**Reading**

The Student Handbook describes the organisation of this module in detail. Everyone *must* read this very carefully!

The following are relevant books that you may find useful. Some are in the library already, some may be not, if you want to read a particular book but it is not yet available in the library, please let us know, we will order it.

|  |  |
| --- | --- |
| http://www.cs.nott.ac.uk/%7Enhn/G52GRP/Pics/Bot05.jpg | The Professional and Quality Issues (PQI) part of the module is based on the book *Professional Issues in Information Technology* [Bot05] by Frank Bott, which is recommended reading. |
| http://www.cs.nott.ac.uk/%7Enhn/G52GRP/Pics/Daw05.jpg | The book *Projects in Computing and Information Systems: A Student's Guide* [Daw05] by Christian W. Dawson discusses various aspects of carrying out a project, such as planning, risk management, basic software engineering issues, and presentation. It adopts a student's perspective (whereas many other books on software projects focus on large-scale, industrial projects), and it contains information relevant both for group projects and individual projects. While it is not covered directly in any of the lectures, it would be worth to have a look at it for some concrete advice on how to manage student projects, and as background reading for the (guest) lectures on project management. |
| http://www.cs.nott.ac.uk/%7Enhn/G52GRP/Pics/CSFP08.jpg | I highly recommend the book *Version Control with Subversion* [CSFP08] by Ben Collins-Sussman, Brian W. Fitzpatrick, and C. Michael Pilato if you are going to use Subversion, or even to just get an understanding of what version control systems are, and why they are useful. The book is very well written and it can both be bought if you want a hard copy or downloaded for free. And yes, you should use a version control system for your projects! |
| http://www.cs.nott.ac.uk/%7Enhn/G52GRP/Pics/McC04.jpg | Dr. Steven Bagley recommends *Code Complete 2* [McC04] by Steve McConnell:  “It's a nice book looking at how to write large software well covering both the design and coding of the project. It's intensely practical and I find you can just dip into a particular topic and come away with useful knowledge — including a lot of examples of bad practice and explanations of why its bad. There's a website for the book at [www.cc2e.com](http://www.cc2e.com) which lists the contents.” |
| http://www.cs.nott.ac.uk/%7Enhn/G52GRP/Pics/Mar08.jpg | Our friends at [Esendex](http://www.esendex.co.uk/) and [7digital](http://www.7digital.com/), experts on agile software development, warmly recommends *Clean Code* [Mar08] by Robert C. Martin. Paul Shannon, 7digital, says:  “Robert C Martin, or 'Uncle Bob'; as he's better known, has written a definitive guide to quality code in this book. It provides a gateway to the world of software craftsmanship and helps you on your journey to writing high quality software that can be easily maintained, is easy to read and simple to change. You'll learn that comments are considered a bad idea and why tests and good naming will always win the day, but also how best to handle errors, use emergent design and learn tricks around concurrency and refactoring. Most of all though, the book contains a complete guide to detecting code smells so you'll know what bad code looks like and why it's important to clean it up!”  Neil Kilbride, Esendex, says:  “Poorly written code may function correctly but costs organisations time and money. The importance of code quality is presented in Clean Code by the original software craftsman Robert C. Martin (Uncle Bob). This book uses clear examples to outline the principles, patterns, and practices of writing clean code. This book will make you a better software developer.” |
| http://www.cs.nott.ac.uk/%7Enhn/G52GRP/Pics/Dup98.jpg | The book *BUGS in Writing: A Guide to Debugging Your Prose* [Dup98] by Lyn Dupré provides plenty of concrete and useful advice on how to improve various aspects of your writing. It focuses specifically on technical and scientific writing, with many examples from Computer Science. It is also very enjoyable to read. Not only useful in the context of the group project, but also for other report writing such as your individual dissertations and almost certainly also later in your professional careers. |
| http://www.cs.nott.ac.uk/%7Enhn/G52GRP/Pics/KD03.jpg | You may be considering using LaTeX for writing your group project (or other) reports. If you do, I can highly recommend *Guide to LaTeX* [[KD03]](http://www.cs.nott.ac.uk/%7Enhn/G52GRP/#KD03) by Helmut Kopka and Patrick W. Daly. Shown here is the fourth edition, but the third edition is also OK. While the learning curve of LaTeX can be a bit steep, and while LaTeX certainly isn't perfect in all ways, on balance, it works quite well for structured technical documents, largely automates tedious but important aspects such as managing references, and it really helps with producing a finished result that is up to top-notch typographical standards. In addition, it works quite well for co-operative writing as everything basically is just text files, just like program source code, and thus can be put under effective version control. This book makes the learning curve substantially less steep and and also provides concrete examples showing how to do most things you might want to do, and how to work around common problems. While on the topic of LaTeX, there are sophisticated IDEs for LaTeX providing good error reporting and quick previewing, which helps reducing the learning curve. There are even on-line services, such as [writeLaTeX](https://www.writelatex.com/) or [ShareLaTeX](http://www.sharelatex.com/), providing similar functionality as well as support for collaborative writing. Like Google Docs, but for LaTeX. |

[Bot05] Frank Bott. *Professional and Quality Issues*. The Britsish Computer Society, 2005. ISBN 1-902505-65-4

[CSFP08] Ben Collins-Sussmann, Brian W. Fitzpatrick, C. Michael Pilato. *[Version Control with Subversion](http://svnbook.red-bean.com/)*. O'Reilly Media, 2008.

[Daw05] Christian W. Dawson. *Projects in Computing and Information Systems: A Student's Guide*. Addison Wesley, 2005. ISBN 0-321-26355-3

[Dup98] Lyn Dupré. *BUGS in Writing: A Guide to Debugging Your Prose*. Addison Wesley, 1998. ISBN 0-201-37921-X

[KD03] Helmut Kopka and Patrik W. Daly. *Guide to LaTeX*, fourth edition. Addison Wesley, 2003. ISBN 0-321-17385-6

[Mar08] Robert C. Martin. *Clean Code: A Handbook of Agile Software Craftsmanship*, first edition. Prentics Hall, 2008. ISBN 0-132-35088-2

[McC04] Steve McConnell. *[Code Complete: A Practical Handbook of Software Construction](http://www.cc2e.com/)*, second edition. Microsoft Press, 2004. ISBN 0-735-61967-0