

L1: Introductions

Dr. Julie Greensmith

julie.greensmith@nottingham.ac.uk



Welcome!

- Introductions
- What is software engineering?
 - why you should care
- Overview of the module
- Lectures and Moodle
- Methods of Assessment and Labs
 - coursework and exams



Who am I?

- Undergrad studies in Biomedical Sciences & MSc in CS @ Leeds
- Junior Software Engineer for HPLabs Bristol
- PhD in Artificial Intelligence for Computer Security @ Notts
- Currently lecturer in CS
- Soprano Cornet in Ilkeston Brass



What is Software Engineering

-apart from the first two pages of waffle in every Software Engineering text book!!
- No agreed definition, but generally involves (big shock) the **engineering of software**
- Software first conceptualised by Charles Babbage for his programmable analytical engine
- A unified process or method used in the construction of software systems



Your unrealistic scenario

- Have to work on a piece of code only once
- Maximum of a couple of thousand lines
- Your bugs are unlikely to kill anyone
- Your errors are unlikely to cost anyone money
- Can sit in your own little bubble and do your own thing
- *The reality is the opposite...*



The real scenario of SE

- Have to work on a piece of code **many times**
- A maximum of a couple of **million lines**
- Your bugs may **kill the users**
- Your errors **will cost your company**
- Can **not** sit in your own little bubble and do your own thing, **will be expected to work as a team**



Experiences in industry

- In industry you will be working with a team
- You will be working to someone else's deadlines
- It is expected that you understand this jargon and the standard methods for development
- Different companies use many different methods
- Source code is often edited by multiple people, across lengthy time periods and *customers!*



When software goes bad...

- As software grows, so too does its complexity
- Incorrectly engineered software can lead to tragic consequences:
 - Ariane 5 - it went bang!
 - NHS Infrastructure - cost us MILLIONS!!
 - Beagle Mars Rover - landing system bug



What we will learn

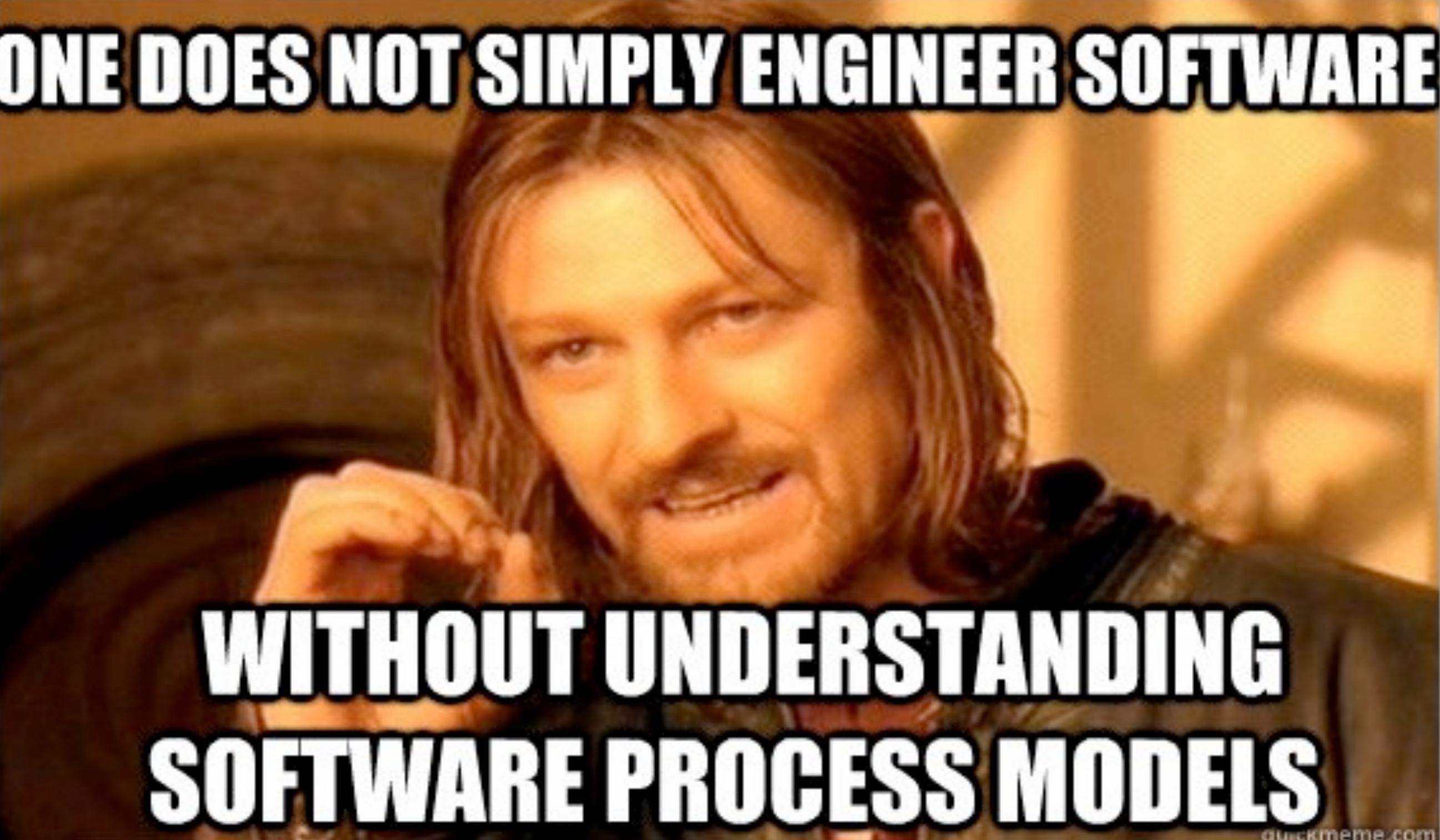


The Software Engineering Process

The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software
- Wikipedia



ONE DOES NOT SIMPLY ENGINEER SOFTWARE



**WITHOUT UNDERSTANDING
SOFTWARE PROCESS MODELS**

quickmeme.com

There are numerous stages

Requirements analysis

Software design

Implementation

Integration

Testing

Maintenance



In this module...

- Introduce the software lifecycle and place SE in context within a CS framework
- Give a taster of various different methodologies
 - to select the most appropriate one!
- Experience the practicalities of SE
- Use tools which assist with the process
 - version control and debugging systems



Why you should care about this?

- The majority of our students get jobs as software engineers
- A lot of software is rubbish, we'd like this to be better in future
- The principles taught this module apply to large systems, much bigger than anything you will write for your degree



Module Structure

- We have 11 weeks = 16 taught lectures, 2 revision
 - pair up (roughly) with the first chapters of each section of *Software Engineering by Sommerville*
 - plus debugging and collaborative working sessions
- Theory reinforced in lab sessions
 - practical individual and pairs exercises
- Use of online interaction with moodle
 - make sure you can log in!



Oh and you might want...



Ian Sommerville
Software Engineering
8th Edition
2006

Addison Wesley
About £30

9th Edition out on Kindle
About £20



Don't underestimate the need to read!

- In the “olden days” when I was an undergrad we were taught to “read for your degree”
- Not necessarily books, but it doesn't hurt!
- Lots of resources online
 - including industry whitepapers
 - don't forget technical developer forums



We are studying this as software is still rubbish!



What we will cover

- The aim of this module is to provide a general understanding of Software Engineering
- The typical phases of the software lifecycle, practical specification, design and testing
- Prepare students for software development projects undertaken throughout their studies
- Appreciation of software development methodologies



Learning outcomes

(stuff you should know by the end)

- **Knowledge and Understanding:** The different approaches to managing the software development process. The practice of producing specifications from informal briefs. Knowledge of how to test, debug and change programs.
- Taking a project from start to finish
- Keeping control of your source code



Learning outcomes

(stuff you should know by the end)

- **Intellectual Skills:** *Understand how to represent formal program requirements. Understand how to create and deploy an effective plan for testing software systems.*
- Writing a software requirements specification
- Producing testing documents
- Learning to estimate how long tests will take



Learning outcomes

(stuff you should know by the end)

- **Professional Skills:** *The ability to apply software engineering methodologies in practical scenarios. The ability to evaluate, select and deploy appropriate tools and techniques.*
- Learn the different methods and what they are good for
- Picking the right methods for a particular task
- Knowing the ‘language’ and terminology of SE



Learning outcomes

- **Transferable Skills:** *The ability to deploy a software development methodology, and to test and debug software, independent of programming language.*
- To be able to work as a software developer
- To be a good member of your team despite what language or method you are given to work with
- To be a SE pro by your 3rd year :-)



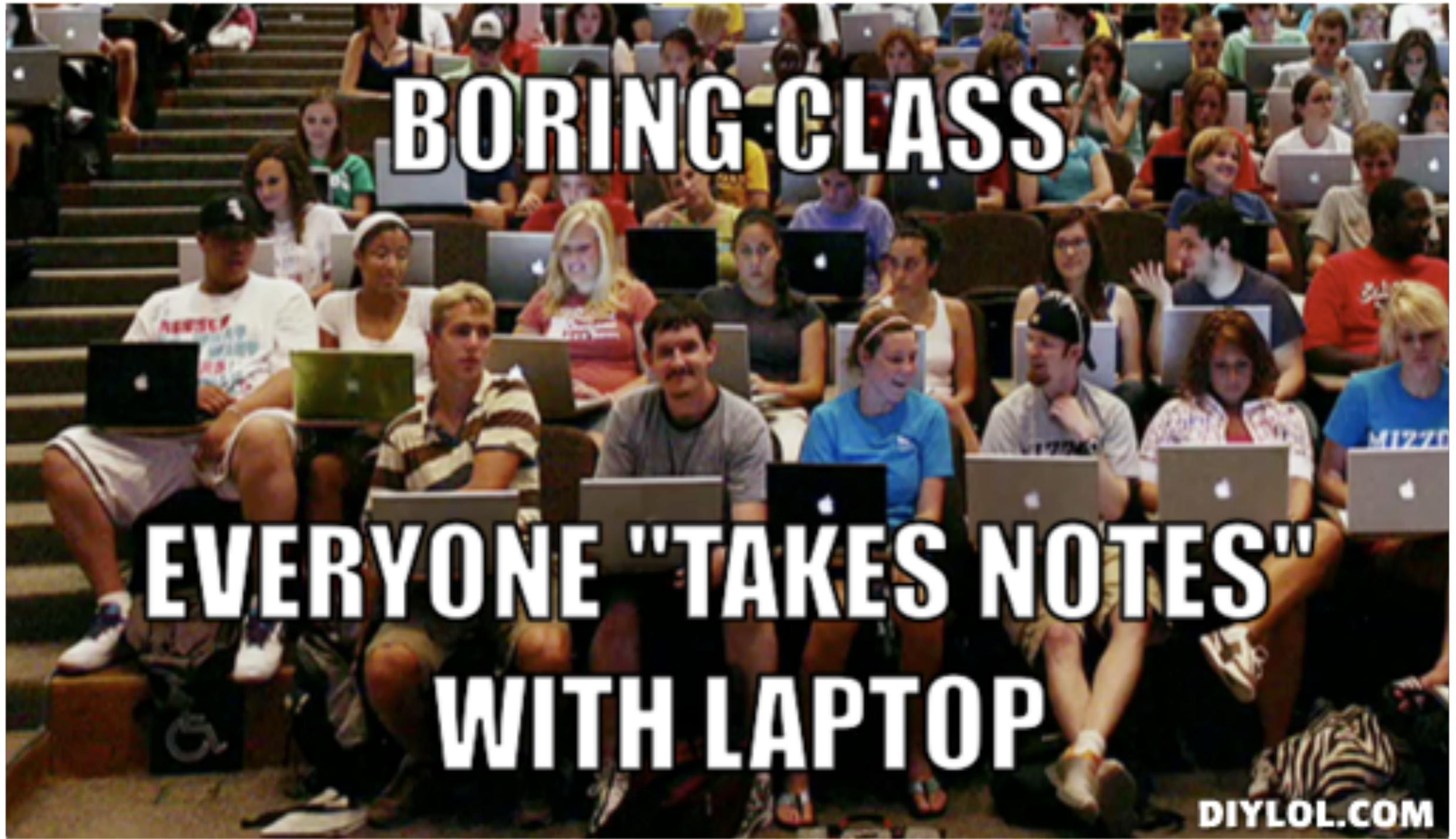
Lectures and Support



Lectures are instructive

- *What's the point of going to lectures if I can just download the slides?*
- Go over what you need to know, not necessarily teach you everything!
- Take note of which topics you need to study
 - processes to understand
 - things which are likely to come up on the exam
 - opportunity to ask questions formally or informally





My job in these lectures

- To highlight and explain key concepts
- To clarify and pull together information not contained within a single source
- To provide support in helping you *learn to learn*
- To provide practical examples of taught theory
- To turn up on time for all lectures!
 - *I expect you to turn up on time too - else I get grumpy!*



I will do my best to be on time

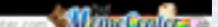


Timetable changes

- Lecture A: Monday 2pm, C3
- Lecture B: Wednesday 4pm, LT2
- Labs : Fridays 4-5pm
 - things to try out in pairs
 - game dev tutorials
- *Lecture attendance is highly recommended*
- ***Lab attendance is compulsory***



Time
Use It Wisely

memecenter.com 



Online support for FSE

- We will use Moodle to deliver online content
- Please please make sure that you can see the resources - if not please see me!
- Contains the lecture slides and online material
 - there is no other official web page for this module!!
- Please discuss ideas amongst yourselves using the news forum on Moodle
- Fingers crossed it will work!!



Dont forget to show them moodle!!

Use it responsibly!

- These services are there for your own good!
- Inappropriate language, posting or behaviour will not be tolerated. I mean it. I name and shame.
- It would be great if you could contribute but I'm not forcing anyone!
- I will post snippets and blogs here too
- *Remember: dont post anything that you wouldn't say to someone's face!*



Meeting me

- I am available in the timetabled lab session to answer questions and at 3pm Mondays in B34
- Happy to answer questions in the forum
- **I dont accept drop-ins at my office!**
 - make an appointment via **email** if you wish to discuss any issues with me
 - if you turn up at my office, I will ask you to make an appointment



Assessment & Labs



Assessment

- According to Saturn:
 - 70% - 2 hour written examination
 - 30% - Coursework assessment through assessed lab portfolio
- If you don't turn up for the labs, you will not pass this module.



Exam assessment

- 2 hour written exam
- Three compulsory questions
- First question will involve you designing “something”
- Longer questions to test your knowledge



Past papers

- Amalgamation of two modules in 2011:
 - G51ISS and G51IISE
- Examine similar content but will be presented in a different style
- Any practice questions on a subject will form useful revision
- More practice questions given nearer the time
- We will go over recent past papers at the end of the course



Labs based coursework

- You guys have enough coursework already!
- Hence, we are combining coursework with labs
 - work together in pairs on your submission and complete the relevant tutorials
- Please let me know if you have ECF or are off ill via email or school office
- You must attend **six out of eight** to get the 5% attendance marks for your coursework



A note about labs this week

- Coursework completed in pairs
- Come along and sign up, find a partner
- Make sure you can log into Moodle and see the FSE course notes and material
- Contribute with a hello message to the forum
- Complete the preliminary exercise as an icebreaker



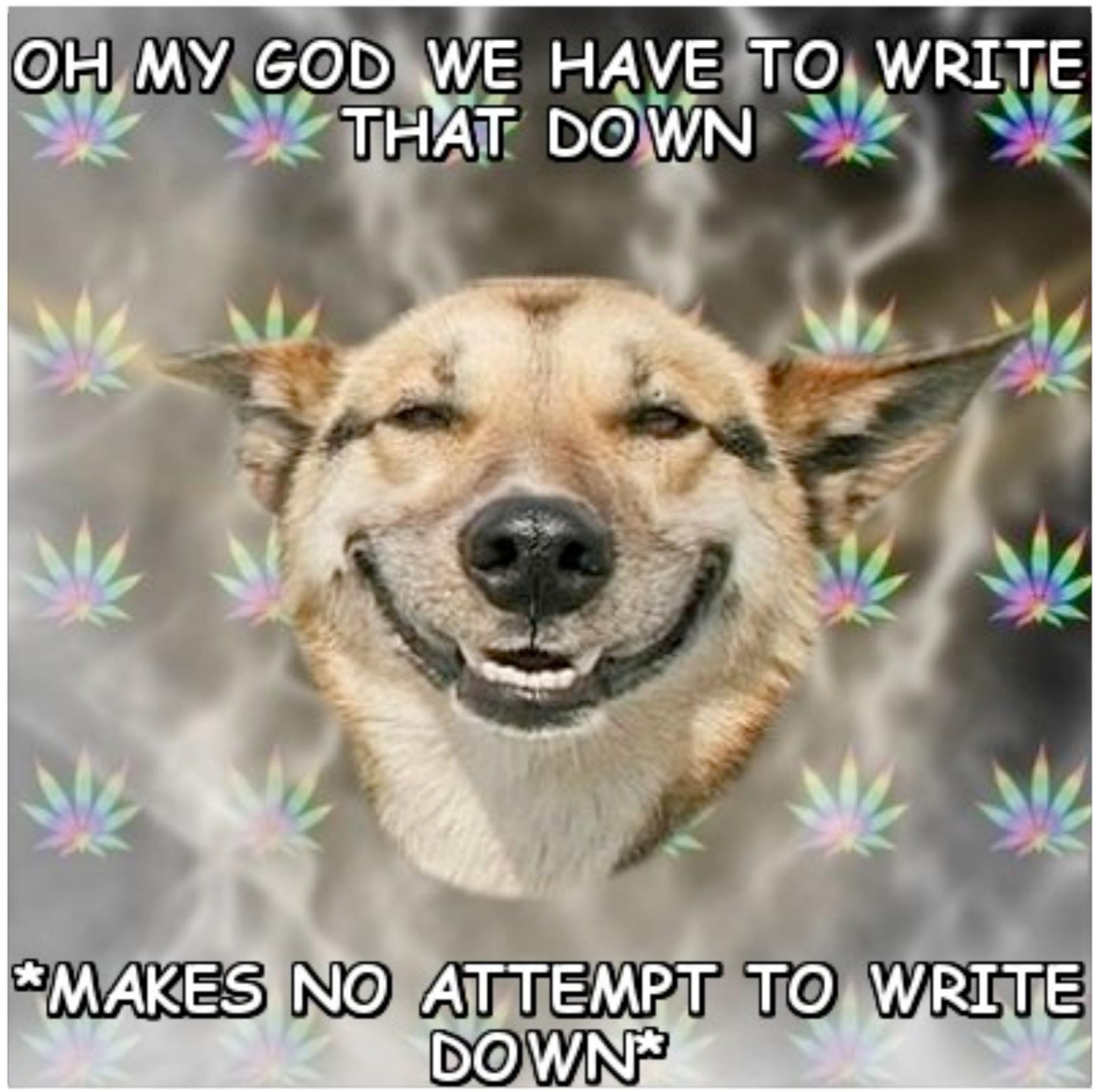
and finally...



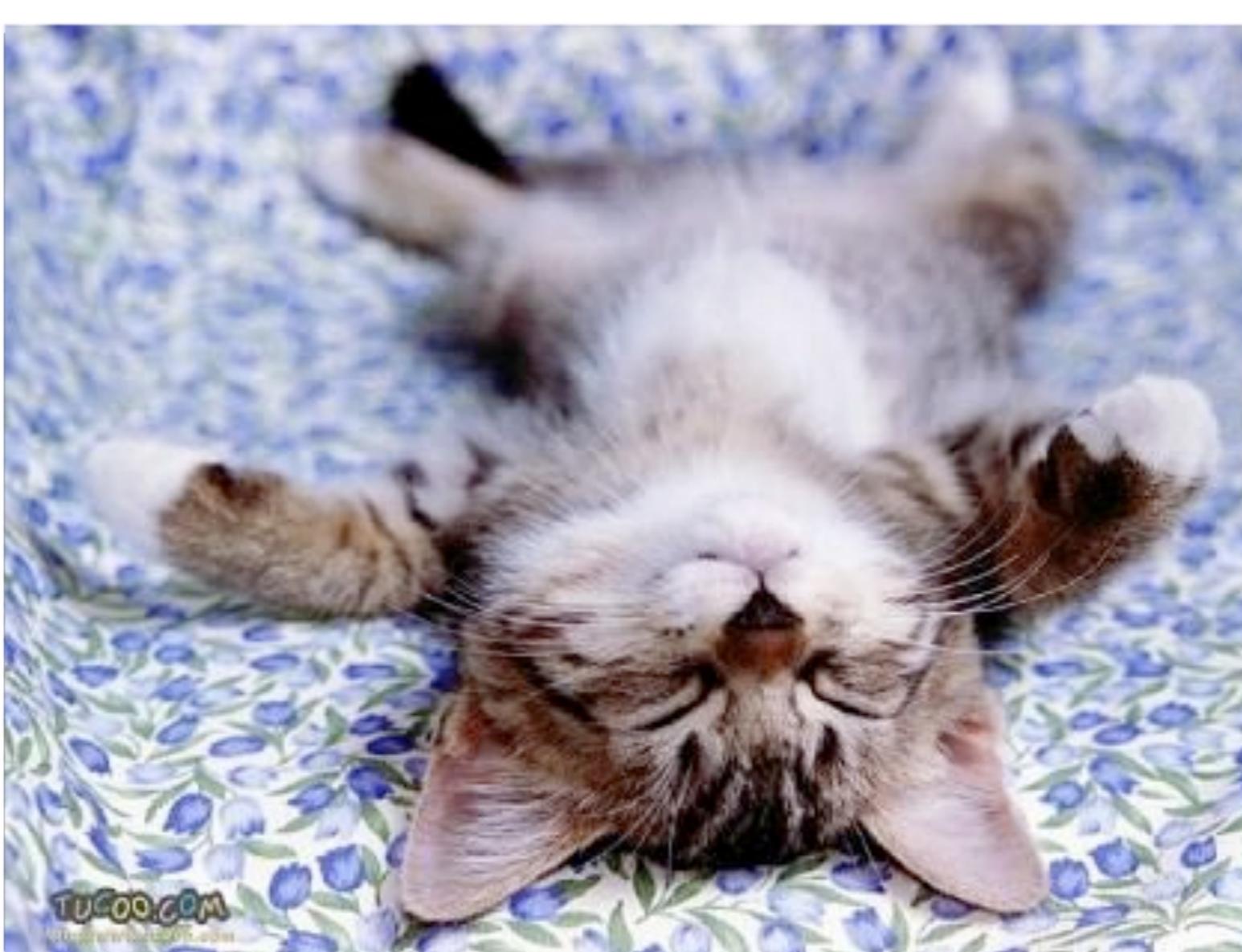
...it should be fun!

- SE should be a rewarding process
- Give me feedback throughout the course
 - if I'm going too fast
 - if I'm going too slow
 - if its too theory based
 - if its too practical
- These are the essential skills you need!!



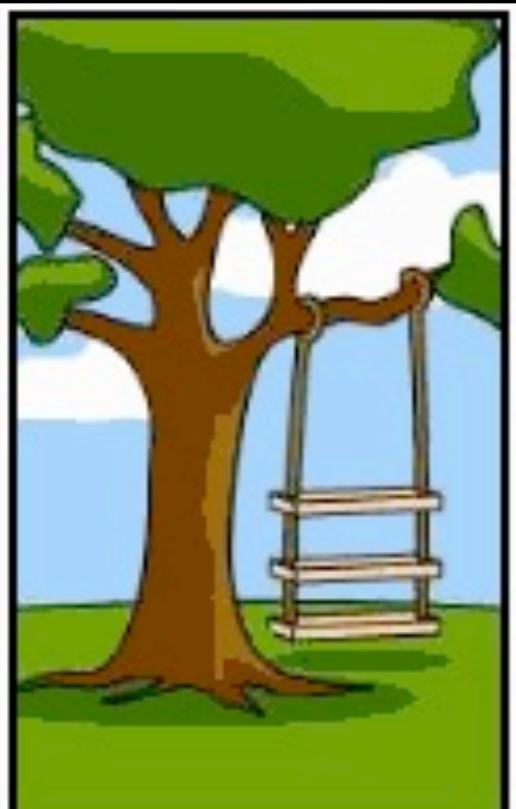


See you all on Wednesday

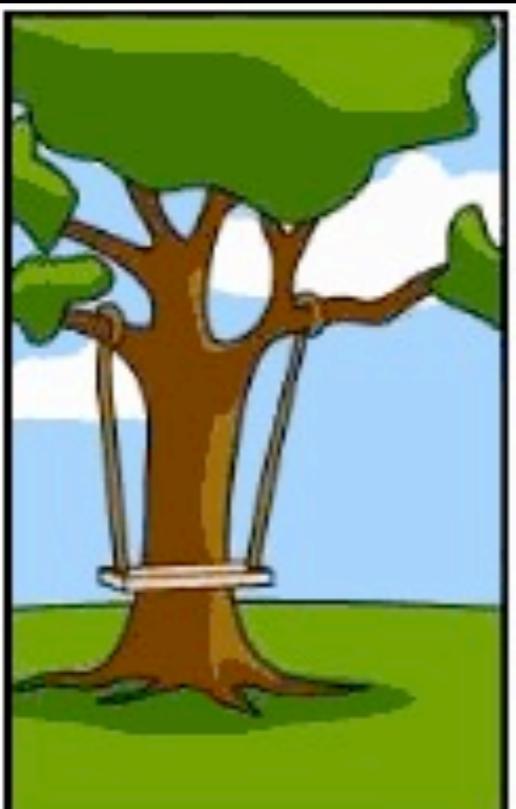


julie.greensmith@nottingham.ac.uk





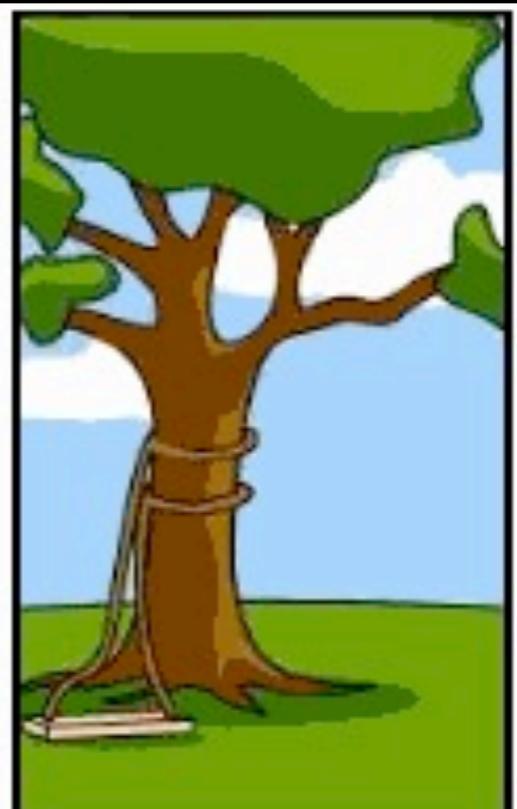
How the customer explained it



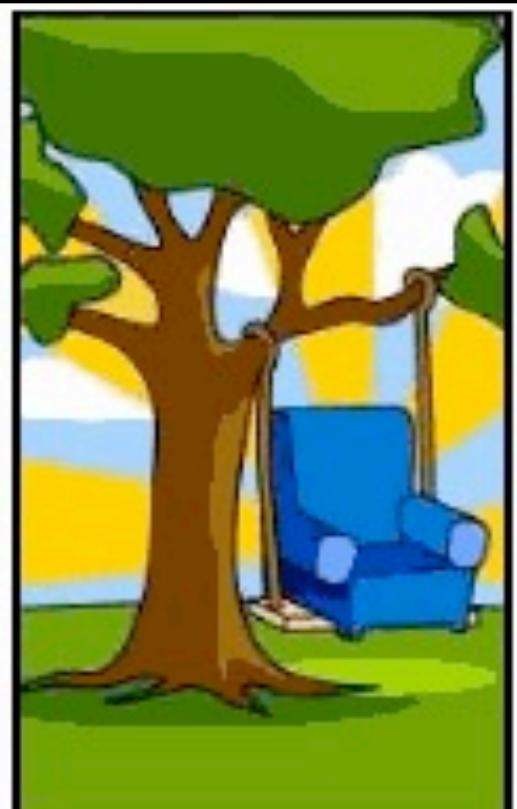
How the Project Leader understood it



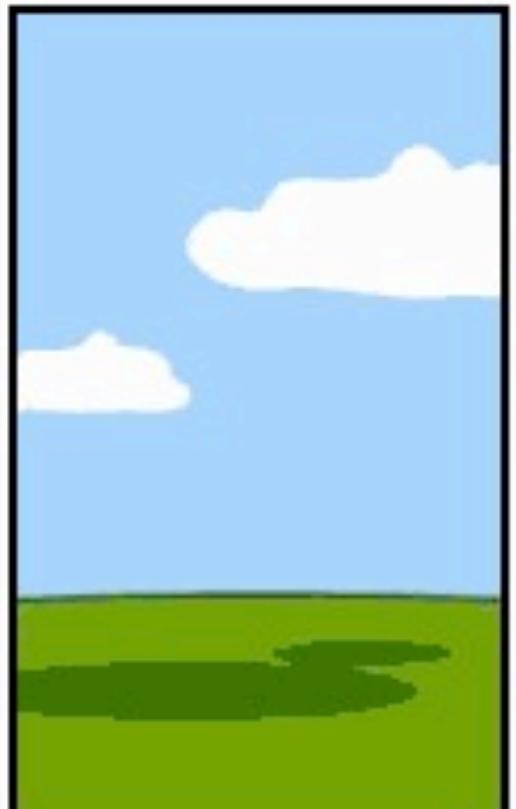
How the Analyst designed it



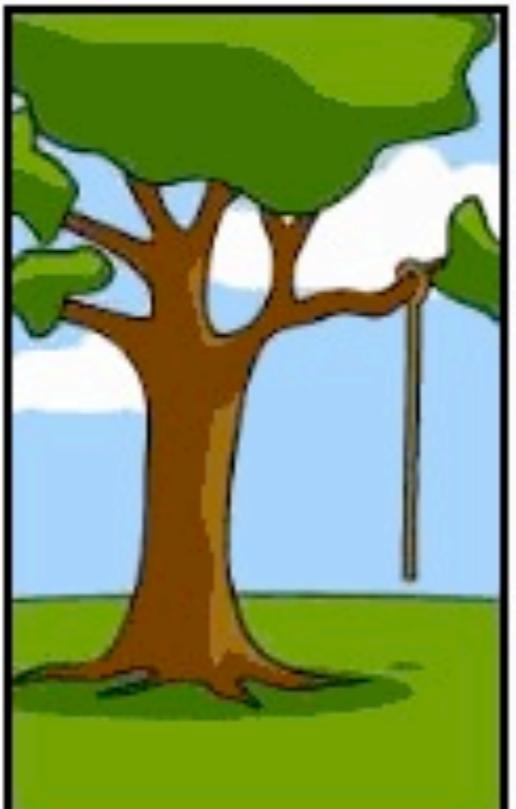
How the Programmer wrote it



How the Business Consultant described it



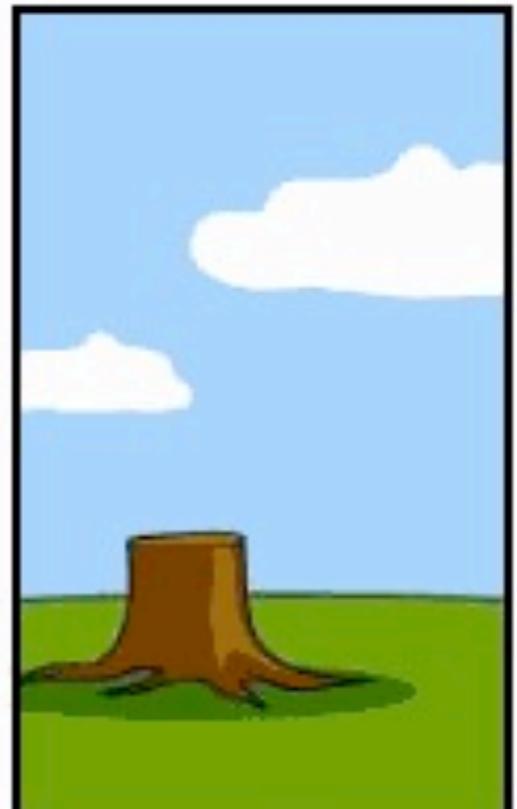
How the project was documented



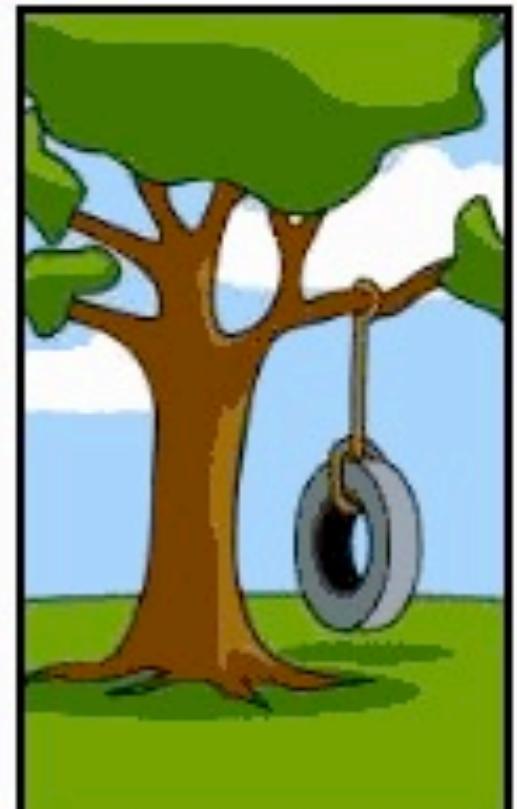
What operations installed



How the customer was billed



How it was supported



What the customer really needed