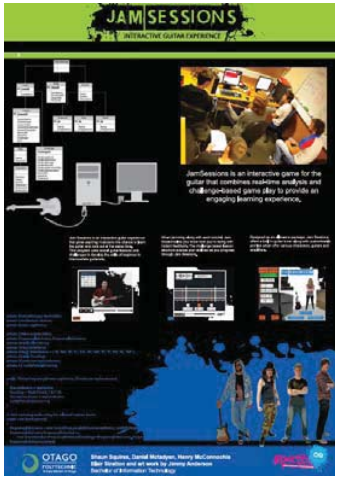
Learning from JamSessions

Samuel Mann
Shaun Squires
Henry McConnochie
Daniel McFadyen
Blair Stratton
James Anderson[1]

Patricia Haden
Hamish Smith
Lesley Smith

Information Technology
[1]Communication Design
Otago Polytechnic
Dunedin, New Zealand
samuel.mann@op.ac.nz



This quality assured paper appeared at the 1st annual conference of Computing and Information Technology Research and Education New Zealand (CITRENZ2010) incorporating the *23rd Annual Conference of the National Advisory Committee on Computing Qualifications*, Dunedin, New Zealand, July 6-9. Samuel Mann and Michael Verhaart (Eds).

Abstract
This paper describes the development of JamSessions: an interactive game for the real guitar that combines real-time analysis and challenge-based game play to provide an engaging learning experience. JamSessions was developed as a capstone project within an information technology degree. This paper explores the factors that made the project successful, with a view to enhancing future projects. These success factors include a strong group with a shared vision, a test-driven development approach, collaboration with design students, high skill and problem solving levels, and a project that maintained the students' passions.

Keywords:
Capstone, technical complexity, agile, project

Introduction

The capstone project is a feature of many computing degrees (Fincher 2001, Clear *et al.* 2001). Mann and Smith (2005,;2006; 2007; 2009) have worked to better understand the process of undertaking a capstone project in computing. Following this pathway, this paper explores the features of a highly successful group project, with a view to further enhance the capstone experience.

At Otago Polytechnic the projects are taught with an emphasis on a "real project for a real client". Students undertake the project in groups, following an Agile Development Framework (Mann & Smith 2006). The group project is assessed by an independent industry panel, largely on the basis of successful deployment. In 2009, two projects received 98% and 100% from the industry assessment panel. JamSessions is one of those projects (the other is described in Farquharson *et al* 2010).

The success of JamSessions can be seen in different ways - in this paper the focus is on the technical complexity and the student experience.

Paper methodology

The primary source for this paper is the evidence portfolio compiled by the student group. This portfolio contains evidence of the process of development along with critical reviews from each of the team. (http://bitweb.ict.op.ac.nz/wiki/_Final_Assessment). An overview of the components of JamSessions is shown in Figure 1.

The opportunity that led to the project will be described first, along with the two iterations. The final iteration: "Robust Delivery" is described according to the major areas of development:

- Gameplay
- Sound capture and analysis
- Learning structures
- Database

For areas of development of JamSessions, we present an overview of design decisions, the technical complexity, quality assurance and deployment (a more complete description is available in the online portfolio). The critical reviews from the individual students and the combined group also provide insights into the workings of the group.

By way of synthesis, the paper finally discuss some of the attributes of this group with a view to learning for future groups.

First iteration

This project came from one of the team's first year business proposal (Shaun Squires). In this he described the concept of 'guitar hero, but for the real guitar'. This idea was further explored when two of the team implemented a multimedia application to teach guitar in their second year of the BIT. The natural progression from this non-interactive system led to the foundation of the project. The group had a good vision of the problem to be solved - to take this static multimedia application to the next level and add some interactive elements to it. The success of the JamSessions would come from interaction that kept the user involved and enthusiastic.

The goal was to fill an identified gap in the market for an interactive teaching tool for guitar. The current hype of the Sony Play station game: "Guitar Hero", portrayed exactly the kind of interaction envisioned. The point of difference was to utilize an actual guitar, instead of a simple button controller in the shape of a guitar which provided no elements of learning anything about guitars.

All members of the group were musically inclined and were enthusiastic about the possibility of designing a system that both entertains and teaches the skills and techniques required to play guitar.

Initial research examined whether this idea of an interactive guitar lesson program was already established. While there was no similar product on the market, several companies had expressed their interest in the idea. However they provided only previews and no release dates.

A stake holder analysis identified key people who would directly influence some project decisions. This project was slightly different from other projects because the group acted as its own client with the project set up from the outset as a business venture. This gave the students more control over the implementation and functional requirements of the system. The outcome was a functioning deliverable that was built with the purpose of being presented to an industry panel as a pitch for obtaining investment for further development funding.

Second iteration: functional delivery

The second iteration release was a combination of the vigorous testing and proof of concept. The release included a very detailed paper based prototype that involved every section of the system (Figure 2).

.

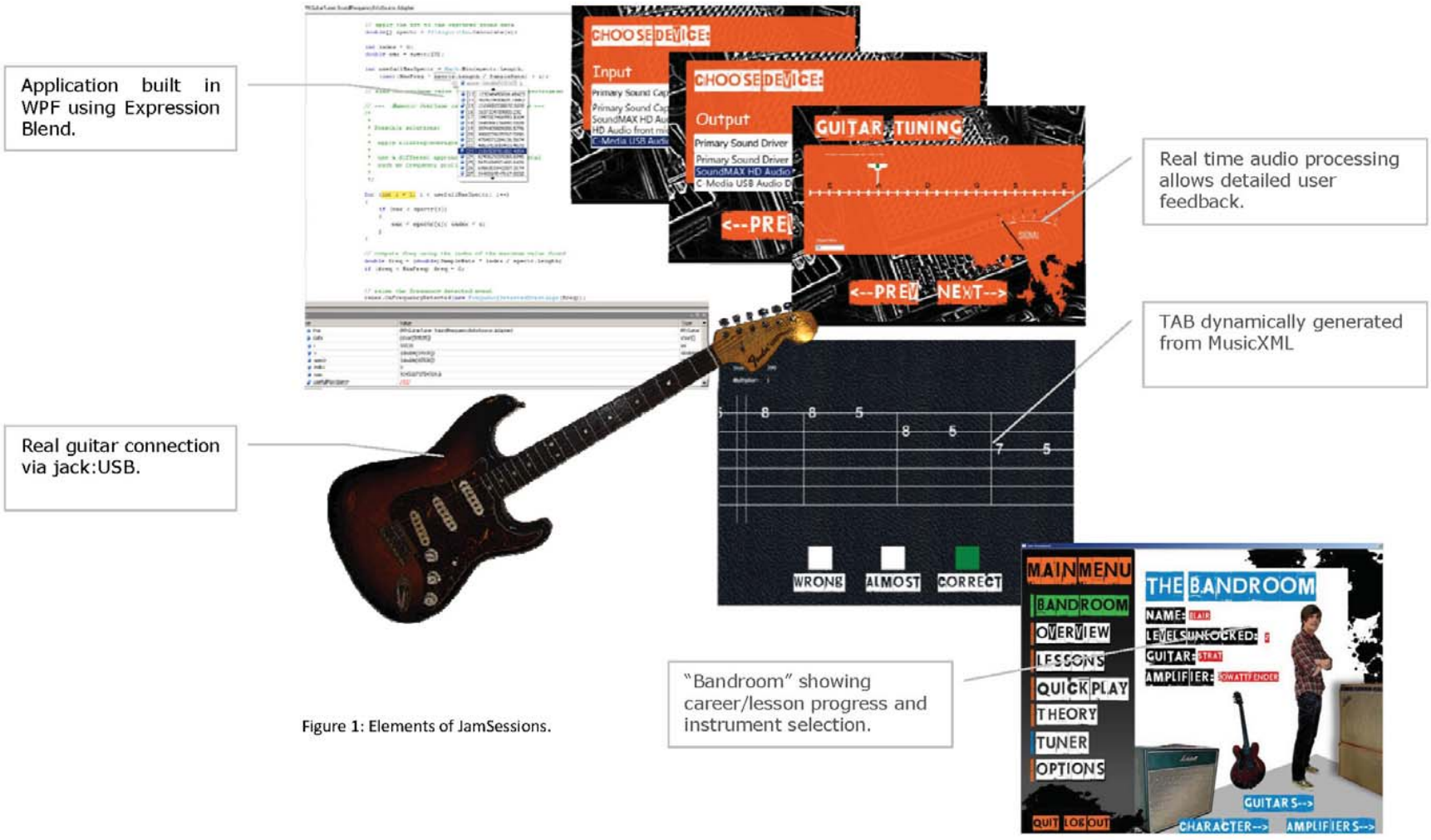


Figure 1: Elements of JamSessions.

The deliverable also included prototypes of the sound capture and frequency detection. It was vital that the group could demonstrate understanding and knowledge of how to implement these crucial requirements. Sound capture and frequency detection would be the pivotal part of the project to differentiate it from other methods of guitar teaching tools. Successful prototypes for both these requirements were created.

System 1: Game play

One of the main goals of the project was to create a challenge based game play environment that included real time feedback for the user utilizing the sound analysis functionality. This is the essential difference between this software and any other currently available teaching tool as it provides real time feedback to the user (note: educational aspects of gameplay considered in Lesson Structure).

Figure 2: Paper based prototype.

The functional requirements of the game-play section of the project can be broken up into several main sections:

- The ability of the user to play along with a song in a lesson and challenge based environment
- To generate tablature (TAB) dynamically to allow for a generic structure for each lesson and expandability purposes.
- To provide real time feedback and a scoring system

Technical aspects of gameplay

MUSICXML

A key functional requirement of the entire system and in particular the game-play element was to utilize a uniform data structure that allowed for the straightforward addition of new songs and expandability for future development. The use of MusicXML supported dynamic generation of the TAB and held song and note data which was compared with user input during a song/lesson.

INTERNAL DATA STRUCTURE

A robust data structure was created and implemented providing a generic base for storing note information that is used in both TAB generation and note comparisons in the real time feedback that is expandable to any piece of music loaded. On top of this it includes the functionality for utilizing all the features outlined on the functional requirements needed for our proposed system including dynamic tab generation, audio playback, sound capture and analysis, real time note comparison with feedback and a scoring system.

TABULATURE

The requirement for having TAB displayed scrolling across the screen when the user is playing was best satisfied by 'dynamically' creating the image for each song/lesson. This is to achieve the most expandable structure. Not having this dynamic and expandable system would lead to huge overhead creating and loading in new songs which is unacceptable as are developing this with further development and additions to content in mind.

When a user chooses to play a song the information from the MusicXML is filtered out to and stored in our own data structure. From here before the lesson starts the song data is traversed in a way where every note and chord is drawn to a 'canvas' and held in

memory. When the lesson/game starts the image is drawn and scrolled across the screen synced up with the audio.

The complexity of dynamically drawing TAB was huge; there are many factors in drawing TAB that must be accounted for. Not only must it be done dynamically but in a way from a structure that allows any piece of music to be imported and produce its corresponding TAB.

SCORING

After struggling with deciding on a formula to determine the hits and misses ratio, the group decided to test with a hit value of 1 as: (the capture frequency rate happens at 16 times a bar with the tempo set to 120bpm. So this means a 16th note only has 1 capture during its note). So needing to achieve the hit value 1 means you have to get it right on the first attempt as you will only get 1 attempt.

Gameplay success

The requirements outlined for this functional set for the 'game play' have all successfully implemented. The dynamic generation has allowed for expandability options along with generic coding structures that need only the source file for the notes and audio to play the music, draw and scroll the music in time with the audio while providing real time feedback to the user.

The structure has met requirements for implementing the functionality into the Jam Sessions final version: to produce the tablature dynamically, play the backing track, holds the current song notes data for real time feedback comparisons with user input. Finally the ability to use the sound analysis functionality to determine the user input (what they played on the guitar) coupled with the real time feedback and scoring system provides the basis for our 'challenge based game-play'.

System 2: Sound capture and analysis

The sound capture facility of Jam Sessions is a vital point of difference between this project and other computerised tools for musical education. It allows us to combine the interactive game play of successful products such as Guitar Hero and the real guitar experience of traditional music training.

The goal of providing real time feedback are captured in the following functional requirements:

- Analyse the notes played by the user in real time. This will enable the system to analyze the timing and pitch of the user's input
- Provide real time feedback to the user. The system will provide adequate feedback, based on the user's accuracy

The sound capture and analysis is the framework that allows us to combine the interactive game play with the real guitar experience of traditional music training. The sound capture facility uses a Direct Sound to take the input from the guitar, detects the fundamental frequency using spectrum analysis and provides feedback in real time. This feedback forms the basis of the game play and the guitar tuner.

Technical aspects of sound capture

Primary Implementation issues:

- Connection hardware
- Direct Sound
- Fast Fourier Transform (spectral analysis)
- Multi-threading
- Event architecture
- Fundamental detection
- Harmonics and chord identification

- Playback of sound data

Algorithms for sound capture and analysis were researched throughout each of the iterations of the development process. In the final implementation, the frequency analysis was performed using a standard Fast Fourier Transform. The fundamental frequency, as identified by the FFT is compared to the user's input and feedback as to accuracy is generated in real time.

System 3: Learning structures and content

One of the first functional requirements the group settled upon was that at the core of the project was to educate the user to learn the guitar. The JamSessions system contains a structured lesson plan that aims to meet the needs of all types of guitarists - whether a beginner, or a well accomplished guitarist. Given its core status, throughout the three iterations, this functional requirement has remained as the number one functional requirement to accomplish.

Lessons technical complexity

The group worked with a music teacher, John Dodd and decided that best approach was to align with the Year 9 curriculum. Mr. Dodd demonstrated each of the key elements that had successfully helped his students learn the guitar. This collaboration resulted in a progressive learning structure (http://bitweb.ict.op.ac.nz/wiki/Lesson_Structure).

The lessons were written up into sections, identified for gameplay as levels. The levels were divided up into sections that contained similar elements. This took into account that people using our system might have never even picked up a guitar. In other situations, there will be users that may have played guitar for several years. This was crucial to our structure of learning. Starting from the very basics, lessons were developed to include more complex techniques throughout the system.

The lessons also utilised multiple means of teaching technical skills. A written form of lessons provided the foundation for the film and interactive categories. Finally the lessons would be incorporated into the interactive feature of our system. Some lessons were basic introduction elements about learning the guitar and not actually learning to play the guitar – for these modules a quiz section was developed.

The lesson content was slightly altered to incorporate the content into film scripts. The group had never developed any kind of filming production or scripting so this was certainly a challenge. Scripts were tested by repeated reading and low-fi videoing many times before the actual filming, as we only had limited time to work in the production studio (Figure 3). The group collaborated with an actor and a design student to help us produce and direct the filming of the lessons. The editing process took a lot longer than expected but the finished tutorials were easily incorporated into JamSessions.

System 4: Database

The database is a key part of Jam Sessions as it holds everything from the players name to path names for media content to the unlocking of levels.

A data structure was created and implemented that provides a generic base for storing note information that is used in both TAB generation and note comparisons in the real time feedback that is expandable to any piece of music we need to load during the lifetime of the application. On top of this it includes the functionality for utilizing all the features outlined on the functional requirements needed for the system including dynamic tab generation, audio playback, sound capture and analysis, real time note comparison with feedback and a scoring system.

The interface design was tested at each stage of development (Figure 4) with the goal of making the navigation flow was flawless. Testing was also carried out on data binding. This required testing that the users profile and contents could be saved and accessible throughout the whole system.

Figure 3: Project team in the studio; and the final lesson view.

Figure 4: Testing of nearly completed system.

Learning from the JamSessions.

In this section we allow the 'student voice' to tell the story of their experiences of the project. This is interspersed with insights for future capstone projects.

The group followed the Agile Development Framework taught in the prerequisite Software Engineering course. In this course students are introduced to the agile concepts of an iterative approach, deep commitment to communication with stakeholders and embracing change. This is clearly evidenced in the student reviews:

> *Software Engineering reinforced the critical processes and methods to follow and understand in the production of our system. …. We were able to embrace change at critical points in the system, without letting the focus of quality and time hinder our progress.*

They were able to stay focused on the end goal:

> *Our main objective and primary functional requirement was to educate the user of system. We strived to understand and incorporate the goals of the user into our project.*

While still embracing the many changes inevitable in a project development:

There were many times in the projects lifecycle that the concept of embracing change occurred.

Our major functional requirements never changed throughout the lifecycle of our system. We were always intending on deploying the system with the main requirements implemented. We had extra functional requirements and ideas that didn't get executed because of the time frame. However this did not deter the user experience of our system.

They utilized a defined agile framework (Mann & Smith 2006) as the basis for the development but were able to incorporate alternative methodologies when necessary:

The first two iterations were followed very closely to the agile framework. We held scrum meetings and group discussions and collaboration several times a week. The third iteration however took more of an extreme programming (XP) approach. We still adhered to the agile approach, but we intended to improve the software quality by rigorous testing and frequent release versions. This method of short development iterations lead to the improvement of productivity and allowed us to implement new ideas and changes to the system.

While the group had no external client, the agile practice of engagement with stakeholders was followed closely:

Our group had no client. We established our project as a business venture that could possibly be marketed and implemented for commercial release. To substitute the client figure, we sourced several major stakeholders to have an active role in the overseeing and decision making of our project. We valued the relationship with the stakeholders and always kept them informed on our major developments. We held regular meetings and informal discussions to further the quality of our project.

This situation led to considerable reflection within the group on the role of the client and the nature of client/stakeholder communications:

Considering the unique nature of the product of creating something where we ourselves are the client had an interesting effect on the project. Client- employee communication was not documented as we would just discuss topics as they arose. Also it lead to us looking to outside sources to act as stakeholders with high input into the project in terms of defining the functional requirements and advising us on the learning structure we implemented for example.

Prototyping was a key element of the methodology:

The first two iterations of our project were primarily focused on prototyping. We knew the complexity of the problem we were trying to solve was a highly complicated challenge.

Resulting in a focus on quality assurance:

The immense amount of time we spent on prototyping and testing before actually implementing the software was our indication that we would meet the essential functional requirements.

We tested each functional requirement for robustness and stability. All of the testing was performed in the prototyping stages which allowed for change and had good impact on the implementation in the third iteration. The testing was accomplished with having multiple users navigate through the prototype. We observed every test thoroughly and appropriately changed the necessary parts of our system accordingly.

Previous Otago Polytechnic BIT capstone projects have benefitted from working closely with design students (eg Goodsir *et al.* 2005, Nicholl *et al*. 2007). Sometimes, however, there have been tensions between the IT and communication design students, with neither side really sure of the others' skills. The partnership for JamSessions was very successful, with both sides clearly understanding the requirements of their colleagues:

This proved to be a massive success as we had already worked out our overall interface navigation and only required the actually screen designs allowing the designer to focus solely on the artwork.

The Agile Development Framework places high importance on communication within the development team rather than a document-centric approach (Noble et al. 2004, Mann and Smith 2006). The "barely sufficient documentation" to support that communication can be seen in the group's use of the wiki both as a

collaboration tool and a self generating evidence trail, and the heavy use of the whiteboard as an extended planning game and central hub:

> *During the lifetime of our project the development process followed the same generic theme. On a regular basis (usually one to two times a week) we discussed the current project status identifying and allocating tasks for the coming days.*
>
> *The allocation and discussion of potential solutions for each topic was done so in a informal scrum like manner in the way we did not always record the discussion for record purposes but rather keeping notes and goals written on whiteboards to give the group an overview of what is going on. This also allowed us to cross off completed tasks to show visual conformation of the work done.*

Many authors of agile persuasion promote the use of sandbox development (Richardson & Gwaltney 2006) or always having a shippable version (Tate 2005). In discussing revolutionary versus evolutionary processes, Mann and Smith (2006) identified that student groups can struggle with these concepts. The JamSessions group got it right though and provide an exemplar for test-based sandboxed development:

> *Whenever we had made significant improvements in development we would thoroughly test the product with lecturers and other students, we would then make a new version of the Jam Sessions and develop form that, meaning we always had a working copy of the database that we could always go back to.*
>
> *… we made an effort to attack each task in a separate 'mini' projects allowing continuous development towards the overall target without having to fight for time on the master version. This way we could test and develop each section and then have 'combine sessions' where we would update the latest/master version.*

A capstone project is more than a big project at the end of the degree. It is supposed to pull together the learnings from across the degree, providing a platform for integration and application:

> *Another point of interest was the overall structure and implementation of the 'Game play' within our project where I was able to utilise techniques learnt throughout the year from 'Object Orientated Programming' and 'Advanced Algorithms and Data structures' classes. This structure caters for the entire game play system from generating the TAB, playing the Audio, utilising the Frequency analysis, scoring, feedback and syncing the lot together.*
>
> *Overall it was a pleasure to see how our initial sketches of the system and ideas became reality. It was tremendous to be able to apply the skills we have learnt over the past three years of the B.I.T and create an amazing project.*

Having completed the capstone, the group expressed value in areas outside traditional technical areas- here they describe educational content:

> *By adhering to the knowledge of musicians we consulted for our lesson structure, we managed to implement the lesson structure to a very high standard. With the knowledge we have gained over our time at polytechnic, we were able to create a structured method of applying the lessons into a multimedia platform. The code created to generate the features of our lessons was implemented elegantly.*

It is interesting that the students saw the areas of educational structure, editing films and collaborating with a designer as "outside the IT curriculum". It is difficult to predict the areas their careers will take them but those areas are not unlikely. Perhaps as academics we need to do a better job of giving the message that the skills they have learnt are not just about specific technology but more generic and that few areas are outside the reach of IT.

Another feature of JamSessions, is that was a group project – like most of Otago Polytechnic BIT capstones. Critical to the success of group work is an aligned vision. This clear project vision must be both of the project itself and the context in which it is carried out (Augustine 2005). JamSessions demonstrated this coherence, both of the task and the process:

> *The cohesion of our group and our ability to take and give advice, work together in peer programming environments and work together to attain a product that is better than any*

one of us could make on our own is something I am proud to a part of.

…

The chance to work in a group with likeminded and talented students was a great pleasure. Everybody in the group was very reliable and we created a great atmosphere for learning and developing our system. Everyone was willing to help each other out, as we each brought different skill sets into the project.

This vision set high standards:

I think the need and willingness to embrace change relates to the fact we all realised the potential of our project not only for potential business opportunities but a certain 'cool factor' that many other projects can lack. Having realised this potential in every aspect of the project we did not want to do anything by half standards and were aiming for high marks right from the start.

Conclusions

JamSessions was a technically challenging capstone project. It required: the use of C#.NET audio processing engine using a Fast Fourier Transform algorithm for frequency detection; a WPF front end, user controls designed in Expression Blend; dynamically generated guitar tabulature from underlying musicXML files; high quality digital video for lessons and demonstrations; and an underlying SQL Server database stores user information and student progress.

Despite - and perhaps because of – these challenges, the student group performed extremely well.

Insights for future project groups include:

- The importance of always having a working deliverable
- Heavy use of peer programming.
- Working with designer
- Multilayered communication supported by as light as possible tools.
- Cohesive group, aiming high
- Integrating all degree
- The "real project for real client" can be simulated with a passionate group and a stakeholder structure to provide objective guidance
- Empowerment of project
- Excitement of computing

The Agile Development Framework is working for such high performing groups. Especially useful are the expectations of early deliveries and a focus on transparent communication. It is interesting that students thought they changed from ADF in last iteration – instead adopting Extreme Programming for the robust built (despite this being promoted in Software Engineering).

Perhaps the conclusion is best encapsulated in these final quotes from the student group:

It was one thing to dream up the ideas but to see them all implemented and working is just awesome.'

The most pleasing aspect of the project was that it always remained interesting. The chance to create an interactive guitar teaching tool was something that I never thought I would be doing when signing up for an IT degree.

References

Augustine, S. (2005). Managing Agile Projects. Upper Saddle River, Prentice Hall. 229

Clear, T., Young, F. H., Goldweber, M., Leidig, P. M., & Scott, K. (2001). Resources for instructors of capstone courses in computing. ACM SIGCSE Bulletin, 33(4), 93-113.

Dawson, C. (2005). Projects in Computing and Information Systems: A student's guide. Harlow, UK, Addison Wesley. 253

Fincher, S., Petre, M., & Clark, M. (Eds.). (2001). *Computer Science Project Work: Principles and Pragmatics*. London: Springer. 267

Goodsir, K., Douglas, A., Holo, S, Mann, A., Smith, L., Sewell, A. and Mann, S. (2005) SciCards Proceedings 18th Annual NACCQ,

Mann, S. & Clear, T. (eds). Tauranga. July 10-13th July 2005. p352

Mann, S. and L. G. Smith (2006). "Insisting on best of practice within capstone projects." New Zealand Journal of Applied Computing and Information Technology: 57-63.

Mann, S. and L. Smith (2006). Arriving at an agile framework for teaching software engineering. 19th Annual Conference of the National Advisory Committee on Computing Qualifications, Wellington, New Zealand, NACCQ in cooperation with ACM SIGCSE.

Mann, S. and L. Smith (2006). A value proposition model for capstone projects. 19th Annual Conference of the National Advisory Committee on Computing Qualifications, Wellington, New Zealand, NACCQ in cooperation with ACM SIGCSE.

Mann, S. and L. Smith (2007). Software engineering class eating its own tail. Ninth Australasian Computing Education Conference (ACE2007), Ballarat, Australia, ACS.

Mann, S., L. Smith, et al. (2009). Project Wiki. Snapshot paper in 22nd Annual Conference of the National Advisory Committee on Computing Qualifications. S. Mann and M. Verhaart. Napier, NZ: 192-193

Nicholl,A. Coup, L., Labes, J., Haden, P., Mann, S., Bagrie, J. (2007) MARINE QUEST: Environmental E-Learning for Primary Students 20th Annual Conference of the National Advisory Committee on Computing Qualifications, Nelson, New Zealand, NACCQ in cooperation with ACM SIGCSE. 287

Noble, J., S. Marshall, ., Marshall, S., & Biddle, R. (2004). Less extreme programming. Proceedings of the sixth conference on Australian computing education - Volume 30, Dunedin, New Zealand.

Richardson, J., & Gwaltney, W. (2006). Ship It!: A practical guide to successful software projects. Raleigh, NC: The Pragmatic Bookshelf. 198

Tate, K. (2005). Sustainable Software Development: An Agile Perspective, Addison Wesley Professional, NY.