THE UNIVERSITY OF WESTERN AUSTRALIA

CITS3200
Professional Computing

Coordinator: Michael Wise
Email: Michael.Wise@uwa.edu.au
http://teaching.csse.uwa.edu.au/units/CITS3200/

Learning Outcomes

- To develop awareness of the ethical and social responsibilities of computing professionals
- To develop experience using professional practices in a **teamwork** setting
  - *Note: Teamwork, not Groupwork*
- To provide a "programming in the large" experience as far as practical within semester boundaries

Learning Outcomes

- To allow for the integration of and reflection on previous computer science knowledge
- **Capstone**
- To develop student capability, confidence and maturity
- To encourage students to be responsible for their own learning
- To model industrial practice regarding commercial software development, and effective client relationships

Lectures

- Lectures aren't the primary focus of this unit, but nonetheless very important.
  - *Talks about important issues by high profile people*
  - *Are being assessed (see below)*
- Lectures covering Software Engineering, Ethics, Professional Practice
  - *I.e. How to be a Professional*
  - *Strong industry focus*
- Mostly face-to-face, some past recordings.
  - *Live lectures will be recorded*
- Details of topics and times on unit web page
  http://teaching.csse.uwa.edu.au/units/CITS3200/lectures.html

The project

- A major piece of work, about 300, 360 hours
- Group size will normally be 5-6 (i.e. 60hrs/person)
- Programming is one skill amongst many
  - *All the things you learnt in previous units will potentially be useful.*
  - *Learn to LISTEN*
- Requires commitment and maturity
- Worth 80% (!!!) of unit mark
  - *most of your learning happens here*
  - *Agile methodology*: *Scrum*
  - *35% group/product based*
    - Instead of one mark per project group, moderated mark per person based on group mark
  - *45% individual*

A Brief Word about IP

- Teams have IP in the things that they create
- Project proposers also have IP (not least because the project is their idea).
  - *Possible third-party IP (e.g. tools, libraries)*
- Each project has a statement about the IP model preferred by the project proposer
  - *Very rarely a problem*
  - *Stated up front*

Team Allocation

- I will be assigning people randomly to teams, though I also aim to balance teams.
  - *Random is not necessarily fair*
- The mapping of people to groups will be posted by Tue morning of Week 1 under the *Files* tab for Professional Computing on MS-teams.

Project Allocation

- Arrange a meeting with your team members asap,.
- The list of projects can be found on the Project Proposal Management System, which services all the capstone units
- You can view the projects at: https://teaching.csse.uwa.edu.au/units/capstone/projects/submit-preference/1/
- Look through the list of projects and get one person from your Team to register your references by 5pm on Thursday using the top two parts of the projects page linked above.
- I'll post project allocations on Friday, attempting to maximise overall happiness based on lists of Teams' preferences
- Arrange to see your project Client asap

Team Requirements

- Teams must have at least one formal meeting weekly.
  - *Meetings must be minuted*
- Files of Booked Hours spreadsheets, Time sheet and minutes are to be submitted to your Project Team Auditor on Fridays
  - *Late Booked Hours spreadsheet, Minutes, Time-Sheet will result in reduced Professionalism mark*
- First mentor session is in Week 3
  - *4 sessions across semester with a mentor from industry*
  - *Mentor meetings are significant learning opportunities - Don't miss them!*
- When everything is due can be found on the unit Timetable:
  - *https://teaching.csse.uwa.edu.au/units/CITS3200/timetable.html*

Group-Based Assessment

- Project 80 % overall; 35% moderated group-based
  - *Moved from Waterfall to Agile(like)*
  - *3 Sprints with deliverables, (35% all together)*
- Timeline – see Timetable link
- Descriptions of the deliverables linked to Project page
- There are no formal labs or tutorials; no exam

Individual Assessment

- Project 45 %
  - *Professionalism, (10%)*
    - *Assessment of contribution (20%)*
      - The number of hours you contribute, div 10.
      - The mean/expected value is 6 (i.e. 60hrs)
    - *Structured personal reflections at the end of each Sprint (5% each, so 15% across the 3 Sprints)*
- Peer assessment (**)
  - *Using SPARK**PLUS** after each sprint. Produces an individual multiplier which is applied to group mark*
- Professional Development Portfolio 20%
  - *Reflection on each lecture (with reference to the content) on how that information is informing your practice now, and may help your career going forward*
    - *NO ChatGPT*

Role of Auditor

- An Auditor appointed by the Unit Coordinator will attend 3 Team meetings (see Timetable).
  - *Not there to advise on technicalities. Auditor in sense of one who listens*
  - *Role is to look at leadership (of current Team Manager) and input by team members; perhaps comment on team processes*
- Auditor will also check the veracity of claims for Booked Hours against evidence on GitHub
  - *GitHub to be use for all outputs: code, documents and testing logs*
  - *Auditor in the sense of auditing accounts*

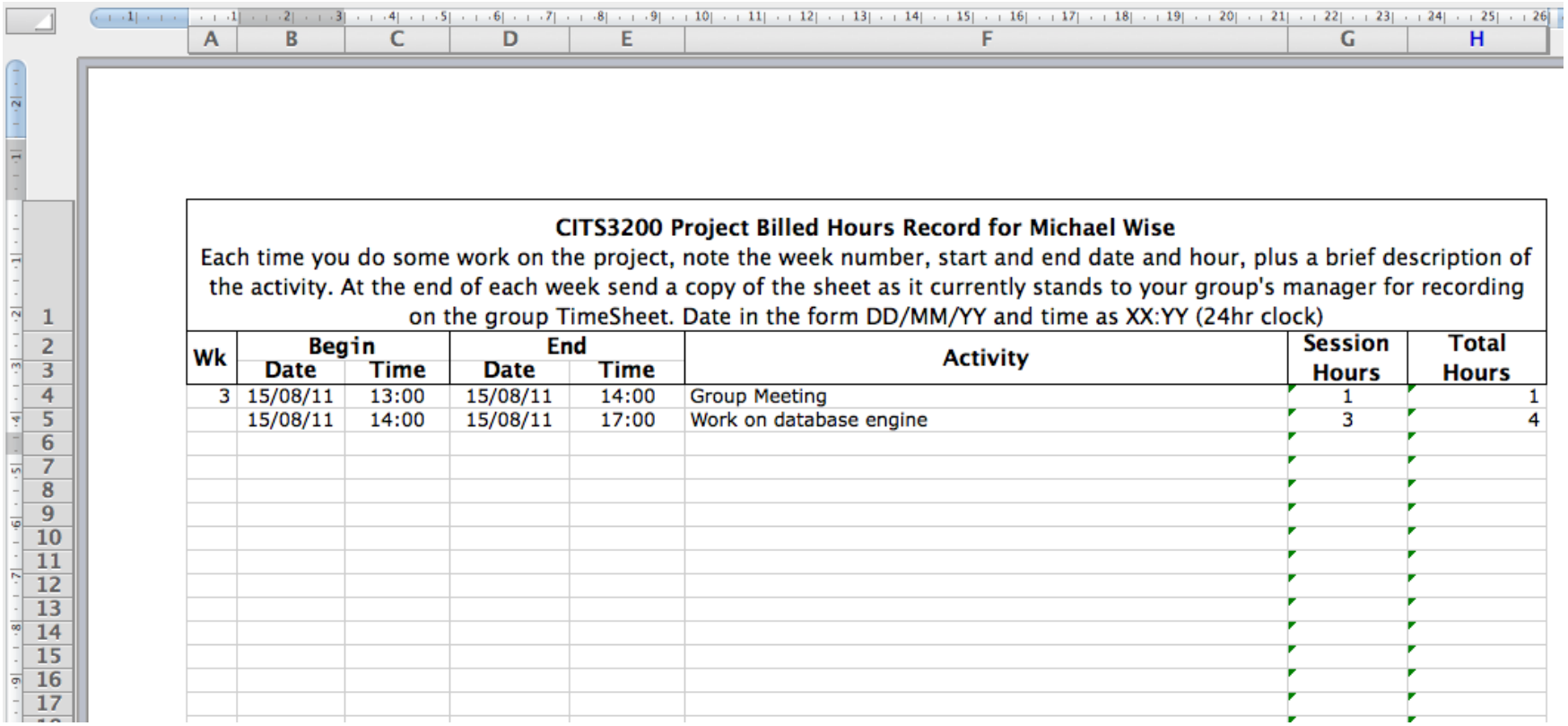
Resources and Professionalism

- Unit web site has a Resources page
- Many useful resources, including links to resources to bring you up to speed with Git.
- Please note the resources related to Ethics, particularly UWA Code of Ethics and Code of Conduct.
  - *Professionalism, both in person and electronically*
- If you have a Conflict of Interests with either your project Client or team's Auditor, please let me know.
  - *Personal/professional connection*
  - *Need not prevent you/your team working with that Client/Auditor, but must be disclosed and managed*

Time management

- Managing your time is a crucial skill
  - *Watch out for spending 15 minutes here and there without getting anywhere*
  - *Minimise wasted meetings*
- Two reasons:
  - *Don't overdo things, as CITS3200 not your only unit*
  - *Will be doing an analysis of time spent versus estimates for Sprint 2, 3 Retrospectives*

Booked Hours Spreadsheet



CITS3200 Project Billed Hours Record for Michael Wise

Each time you do some work on the project, note the week number, start and end date and hour, plus a brief description of the activity. At the end of each week send a copy of the sheet as it currently stands to your group's manager for recording on the group TimeSheet. Date in the form DD/MM/YY and time as XX:YY (24hr clock)

Wk	Begin		End		Activity	Session Hours	Total Hours
	Date	Time	Date	Time			
3	15/08/11	13:00	15/08/11	14:00	Group Meeting	1	1
	15/08/11	14:00	15/08/11	17:00	Work on database engine	3	4

- Booked Hours Spreadsheets are to be completed by **each member of the Team** showing what you did that week, both General and Requirements-related tasks. Just add to the next line and resubmit
- Please don't mess with the formatting; not a robust document

Handling the Timesheet

- MS Excel spreadsheet
- Submitted by Team Manager every Friday *starting Week 2*, based on Booked Hours spreadsheet sent to him/her by Team Members
- Read the Instructions on the worksheet
- Your estimates will change over time – this is a good thing: it shows you are monitoring your project and constantly reevaluating!
- You will be referring to these estimate in your team reflections at the end of Sprints

TimeSheet - General Tasks

- List of general tasks are there already.
- You can add tasks, but don't remove them. If you add a task fill in the week added column
- Fill in the total hours your group has allocated to the project
- Fill in the team member responsible
- Each week fill in
  - the Actual time you spend on each task (A)
  - the time you estimate *is remaining*. (E)
  - If you complete a task put the week number in the completed column

Fill in the actual time spent

Fill in the estimated time remaining

If you compete a task fill in the week

Total Hours our group has allocated =																	
Task			ed time remaining, A=Actual time spent this week)														
Task Name	*Week Added*	*Team Members Responsible*	*k 5*	*week 6*		*week 7*		*week 8*		*week 9*		*week 10*		*week 11*		*Completed (Week Number)*	
			A	E	A	E	A	E	A	E	A	E	A	E	A		
Learning Techniques and tasks (eg bugzilla and CVS)	0																
Research and Investigation	0																
Requirements Gathering	0																
Requirements Analysis	0																
System Design	0																
Test Design	0																
Documentation	0																
Project Meeting and Communication	0																
Review	0																

Fill in total time allocated

Fill in the person responsible; be specific

Fill in the week added

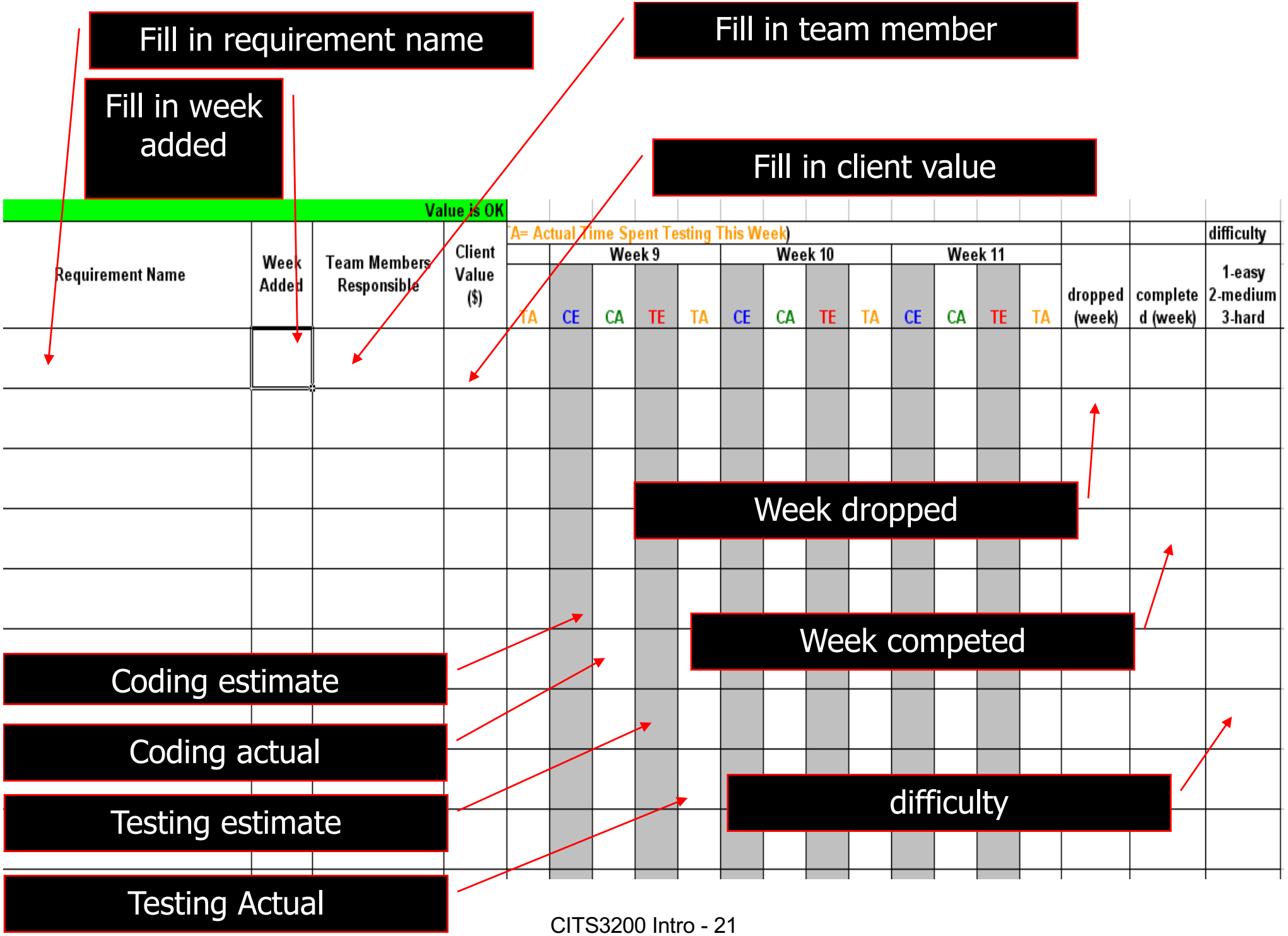
Add tasks here

Requirements

- Add the requirements in when you know them. Put in the name of the requirement
  - *The week added*
  - *Who is responsible*
  - *The client value*
  - *How difficult you think it is (easy, medium, hard)*

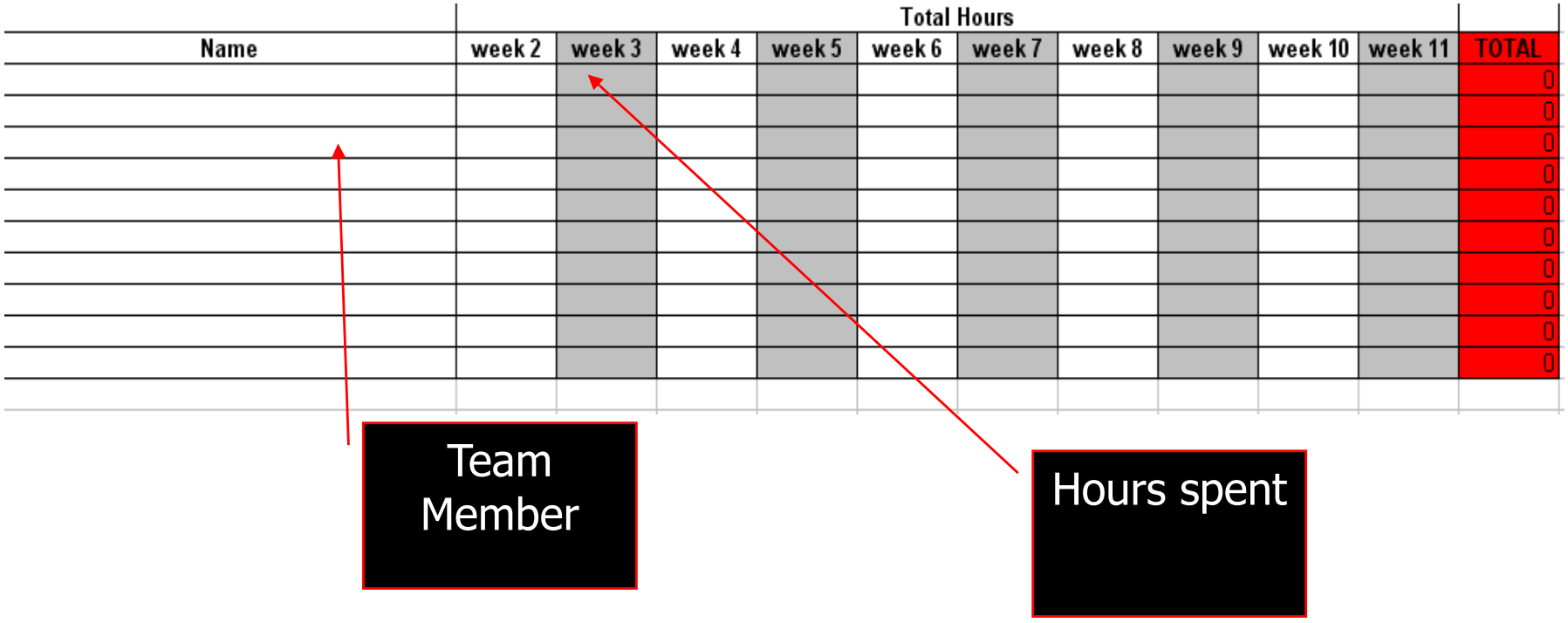
Requirements

- Each week fill in
  - *the Actual time you spend on coding (CA)*
  - *The Estimated time you have left on coding (CE)*
  - *The Actual time you spend testing (TA)*
  - *The Estimated time you have left on testing (TE)*
  - *If you complete a requirement put the week number in the completed column*
  - *If you drop the requirement put the week number in the dropped column*


Fill in requirement name
Fill in team member
Fill in week added
Fill in client value
Value is OK
A= Actual Time Spent Testing This Week)
Requirement Name
Week Added
Team Members Responsible
Client Value ($)
Week 9
Week 10
Week 11
TA CE CA TE TA CE CA TE TA CE CA TE TA
dropped (week)
complete d (week)
difficulty
1-easy 2-medium 3-hard
Week dropped
Week competed
Coding estimate
Coding actual
Testing estimate
Testing Actual
difficulty

Per Person

- Add in each team member
- Record how many hours they spent each week (Based on Bookable Hours)



Results

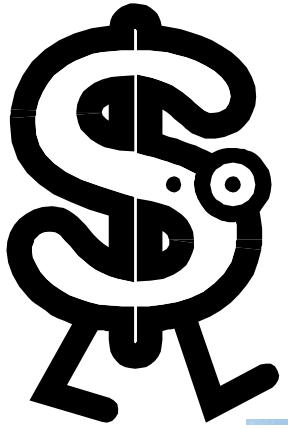
- Don’t change any existing formulas on this sheet but feel free to add anything that you think will help you
- Total Time Spent shows
  - *How your time differs with the recommended 60hrs*
  - *How your time differs with the time you agreed to spend*
  - *How your time differs with the estimates you gave when deliverables A and B were due*
- Requirements
  - *Number of requirements met*
  - *The value of the requirements met*
  - *The number scrubbed and the value scrubbed*
- Tasks
  - *The number of tasks remaining and the number completed*

TOTAL TIME SPENT			
Suggested Total Time to be spent by group =	0	(Computed as 60 hours per person)	
Actual Total Time Spent By Group =	0		
Hours Remaining From Suggested Time Budget =	0	within budget	
Hours Remaining From Your Groups Chosen Time Budget =	0	within budget	
Initial Time Estimate compared to Actual Time Spent =	0	within estimate	(Taken from week 4)
Initial Time Estimate compared to Actual Time Spent =	0	within estimate	(Taken from week 8)

REQUIREMENTS		
Number of Requirements Met =	0	
Value of Requirements Met =	0	less than 60% met
Number of Requirements Scrubbed =	0	
Value of Requirements Scrubbed =	0	less than 40% scrubbed

TASKS	
Number of Tasks Completed =	0
Number of Tasks Remaining =	9

Determining Client Value using the hundred-dollar test



What is it?

- A quick and easy method of getting your client to indicate the importance they place on a requirement. We use money as its something most people are used to thinking about!
- This will show you the relative values of each requirement – i.e. you can see how much more important one requirement is to another so you will know where your time is best spent!

How to use it

- Develop the list of your requirements with the client
- Go back to the client with the final list of requirements
- Tell them that they have $100 to divide over the requirements.
  - *There is no point distributing the money evenly!*
- The way that they spend the $100 indicates the priority that they put on each requirement

How to use it – An Example

Requirement	Value
Authentication System	$20
Database	$40
GUI	$20
Web Access	$15
Command line access	$5

Why we use it

In your project there will be more requirements than it is possible for you to meet with your time constraints

The $100 test shows us:

- where our time is best spent
- how much of the value of the total project we are meeting by fulfilling one requirement

Lost?

- Everything you need to know about the unit can be found at: http://teaching.csse.uwa.edu.au/units/CITS3200

Acknowledgements

- Members of the University community and industrial partners providing projects
- The companies and individuals that have provided mentoring
- Lecturers from across the University and from industry
- UWA Community of Practice for Teamwork