Team 4

Sprint 1 Retrospective

**Action Points**

* Scrum 9am on Monday morning
* Be more disciplined about swapping roles in pair programming
* Better distribution of responsibilities
* Use auto-format in Eclipse
* Fully install and learn to use git hub
* At the end of each user story, contact customer to check work and maintain contact
* Each member must update sprint backlog by midnight each night
* Sprint backlog added to git hub repository for proper version control
* Error tests should be considered at the sprint planning meeting and discussed with the customer
* Acceptance tests should be drafted and approved by customer at the very start of the sprint

**Positives**

Pair Programming

* Worked well as a team pair programming. Got a lot done & was more efficient than alone
* Pair programming was productive

Customer Meetings

* User stories were clearly established at the start
* Positive, constructive meetings with client
* Communicated well with customer
* Good communication with customer

Team work

* Good communication: everyone was aware of what needed done & when
* Good communication between team members
* Group got on well together
* Team worked well together (no fights)
* High attendances. Everyone attended when possible and when asked to do so

Sprint Planning

* Time estimates as a whole were accurate
* Most estimates were close to reality
* Well planned, sprint backlog well constructed, good time estimates
* Scrums went well, always knew what each other was doing

**Negatives**

Time Management

* Slightly slow to start to sprint, didn’t do as much as we could have first couple of days
* Finding time to scrum

Solution:

* Scrum Monday morning
* Discount Thursday

Better spread of responsibilities

* Too much reliance on one or two particular team members

Solution:

* Be more open to swap roles in pair programming

Code formatting

* Should have agreed to work to a coding standard, code formatted differently in files (x3)

Solution:

* Better use of version control
* Frequently use the Auto-Formatting tool in Eclipse

Version Control

* Problems with having 2 versions of code. Code overwriting.
* Version control did not work
* Version control!!! Disaster

Solution:

* Sit down together to jointly learn and install git hub

Communication with Customer

* Communication with customer was lacking during middle of sprint
* Could have made better use of access to client in the middle of the sprint
* Should have continued meeting the client more often during the sprint
* Should have checked interface more with customer
* Should have written and approved with customer the full acceptance tests, right at the start of the sprint
* Error tests should have been considered and discussed with customer at the start of the sprint

Solution:

* Set regular checkpoints to consult with customer during sprint
* Acceptance need done at start of sprint
* Error handling considered from start of sprint

Sprint Backlog

* Could be better at updating sprint backlog every day
* Should have added in extra tasks on sprint backlog as soon as they were thought of
* Underestimation of what tasks were involved in user stories (sprint backlog)

Solution:

* Set deadline for sprint backlog to be updated by each night
* Need to use git hub for version control to ensure team members aren’t simultaneously updating spreadsheet and overwriting each other