SOFTWARE ENGINEERING

AGILE CLASSROOM

WHY SOFTWARE ENGINEERING?

- "You might think that the biggest part of a software project is programming, but in a typical project, programming usually takes up only about 20% of the total time! 40% is spent on analysis and design and another 40% on testing. This shows that software engineering is so much more than programming." (Bell & Morgan, 2014)
- The topic is especially relevant because students can see that software development is always situated within a context.
- More "human" side of computing – analysis, communication, solving a problem for a group of users.

WHY SOFTWARE ENGINEERING?

- There is a strong correlation to the NCEA Generic Technology Internal Achievement Standards, so this topic underpins a level 3 DT programme with a focus on developing stakeholder oriented projects:
  - Brief development – determining requirements from different stakeholder groups
  - Project management – researching project management methodologies, applying a project management schedule
  - Technological modelling – risk mitigation through determination of requirements and iterative development
  - Prototyping – trialling, rigorous testing, iterative development

VISIT THEN IMPLEMENT

- In order for the students to have an adequate grasp of the topic, it is important for them to have some "real life" exposure to software development in the real world.
- If possible, arrange a visit to a company for the students to see the processes in action (with the added benefit of promoting careers in IT).
- Alternatively, have a speaker visit your classroom to discuss their process.
- Then, have the students implement as much as the process as possible within the confines of NCEA and the classroom.

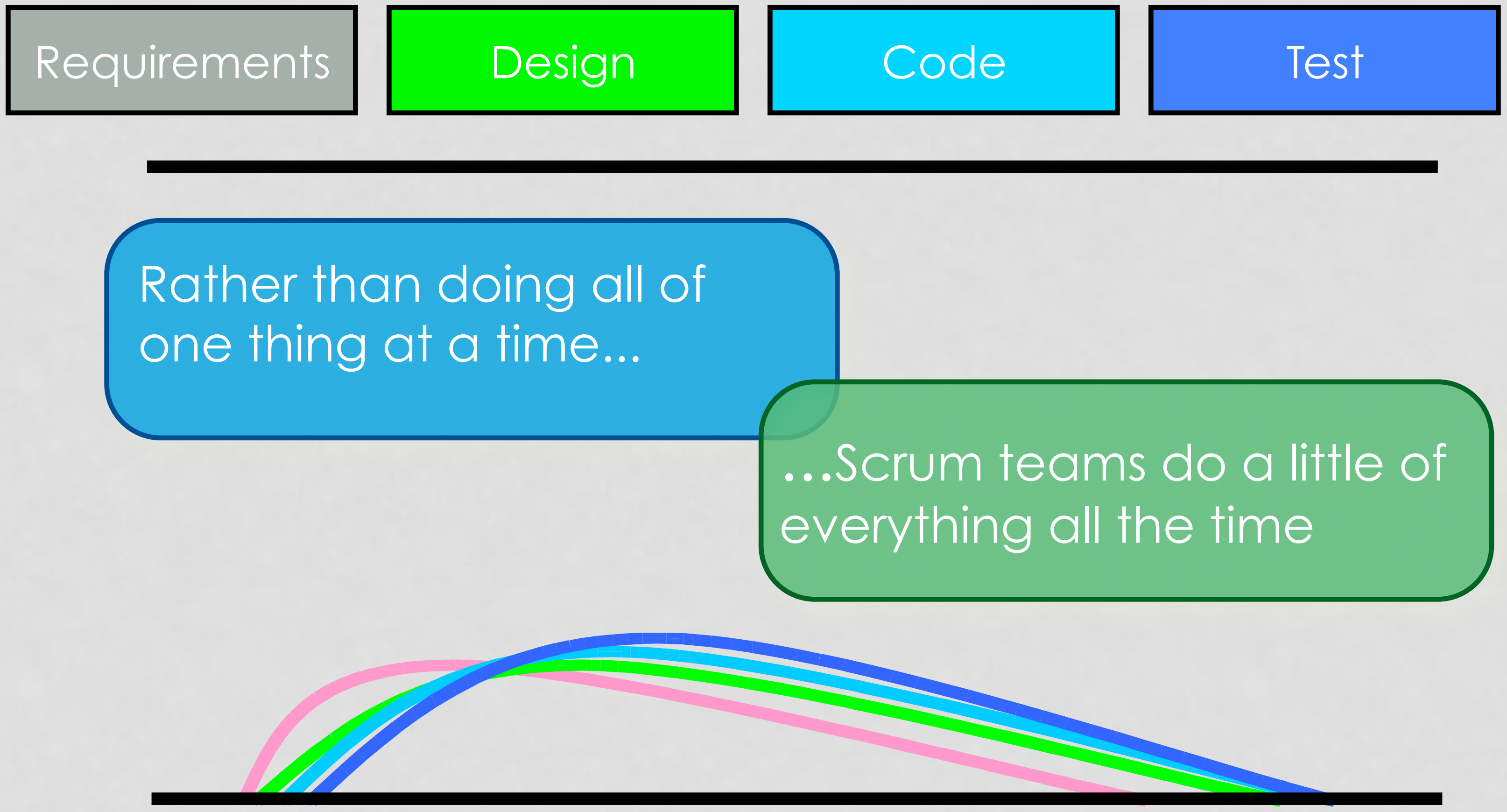
KEY CONCEPT: STRUCTURED METHODOLOGY

- Software in the real world is large and complex – generally developed by teams as it is impossible for any single person to understand or code.
- A structured methodology for development is necessary to manage the complexity of the software and to ensure it meets the requirements of the stakeholders, is delivered on time and within budget.
- Without a structured process for development of software, the probability and severity of risk for failure/bugs in the software increases.

KEY CONCEPTS: ANALYSIS | DESIGN | CODE | TEST

- Analysis/gather requirements
- Design the software
- Code/Implement the design
- Testing of the code

SEQUENTIAL VS. OVERLAPPING DEVELOPMENT


Requirements
Design
Code
Test
Rather than doing all of one thing at a time...
...Scrum teams do a little of everything all the time

Source: "The New New Product Development Game" by Takeuchi and Nonaka. *Harvard Business Review,* January 1986.

WATERFALL (SEQUENTIAL) VS AGILE (OVERLAPPING)

Waterfall (of tears)

Old NCEA Standards

- Each phase in the process is completed before moving on to the next.
- There is little change once the project design phase is complete.
- Software development work completed in "silo" environment.
- Coders and testers work in opposition.

Agile

New NCEA Standards

- Each phase is completed iteratively in a short sprint of 2-weeks.
- Flexibility is the key, so that the software can be changed if there is a design issue or change to requirements.
- Software development is completed in cross-functional teams.
- Coders and testers work in tandem.

WHAT WE LEARNED AT ORION

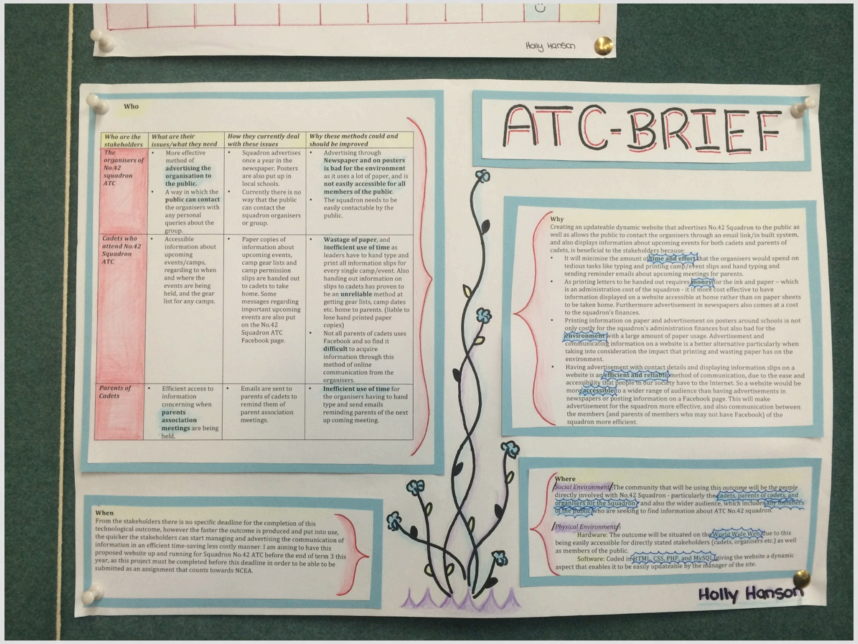
- Importance of user stories for requirements analysis
- Importance of cross-functional teams:
  Business Analyst, Coders, Testers, UX Designer
- Use of SCRUM boards for visual planning
- Breaking down software development into manageable chunks via “sprint goals”
- Importance of testing at each sprint and writing test conditions for that sprint goal
- Iterate until awesome!

HOW WE ARE IMPLEMENTING WHAT WE LEARNED

- Each student develops their project brief and tasks in terms or user stories.
- Each student has a scrum board for visual planning.
- Each student must do a stand-up after a two week sprint.
- Testing is incorporated into each sprint cycle.

CLASSROOM IMPLEMENTATION SCRUM BOARD

A3 Poster for Initial Brief



ATC-BRIEF

Who

Who are the stakeholders	*What are their issues/what they need*	*How they currently deal with these issues*	*Why these methods could and should be improved*
The organisers of No.42 squadron ATC	• More effective method of **advertising the organisation to the public.** • A way in which the **public can contact** the organisers with any personal queries about the group.	• Squadron advertises once a year in the newspaper. Posters are also put up in local schools. • Currently there is no way that the public can contact the squadron organisers or group.	• Advertising through **Newspaper and on posters is bad for the environment** as it uses a lot of paper, and is **not easily accessible for all members of the public.** • The squadron needs to be easily contactable by the public.
Cadets who attend No.42 Squadron ATC	• Accessible information about upcoming events/camps, regarding to when and where the events are being held, and the gear list for any camps.	• Paper copies of information about upcoming events, camp gear lists and camp permission slips are handed out to cadets to take home. Some messages regarding important upcoming events are also put on the No.42 Squadron ATC Facebook page.	• **Wastage of paper**, and **inefficient use of time** as leaders have to hand type and print all information slips for every single camp/event. Also handing out information on slips to cadets has proven to be an **unreliable** method at getting gear lists, camp dates etc. home to parents. (liable to lose hand printed paper copies) • Not all parents of cadets uses Facebook and so find it **difficult** to acquire information through this method of online communication from the organisers.
Parents of Cadets	• Efficient access to information concerning when **parents association meetings** are being held.	• Emails are sent to parents of cadets to remind them of parent association meetings.	• **Inefficient use of time** for the organisers having to hand type and send emails reminding parents of the next up coming meeting.

Why
Creating an updateable dynamic website that advertises No.42 Squadron to the public as well as allows the public to contact the organisers through an email link/in built system, and also displays information about upcoming events for both cadets and parents of cadets, is beneficial to the stakeholders because:

- It will minimise the amount of **time and effort** that the organisers would spend on tedious tasks like typing and printing camp/event slips and hand typing and sending reminder emails about upcoming meetings for parents.
- As printing letters to be handed out requires **money** for the ink and paper – which is an administration cost of the squadron - it is more cost effective to have information displayed on a website accessible at home rather than on paper sheets to be taken home. Furthermore advertisement in newspapers also comes at a cost to the squadron's finances.
- Printing information on paper and advertisement on posters around schools is not only costly for the squadron's administration finances but also bad for the **environment** with a large amount of paper usage. Advertisement and communicating information on a website is a better alternative particularly when taking into consideration the impact that printing and wasting paper has on the environment.
- Having advertisement with contact details and displaying information slips on a website is an **efficient and reliable** method of communication, due to the ease and accessibility that people in our society have to the Internet. So a website would be more **accessible** to a wider range of audience than having advertisements in newspapers or posting information on a Facebook page. This will make advertisement for the squadron more effective, and also communication between the members (and parents of members who may not have Facebook) of the squadron more efficient.

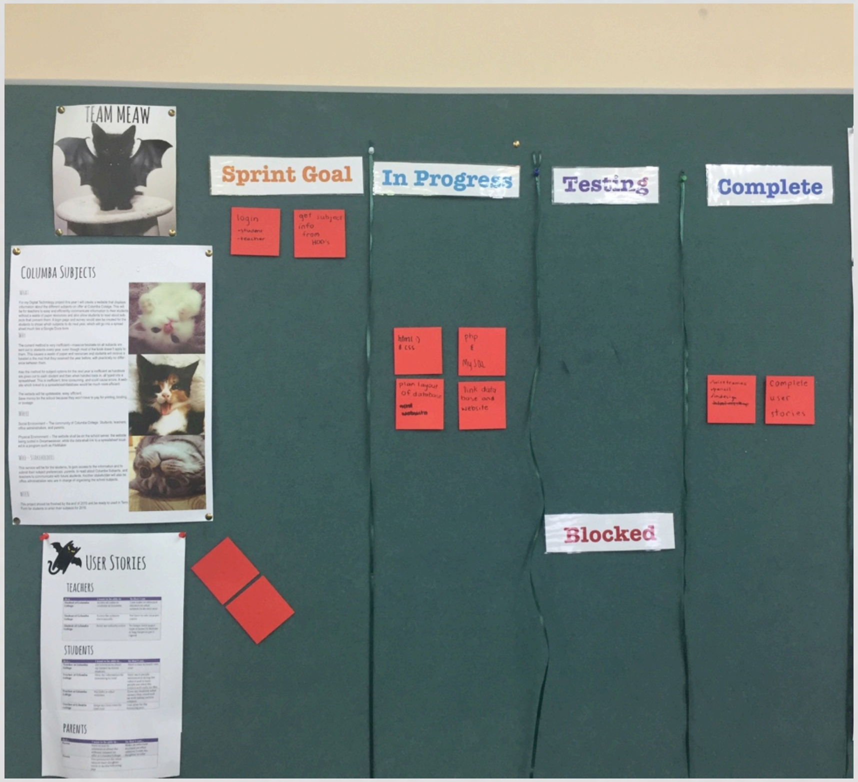
When
From the stakeholders there is no specific deadline for the completion of this technological outcome, however the faster the outcome is produced and put into use, the quicker the stakeholders can start managing and advertising the communication of information in an efficient time-saving less costly manner. I am aiming to have this proposed website up and running for Squadron No.42 ATC before the end of term 3 this year, as this project must be completed before this deadline in order to be able to be submitted as an assignment that counts towards NCEA.

Where
Social Environment: The community that will be using this outcome will be the people directly involved with No.42 Squadron - particularly the cadets, parents of cadets, and organisers for the Squadron, and also the wider audience, which includes any members of the public who are seeking to find information about ATC No.42 squadron.

Physical Environment:
Hardware: The outcome will be situated on the World Wide Web due to this being easily accessible for directly stated stakeholders (cadets, organisers etc.) as well as members of the public.
Software: Coded in HTML, CSS, PHP, and MySQL giving the website a dynamic aspect that enables it to be easily updateable by the manager of the site.

Holly Hanson

SCRUM Board with user stories and tasks



CLASSROOM IMPLEMENTATION STAND-UPS

“Regular stand ups were also a feature of Agile which was adopted in our projects. A presentation on the progress made recently on each of our own projects was made to the rest of the class weekly/fortnightly. Although this is designed to assist teamwork and inter-team communication on progress state for each other, this worked really well for out class in that it allowed ourselves to understand more concretely the progress we were making, just like the SCRUM boards did. Also, this allowed our class as a whole to judge our progress in relation to one another, so that a good indication of whether we were ahead or behind of everyone else could be seen.

This also promoted teamwork, even though these were individual projects, in that we were aware of the projects and progresses of one another and so were able to encourage and help one another if needed. ***In fact, we were so involved in each other’s projects that a sort of ‘culture’ had developed, where every time someone moves a task on the SCRUM board to the ‘complete’ section, we would all stop what we were doing and clap - because we all understood the achievement of the completion of even just one task.”***

KEY IDEA: REQUIREMENTS ANALYSIS

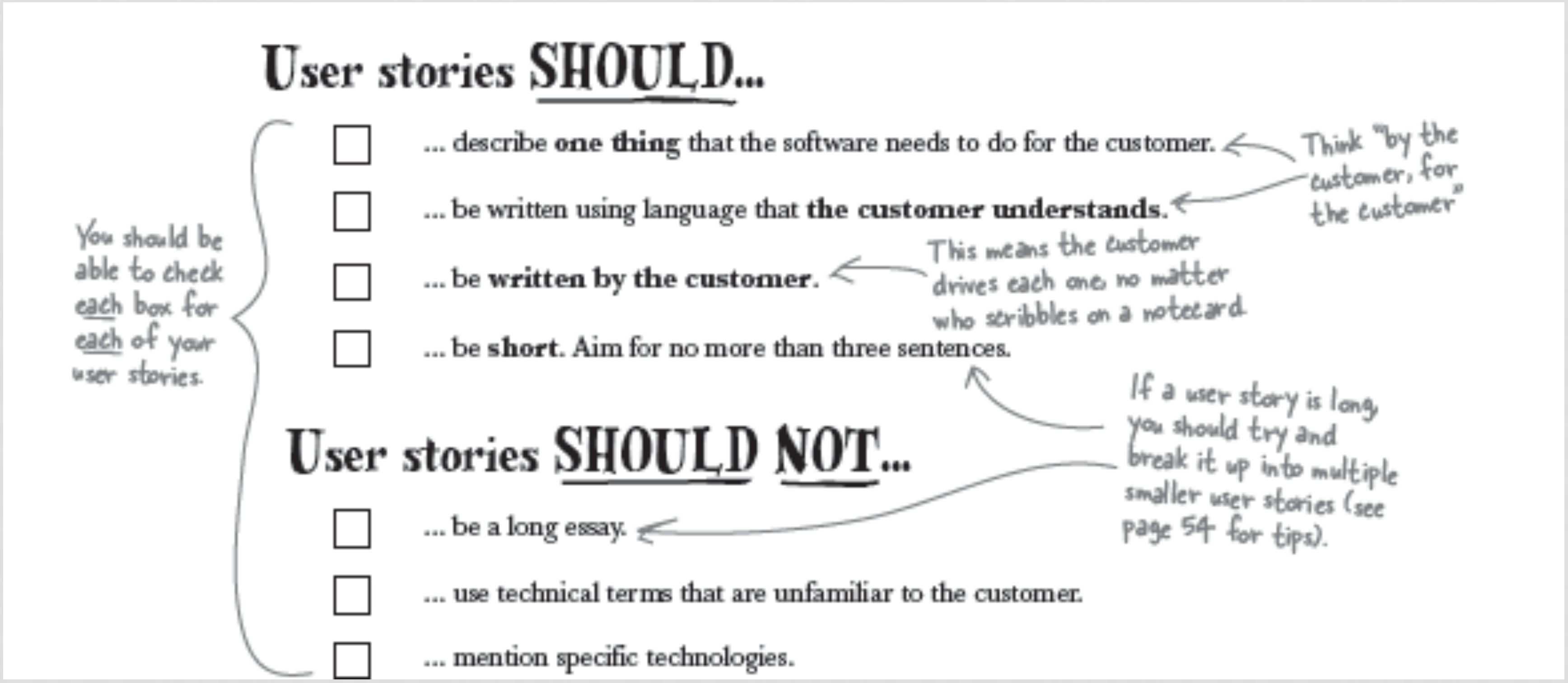
- Crucial to ascertain *functional and non-functional requirements* before launching into a software development project
- However, it is often difficult for the *stakeholders* to communicate what their requirements are
- Furthermore, different stakeholder groups may have conflicting requirements that need to be negotiated so that the software suits a range of needs.
- The hardware/physical environment that the software is to be run on will need to be considered and may also even change as the project develops.
- Finally, requirements need to be revisited regularly and the dialogue with stakeholders should be on-going.
- *Very strong correlation to NCEA brief development standard at Level 3.*

USER STORIES

- 16.2.1. PROJECT: FINDING THE REQUIREMENTS from the CS Field Guide could be a useful guideline for students who are developing their own project brief and need to research a variety of stakeholders.
- At this stage, it would be helpful to introduce the concept of 'User Stories', which are used in AGILE methodology:
  - A requirement should be written in the stakeholder's language and read like a user story: a story about how their users interact with the software that is being developed.
  - Requirements must be stakeholder-oriented and are written from the stakeholders' perspective describing what the software is going to do for the stakeholder.

USER STORIES

Excerpt from *Head First Software Development* (pg. 39)



User stories SHOULD...

- ☐ ... describe **one thing** that the software needs to do for the customer.
- ☐ ... be written using language that **the customer understands.**
- ☐ ... be **written by the customer.**
- ☐ ... be **short.** Aim for no more than three sentences.

User stories SHOULD NOT...

- ☐ ... be a long essay.
- ☐ ... use technical terms that are unfamiliar to the customer.
- ☐ ... mention specific technologies.

You should be able to check each box for each of your user stories.

Think "by the customer, for the customer"

This means the customer drives each one, no matter who scribbles on a notecard.

If a user story is long, you should try and break it up into multiple smaller user stories (see page 54 for tips).

USER STORIES

One format for user stories that we were introduced to at Orion Health is:

As a ……… I want………. So that I can………..

Examples from student projects:

As a reader of the eBook

I want to be able to choose what decade to read

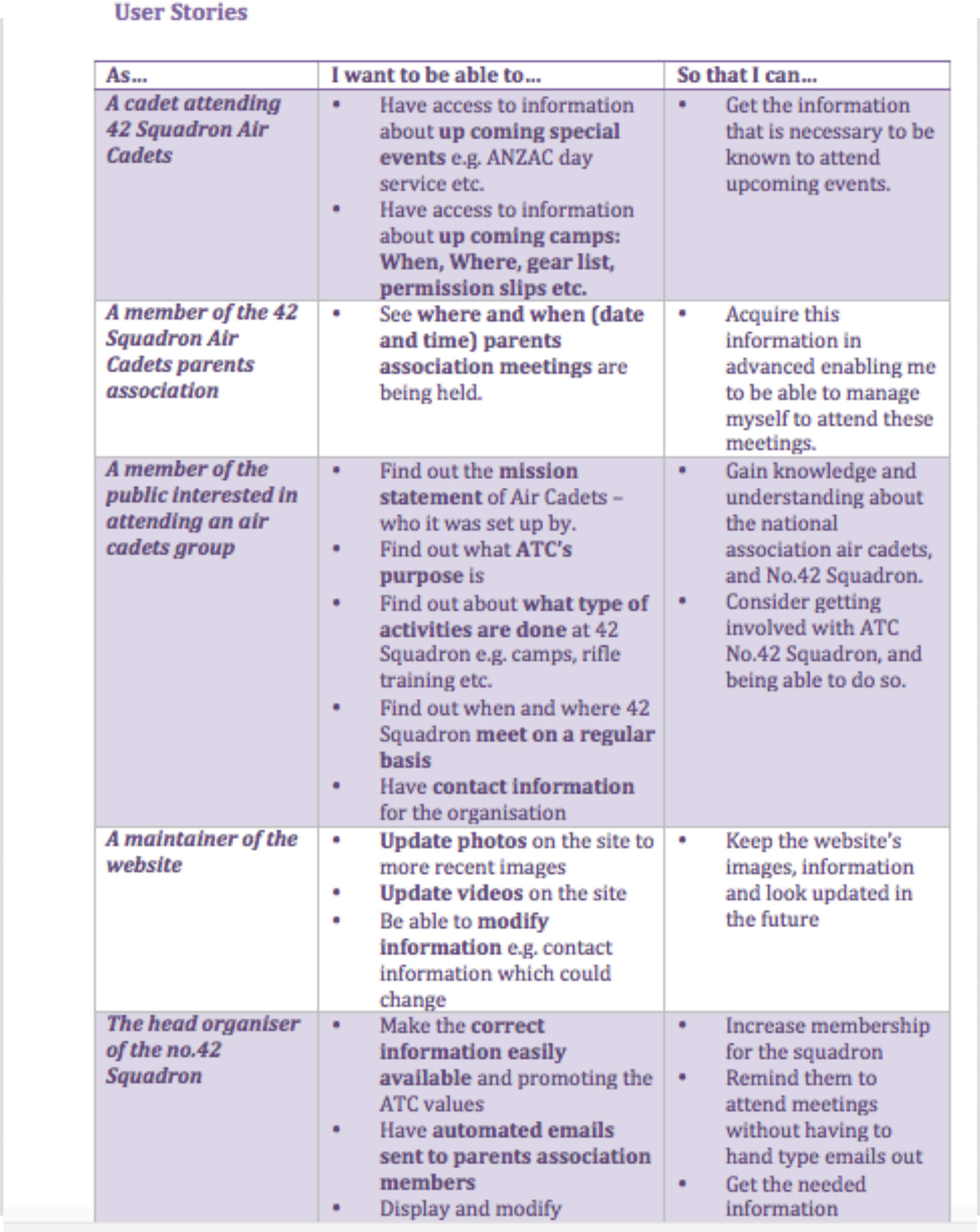
So I can skip to certain decades that I was in or have family and friends involved with so I don't have to go the through the whole presentation to see it.

As a customer of the restaurant

I want to be able to narrow down dietary requirements (for vegetarian and gluten free options)

So that I can determine if there is food that would agree with my diet.

USER STORIES



User Stories

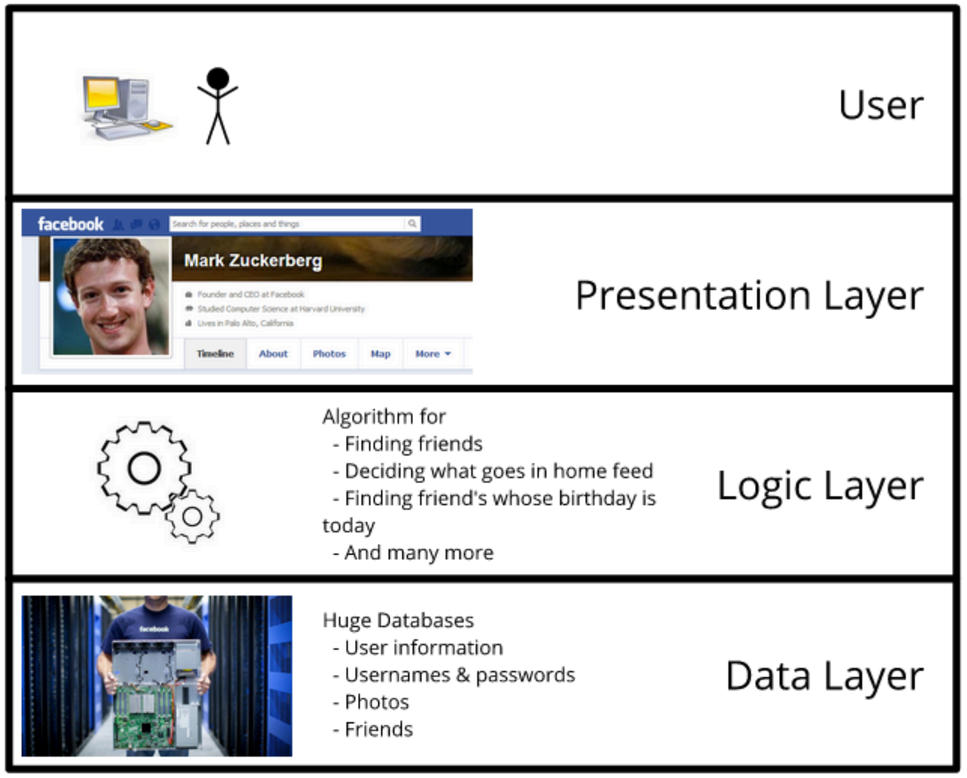
As…	I want to be able to…	So that I can…
A cadet attending 42 Squadron Air Cadets	• Have access to information about **up coming special events** e.g. ANZAC day service etc. • Have access to information about **up coming camps: When, Where, gear list, permission slips etc.**	• Get the information that is necessary to be known to attend upcoming events.
A member of the 42 Squadron Air Cadets parents association	• See **where and when (date and time) parents association meetings** are being held.	• Acquire this information in advanced enabling me to be able to manage myself to attend these meetings.
A member of the public interested in attending an air cadets group	• Find out the **mission statement** of Air Cadets – who it was set up by. • Find out what **ATC's purpose** is • Find out about **what type of activities are done** at 42 Squadron e.g. camps, rifle training etc. • Find out when and where 42 Squadron **meet on a regular basis** • Have **contact information** for the organisation	• Gain knowledge and understanding about the national association air cadets, and No.42 Squadron. • Consider getting involved with ATC No.42 Squadron, and being able to do so.
A maintainer of the website	• **Update photos** on the site to more recent images • **Update videos** on the site • Be able to **modify information** e.g. contact information which could change	• Keep the website's images, information and look updated in the future
The head organiser of the no.42 Squadron	• Make the **correct information easily available** and promoting the ATC values • Have **automated emails sent to parents association members** • Display and modify	• Increase membership for the squadron • Remind them to attend meetings without having to hand type emails out • Get the needed information

KEY CONCEPT: SOFTWARE DESIGN

- "Software design is all about managing this complexity and making sure that the software we create has a good structure." (Bell & Morgan, 2014)
- Subdivision – breaking the software into small parts that can be built independently.
- This concepts works well with Agile methodology and having small sprint goals.
- Although students don't generally work in teams, they can still break the development into subdivisions, e.g. creating a database back-end for a website, creating a menu system, etc.

KEY CONCEPT: SOFTWARE DESIGN

- Abstraction – breaking the software into layers. Each layer only needs to know how to communicate with the layer above it, but not how it works.

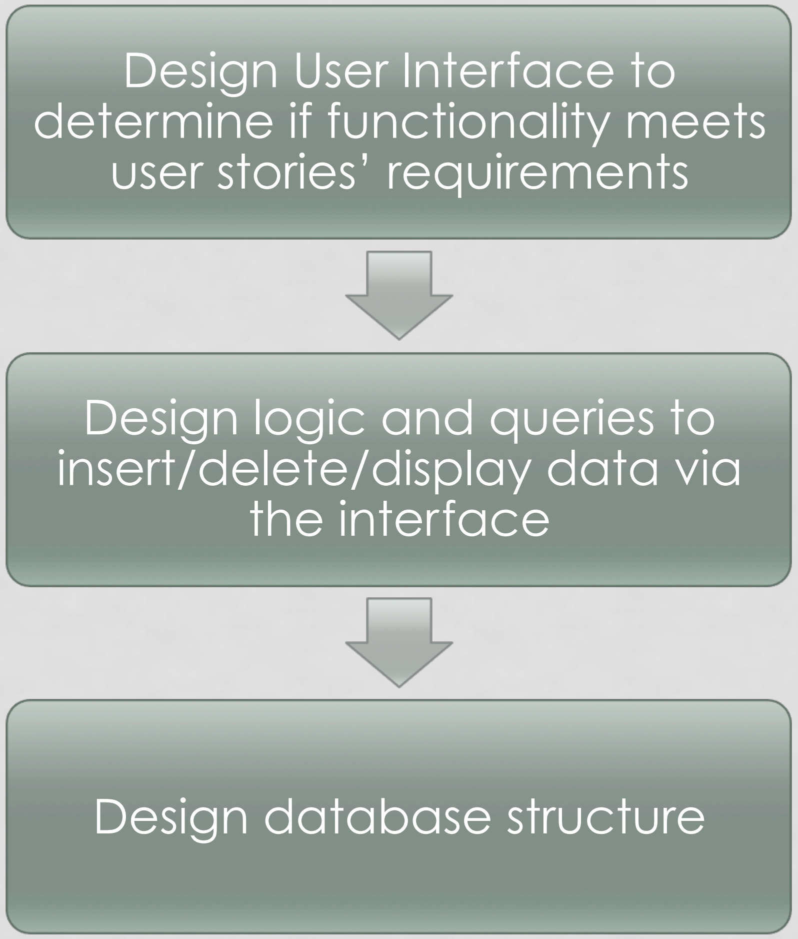


Facebook as a three-tier system

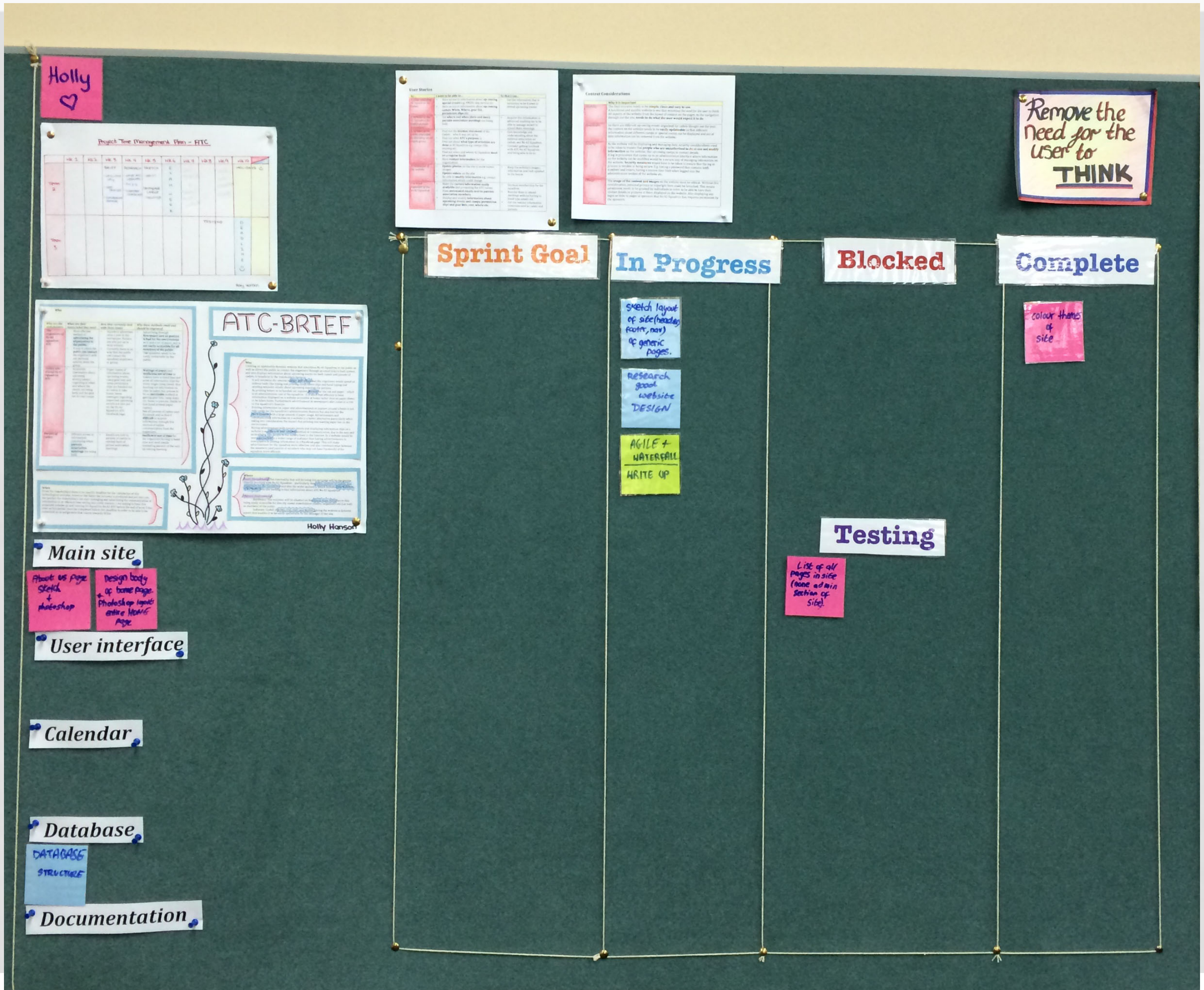
Image from CS Field Guide, Software Engineering

KEY CONCEPT: SOFTWARE DESIGN

- Although in the real world teams would be working in parallel on each layer, it is a concept that can be applied to independent student projects.
- They can use the concepts of subdivision and abstraction to break their projects into manageable layers and develop the design for the various layers.
- 16.3.1. PROJECT: DESIGNING YOUR SOFTWARE from the CS Field Guide could be used with the generic technology standard 3.2 Conceptual Design Development


Design User Interface to determine if functionality meets user stories' requirements
Design logic and queries to insert/delete/display data via the interface
Design database structure

SCRUM BOARD


Holly
Project Time Management Plan – ATC
ATC-BRIEF
Holly Hanson
Remove the need for the user to THINK
Sprint Goal
In Progress
Blocked
Complete
Testing
sketch layout of site (header, footer, nav) of generic pages.
Research good website DESIGN
AGILE + WATERFALL WRITE UP
colour themes of site
List of all pages in site (none admin section of site).
Main site
About us page sketch + photoshop
Design body of home page. + Photoshop layouts entire HOME page
User interface
Calendar
Database
DATABASE STRUCTURE
Documentation

KEY CONCEPT: TESTING

- Software must be tested before being released as clients and end-users wouldn’t be happy with a product that is full of bugs!
- Some bugs can be very severe and cause harm or even death, thus testing must be thorough, iterative and planned.
- **Automated testing** can be programmed to run repetitively find the probability of a bug occurring. Automated tests can also be run after each update to determine if an update has caused a bug to occur.
- **Manual testing** is also necessary. It may be exploratory in nature and completed by a member of the development team, or it may be carried out by an end-user.

KEY CONCEPT: TESTING

Unit Testing is another key concept of testing, meaning that as each discrete portion of the software is being developed, it is tested as a unit to reduce the complexity of finding a bug in the final code.

Integration testing takes place once all the units have been tested to determine if all the parts work together properly as a whole.

An example of unit test conditions within the Orion process is shown here:

As a Collector (Phlebotomist / Nurse)
I want to be able to easily search for collections
So that I can easily locate those collections I want to reprint labels for

Scenario: Collected orders are listed in the recent collections screen sorted by date

Given a patient has had two orders collected

When I open the recent collection screen for that patient

Then the collections for those orders are displayed
And the most recent collection is displayed first

Scenario: Labels are not available for reprint once the specimens have arrived at the lab

Given a patient has had an order collected
And the specimens from that collection have arrived at the lab

When I open the recent collections screen for that patient

Then the collection is not displayed

KEY CONCEPT: TESTING

- **User Acceptance Testing** is important to determine if the software is meeting user/stakeholder expectations.
- Whilst unit/integration testing may reduce bugs, it doesn't guarantee that the software is what the user actually needs.
- Acceptance testing left until the last stage before product release can be disastrous to software projects.

“I also prefer Agile as it ensures a better result. It means that I can change small things as they come, making it more manageable than having a lot of big things to be changed all at the end as I would with waterfall. The constant testing and feedback ensured that I would not overlook any problems as they were constantly being reviewed by me and my stakeholders. This ensured that I had the best possible outcome in terms on not only organization but fitness for purpose.

Rather than me working away at it then giving to others to pick apart in the design and usability of it, I worked with them and changed as the advice came in. Often in waterfall there is conflict between creators and testers and the testers job is purely to pick out any small or large problems, which could easily get frustrating. By taking the agile approach, I opted to work with my stakeholders, and I knew that at the end of the day the project’s outcome had to fit what they wanted. “

RESOURCES

Bell, T., & Morgan, J. (2014, March 25). *Software Engineering*. Retrieved March 1, 2013, from Computer Science Field Guide: http://www.cosc.canterbury.ac.nz/csfieldguide/teacherguide23012014/SoftwareEngineering.html

Pilone, D., & Miles, R. (2008). *Head First Software Development*. Sebastopol, CA, USA: O'Reilly Media.

Full Day Immersion Workshop at Orion Health (Christchurch) on Agile Development (SCRUM)

ADInstruments (Dunedin) testing procedures presentation to class and participation in testing process

Matthew Smart, University of Otago eLearning and eReseach Consultant, presentation to class on software development