MSc Project

MSc Computer Science
MSc Information Technology
MSc Advanced Computing Technologies

2014-2015

Overview

- Introduction
- Choosing an idea
- Getting a supervisor
- Writing your project proposal
- Literature review
- Writing the report
- Plagiarism
- Working on your project

Important Contacts

- Project tutor: Oded Lachish
- Program administrator: Liam Simmonds
- Program directors (CS, IT, ACT)
- Project supervisor
- Intranet pages

  `http://www.dcs.bbk.ac.uk/dcswiki/index.php/MSc_CS_project`

  `http://www.dcs.bbk.ac.uk/dcswiki/index.php/MSc_IT_project`

  `http://www.dcs.bbk.ac.uk/dcswiki/index.php/MSc_ACT_project`

Project Aims

- Why do we make you do a project?
- Main aims of the MSc project: offer students the opportunity to
  - plan and execute a major piece of programming work
  - critically present existing approaches, place their own approach in the wider area, and evaluate their contribution
  - gain experience in communicating complex ideas/concepts and approaches/techniques to others by writing a comprehensive, self-contained report
  - **develop their portfolio**

Organizational Matters

- Two important documents have to be submitted:
  - Project proposal
    - Due: two weeks after the end of the spring term
    - About 2000-3000 words in length, providing essential **background research**, **problem presentation** and **development plan** for carrying out project
  - Project report
    - Due: two weeks before the start of the autumn term
    - About 10,000 words in length, explaining what you did in the project: **design**, **implementation**, **testing** and **evaluation**
- More details about these documents in just a moment. . .

Submission of Proposal

- The proposal must be submitted electronically (for plagiarism detection) via Moodle
- The file name should begin with PROP_ followed by your surname and an initial and the programme (e.g., PROP_SmithJ_CS.doc)
- Plain text, Word, postscript, PDF, HTML, or RTF formats are accepted
- A filled-in project proposal form (can be obtained from the intranet page of the module) must be also submitted
- If you need the installation of particular software on Department machines, please note so on the form
  - If you're not sure about your software requirements, please contact someone from the systems group

Submission of Report

- You submit *two* copies of the printed report to the programme administrator
- The submission date will be recorded, late submissions can affect the project's grade
- There is no provision for formal extensions, however,
  - you can include a letter explaining the reasons for the late submission (within ten working days of the deadline), which the exam board may take into account
  - providing sufficient written evidence, you can defer your project to the following year — this has to be done **before** the submission deadline
- If you are working on a non-Department computer, you are responsible for back-ups

Submission of Report(2)

- You also submit an electronic copy of the report via Moodle
- Plain text, Word, postscript, PDF, HTML, or RTF formats are accepted
- The file name should begin with REP_ followed by your surname and an initial and the name of the programme (e.g., REP_SmithJ_CS.doc)
- This document will be run past the JISC plagiarism detection service

Assessment

- The proposal and the report are assessed by your supervisor and a second (occasionally third) marker
- The overall mark for the project is made up like this:
  - 20% for the proposal
  - 80% for the report

Assessment(2)

- The proposal is judged according to the following criteria:
  - Background research
  - Presentation of the problem
  - Plan for developing the solution
  - Presentation of the proposal
  - Any other aspect (optional)
- Some MSc programmes may have additional criteria. Check the programme intranet pages.

Assessment(3)

- The report is judged according to the following criteria:
  - Specification and design
  - Implementation
  - Testing, results, analysis, and critical evaluation
  - Presentation of report, documentation
  - Any other aspect (optional)
- Some MSc programmes may have additional criteria. Check the programme intranet pages.

Feedback

- After the supervisor and the second marker have marked the proposal, you'll receive their comments in en e-mail. You should take into account their comments during your work on the project.
- After the exam board in mid-November, you'll receive a letter from the Department telling you the result of your project
- You will also receive a copy of the examiner's comments on your project
- An "official" transcript of your results is sent out in January or February by the College
- One copy of your report stays with the Department, you can collect the other copy

Choosing an idea for the Project

- What is your goal?
  1. To build your IT confidence.
  2. An impressive CV.
  3. Implement an idea you already have.

1. Building IT Confidence

- Suggested approach

  1. List the technologies you have been or will be taught (programming languages, data bases etc.)

  2. Choose your idea so that you can practice this knowledge.

- Remarks
  - This approach minimizes risk.
  - The challenge is to demonstrate strong capabilities in the taught material.

1 Finding an Idea, Generic Advice

- **The projects on the MSc programmes don't have to be novel ideas.**
- **The work you do has to be your original work.**

This means:

- If you have a problem coming up with your own idea pick:
  - some application,
  - maybe a mobile application, or
  - a game you like to play.
- Pick anything that will get you started and enable you to practice the knowledge you've gained.
- Once you start working you or your supervisor may suggest other options.
- Your first choice is not irrevocable.

2. Impressive CV

- Suggested approach
  1. List the technologies you want to add to your CV.
  2. Choose your idea so that you can practice these technologies.
- Remarks
  - This is a high risk approach that requires a strong background.
  - When choosing specific technologies ensure their stability and the availability of sufficient support.
  - Before finalizing your choice, check that you can actually use the chosen technologies.
  - Take into account a steep learning curve.

3. Pre-existing Idea

- Check
  1. Can your idea be implemented?
  2. What are the technologies required for the implementation?
  3. Are these technologies available?
  4. Is it reasonable to assume that you will be able to use these technologies?
- Remarks
  - This is a medium risk approach.
  - Find a supervisor to help you answer the above questions.

3. Generic Advice

- **The projects on the MSc programmes don't have to be novel ideas**.
- Ensure you are aware of Intellectual Property issues when
  - developing code for an external company and
  - planning to use code for commercial purposes.
- There are lists of the topics of previous projects available.
- Copies of selected project reports are available in the library.
- This can give you an idea what to do, e.g., by expanding or extending a previous topic.
- It is your responsibility to ensure you have an idea.

Finding a supervisor

- Follow the slides on choosing an idea for your project.
- Pick a member of staff by:
  - having a look at the research interests of staff (published on their web pages),
  - choosing a lecturer whose module you have enjoyed
  - any other reason
- The choice does not need to be clear-cut
- Set a meeting, by e-mail, with the staff member of your choice
- The result of the meeting should be
  - You have found a supervisor
  - You have a recommendation for a more suitable supervisor
- You should find a supervisor as soon as possible.

Finding a supervisor

- Once you have found a supervisor send an e-mail (with cc to supervisor and postgraduate administrator) to the project tutor. The e-mail should contain the name of the supervisor.
- Wait for approval
- The allocation of supervisors is here:
  `http://www.dcs.bbk.ac.uk/r/doc/studentprojects.php`
- **Important: if you haven't received an explicit approval by the Project Tutor (me), you are not supervised!**

External supervisor

- Given permission from the relevant program director, it may be possible to have an external supervisor.
- In addition to the external supervisor a student **must** have a supervisor that is a staff member.
- This requires an explicit agreement of the staff member and the project tutor.

Writing Your Project Proposal

- The purpose of the project proposal is to demonstrate that you have put some thought into choosing your topic and you know what you are talking about
- It is not a full-fledged project report, but should cover the areas shown on the next slide

Writing Your Project Proposal(2)

- Your project proposal should consist of the following parts:
  1. A brief description of the topic and where it fits into the field
  2. An account of the current work/applied technology in this area
  3. A high level description of the proposed project. May include
     - Your vision of the results
     - The identification of the question you would like to answer
     - The practical problem you would like to solve
  4. A suggested means of answering this question/solving the problem. May include:
     - High level software architecture
     - Methodology and work plan
     - Technologies

  Remark

  - This does not mean that you have to have all the answers at this stage
  - This is more about how you plan to find/develop the solutions

Writing Your Project Proposal(2)

- Bullet point 3. from the last slide defines the *aim* of your project
- Bullet point 4. is about the *objectives* that support the aim
- Often these two points are too vague and imprecise
- You can use the SMART method to make them more concrete

SMART

- SMART is an acronym standing for:
  - Specific: be as specific as possible
  - Measurable: try to establish measurable indicators of progress
  - Assignable: even though you're working on your own, formulate objectives as if you assign them to someone else for completion
  - Realistic: state what can realistically be achieved within the budgeted time (and resources)
  - Time-related: set milestones for the objectives

SMART(2)

- The last two points may cause the greatest difficulty
- Your supervisor can give you some help in establishing the scope of the project
- Students tend to wildly underestimate the amount of time needed
  - Leave yourself an **ample margin of time** to deal with this
  - It is perfectly acceptable to **review the scope** of a project as you progress and make adjustments
  - It is a good idea to prioritize the objectives: start with the **core**, you can always add additional functionality if you have some spare time

Writing Style

- Your proposal should be well-structured and written in an understandable way
- As many of the aspects of good writing are also relevant for the final project report, they will be covered later on

Literature Review

- In order to hand in a proper project proposal, you need to do some background research
- Usually this involves having a look at existing systems, ideas, algorithms, and approaches
- The hardest part is getting started
  - Once you have identified a couple of sources, they'll point you to other sources
- Here we'll discuss how and where to start your search

Search Engines

- Web search engines like Google are a good way to find lots of online resources
- However, not all of these resources are reliable, so this can only be a starting point
- You might also have to try out different combinations of search terms before finding the right terminology
- There is a special service called Google Scholar for academic texts: `http://scholar.google.co.uk`

Libraries

- You can also have a look at what's present in the College library
  - You could look for an (introductory) textbook about the area you are interested in
  - The library also provides online resources (`http://www.bbk.ac.uk/lib/`)
    - This includes their whole catalog
    - and access to electronic versions of journals
- The College also has subscriptions to libraries of professional bodies, e.g., the ACM Digital Library (`http://www.acm.org/`)

Other Online Resources

- There are other resources available on-line where professionals discuss certain topics
- Some examples are
  - mailing lists
  - newsgroups
  - discussion forums

Your Supervisor

- Your supervisor
  - might be able to point you to some sources
  - give you some suggestions on what to read first
- Obviously, your supervisor will not do the literature research for you, but can help you in getting started
  - So don't expect complete reading lists

Some Further Suggestions

- You don't have to read complete articles or books
  - Have a look at the abstract or summary first
  - Browse the article/book if it seems to be interesting
- Don't overdo the literature research
  - There is a huge number of publications out there, nobody expects you to read them all (this could take years)
  - At some point you have to decide on what you want to do and write up your proposal

Aims of Project Report

- Your project report is not simply a project management report or system documentation
- The main aims of your project report are:
  - To present your project in a meaningful way
  - To demonstrate that you can produce a document written in a well-structured and intelligent way

Aims of Project Report(2)

- In more detail, it should demonstrate that
  - you can apply the techniques taught in the MSc programme to the problem you are addressing
  - you can explain your project and its background clearly and concisely to third parties who may not have expertise in the specific area of the project
    - You may assume that a reader has a Computer Science background
  - you can relate your project to the wider context of IT

Why Is This Important?

- At least one of the people marking your project will not have followed your project closely (2nd/3rd marker)
- If you are on the boundary between two different marks, a well-written project report can make a difference
- The external examiners on the exam board only have your project report by which to judge your project
- To get good marks for your project, you need to do both:
  - Produce the software
  - Turn in a good report
- There are projects that have been graded below their potential due to an indifferent or poor write-up

Physical Appearance

- A tidy, well laid-out and consistently formatted document makes for easier reading
  - Use word-processing software
  - Leave margins to allow for binding
  - Use headings for chapters, sections, and subsections consistently
- Quantity does not automatically guarantee quality
- Project reports need to be **concise**, clear and readable
- The assessment is not about page count or word count

Mandatory: Title Page

- The title page should contain
  - (obviously) the title of your project
  - your name
  - *MSc* YOUR PROGRAMME *project report, Department of Computer Science and Information Systems, Birkbeck College, University of London* YEAR
  - *This report is substantially the result of my own work, expressed in my own words, except where explicitly indicated in the text. I give my permission for it to be submitted to the JISC Plagiarism Detection Service.*
  - *The report may be freely copied and distributed provided the source is explicitly acknowledged.*

Table of Contents

- Gives the full headings of all chapters (and the sections within them) with the appropriate page numbers
- Page numbers should be right-margin aligned

Optional Sections

- Acknowledgements: if there are persons who you would like to thank for their support and help
- List of Figures/Tables: if you have used lots of figures and tables
- Abbreviation list: if you have used abbreviations (it's also a good idea to spell out the meaning the first time you use an abbreviation in the text)

Report Structure

- A good document structure takes into account
  - the purpose of the document (i.e., to report on your project)
  - its target readership (i.e., the examiners)
- Many reports are too long, too unstructured, and lack purpose
- You should aim for
  - continuity: order your material in such a way that a reader is able to follow your descriptions
  - completeness: do not leave out significant parts
- Find a balance: cover the important things without overwhelming the reader with unnecessary details

Report Structure(2)

- Unlike an essay, a report contains headings and subheadings (to make its structure explicit)
- Each subheading may be further divided into subsections or subdivisions
  - Usually it's a good idea to number each section and subsection
- To develop and improve the continuity, it helps to pay special attention to the report's structure

Starting to work on the report

- Before starting to write, think about the structure of your report (in outline or even just as subheadings)
- Suggested outline:
  - Abstract
  - Introduction
  - Background (may be a subsection in the introduction)
  - "Project Trailer" and specifications
  - Software architecture
  - Implementation
  - Testing
  - User Manual
  - Summary and Conclusions
  - References
  - Appendix: Code
- Remark:
  - The final structure must be adjusted to the project.

Abstract

- The abstract is a brief synopsis of your work, a bit like an executive summary
- It should be no longer than about 250 words
- It's usually a good idea to write this at the end (when everything else is known)
- Beneath the abstract, put the name of your supervisor

Introduction

- Contains a brief outline of the topic as a whole
- Then state the aim and objectives of the project
  - What was the purpose of the project and what did it set out to investigate?
- At the end of the introduction, provide a road map for the remainder of the report

Background/Literature Review

- This chapter should focus on the context that you are operating in, e.g., by describing
  - typical applications
  - alternative tools and development approaches and how they have been used in practice
  - alternative systems and what they achieve and do not achieve
- This should be a synopsis of the relevant part of your project proposal (do not just copy your proposal)
- Restrict yourself to what's relevant to the specific context of your project (the proposal can have a more general look at the state-of-the-art)

"Project Trailer" and Specifications

- An exhibition of selected features of the software you developed:
- It is recommended to add screen-shots of interesting stuff.
  - the login page and long menus are usually not interesting.
- A specification of your project (details that you didn't show in your project trailer)

Software Architecture

- High level description of the system
- n-tier architecture
- Key parts of each layer
  - how you applied design patterns (how they fit together)
  - It is recommended not to have **overcrowded** or **under-crowded** UML diagrams.

Implementation

- Software development process
- Describe the technologies used in the project, why they where chosen and what were the other options:
  - Tools and programming languages.
  - Data-base, servers etc.

Testing

- Describe the verification process you used in your project
  - Unit Testing, Static Analysis etc.
  - Manual testing, Selenium, Cucumber etc.

Summary and Conclusion

- The final section
  - summarizes the project as a whole
  - A critical evaluation by the student, emphasizing
    - strong points and weak points
    - lessons learnt
    - design decisions which, looking back, would be made differently
    - ways in which the project could be improved or extended
    - etc.
  - recommendations for the project
- You can also describe possible future work in the area of your project

References

- You have to provide a complete list of all the works mentioned in the text
- For a book, this normally includes the name(s) of the author(s), the title, the publisher and date of publication
- For an article, it would include the name(s) of the author(s), the title of the article, the name of the journal, the volume/issue number and date and page numbers
- Examples:
  - Bloggs F, Advanced Widget Design, Gargoyle Press, 2002
  - Hardy O and Laurel S, "A software approach to fine mess avoidance", Journal of Disaster Studies, Vol 4, 2000, pp 123-134

Appendices

- Additional relevant material which did not make it into the main sections should appear in an appendix
- It can also include lengthy items such as
  - program code
  - raw data
  - detailed statistical analysis
- If you have very lengthy items, you can include a CD or DVD that contains these items
  - Usually, there's no need to print out the complete source code

Adjusting Structure

- You should avoid situations where the reader needs to jump forward in order to understand something.
- The length and detail of each section should be proportional to its importance.
- Try to arrange the sections from most interesting to least interesting.
  - This is one of the reasons you want the project trailer as early as possible.
  - Example: if for some reason your testing is very interesting, try to push it forward.

Presentation Techniques

- Top-down
  - Usually the proper technique for project specific information, for example,
    - in the introduction: start from the context of project and work your way towards the details of the project
- Bottom up
  - Usually the proper technique for technical information, for example,
    - when describing the programming language chosen: start with the choice, then explain why and finally mention other options
- Linear Story
  - Usually the proper technique for the "project trailer"
    - describe how the software you developed is used in a linear manner.

First Attempt, Initial Project Structure (Joseph Morrison)

1. A brief description of the topic and where it fits into the field
   This has to come from two angles; the computing framework angle and the automated framework angle
   For brevity I will start with small introduction on computational finance
   I will write about automated trading with in the financial industry
   I will talk about the continuing growth of the automated trading industry
   I will give an introduction about retail trading and the wider public access to the markets. This could then be referenced as a "target audience for the framework"
   I will talk about both the fundamental and technical analysis.
   I will then talk about this in the context of foreign exchange and equities.
   Noting that automated trading is applied in a huge spectrum of markets but focusing on these two.
   I will follow this with an explanation on how floating exchange rates were introduced and give a short back ground on the London stock exchange (not the history of stock markets)
2. An account of the current work/applied technology in this area
   The three topics I will focus on are: High frequency trading (HFT), MAN Groups AHL fund and direct market access (DMA) I will talk about high frequency trading and the regulation surrounding it...

Second Attempt, Initial Project Structure (Joseph Morrison)

1. A brief description of the topic and where it fits into the field - in this section I will:
   - Introduce the two topics of computing framework and automated framework
   - Start with small introduction on computational finance
   - Write about automated trading and its role in the financial industry
     - Including retail trading
   - Talk about both the fundamental and technical analysis.
   - Introduce the foreign exchange and equities markets. Noting that automated trading is applied in a huge spectrum of markets but focusing on these two.
   - I will follow this with an explanation on how floating exchange rates were introduced and give a short back ground on the London stock exchange (not the history of stock markets)
2. An account of the current work/applied technology in this area
   - Focus on three topics: High frequency trading (HFT), MAN Groups AHL fund and direct market access (DMA)
   - High frequency trading and the regulation surrounding it...

Third Attempt, Initial Project Structure (Joseph Morrison)

1. A brief description of the topic and where it fits into the field
   - Automated Trading
     - Its role in the finance industry
     - Include the retail industry
     - High frequency trading
     - Direct market access
     - A framework to accommodate the ability to access the markets
   - Fundamental and technical analysis.
     - What they are
     - How they are used to trade the markets
   - how it has changed with computational finance
   - Foreign exchange and equities markets.
     - Foreign exchange moving to a floating exchange rates
     - Lightweight introduction on the London stock exchange
     - Briefly mention the others markets
2. An account of the current work/applied technology in this area
   - MAN Groups AHL fund
     - Its success - starting in 1987
     - Recent poor performance
     - Research into systematic trading
   - Co-location ...

How to Write the Introduction in a Top Down Approach

1. Start with the top level, for example:
   - Dedicated social network project: Start an overview of social networks; Do not start with a long description of the internet.
   - time management application: Start with an overview of time management applications; Do not start with a long description of applications in general
   - Lowest level that is sufficient for explaining the context of the project.
2. Explain at a high level what you plan to implement.
3. Unless really essential, the high level should not include technical details and exact specifications.

Advice for related work

1. Select about three similar application.
2. Describe each application shortly.
3. Elaborate on the application closest to your project. You may want to compare it to what you plan/achieved.

Remarks

- If your project is unique then explain this instead of doing the above.
- You should not have a long list of similar applications, each with its own description. If you feel you need to have one, then minimize the extra details, ideally have a list with names and references

Avoiding Common Pitfalls – Style

Be to the point! Be explicit! Be concise!

- Starting sentences with 'In my opinion', 'I think', or 'When I did'
  - An examiner is not interested in your opinion, but in how you support your argument
- Using the phrase 'It is obvious'
  - What is obvious to you might not be obvious to someone else
  - If it is really obvious, you can explain it in a few words
- Broad generalizations
  - 'All generalizations are untrue'

Plagiarism

- Unfortunately, there have been a few cases of plagiarism recently here at the Department
- This is an important topic, as the penalty for plagiarizing work can be very severe
- Plagiarism is using words and ideas from another text without proper acknowledgement
- The College's Policy on Assessment Offences lists examples (`http://www.bbk.ac.uk/reg/regs/`) and penalties

Examples

- Plagiarism can take many different forms, the policy mentioned above gives examples:
  - copying a whole or substantial parts of a paper from a source text, without proper acknowledgement
  - paraphrasing of another's piece of work closely, with minor changes but with the essential meaning maintained
  - piecing together sections of the work of others into a new whole
  - procuring a paper from a company or essay bank
  - submitting another student's work, with or without that student's knowledge
  - submitting a paper written by someone else, and passing it off as one's own
- **You should not quote large chunks of text from another source!** doing so may raise a Copyright issue.

Trying to Get Away With It

- It is not that easy to pass something off as your own work
  - Your report will be run through a plagiarism detection software
  - Most supervisors have seen plagiarized work before and can identify typical give-aways
  - Markers will also be familiar with the main ideas and publications in certain areas
  - We get alerts if suspicious jobs are posted on the net

Using Existing Material

- However, nobody expects you to reinvent the wheel
- You are allowed to use existing work as a basis for your project, provided that
  - you reference this work properly
  - there is enough of your own work in the project
- **You should not quote large chunks of text from another source!** doing so may raise a Copyright issue.
  - in that case it's better to summarize it in your own words and reference the source

Using Existing Material(2)

- When in doubt, ask your supervisor
- See
  `http://www.bbk.ac.uk/mybirkbeck/services/facilities/support/plagiarism`
  for College guides and policies on plagiarism

Working on your project: Project Report

- Start to work on the project report as soon as possible. If possible, even before you start coding.
- Use an horizontal approach
  - Start with the structure of the report.
  - Proceed with the structure of each chapter etc.
  - At the paragraph level, write what you plan to write in the paragraph.
  - Write the actual text when it is clear if it is required and where.
- Remember
  - Writing actual text may require significant effort. So, doing so prematurely may result in a reluctance to remove or move redundant or misplaced text.
  - Meta-text is easier to move around.
  - Parts of the introduction and background may require early writing, since in general writing them is far from easy.

Working on your project: Coding

- Start as soon as possible.
- It is suggest to use an incremental approach:
  - Start with the minimal implementation that uses all the required technologies. For example,
    - A GUI with one button and one text box. When the button is pressed something happens.
    - Proceed by involving the data base
  - Once you have all the technologies working together start adding features and refactoring.
- Advice:
  - If you started to work on the report you can choose which features to add according to what makes your report look better.
  - It is better to implement some features two weeks before the deadline, than to have a million features and two weeks to write the project report.

Common Pitfalls(1)

- Pushing in the wrong direction
  - To avoid this problem, arrange meetings with your supervisor and show up on time.
- Getting stuck
  - If you are stuck for any reason (and have no meeting scheduled), let your supervisor know immediately.
- Your supervisor can do nothing for you if they are unaware of your situation.

Common Pitfalls(2)

- Trying to satisfy an external customer at the expense of your grades
  - This is especially true for work-related projects
  - Do not let outside interests interfere with your project
  - The guidance for your project should come from your supervisor

Common Pitfalls(3)

- Over-/under-ambition
  - Try to be realistic what you can achieve, a good project requires a lot of effort
  - However, it is better to do a smaller job well than it is to fail to do a big job at all
  - Ask your supervisor for guidance on the scoping of the project

Common Pitfalls(4)

- Failing to plan a fallback position
  - Have a plan B if you are not able to complete the planned work in time
  - Try to plan your project in stages, so you have a complete stage to fall back on

Common Pitfalls(5)

- Inadequate literature review
  - References should cover the relevant theory and/or technology that you use
  - The literature review should demonstrate that you have an understanding of the current state-of-the-art and show how your project fits into it
  - Cite your sources properly (more on this in the section on plagiarism)

Common Pitfalls(6)

- Deliverables of unknown quality
  - The work appears sound but there is no evidence of its validity
  - Include summaries of test results
  - Demonstrate an executable to your supervisor

Common Pitfalls(7)

- Deliverables of unknown origin
  - Sometimes a project seems to be of good quality but too extensive to have been done in a few months
  - If work existed before the start of the project, define what existed and who produced it
  - Nobody expects you to reinvent the wheel, but you have to document where pre-existing work comes from (more details in the section on plagiarism)

Common Pitfalls(8)

- No acknowledgment of sources
  - Similar measures are applied to the report: you have to cite your sources
  - For every part of your report it should be clear
    - if this was the result of your own work
    - or if it has an external source

Common Pitfalls(9)

- Perfectionism
  - Don't be too hard on yourself: try to avoid perfecting each and every task
  - A 'good enough' project is better than the promise of unfinished 'perfection'
  - Sometimes you just have to get on with it

Summing It Up

- Working on your project may be stressful at times (it's not going to be a 'walk in the park')
- However, if you experience exorbitant amounts of stress, this is usually as sign of things going wrong
- If you have any worries about your project, discuss them with your supervisor as soon as possible
- Additional reading
  - *AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis*, by William J. Brown, Raphael C. Malveau, Hays W. "Skip" McCormick, Thomas J. Mowbray.
  - *Antipatterns: Identification, Refactoring and Management*, by Phillip A. Laplante, Colin J. Neill.