CS3305/ CS3505 Team Software Project - 2017 Report Guidelines

All reports should be submitted electronically via Moodle, both in .odt and in .pdf to ensure flexibility and platform independence in production, compilation and evaluation.

Moreover, .odt supports basic code syntax highlighting via the COOoder extension.
Hardcopies are not required, so there is no excuse not to use colour, as there are no print restrictions.

A class share is also in place, which contains a folder for each registered for the module:-

/users/coursework/TYP_17_demo

and each can write to their own folder there, and share with the class files, both code and documentation provided permisisons are set correctly on the files.

File format
.odt format: Open Office document text.. is mandatory, as it is entirely cross platform, for others to read, and Google docs and other cloud offerings should support it, so you can collaborate in writing.

Since submission sizes are specified in wordcount this year, the choice of font and size are less important, but the following guidelines may ease things for all.

Fonts
To minimise risk of wasting time reformatting when downloading from Google Docs to .odt, use Arial, Calibri or Times New Roman as they are clear, common and available across Google, MS & Libre docs.

Size
Use 12 point, but can reduce to a minimum of 10 point, if needed for page reformatting, but try to keep it as near 12 as possible as reading large quantities at 10 point can be tiring.

Length.
However, as rough guidelines, here are my estimates. Rather than give pages, I'll use a word count, assuming 250 words per page, using 12 point of fonts specified, with line spacing from 1 to 1.5.

There's no point in waffling, but we need to write something. To help you think about things I'll upload a few papers about agile documentation, but you don't really need to read them unless you are stuck for ideas. Of course it might impress the world that you appear to have thought about such things, even if you have no firm conclusions, just like the rest of the world!?

After thinking and rethinking, about page lengths, I think the easiest is to follow the mark breakdown in the spreadsheet, with grading product and grading process sheets...and of course the descriptions of both may be intermingled in your report...

Grading Process...assume
- 30-50 pages excluding code/API documentation, (a page is about 250 words of size 12 Arial, Calibri or Times New Roman font as they seem to work across platforms and should save you reformatting)
- proportional to marks for documentation
- so that translates to about 100 words per mark on average for the grading process assuming 40 pages
- and the accounting section can

Grading Product.
- I'd estimate you could describe the actual final product, and what it does in far less,
let's say 10-20 pages max, about 1/2 to 1/4 the length of describing the process, so that translates to around 20-50 words per mark for that product documentation.

Wrapping up and How long is a piece of string!?

People are always looking for specifications as to how long the documentation should be, and these guidelines exclude specific code & API documentation, which should accompany the code.

However, in the course of the report, reference can and should be made to any particularly interesting approaches or solutions, with code and accompanying explanations in the report.

As a matter of urgency, please prioritise code and documentation with respect to interoperability to help other groups finalise their documentation on that aspect. Both the original documentation on your interface, and subsequent documentation on making pairs of projects interoperate should each only run to a few pages.

Documentation requirements should be fairly obvious, what would a typical peer, not in your group need to appreciate the product (code and design) and process involved in your group project. Here are some extra guidelines and suggestions, which you can use in conjunction with your own. If you are still absolutely stuck for a checklist or inspiration it may be helpful to refer to

- the Main Sequence of the project on a following page,
- relevant sections of the GradeAll spreadsheet uploaded with this
- and some papers on documentation in an agile setting, also uploaded with this

But common sense should apply for

- the group report,
  - product
    - documentation - what you did,
    - code – how you did it
  - process - how you decided and divided the work,
  - accounting – with graph of distributions of time to various activities vs time
  - weightings – publicly agreed fraction allocated to each group member
- individual –
  - unique contributions and angles on product and process
  - accounting
  - weightings – for privately disagreed fractions allocated to each group member!

Basically, documentation should be long enough to cover the subject and tie it all together, and that depends on the project, problems and approaches adopted.

These are only guidelines, you might get into detail or be on a roll, and write for the night, or be as dry as a desert, and give the bare minimum...whatever it takes.

Last year's guidelines are also uploaded for reference, but it was a different and bigger project, with slightly different emphasis on a relatively documentation heavy waterfall approach web apps, project management tools etc., whereas this year the emphasis was on a documentation light flexible agile approach with interoperability, for code, as groups and as a class, with smaller simpler code, but more flexibility, decision making, resulting in changes to aims and deadlines. So, this year, the product was simpler, but the process more challenging, which should make career easier afterwards.

Group Report:

In other words, while other projects may have independent documentation for various subdivisions, in our simplified case, it seems reasonable, to focus mainly on the following types of product documentation. However, you can submit whatever you like or have written, in addition to covering the following basic types of documentation, for the **group report**:-

- **code documentation**: accompanies the entire code base, so should be with it, suitable mostly for use & minor maintenance, whether
  - clear code
  - comments
  - associated documentation for code, API etc., whether generated manually or automatically
- **product documentation**: explains the function, operation and design of the product (not the process) and would be generally suitable for redesign or extension;
  - basically section 9 only of the Main Sequence on the next page
  -
- **process documentation**: in the face of
  - changes to requirements
  - agile adaption to goals with limited time,
  - realisation of limitations to technologies etc

  please give an outline of
  - the management process adopted,
    - communication and decision making,
    - issues in relation to changes in design and implementation;
    - work load distribution and practices... independent to pair programming et.c.
  - basically everything other than section 9 of the Main Sequence on the next page, but within a general group context, rather than an individual context.
  - accounting – with graph of distributions of time to various activities vs time
  - weightings – publicly agreed fraction allocated to each group member

Individual Report.

Additionally, for the **individual report,** you can highlight unique and individual contributions and perspectives, too detailed to include in the group report, such as

- initial individual plans and aspirations before groups were formed
- subsequent changes and compromises after forming groups, and why they were made...basically had to compromise and work together as a team, each working to their strengths and
- accounting – with graph of distributions of time to various activities vs time
- weightings – for privately disagreed fractions allocated to each group member!

Main Sequence

1. Initial own, private, personal, individual, understanding
    1. Plans
    2. Preferences
        1. Role / work preferences
        2. Language / IDE
        3. Management style
    3. Time estimates

2. Group meeting with lecturer there to attempt to
    1. clarify direction
    2. specify & standardise interfaces

3. Group formation

4. redo of 1 above
    1. Plans with a little more detail
        1. Overall software architecture
        2. Overall role allocation & decisions for
    2. More specific individual work/role Allocation
        1. Role / work preferences
        2. Language / IDE
        3. Management style
    3. Time estimates

5. Another Group meeting (as expected), with lecturer leaving to attempt to redo 2 above, in view of possible strict enforcement of interoperability and autoplay – to give appreciation of trying to get a group of groups to
    1. communicate : with conflicting (mis-)understandings,
    2. co-operate : with preferences and work invested

6. Optional - redo of 4 as in view of apparent enforced interoperability

7. Redo again, with relaxed requirements, with no enforced
    1. interoperability
    2. autoplay

8. Time constraints enforcing selective deployment / inclusion etc. how was it handled
    1. on marks
    2. on time
    3. on personal preferences
    4. on optimistic estimates of what could be done

9. Final documentation on Final product : for next Thur
    1. Design : what you eventually did, not how you got there
    2. Implementation : as above, but can comment on changes to choice of technologies etc
    3. User's guide
    4. Future Work : plans to finalise if you had time

10. Reflection & review : lessons learned… the only mistake, is not to learn from mistakes
    1. your initial plans, roles and estimates
    2. same for team
    3. finally adopted… lessons learned, reasons for differences with initial personal & group plans

11. Appendices
    1. accounting :
    2. Presentations

Individual Report – *auxiliary guidelines adapted slightly from previous years – only follow if you must.*

Please follow the structure outlined below in preparing your individual report. The individual report should be submitted separately to your group report. Individual reports that are not submitted separate to the group report will not be accepted.
Length guideline: whatever you think is necessary to report professionally on your part of the project; in other words : comprehensive, concise, informative and relevant reporting.

1. Individual Reports – the broad outline and breakdown of your activities is mostly available at a glance on spreadsheets, but the details should be given in your individual report. Give a brief textual overview of your main activities, contributions and outcomes, highlighting any issues not obvious in spreadsheets. For example, if you managed / designed / coded / tested / integrated front-end : GUI/middle/backend dB's/ what tools, methods, techniques did you use, and what were the main obstacles, surprises, workarounds, alternatives, successes and methods used to achieve those etc

2. Lessons you learned

   which is, after all, the main aim, and is not done on spreadsheets, although they may give you an overview & help formulate ideas about lessons learned!

3. Suggestions for future years

   probably in light of preceding... but each situation different

4. Attach the 2 private spreadsheets as appendices 1 & 2 (& Gannt if you want!)

- for your benefit
  - need to learn about scheduling, work (re-)distribution and load balancing
  - Gives historical progression, which you can reference
  - Affords 'at-a-glance' review to generate / validate lessons learned
  - Somewhat depersonalised role evaluation done on spreadsheet.
    - But will be consulted in the event of significant load imbalance.
    - (reviewed in the light of potential problems highlighted after lecture!)

- And for our benefit : can see at a glance
-

  (Please write text rather than just dumping logfiles of meetings etc. to us, which you can reference in appendices if relevant. No professional would present or accept logfiles in place of a readable account which could be used to present or justify the project to management.)

Group Report – ***auxiliary guidelines for reference only, not applicable here, do not follow.***

(All page estimates are approximate guidelines, but please write text rather than just dumping logfiles of meetings etc. to us, which you can reference in appendices if relevant. No professional would present or accept logfiles in place of a readable account which could be used to present or justify the project to management. All should contribute and help (leader / documentor) compile it. Please follow the structure and the sections outlined below in preparing your Group Report.

1. Project Overview - description of the existing methods to arrange meetings in UCC (2 pages)
2. Justification - a statement of the benefits of a user-accesible online web based system. (2 pages)
3. Statement of Requirements - Function/Non-functional and domain requirements, a simple Use Case Diagram and a narration of the proposed system (2 – 4 pages)
4. Planning
how and why roles were allocated between team members
first steps
outline planning
Gantt and (PERT – omit if not covered for start of project) charts –
risks
identification
avoidance/minimisation plan
technical problems that occurred - what were they, how great was their impact, how were they solved (group conflict and individual issues should be discussed in the individual report).
minutes of all meetings should be put into an appendix not included in the main body of the report (10 – 15 pages)
5. Design
GUI design and example's (either layout diagrams manually drawn or screenshots of the final system) (10 pages)
database design & ER diagram (2 -3 pages)
UML (Sequence, Activation and (if appropriate) Class diagrams) (10 pages)
Definition of all fields (i.e. inputs) and associated validation rules (e.g. Dates, what format is used (e.g. DD/MM/YY) and how does the system make sure that the user has entered a 'correct' date. (4 pages approx)
6. Software Deployment
target environment
deployment of components
client-server configuration
installation
updating
de-installation (3 pages)

7. Testing

test plan
test data
test results
log of bug report system should be placed in appendix B (3-6 pages)

8. User Manual
Beginners manual. System managers manual, Departmental managers manual all containing requirements, installation/setup, how to use, error messages (including logfiles where necessary) and comments on maintenance. In here would go details on the recommended browser(s), the browsers settings (i.e. need cookies and javascript), the web server used, how a system is set up from scratch, what needs to be backed up in a running system, how to setup an appropriate payment

system account. Basically all the information needed for someone to setup and operate the system. You can assume that the person who is the system manager knows how to setup a web site, an email system and a database. (10-15 pages)

8 b. Video...option to augment simplified manuals in section 8.
Various screen grab options exist for various platforms, which could considerably clarify and simplify manual generation.

9. Evaluation and Conclusions
evaluation of the product
evaluation of the product-development activity (incl team-work)
evaluation of any software used for project support
conclusions (7 pages)

Appendix A – Log of meetings
Appendix B – Log of bug system : if bugs are handled manually then a record needs to be kept of the bug, the version of the segment containing the bug, how the bug was discovered (testing/accident/etc.) and how the bug was fixed

Any other appendices consisting only of extracts you deem necessary and relevant to support your arguments in the report.

The “number of pages” is given simply as an indicator. The total maximum count of pages from the estimations given above is 83 pages (for the group report). If your report is at this boundary or well above then you are almost certainly providing unnecessary detail. On the other hand, if your total is around 45 pages or less then you should refer back to the outline suggested above and check if you are missing or underweight in any section.
If you carried out work that does not fit in with the headings above then include that work in an Appendix. Examples might be focus group setup and results or a more extensive guide to product maintenance.

IMPORTANT:
(i) We are awarding marks for the work you put in as a group and individuals. If you complete an enormous amount of work but neglect to put it in the report then it makes it almost impossible to award you the marks you deserve.
(ii) Mistakes are work. We all make mistakes - in this project a mistake is taken as part of the learning process and is therefore counted as work. So if you describe your mistake and discuss how you fixed it or bypassed it you will be awarded marks (we will not judge you on your mistakes because we also make mistakes - it's how you learn how to do it right next time).
(iii) Have your reports proofread by someone other than the author - it might even be better to find someone outside the group to read the report. This suggestion is not just for those students for whom English is not their primary language. If we find a report riddled with mistakes we have to wonder how much care was taken with other parts of the project, such as the code. The mistakes that really stick out are spelling mistakes, every modern word processor checks the spelling of a word once you complete it, if you ignore immediate feedback like this where else have you been lazy? In industry a high level manager is not going to look at your code but they will need to read relevant sections of your report. If they feel you are not competent to run a spell checker or committed enough to proofread the report how eager do you think they will be to give you more responsibilities ?
(iv) Omission of references will result in significant penalties. You have been given the freedom to use the work of others in whole are in part provided that the people involved have been given appropriate credit and you have obeyed the terms of the licensing agreement associated with the referenced work. If licensing is not clear then write to the author stating that you wish to use their work and put a copy of the correspondence in an appendix.

Report Format: The entire set of deliverables (documentation, maintenance information, group report, individual reports, libraries, etc.) should be supplied in a standard pkzip file.
You are required to create a contents/index.html document (or manifest) that would be placed at the top level of the directory structure and would provide a list of the contents of your submission and a hyperlink to their location in your file tree. Everything should be linked, as well as being clear from the directory structure so that graders do not have to waste time looking for material to grade, as it will result in omission of marks for you.
Such omissions will be heavily penalised because our own deadlines for mark submission to the exams office are very tight - each examiner will have to read and score 1,400 pages of reports.

Deliverables: When we are marking the deliverable we may attempt to try out your program to see that it works. Your report should contain a handbook to facilitate this. You should provide the required libraries and packages and your own servers (if you customised them). Make sure you tell us what we will need to install on our machine (JRE, PHP, etc). Servers external to the supplied environment (e.g. a mail server) should be referenced and their use explained. An attempt will be made to setup the environment for your project, install it and then launch it. You should therefore provide a sample DB populated with sample data so we do not have to start a DB from scratch. A file in 'SQL' format exported from mySQL via phpMyAdmin would be suitable. Again it's a good idea to give these instructions to someone outside the group to test that they work correctly.