit for this task is 20 pages (not including references).

Please include your page count at the end of your assignment.

## **Section 3: Task Specification for Software Deliverables**

This assignment (**Project 2**) is worth **75%** of the overall mark for the module.

This is a group-based assignment.

Broadly speaking, the assignment requires you to **work within a group of 5 to 7 students** i.e. for the hospital based system described in section 2.

As a group, you will implement either a stand-alone, command line or web application for the hospital based system described in section 2 in any programming language of choice.

Working on this assignment will help you to:

- Learn about the link between requirements, design, implementation and testing
- Learn how to transform stakeholder and user requirements into a technical view of the system.
- Create a complete architecture for system elements (or components). System architecture is concerned with relations and structure i.e. how are system units related?
- Learn system design. Define system elements in sufficient detail to enable implementation i.e. using divide and conquer approach.

- Learn how to test and verify your concrete implementation for correctness.
- Practice how to manage complete Software Development Lifecycle (SDLC)
- Create professional report based on your implementation.

If you have questions about this assignment, please email the teaching team or post them to the discussion board **Assignment forum** on Blackboard.

## Section 3.1: Deliverables

Your final submission zip file for the group-based project need to contain the following:

- A named folder containing the contribution of each team member (e.g., John Forest\_Group1\_Contribution)
- A folder containing final / complete deliverables for each unique project group (e.g. Group1\_Final Deliverables). This should be well organised for easy access for the module team.

**The final submission deadline for both group-based software deliverables and individual reflective report is 7th August 2023.**

Submit a zip file of your implementation for the hospital system. This must be submitted on Blackboard at 2.00pm. The zip file should contain what is described in sections 3.1.1 and 3.1.2 below.

### 3.1.1 A .doc/.docx file containing a brief discussion of your implementation choices

A .doc/.docx file containing a brief discussion of your implementation choices. The discussion should include a description of the limitations of your implementation, any alternatives that you considered and discarded and the rationale for your solution. Please ensure that you use the format for this that has been discussed in class (i.e. as covered herein below in **Section 3.1.3**):

### 3.1.2 A zip file containing the following set of information about your project

- Product Backlog (and evidence of history and management)
- Iteration Plan
- Risk Register (and proof of history)
- Design Documentation
- All source Code

- Runbook (this should tell the user how to run your code)
  - ✓ Any technical requirements
  - ✓ E.g. OS
  - ✓ Programming Language Version e.g. Scala 2.10.3
  - ✓ Database and version information
  - ✓ Any required tools e.g. Eclipse Scala IDE / SQLWorkBench
  - ✓ Where to find all resources
  - ✓ Code, models, documentation, data etc.
  - ✓ All libraries used
  - ✓ Names, versions
  - ✓ Where to find, how to configure if appropriate
  - ✓ Steps to build, deploy and test database
  - ✓ Steps to build, test domain model
  - ✓ Steps to build, deploy and test PPL
- Any Limitations & Enhancements of your proposed tool
- Group Assessment (also append your group's contribution matrix)
  - ✓ What went well
  - ✓ What did not go so well
  - ✓ How the group could improve / do better next time
  - ✓ Any Legal, Ethical or Professional Issues (e.g. licensing)
  - ✓ How well the Development process worked (e.g. would you use Agile again)
  - ✓ Recommendations on tooling
  - ✓ Experience with environments / tooling you used

## Section 4: Task Specification for Reflective Report

This assignment (**Project 3**) is worth **25%** of the overall mark for the module.

This is an individual-based assignment.

You need to provide

- Review of work you did on the hospital system project
- *Critical* appraisal of your own role for the hospital system project
  - What you did well
  - What you did not do so well
  - How would you improve next time
- How well you adapted to the development process and to team working
- Note – not just a list of things I did!

## Section 4.1: Deliverables

Submit a report not more than 2,500-word count essay based on the task specified in section 4.

Cite relevant work of others in the body of the report and, at the end of your report, include a list of the cited references (expressed using the Harvard standard).

The word count for this task is 2,500 words (not including references). See UWE's policy on word count by accessing the Word Count Policy link at <http://www1.uwe.ac.uk/aboutus/policies>.

## Section 5: Feedback mechanisms

Feedback will be offered after the submission date in two ways:

1. A group report for the whole cohort on the important strengths and weaknesses that are common to most, or all, of the submissions.
2. An individual report for each student or group created by customising the CSCT feedback template.

## How your Group-based Project will be assessed using Contribution Matrix

The summative assessment is carried out according to the assignment brief with marking carried out by module team against pre-determined criteria (for this, please see the Module Marking Scheme). This generates the underlying group mark. There follows a team evaluation process which generates the weighting factor that will be applied in order to assign individual grades. The evaluation requires all team members to assign their partners and themselves a mark from 0 to 5 according to the following descriptions outlined in Table 1 **for each task completed for the group-based project**. This mark gives a weighting as to their contribution and effort within the team, where their fellow team members have good authority. Contribution is defined as a quality measure: good work is influential and can strongly influence the team dynamic and assignment outcome. Effort is a quantitative measure: it may influence general group dynamics and the motivation of the team.

Table 1. Classification and descriptors for students to score themselves and each other

Classification	Descriptor
0	Rarely present at sessions and made no contribution, or has been present but actively destructive in their contributions
1	Has been present but made no attempt to contribute
2	Has contributed under duress and/or quality was unsatisfactory
3	Team player, doing what was required and of usable quality

4	Volunteers activity and contribution, engaging in the assignment
5	Very pro-active, initialising activity and taking on important roles with high quality outcomes

### Processing and interpreting results

Marks are calculated using a spreadsheet, but the module team needs to remember that these are “decision numbers” so care must be taken about their implied rigor in the processing, as they come from a subjective source. The rows of the table represent how an individual is evaluated, whilst the columns show each student’s evaluation of themselves and others in the team. The average of each row divided by the average of the table gives the individual weighting factor. When multiplied by group mark this produces the individual student mark.

	A	B	C	D	E
1		James	Anna		
2	James	4	4	=average(B2:C2)	=D2/\$DS\$4
3	Anna	4	3	=average(B3:C3)	=D3/\$D\$4
4		=average(B2:B3)	=average(C2:C3)	=average(B2:C3)	

  

	A	B	C	D	E
1		James	Anna		
2	James	4	4	4.00	1.07
3	Anna	4	3	3.50	0.93
4				3.75	

Figure 1. An example formula sheet in MS excel for a 2 person group (top), with the resulting data with weighting factors for each student highlighted (bottom).

In addition to processing the table, the module team needs to carefully analyse the results and to be alert to the telltale signs of:

- **Collusion** between some group members and the formation of cliques.
- **Vendettas**: sometimes tribal splits open up along lines of technical discipline, cultural or even racial differences. These need to be actioned appropriately as per University policy.
- **Outlying individuals**: proactive and effective leaders are easily highlighted, as are those who have contributed little. Other typical profiles that can be identified are self-promoters who exaggerate their input and diffident students that undervalue theirs.

## How the module team will assess your Group-based project where there is no agreement for the Contribution Matrix by members of the team

**Reference** to the teams’ shared workspace (**Blackboard File exchange contents**) can help the module team better understand the politics of the group interactions and give insights into some of the

situations describe above. There should be file sharing records, where contributions are made and action lists or group meeting minutes are available for scrutiny.

If the module team decides to override any of the students' self or peer classification, notes/comments/feedback will be kept that justify the reasons either to students or Examination Boards. Academic judgment is completely in the module team domain, so although the students do not have a right to appeal the mark, they are perfectly entitled to have decisions clarified. Consequently, there need to be clear records.

Finally, feedback and review is an important stage. Students marks and Individual Reflective reports are returned with comments. Generally, the module team will also need to put aside some time to address misunderstandings and queries both on a group basis as well as one-to-one. This part of the process can be difficult as some individuals come to realise that their perception of the role in the group is not shared by others.

	Percentage mark	0-30	31-49	50-60	61-69	70-79	80-100
	Overall Descriptor	Very poor	Poor	Satisfactory	Good	Very Good	Excellent
<b>Business Case</b>		Barely understood / case unclear	Business poorly specified and unclear	Business case exists but understanding is limited and scope unclear	Business case made but general and only limited scope outlined	Business Case clearly states and scope expressed.	Mature business case with profound research.
<b>Initial Requirements</b> (Core requirements, Prioritisation, functional and non-functional)		Little or no requirements presented	Minimal, low quality requirements, shows little analysis of thought.	Limited set of requirements that just meet the needs of client provided with evidence of process followed.	Complete set of requirements but lack clarity ideally required in requirements.	Complete set of requirements covering the broad business needs	Complete set of requirements covering all business needs presented in detail and critically examined with examples.
<b>Outline Solution</b>		Essentially non existent	weak showing little understanding of the problem	Essential components present but lacking any clear roles or grasp of main issues.	All main components present.	All main components present with limited evidence of consideration of strengths and weaknesses	Outline of solution comparable to industry standards along with evidence. Strong evidence of consideration of strengths and weaknesses.
<b>Introduction</b>		Little or introduction	Introduction is incoherent, lacks narrative and unclear.	Introduction presented is coherent and clear.	Holistic introduction provided covering most of the areas.	Introduction covers all areas and is coherent.	Holistic introduction provided covering all of the topics.
<b>Literature review</b>		Little or no literature review	Literature review is incoherent, lacks narrative and unclear. Largely relies on old literature on and on	Literature review is relevant, up to date and relates to the concepts of the project chosen.	Literature review is relevant, up to date and relates to the concepts of the project chosen. Gaps in literature	Literature review is relevant, thorough and up to date. Relates to the concepts of the project chosen. Gaps in literature has been	Thorough and critically examined literature review that provides a holistic view of the concepts being dealt within the project. Gaps identified and justified.

			topics irrelevant to the chosen project.		has been identified but not justified.	identified, justified in the context of the chosen project.	
<b>Method</b>		Little or no consideration to methods	Methods presented is descriptive without any sound justification for the choices made. Lacks wider consideration in the context of project.	Methods presented has a sound rationale. Wider implications have been clearly considered. Choices have been justified based on research/ literature in the context of the project. Demonstrates effective use of appendix.	Methods presented is holistic. Wider implications have been explored. Pros and cons of methods carefully evaluated in the context of the project.	Methods presented is critically examined. Justification of the method chosen is provided on the basis of research and is within the context of the project.	Method is thoroughly examined in the context of the project and sound rationale provided for the decisions taken.
<b>Design and development (includes source code, run book)</b>		Missing or virtually missing	Some source code available but poorly structured or missing significant sections (such as suitable tests, run book)	All source code available including tests, compiles and runs. The design and development is consistent with the client requirements. Rationale for design and development embedded. There is evidence of application of process. Demonstrates	All source code is clear, well structured, organised appropriately, compiles and runs and is consistent with the client requirements. Little more than key functionality delivered including documentation covering maintainability..	Source code is complete, suitably commented, structured, with clear separation between technologies, clear self checking tests, compiles runs but some what lacks in the application of method. Partially implemented system delivered including documentation covering maintainability..	Source code is complete, suitably commented, structured, with clear separation between technologies, clear self checking tests, compiles runs. The section demonstrates adequate application of the method. Fully implemented system delivered including documentation covering maintainability.



				effective use of appendix. The key functionality is implemented and in working order including documentation covering maintainability.			
<b>Discussion</b>		Negligible or missing	Superficial discussion. Lacks contextualisation	Discussion is critical and effectively utilises literature. Provides examples and evidence of applying the concepts throughout. <b>Demonstrates effective use of appendix.</b>	Discussion is balanced and critical drawing upon various sources. Examples and evidence of applying concepts have been presented throughout.	Discussion is thorough and critical examining various dimensions of the project under development. The submission demonstrates some level of understanding of industry practice.	Discussion is thorough and critical examining various dimensions of the project under development. The submission demonstrates good level of understanding of industry practice and provides evidence.
<b>Group assessment</b>		Negligible or missing	Very limited analysis of limitations and no enhancements (alternatively limited analysis of enhancements and no limitations)	Some analysis of limitations and enhancements. Examples and evidences provided in the context of the chosen project.	Clear analysis of limitations and enhancements providing examples in the context of the chosen project.	Clear analysis of limitations with implications of these explained along with the examples in the context of the chosen project.	Clear presentation of enhancements indicating how enhancements would be provided and the benefit they offer.
<b>Group conduct and management</b>		Negligible or missing	Limited analysis of Group operation. Instances of misconduct or deviation from the module group contract or client	Description of what the group did well, did not do well. No suggestions for improving future group working.	Description of what the group did well, did not do well and some proposals for future improved group working.	Clear analysis of the group as a whole with presentation of what the group did well, what did not go well and proposals for how the group	Thorough analysis of the group. Good practices evidenced and discussed. Good practices adopted and demonstrated throughout the module.

			contract observed. There is little or no evidence of collaborative work.	Little analysis of Social, Legal, Ethical and Professional issues. Some evidence of collaborative work, adherence to module group contract and any client contracts throughout the duration of module .	Some analysis of Social, Legal, Ethical and Professional issues and some description of how well development process and tooling worked. Good evidence of collaborative work, adherence to module group contract and any client contracts throughout the duration of module.	could improve their operation. Analysis of Social, Legal, Ethical and Professional issues with description of how well development process worked and recommendations on tooling.	
User acceptance test		Missing or no user acceptance test	Minimal user acceptance test performed. Lacks consideration of user acceptance models.	Research informed user acceptance models considered, described and applied in the context of the chosen project.	Research informed user acceptance model considered, evaluated and applied in the context of chosen project.	Chosen user acceptance model is fully evaluated, appropriate in the context of the project and well justified.	Demonstrates wider understanding of user acceptance models. Chosen user acceptance model is fully evaluated, critiqued. User acceptance model is adapted or extended appropriately in the context of the project citing appropriate literature and rationale.
Prototype		No prototype demonstrated.	Prototype demonstrated with minimal functionality.	Functioning prototype demonstrated that satisfies limited requirements	Functioning prototype demonstrated that satisfies most requirements	Functioning prototype demonstrated that satisfies all requirements	Fully functioning prototype demonstrated that is intuitive i.e. not only satisfies all requirements provided by the client but also

				provided by the client.	provided by the client.	provided by the client.	research informed additional functionalities that were agreed by the client.
Individual assessment		Missing / Little or no critical appraisal. Missing log.	Minimal critical appraisal. No reflection of individual professional conduct or documented instance of misconduct observed. Missing log or incomplete or inconsistent logs	Review work done but little awareness of / or reflection on role and work done. Limited discussion of adoption of development process. Good reflection on professional conduct throughout the duration of the module providing examples. Fully completed log throughout the module.	Good review of work done and some uncritical appraisal of both own role and of the processes followed. Some discussion of adoption of development process. Fully completed log and evidences signposted appropriately.	Good review of work done and thorough critical appraisal of both own role and of the development processes followed. Critical appraisal of development process. Thoughtful log where the output is realistically measured and demonstrated appropriately.	Critical review of work done and thorough critical appraisal of both own role and of the development processes followed. Critical appraisal of development process mapping onto prevalent current industry/research practices. Log fully complete, output measured and the use of log is fully integrated with the tasks undertaken throughout the module.