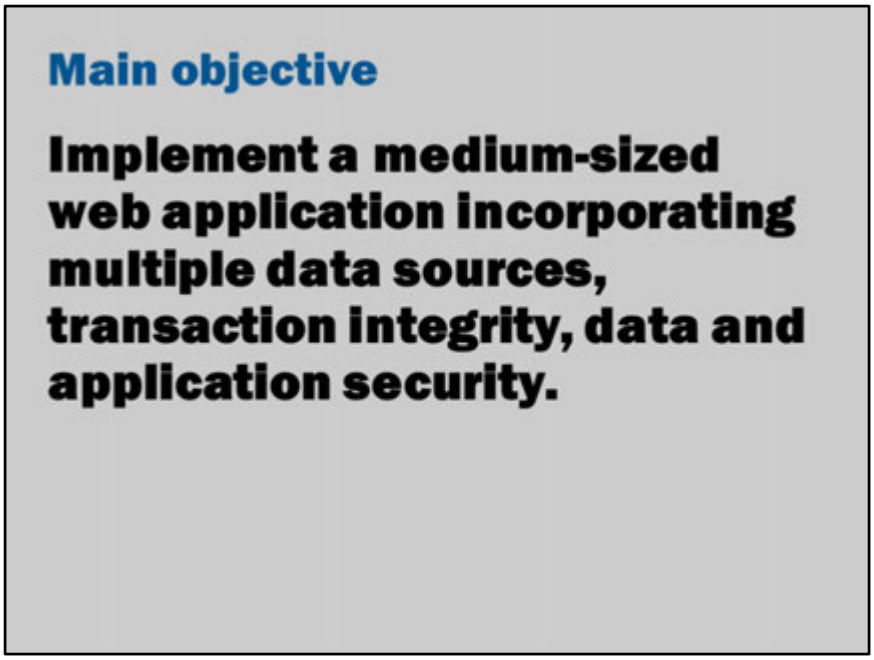
Welcome to AIP



The main objective is for you to demonstrate this objective directly: you will implement a web application for the subject's assessment.

It is not enough to simply 'hack' functionality together. You will need to use principles of good design to create an application that is ready for deployment in production environments.

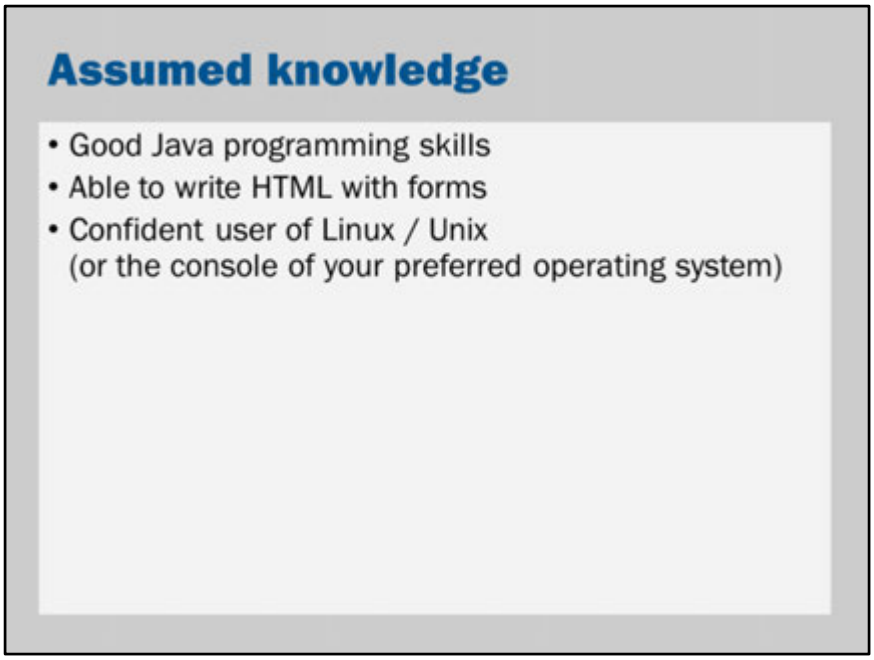
Outcomes

You will be able to:

- Describe, conceptually, a full e-commerce application
- Compare and contrast web application architectures
- Recommend solutions for arbitrary web applications

You will be able to explain concepts relating to:

- Security
- Transactions
- Robustness
- High-availability

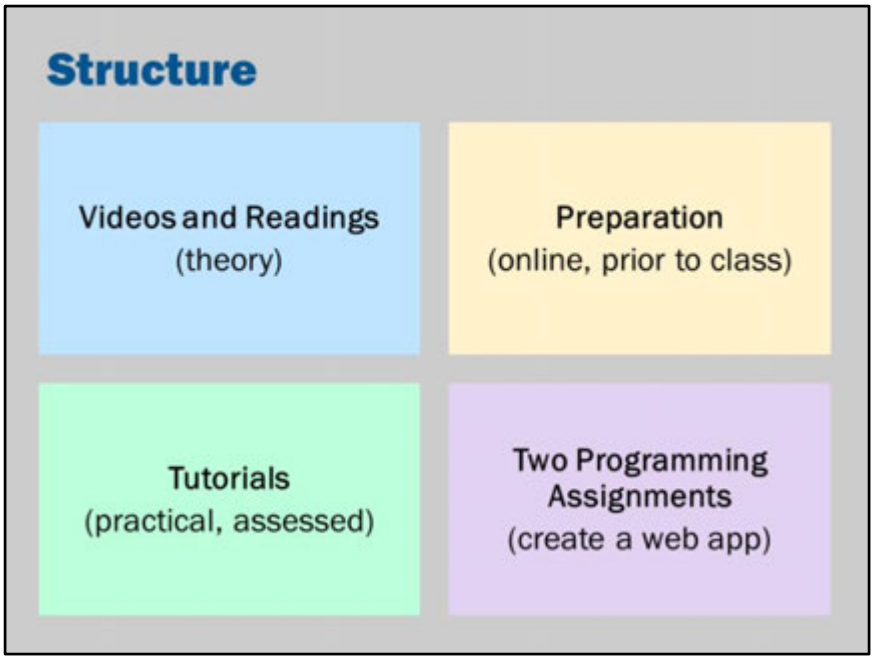


This is **Advanced** Internet Programming. It isn't going to be the basics. You should already know how to develop simple interactive websites.

Java:
You should have a solid grasp of the Java programming language.
You should understand the following features in Java: class, interface, Java's primitive types (boolean, byte, char, short, int, long, float, double), visibility (private, protected, default, public), String, abstract, static, final, synchronized, arrays, loops, exceptions, try/catch and some of the standard libraries (e.g., the collections classes in java.util.*). You should be comfortable with object oriented programming.

HTML:
You should be able to create a simple website using HTML. You should understand the basic operation of HTTP. You should be able to explain how forms work and the difference between GET and POST.

Linux/Unix:
You should be able to edit files, launch programs and use basic commands in the shell (ls, cd, rm).



Videos and readings will focus on the "big ideas" and "theory". Tutorials will focus on applying those ideas using Java.

Videos and Readings:

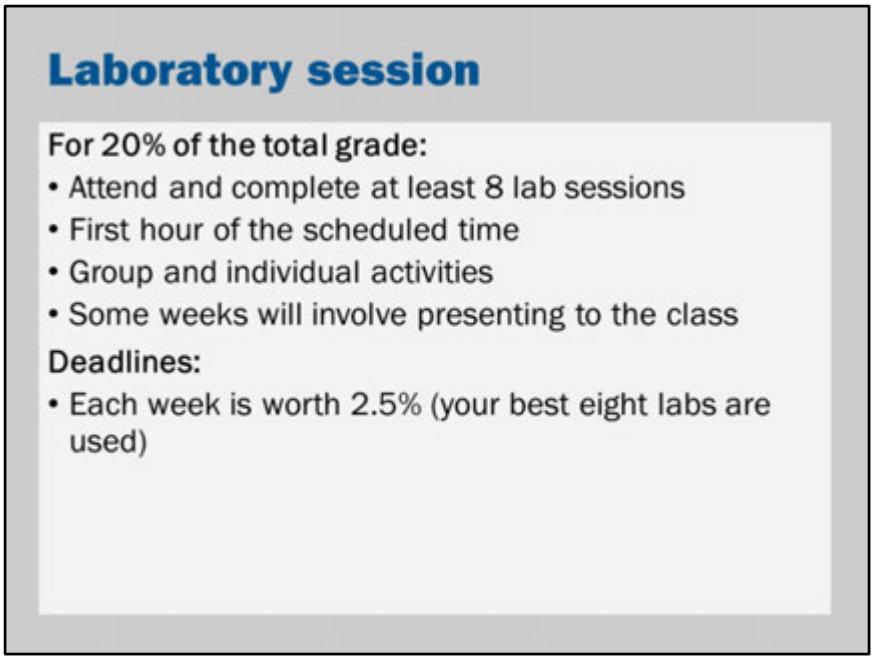
- They will be online each week
- Each week the lecture will cover a separate topic. Roughly one topic per week.
- Complete them before class
- Test your knowledge before the tutorial with a simple quiz

Tutorials

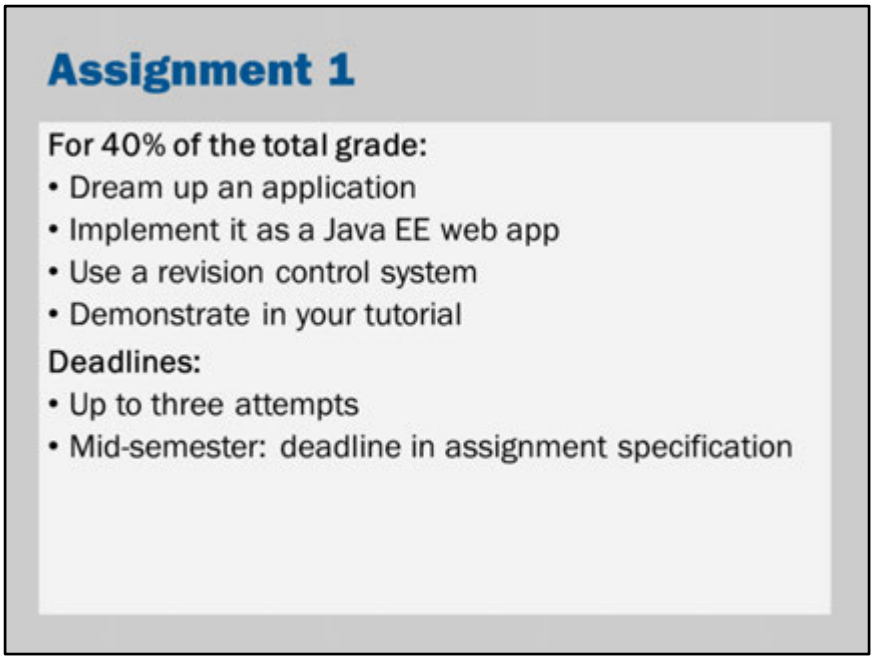
- A 1 hour activity that is assessed
- The tutorials will be where you can apply what you have learned in the lecture and also provides time to discuss your assignment challenges with your tutor.
- Followed by 2 hours hands-on practical activities done alone (but you are welcome to work with a friend)
- In labs, please turn off your phones or put them into vibrate mode. I do not mind if you need to leave the room to answer a call but do not disrupt others.

Assignments:

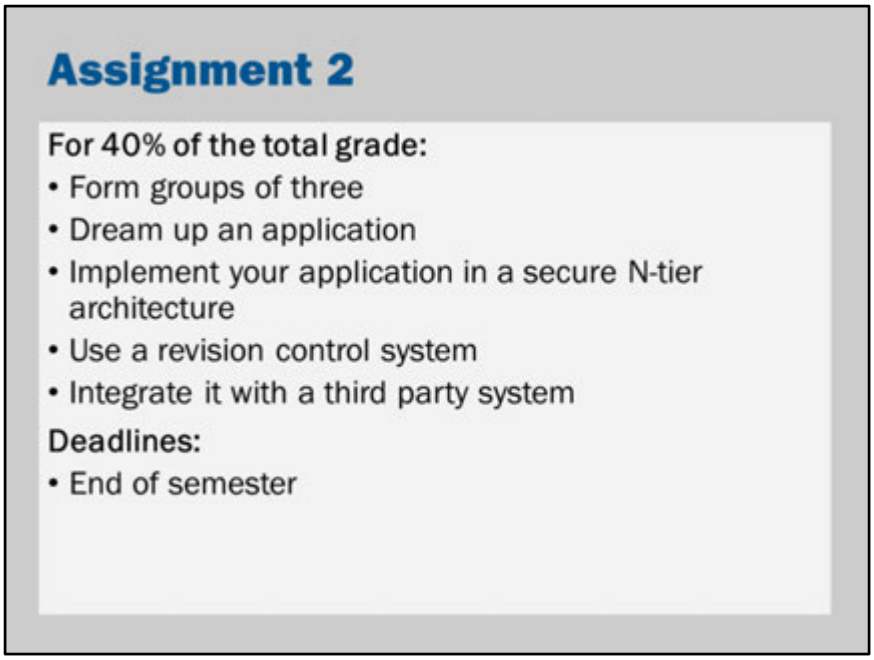
- Two programming assignments



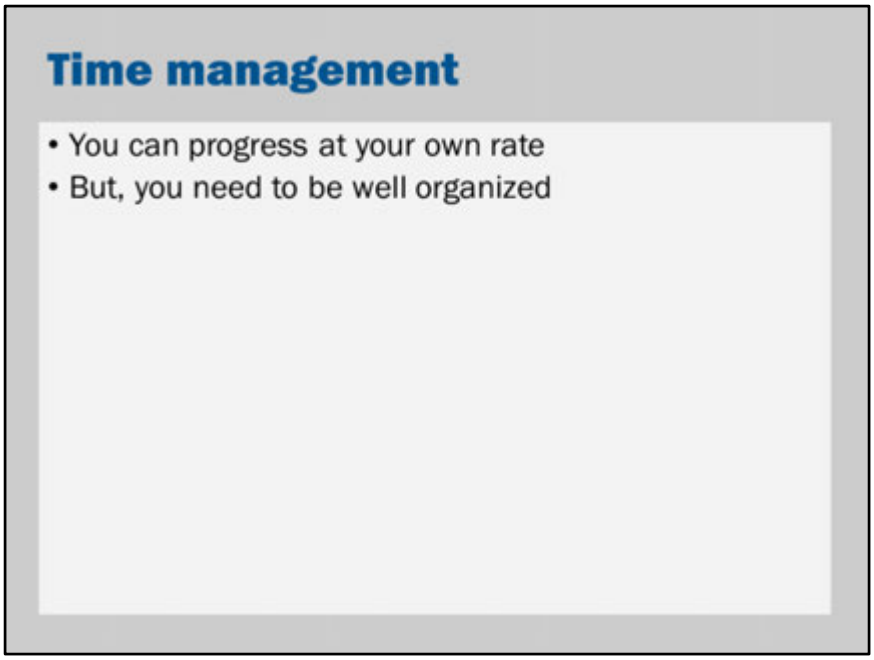
The first hour of the lab session will be a group activity. You will need to work with your fellow students (or alone) to solve a small challenge problem set at the start of the lab session.



Full details of the assignment appear in the assignment specification document.

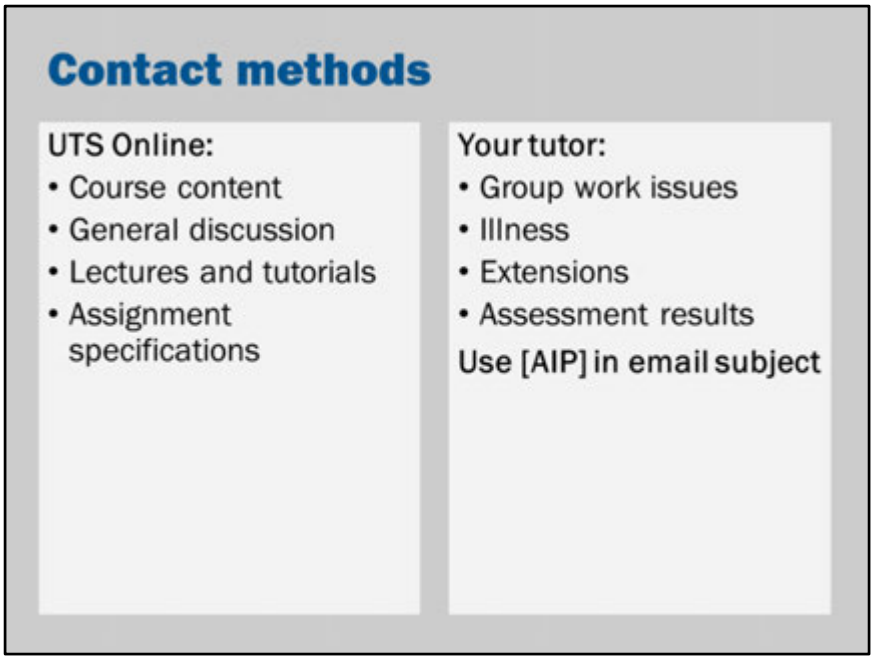


Full details of the assignment appear in the assignment specification document.



You can work at your own pace in this subject. This is very flexible.
But, please don't fall into the trap of leaving everything to the last minute.

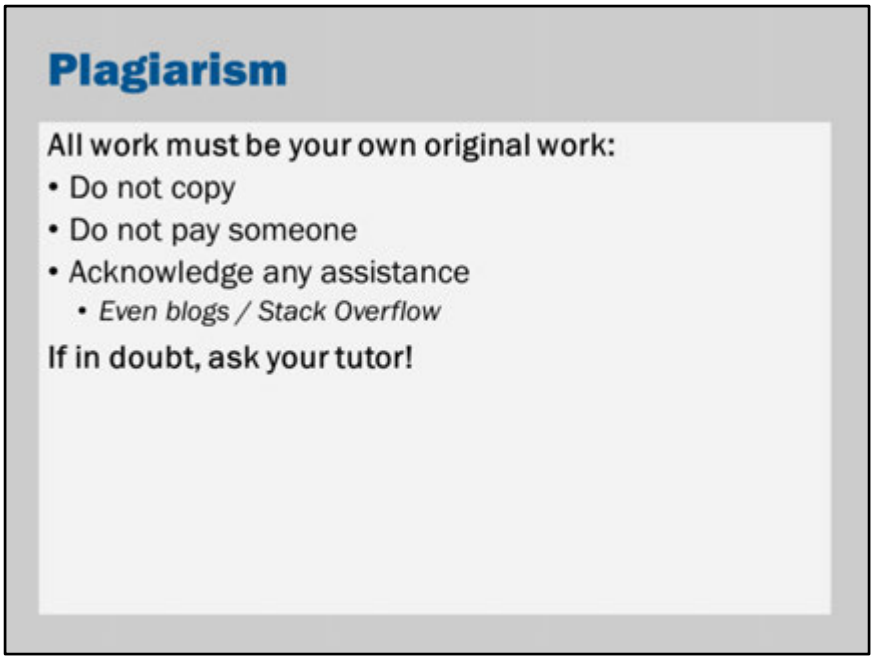
This subject involves a lot of work. You don't need to be brilliant or a Java genius.
Students who work hard tend to succeed the best.



Questions relating to personal matters should be directed to your tutor. If you send an email, include [AIP] in the subject line.

All other questions should be posted on UTS Online.

4 ground rules



Refer to the student misconduct policies and rules for more information:

Engineering and IT:
https://my.feit.uts.edu.au/pages/course/student_policies_rules/student_misconduct

University:
http://www.gsu.uts.edu.au/rules/student/section-16.html

Respect

- Be supportive and encouraging
- Help each other learn and understand the ideas

Enthusiasm

- Be on time
- Be prepared: study each week
- Be enthusiastic
- If you are not prepared, you may be asked to leave the lab session

Fun

Most importantly:

- Have fun!

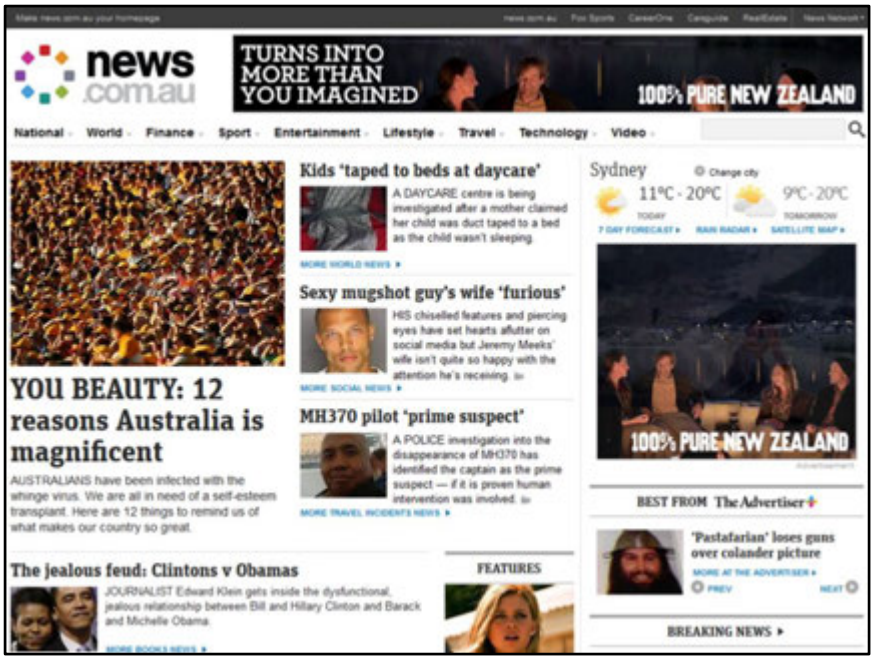
Challenges

Why do we need to learn AIP?

- What about everything taught in Internet Programming?
- Isn't PHP good enough? ("It was good enough for Facebook")

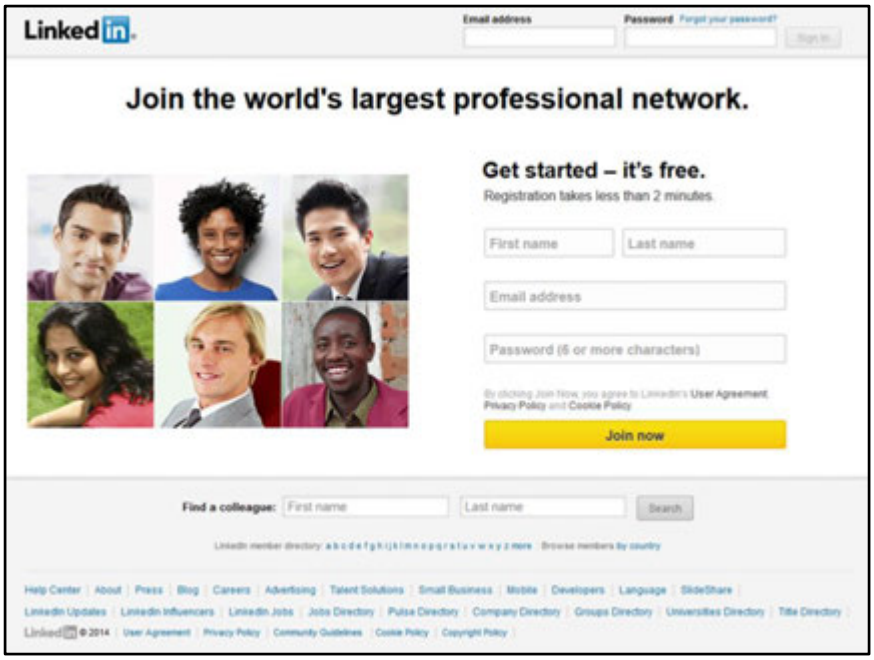
Because we need to deal with:

- Scalability (many users, lots of data, fast response, many servers)
- Reliability
- Security
- Maintainability



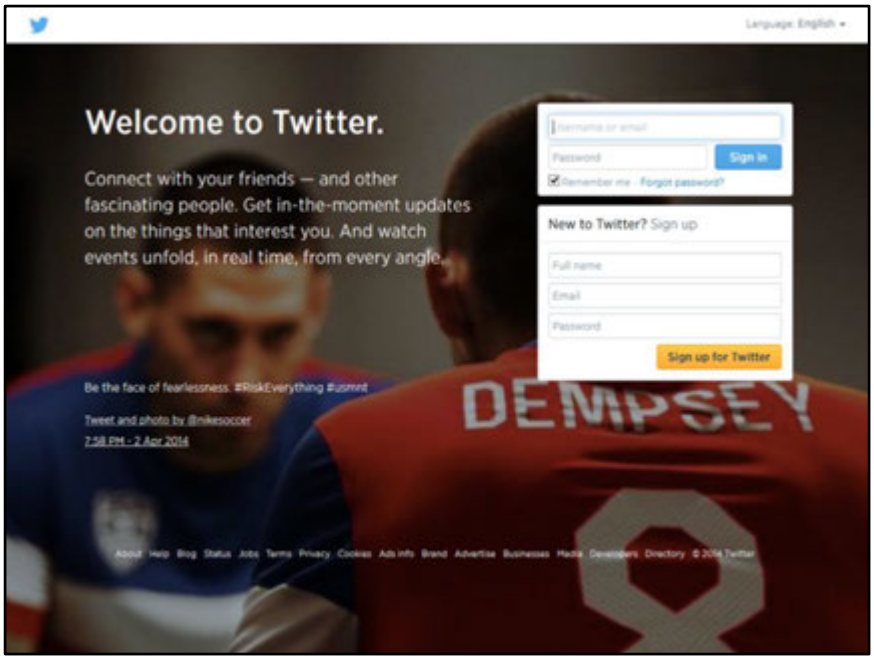
Some challenges facing News.com.au:

- Ensuring quality
- Flexible presentation
- Location-based personalization
- Maintaining old content and links
- Surges in popularity
- Multiple platforms



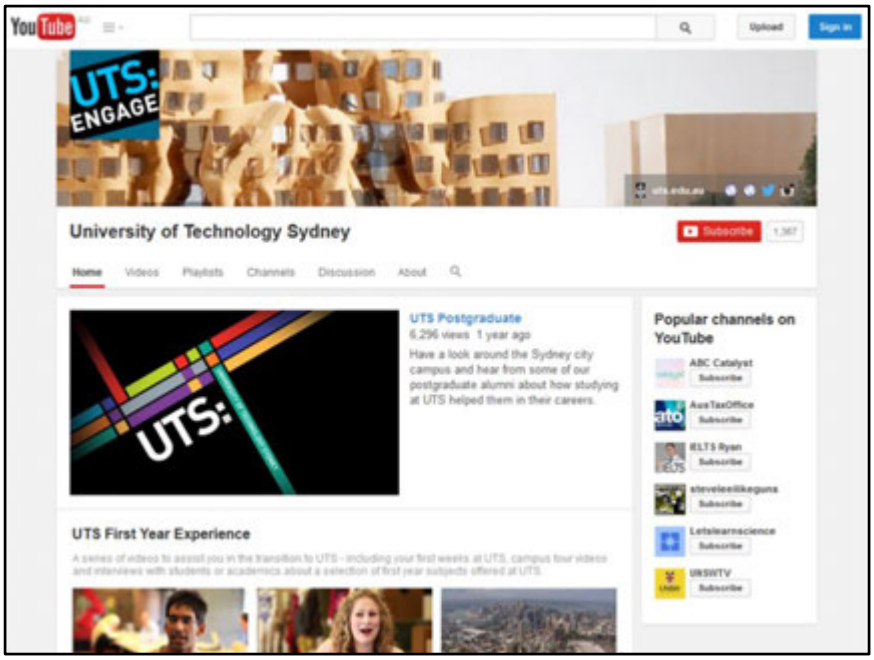
Challenges facing LinkedIn:

- Confidential information
- Real-time interaction
- Highly-personalized content for every individual user
- Multiple languages
- Sophisticated searching



Challenges facing Twitter:

- Many, many platforms including third-party apps
- Expectation of real-time, instantaneous interaction
- Half-a-billion tweets per day (2013)
- Global surges in popularity
- Rapid growth in early years



Challenges facing YouTube:

- Many videos
- Huge files used in upload and download
- 1 billion unique visitors per month
- Transcoding is slow and time-consuming
- Lots of copyright management challenges

Some challenges

- Internationalization
- Reliability
- Performance
- Integrity
- Security
- Maintainability
- and so on...

Bonus slides

Teaching staff

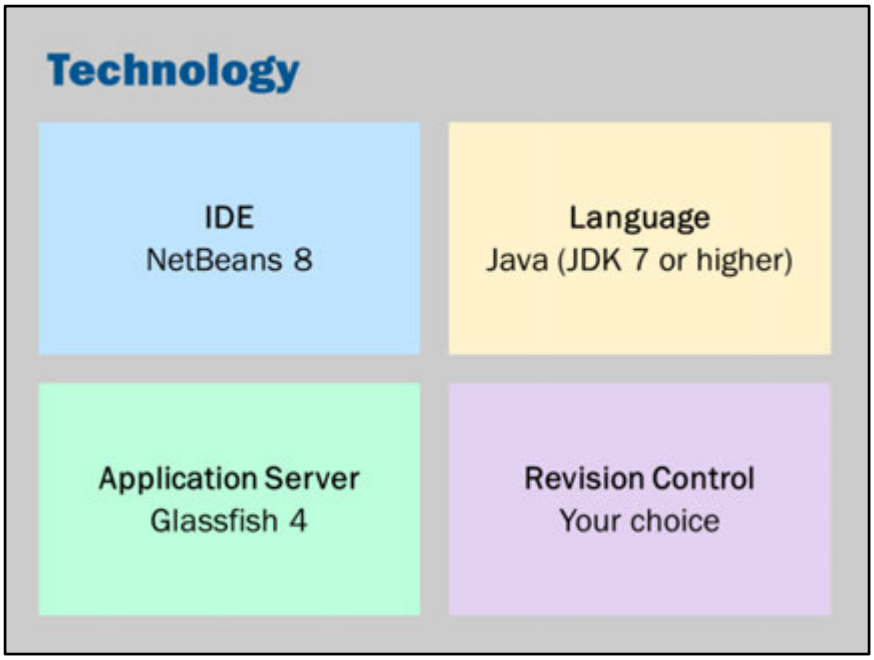
Benjamin Johnston
- Course Coordinator, Tutor
- `Benjamin.Johnston@uts.edu.au`

Ryan Heise
- Tutor
- `Ryan.Heise@uts.edu.au`

UTS Online

Please check UTS Online at least once per week:

- 'Homework'
- Lecture slides
- Tutorial activities
- Announcements and news
- Questions, answers and discussion
- FAQs



While the software for this subject is available on the lab computers, it is very easy to install the software on your own computer or laptop.

In fact, I strongly recommend that you set up a development environment on your own computer.

IDE:
We will be using NetBeans 8.0.2 in lectures and in tutorials. NetBeans makes it very simple to deploy your application. You are free to install and use any IDE you wish, on any operating system you wish.

Language:
We will be using Java 7 or higher.

Application Server:
You can use other application servers. However, your assignments will be tested against Glassfish 4.

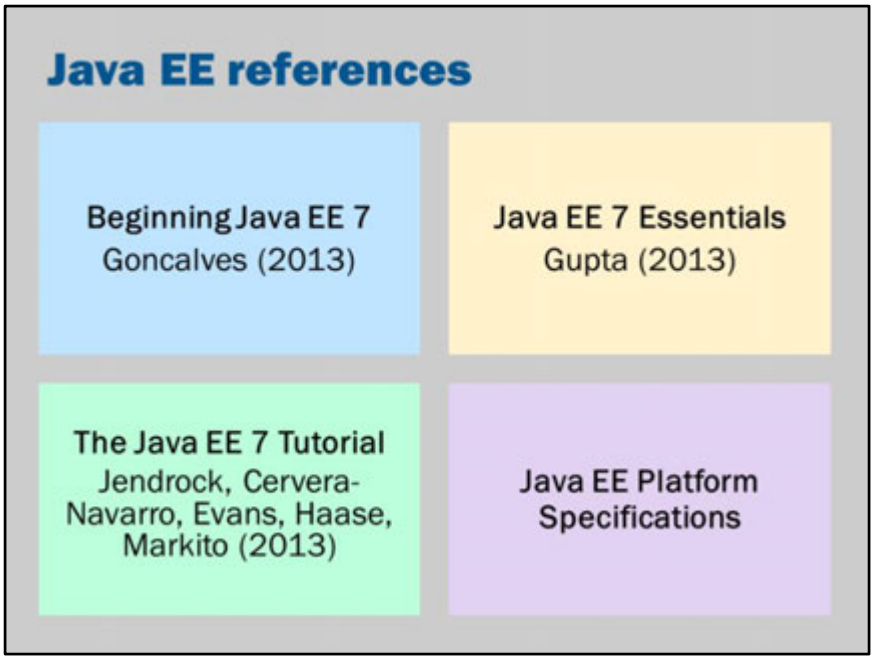
Revision Control:

You must use a revision control system and check in code on a regular basis.

Note that NetBeans has built-in support for revision control and is very easy to use.

You might consider using:

- A free account on develop.eng.uts.edu.au (Subversion)
- A free BitBucket account (Git or Mercurial)
- A paid account on github (Git)
- Your own installation of Git, Mercurial, or Subversion



There is no fixed textbook.

Beginning Java EE 7:
Free download of complete PDF via the library website. A step-by-step guide based on NetBeans and Glassfish 4.
http://find.lib.uts.edu.au/?R=OPAC_b2874770

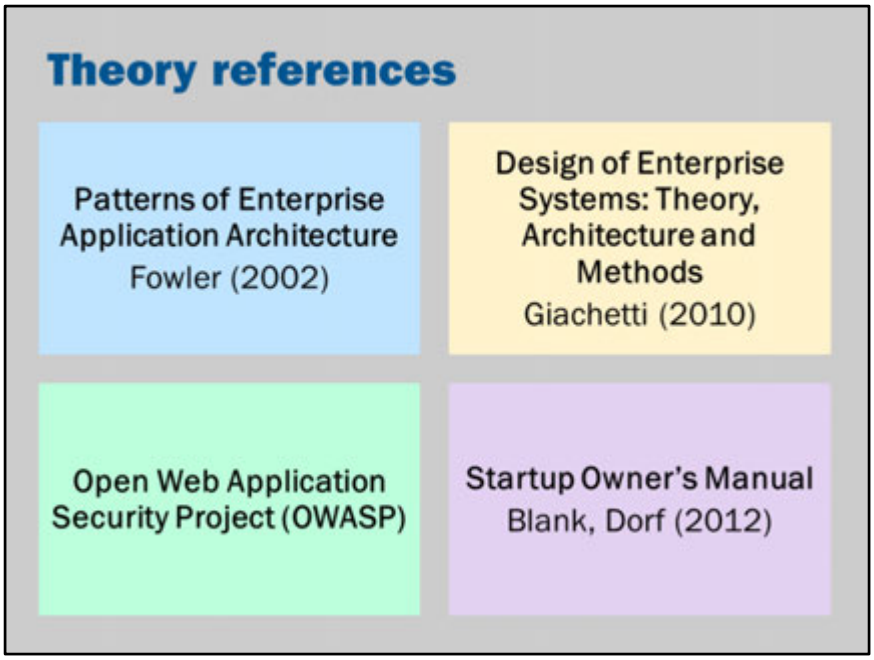
Java EE 7 Essentials:
A shorter book that summarizes key points but is not a tutorial.
http://find.lib.uts.edu.au/?R=OPAC_b2838102

Java EE 7 Tutorial:
Free download, and very comprehensive. However, this is quite an advanced-level tutorial.
http://docs.oracle.com/javaee/7/tutorial/doc/home.htm

Java EE Platform Specifications:
When in doubt about something, check the full specifications!
https://java.net/projects/javaee-spec/pages/Home

Other resources:

NetBeans E-Commerce Tutorial: Slightly out-of-date but an excellent introduction to features of the Java EE platform.
https://netbeans.org/kb/docs/javaee/ecommerce/intro.html



While this subject is hands-on and technical, you may like to take a look at some of the bigger picture questions of internet programming and enterprise development.

Patterns of Enterprise Application Architecture:
A classic book on software architectures for enterprise systems:
http://find.lib.uts.edu.au/?R=OPAC_b1603101
http://martinfowler.com/books/eaa.html

Design of Enterprise Systems: Theory, Architecture and Methods:
Provides high-level overviews of many aspects of building enterprise systems, including data modeling, estimating performance, project management
http://find.lib.uts.edu.au/?R=OPAC_b2549670

Open Web Application Security Project:
Free guides with information about securing and testing websites:
https://www.owasp.org/index.php/Main_Page

Startup Owners Manual:
Information about the “Customer Development methodology”, in which startups are

treated as an experimental process.
http://find.lib.uts.edu.au/?R=OPAC_b2748519
http://www.amazon.com/The-Startup-Owners-Manual-Step-By-Step/dp/0984999302
http://steveblank.com/
See also Udacity course: https://www.udacity.com/course/ep245

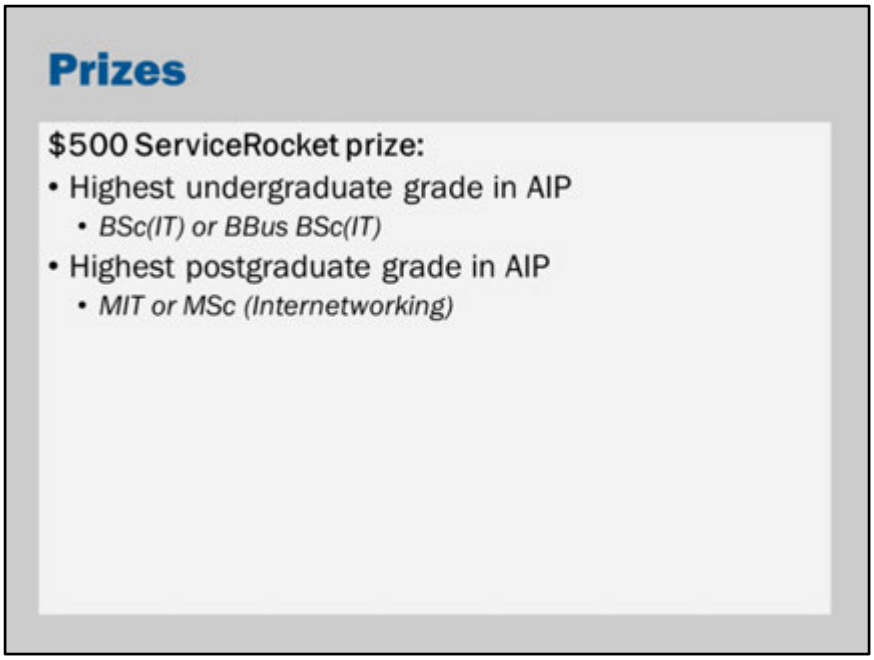
Other resources:

Mozilla Developer Network:
An excellent reference for HTML, CSS, JavaScript:
https://developer.mozilla.org/en-US/

W3C Specifications:
When in doubt, check the full specifications!
http://www.w3.org/
http://www.w3.org/TR/html5/

TCP/IP Illustrated:
The classic book on the low-level protocols underlying the internet.
Stevens (1993) *TCP/IP Illustrated, Volume 1*, Addison-Wesley Professional.

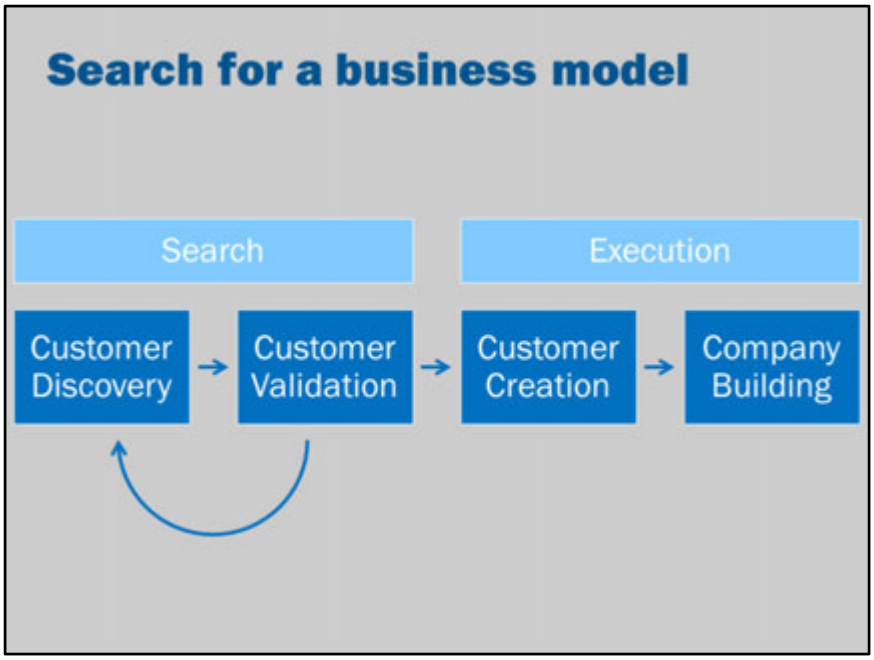
Even more reading ideas:
http://blog.codinghorror.com/recommended-reading-for-developers/



These prizes are managed by the faculty and will be automatically awarded at the end of the year.

What is a startup?
“A startup is a temporary organization in search of a scalable, repeatable, profitable business model.”

For more information on this section, refer to:
Blank, S. and Dorf, B. (2012) *Startup Owner’s Manual*. K & S Ranch.



The customer development methodology is a modern approach to creating start-ups.

It does not start with a brilliant concept, developing it, marketing it and then wait until it has a big launch before knowing the results.

Instead, the idea is that every aspect of the business, including the concept, needs to be tested in an iterative and experimental process. A start up is a *search* for a scalable business model.

Each phase tests an aspect of the business:

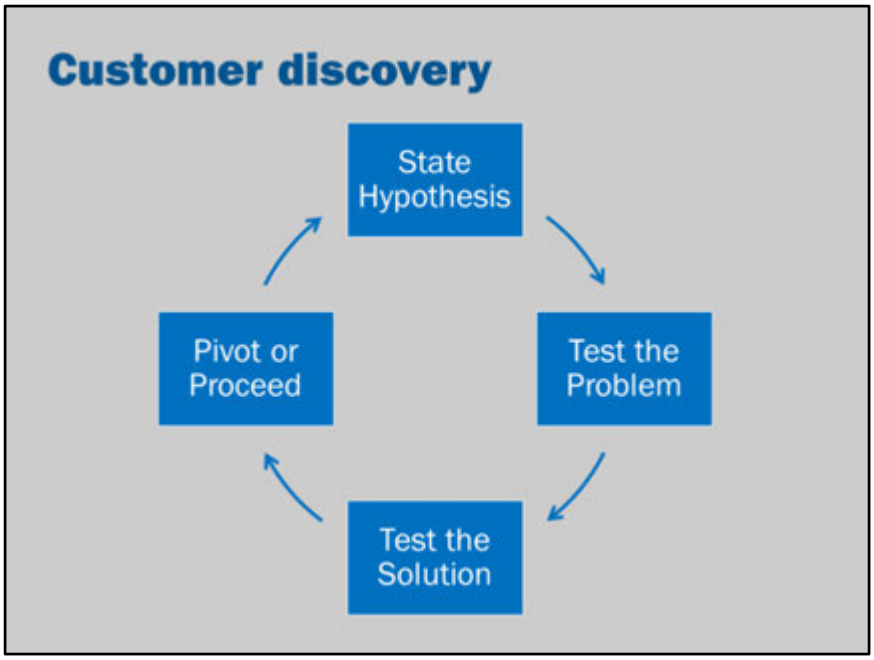
Customer Discovery: Does the product solve a known problem?

Customer Validation: Can the product be sold to enough people to make a business?

Customer Creation: Can the business be scaled through repeatable sales and marketing?

Company Building: Can the business be transitioned into an ongoing long-term

concern to support the scale?

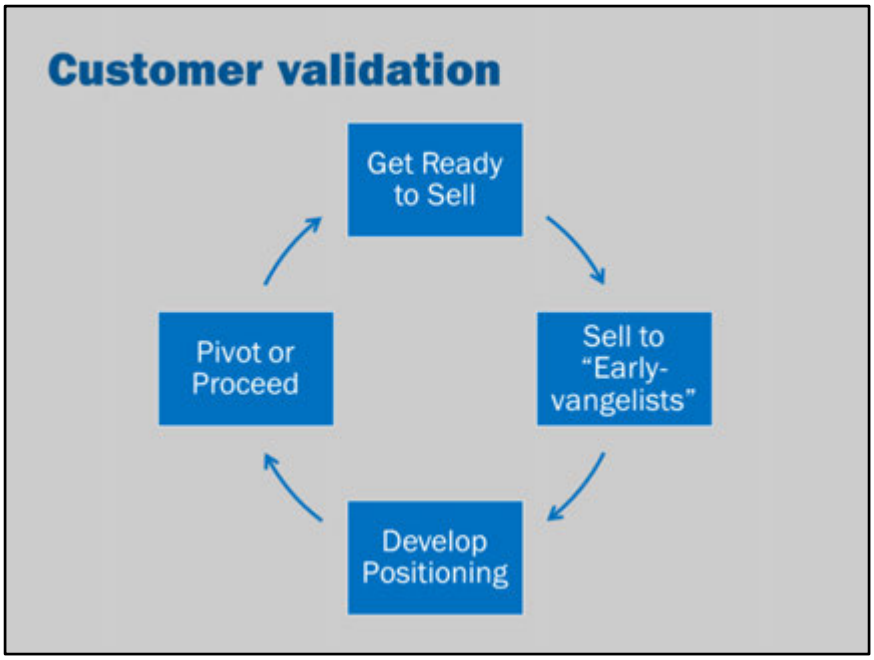


State the Hypothesis: Write down your idea (e.g., on a business model canvas).

Test the Problem: Conduct research and experiments to validate the plausibility of the business model canvas. Do potential customers genuinely face the problem? Is there hard data that supports every aspect of the business model (value proposition, channels, pricing, sales process, etc)?

Test the Solution: Build the minimum viable product that presents your 'value proposition' to potential customers. Are they engaged by the product? Do/will they buy the product? The goal is not sales (yet) but to validate what you've found when you tested the problem.

Pivot or Proceed: Do you understand the customer? Is the market large enough? Are customers willing to pay? Could the product make a profit? If it these questions cannot be answered favorably, then go back and try another hypothesis.



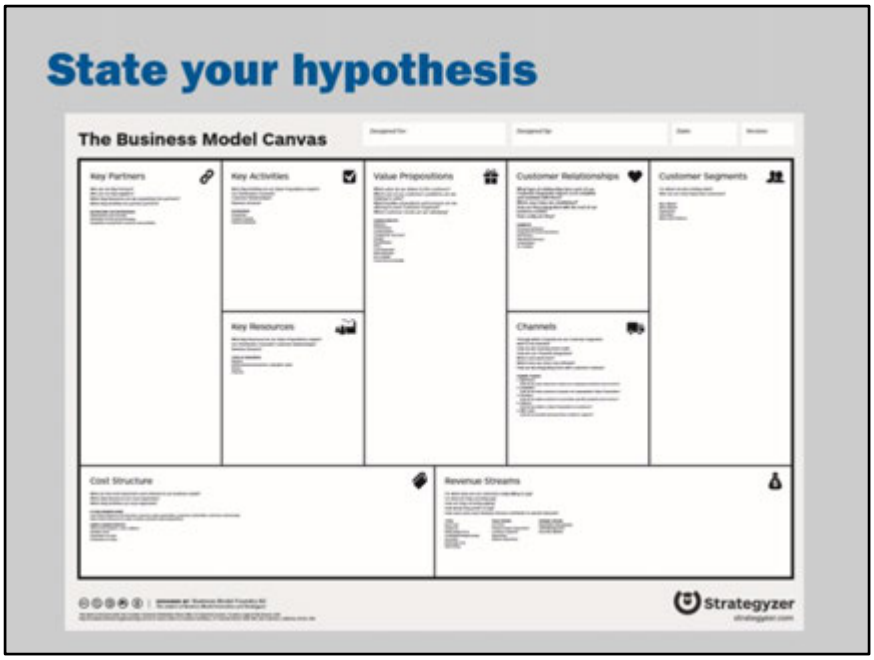
Once we've found an idea that has real traction with customers, the next stage is to see if the product will result in genuine sales.

Get Ready to Sell: Create marketing collateral, build a high-fidelity minimum viable product and decide on product positioning.

Sell to "Early-vangelists": Start selling, especially to those people who acutely suffer the problem the product addresses and are willing to take a chance. Getting feedback is as important as getting sales.

Develop Positioning: Use the early sales as a foundation for refining the product and company positioning.

Pivot or Proceed: "is this a business worth doing?"



This canvas is online here:
http://www.businessmodelgeneration.com/downloads/business_model_canvas_poster.pdf

An explanation may be found online:
https://www.youtube.com/watch?v=QoAOzMTLP5s

This isn't a technical subject, so why are we doing this?

First, this is one way of stating your plans for Assignment 1 and making a quick decision so that you can get on with your Assignment.

But, more importantly, what we do as developers isn't just coding. It is important to see the big picture and relate what you do to your business or to your customers. Completing this canvas is part of practice-based learning.