# **Deliverables 3 & 4 (Increments 1 and 2)**

**Common sections for all groups:**

Deliverables Certificate

Updated Project plan (with revision history + new “Handover Requirements” section)

Updated Requirements/ Scoping Document (with revision history)

**Stream-specific sections (with different criteria):**

Design Documentation

Testing Documentation

Prototype / Product (MVP)

+ deliverable 4 stream-specific document

(13 marks for Deliverable 3, 13 marks for Deliverable 4)

## **All Teams: Deliverable 3 & 4 submission instructions**

PLEASE INCLUDE YOUR TEAM NUMBER in the file name.

The next submission contains multiple documents. However, for marking I would like one .pdf document in this order:

1. Revised Project Plan/Quality (complete plan, with your changes and revision history).
2. Revised Requirements/Scoping document (include complete document with your changes and revision history).
3. Analysis & Design Document including Assumptions at the end of this document.
4. MVP/Prototype Document
5. Testing Document
6. User Manual// Help/Training// Scripts/Model Execution// Recommendations/Monitor Report (Stream specific document **for Deliverable 4 only**)

## **Common sections for all groups**

Add revision tables to the start of all documents just after the TOC to indicate the revision number, date, person and change to the previous version (be specific about what changed / why).

### **Updated Project Plan**

Revised Project Plan (0.5 mark **Del 3 and 0.1 marks for Del 4**)

The following is required:

1. **Include the heading in the project plan: Handover requirements.** This section can be added after your schedule to explain in detail what the client needs. Handover will be different according to what the project is. It's about making sure the client gets what they were expecting and need before the team disappear. So any data, software, algorithms, scripts, test cases, testing tools/environments, outputs and documentation would be handed over together with training/manuals that lets them go forward with it - including reports or test cases showing current output and/or what is fully functional or not.

For all projects, the point is to discuss with the client what they need at the end of the project. Make sure they have everything they need and know how to use what you have given them. There is no set format or requirement for the marker.

a. For games and software projects, think about who will do play testing and/or acceptance testing, asking questions such as when will testing start, when and what devices/sites does the final product need to be deployed on, who will maintain the system, how will the system be maintained, what are the user manual/training manual/installation/installation wizard/administration parameters requirements, what database/file setup is required, what documentation is required.

1. **A revised and updated version of your project *schedule*,** that takes into account any schedule revisions, covering the balance of the project, that have arisen since you submitted the previous schedule. Anything else that has changed such as team structure, resource allocation, risk evaluation, change management processes, communication strategy, etc should also updated in the plan and quality manual and included in the revision history. Resubmit the entire updated document.

### **Updated Requirements / Scoping Document (0.25 marks Del 3 and 0.1 marks for Del 4)**

Add a revision history table at the start of your document to clarify all changes made from Deliverable 2. No new sections are required. Please review feedback to Deliverable 2 submission and consider any project changes or sponsor feedback and update the document accordingly.

### **List of Assumptions**

**(0.25 marks Del 3 and 0.15 marks for Del 4)**

Provide a list of assumptions (relevant to design, requirements, project plan or any of the documentation). This will help you think about them and whether they are reasonable and also help the marker/review team to understand why you have done certain things. Please review the assumptions as a group before submission. A poor assumption will not be a valid reason for poor design decisions.

### **Prototype / MVP Document**

(4.5 marks for increment 1-**Del 3** and 5.5 marks for increment 2-**Del 4**)

**Before** the due date/time you need to arrange a time with the client to demonstrate your system and receive feedback. The team must organise that meeting. All team members are expected to attend. A valid reason for non-attendance by a team member must be arranged before the meeting. When the demonstration/meeting is arranged please let the convenor know of the arrangement via the meetings thread on iLearn.

Under the heading “Prototype”, include at least a page in your increment 1 and 2 documentation to present and discuss your prototype. Screen shots and how the product looks and works should be included.

Also include a subheading “Sponsor meeting, feedback and response to feedback” in the “Prototype section”. Include details of the sponsor meeting and feedback. All sponsor feedback on any aspects of your project should be included in this section. Also clarify how the team have or will respond to that feedback.

If you have not been able to arrange a meeting before submission due to sponsor unavailability state that under this heading and when the meeting is scheduled. After the meeting, include the feedback and response in your next weekly report submitted on iLearn.

I will also be contacting the sponsor. The client will be asked to confirm the meeting and their feedback via a survey sent to them.

### **Prototype Feedback Adjustment**

The marks for your prototype are made up of what you present in your document, the feedback reported and the client’s feedback to me. (For Increment 1 marks are allocated as follows: 0 if no evidence or mention of a prototype, 1 if some evidence identified, 2 if evidence provided, 2.5 if brief, just screens and description or just feedback, 3 if shown with screens and feedback provided. 3.5 if shown in detail with feedback and responses to feedback, 4 if sponsor is happy 4.5 if sponsor is very happy.)

## **Analysis, Design + Testing Documentation for DATA SCIENCE projects**

The main objective with the iterations is to build up as much of the data processing pipeline and describe the decisions / choices made - as well as how any choices and outputs will be evaluated. Someone else coming in late to the project should be able to use this deliverable to understand what the team is doing and where they are up to with their deliverables - including the rationale behind any design decisions.

Students choosing the data science stream can leverage what was built up in data science units (like COMP257 / COMP2200) and discuss what has been done in their project. Much like in the project portfolio and project presentation from COMP257 / COMP2200, the aim here is to communicate different aspects of the project implementation including the data, the goals, the options, the decisions, the choice of models and starting configurations, and the resources available.

For DATA SCIENCE projects, you should seek to establish a baseline model for the project – the simplest model you can think of for the problem. This would be a good deliverable as a MVP since it encourages you to have an end-to-end processing pipeline and to establish a baseline that you can then try to beat with their refined model in the subsequent increments.

Your documentation should include the data manipulation process with clear instructions on how to run them to turn the raw data into the input for modelling. One possible approach could be to load / maintain them in a GitHub / Bitbucket / Firebase repository with appropriate permissions shares. However, this is something that should be discussed with your sponsor and the convenor.

You should present the results of the modelling via evaluations in the form of a report. The report should outline the different models used and the results observed. Interpreting the results and discussing implications for the business questions that prompted the study are also a key part of this. In many cases this can be structured like a traditional scientific paper – material, methods, results, discussion.

### **Feature engineering:** (deciding what you are looking for and how to go about it)

· From the data, are there any trends / ranges to look for?

e.g. if you are analysing time-based movement data, what accelerometer and gyroscope ranges / trends do you want to look for to identify someone that has fallen / collapsed?

e.g. if you are looking at financial data, what does an "upward sale trend" look like?

e.g. if you are looking at population data, or education rates, what "characteristics" would you want to look at and why?

· Describe what data characteristics are being looked for, and how your data pipeline is being processed to generate these features.

· Give each different "feature" or "characteristic" a name and then attribute some form of data ranges / statistical definition.

### **Solution Architecture:** (choice of macro architecture / pipeline)

· Provide an overall description of each section of the pipeline including the data in and out of each "stage". The data details can be properly described in the "Detailed data descriptions" section.

· This would most likely be a more detailed version of the overall pipeline presented in the team's scoping document as some of the stages would now be implemented / finalised.

· Include any resources available / resource processing constraints to each of the sections in the pipeline (e.g. processing / timing limits)

### **Algorithms / models methods:** (detail what is in each part of the solution architecture, including models used and initial conditions / config settings)

· selection of model... there are many different approaches: predictive, supervised / unsupervised, classifiers, ... which are going to be used? why? and why chose those over other approaches?

· Are there any settings needed (eg. in a KNN model, what is the number of the nearest nodes being used in the application of the classifier?)

### **Detailed Data descriptions:**

Data being used, data being generated, data being stored, as well as any summaries and/or reports.

### **Model evaluation:**

· how will any data / outputs be compared / tested / evaluated for correctness and accuracy?

· if you are choosing between models, how will the models be compared / contrasted to see which one has a better performance (e.g. if you are comparing different classifier models, on what basis are you comparing them? Detail each comparison.

### **Performance evaluation results:**

what are the results of any tests run so far and what are the future planned tests for future iterations? (e.g. if this is Deliverable 3, what is planned for Deliverable 4? IF this is Deliverable 4, what is planned before the handover?)

Refer to the rubric / marking scheme (“All Teams: Deliverable 3 and 4 marking schemes”) for sections to be included in the design and testing documentation.

## **Deliverable 4 (Only) stream-specific document: for Data Science Teams**

**Scripts / Model Execution (2 marks)**

The *Scripts and Model Execution Documentation* should not exceed **twenty (20) A4 sides.** This is essentially a user or training manual for the one or more users of your MVP. It might include model parameters, how to set up the models, data sources/format, file locations, etc. The intended user/s may differ according to the project and there could be more than one type of user with different documentation needs.

The content of this document is something that you MUST discuss before **Del 3** when you talk to your client about what they expect at handover at the end of the project (see point 1 under updated project plan for Deliverable 3), ask them what their requirements are regarding support/maintenance. Find out what they need regarding training, help, installation, configuration, etc of your product/output.

Like all technical writing, the document should have a clear Table of Contents (TOC) and well-organised content. In addition to documenting your models and how to run them, this document could involve sections for different types of users (e.g. Admin and end users), installation guide, configuration settings, screen shots with example data, APIs, training, steps describing usage troubleshooting/where to get help, etc. Check you have page numbers, or your TOC will have a limited value.

# **MARKING RUBRIC**



