

Belbin Team Roles, Organisational Patterns and eLearning: A Case Study

Angela Gibson

Christchurch Polytechnic Institute of Technology

gibsona@cpit.ac.nz

Trevor Nesbit

Christchurch Polytechnic Institute of Technology

nesbitt@cpit.ac.nz

Abstract

In 2004 Christchurch Polytechnic Institute of Technology (CPIT) embarked on a project to develop eLearning content for a number of modules from the Certificate in Computing (CIC) that is overseen by the National Advisory Committee on Computing Qualifications (NACCQ).

The purpose of this paper is to describe the process that was used to manage the development team and the key issues that arose, how Belbin Team Roles as described in Belbin (1981) could have been applied at the inception of the project, and how the use of organisational patterns as described in Coplien and Harrison (2005) could have been applied in making decisions about how the team would function.

The paper identifies how some aspects of Belbin Team Roles were extremely helpful in the managing of the team, how some organisational patterns confirm different aspects of how the team was managed, and that had other organisational patterns been applied at the start of the project some aspects of the overall project would have been improved and enhanced.

Keywords: eLearning, Belbin Team Roles, Organisational Patterns.

1 Introduction

In 2004, Christchurch Polytechnic Institute of Technology (CPIT) commenced a project that aimed to develop eLearning content of a number of courses from the Certificate in Computing (CIC) that is overseen by the National Advisory Committee on Computing Qualifications (NACCQ).

The purpose of this paper is to briefly describe the project and to explore how Belbin Team Roles as described in Belbin (1981) and organisational patterns as described in Coplien and Harrison (2005) can be used in the planning of such projects.

The results of this analysis show the usefulness of this application of Belbin Team Roles and organisational patterns to projects of this nature, and should be

This quality assured paper appeared at the 19th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2006), Wellington, New Zealand. Samuel Mann and Noel Bridgeman (Eds). Reproduction for academic, not-for profit purposes permitted provided this text is included. www.naccq.ac.nz

considered as part of the initial planning process by any organisations considering similar projects.

2 Description of Project

The project involved taking material that was (at the time) being taught in a face-to-face environment and develop that material for an e-Learning environment. The content was to be developed by academic staff that it was assumed would have everything they would require at their fingertips. This assumption was based on the perception that everything a tutor uses to teach in a classroom environment was either in a book, word document, or a slideshow. Such a perception takes little account of the reality that much of what is used is actually embedded in a tutor's 'grey-top'.

It was decided that a product should be developed from scratch rather than use commercially available or open source platforms such as BlackBoard or Moodle. This was because the widest possible base of students was to be reached. By giving the students a CD, they need not be connected to the Internet every time they studied. It also meant the developers could produce animations and sound to help illustrate concepts and not be concerned with bandwidth limitations.

Two of the four courses that were to have their content developed for e-Learning had an academic staff member provided who taught these papers face-to-face. This allowed first hand knowledge of the existing course so any potential issues that may have been faced by the e-Learning students could be taken into account in the development of the material. The other two courses were allocated a staff member who had taught at a higher level. By ensuring the teaching staff had taught the subject at the prescribed level or higher, the quality of the content was assured.

This project needed to ensure that while the students did not have a tutor in everyday classroom surroundings, the final product still needed to provide as many different ways of learning for the students as possible. The main aim was to ensure no student was left out or disadvantaged, with this meaning that the final product needed to cater for Kinaesthetic, Aural, and Visual learners. To cater for all the different learning styles there were voice-overs for the aural learners; text and animation for the visual learners, and tutorial that enable the kinaesthetic learners to 'investigate' the software.

The pedagogical design used in the development was a blend of constructivist and instructivist as outlined in McKenna and Laycock (2004). Some of the key aspects of the completed product included many of the key areas of

the constructivist approach as outlined in Vrasidas (2004) who outlined the key assumptions of the constructivist approach which are that learners learn best when they:

- They engage in active learning
- Represent knowledge in multiple ways
- Participate in authentic activities with real-world connections
- Their work is evaluated following authentic assessment
- Collaborate with peers in solving real world problems.
- Have access to distributed tools for meaningful learning

(Vrasidas, 2004)

The example described in McKenna and Laycock (2004) included a “highly interactive Flash artefact” based on the constructivist approach and a “drill-based artefact” based on the instructivist approach.

The nature of the project under review in this paper includes aspects of the constructivist approach, with the learner being able to actively engage, have multiple paths to their learning and being able to participate in real activities. There are also aspects of the instructivist approach, with learners able to engage in repetition or drill-based activities.

During 2005, the first students enrolled in the CIC courses that used the material. It remains to be seen how the completion and success rates of students using the content that was developed, compared with the completion and success rates of students who have enrolled in the same courses taught in face-face mode.

3 Key Issues and Belbin Team Roles

A number of key issues arose in the management of the content development team, with these including the multidisciplinary skills that were required; the delegation required to achieve optimal performance in a small team; estimating the duration of tasks; the imposition of additional requirements; and that initial requirements were not always documented.

3.1 Multi-Disciplining Skills Needed In The Team and Application of Belbin

The project team for the development of the content was a small team, that most of the time was made up of four to five people. If the team player roles of the members of the team are looked at in the light of Belbin’s Team Roles as described in Belbin (1981) the team was made up of predominantly primary roles. Belbin (1981) describes how people generally have team roles that are primary, secondary or occasionally tertiary roles. The different team-role types described by Belbin (1981) are reproduced in Table 1.

Of the four main members of the content development team, the following primary team-roles were present:

- Person 1. Plant
- Person 2. Co-ordinator
- Person 3. Implementer
- Person 4. Implementer

The make-up of this team allowed ideas to be turned into achievable goals. However, it became apparent the team was lacking in certain team-role types. The team was missing people that had natural roles such as:

- Monitor Evaluator
- Completer Finisher
- Resource Investigator
- Shaper
- Team Worker
- Specialist

As with any project, every team member had to be adaptable enough to pick up other roles, and some roles were easier than others to step into. Each member easily became a ‘team worker’. Due to any timelines, all members had to become Completer Finishers, with this possibly being the most important role for the project.

Some of tasks that were allocated within the team fitted well with their primary team-role type. Person 2, who had a natural role of Co-ordinator, had the job title of ‘team leader’. This allowed the natural role to come forward. The team leader was required to interface between the third party software company and the team. This person was also required to maintain the team equilibrium within the development environment.

Each team member was responsible for writing the content of his or her ‘allocated’ subject. The first stage of editing was to check the content, with this in general being conducted by the team leader whose primary role was to maintain accurate content.

After the content was stable the editing process could begin. Each course went through a further one hundred and twenty edits. Each edit took on average six hours. The editing process was done so that the content was as comprehensive as possible. It entailed proofreading, continual checks of changes that had been implemented, with the video scripts requiring checking as well. To achieve the desired outcome, each team member had to take on more roles, such as the Monitor Evaluator. This allowed a wider range of options to be considered.

The most important role of all was one the team had to adapt to, as it was not a primary role of any team member, was the Completer Finisher. Taking up this role meant that tasks could be completed on time. The team functioned well with its natural roles. Having two members with the Implementer role meant that they had a high degree of reliability, efficiency, and that many of their ideas could be turned into action with little effort. Both of these team members however, found it difficult to gel with any ideas from the team member whose primary team-role was ‘Plant’. Person 1 (the Plant) was useful when the team

encountered problems and an answer was difficult to come by. As a Plant, the general contributions that were made to the team were problem solving, and an ability to generate ideas.

This role, according to Belbin (1981), has a flip side. The plant weakness tends to be a lower ability to communicate with other team members. This was yet another 'team role truth' that became evident within the development team. The last team member who was one of the 'main stay' team members was the team leader or Co-ordinator. The Co-ordinator role gave the team direction and became a decision-making role.

By bringing the above roles together the team had a figurehead, an ideas person, and people who were reliable, with this being seen as a good combination.

Although the team managed to produce good work and meet goals, it became more and more evident as time progressed that there should be other key roles. The roles that would have helped the team would have been:

- Shaper – Thriving on pressure and overcoming obstacles
- Resource Investigator – Explores opportunities and develops contacts
- Monitor Evaluator – Judges accurately

By being able to use the above team roles the development of the content could have been a smoother process. However, Belbin (1981) explains that for smaller teams the optimal size is six people. Belbin shows that a team of seven does not perform as well as a team of six. A team of six has enough skills to allow the team to perform. Belbin also makes mention that a team of four cannot cover all of the predicaments that that may be encountered. The development team at a minimum (but for a long period of time) was comprised of four persons. At a maximum it was a six-person team. The 'mainstay' that has been referred to was a team of four people, with the fifth position being filled by multiple people on a rotational basis. So although Belbin's team roles require a balance for a truly harmonious team, Belbin himself suggests that a small team is far more efficient with six members. This would perhaps mean that if the team grew to accommodate other roles, potentially productivity would drop.

3.2 Delegation to Achieve Optimal Performance in a Small Team

To optimise performance within the content development team, each team member became, to an extent, in charge of the ideas and content creation for each course. This gave each team member a degree of responsibility for his or her own subject area. Once the content was written, animations created, tutorials developed etc, a topic was then passed on to the team leader who corrected any errors or gave the go head for the editing process to begin. Whilst the content development team edited any textual errors, the same information was passed on to the people who were developing animations and tutorials, strict rules had to be adhered to. Each member of the content development team was in charge of checking the product as it moved

from software to content and back again. This again gave every team member responsibility and gave them 'buy-in' to their product. The team knew that although they had a responsibility for the development, they were in no way responsible for the content checking. By doing this it allowed the team to be creative, while having no risk associated with their ideas. The 'pseudo' responsibility produced high productivity, as they began to thrive on completing tasks.

3.3 Estimation

When the project started there was an air of apprehension as both parties (content development and software development) had not been in a situation this large and this unknown before. For the most part any timelines that were required were complete estimates. The enormity of the content development project did not become apparent until approximately four months after starting development. When the original timeline was drawn up, it was an estimated four to six weeks from beginning to end. As time progressed, and an idea of what exactly would be required became visible. Time and resource estimates became increasingly more reliable. The final time taken to develop the project was nine months, as opposed to the original four to six week estimate. Within the ever-increasing timeframe, team members became unsettled as to whether their jobs would progress. The affect this had was an ever-present worry for some of the team, and in turn influenced and lowered production times.

3.4 Imposition of Additional Requirements

The content was originally written so that the student would be able to learn Microsoft® Office XP® products. It was soon evident that many students would use either Microsoft® Office 2003® or perhaps have Microsoft® Office 2000® available to them. After developing for a period of three months, it was decided that development should not only encompass Microsoft® XP, 2000, and 2003 Office products, but Office Mac as well.

3.5 Initial Requirements Not Documented

As it was decided to encompass each version of Office products so 'no-one person was disadvantaged' it meant that the development team were to be contracted for further development, and the timeline was extremely underestimated.

If the original scope of the project had been documented and adhered to, development would have happened a lot faster. This would be due to every person involved in development knowing where the time was going and what exactly was required of them.

4 Application of Organisational Patterns

Coplien and Harrison (2005) describe a large number of organisational patterns that can be used to make decisions about the running of a software development team in general. For the purposes of this paper the following organisational patterns are briefly described and related to the organisation and management of the content development team:

- Patron
- Apprenticeship
- Architect Controls Product
- Developing in Pairs

4.1 Patron

Coplien and Harrison (2005, p 133) describe a scenario in which it is important for a project to have continuity, but where centralised control can be a problem, as can complete anarchy, and that as such it is important to have an ultimate decision maker. To solve this problem, the Patron pattern suggests that the project should be given a visible high-level manager who is the champion of the project, who also has the responsibility for removing project-level barriers that hinder process.

In the project that is the subject of this paper, there were times when decisions had to be made quickly, where there was not an obvious decision maker to champion the cause of the project, with a particular example being the need to extend the contracts of members of the content development team. As time went on with the project, a person in a high-level position was identified who both could make decisions and resource the decisions so as to provide continuity to the project.

In hindsight, it would appear that if the Patron pattern had been considered at the start of the project, then some of the decision-making that needed to be made would have been a smoother process.

4.2 Apprenticeship

Coplien and Harrison (2005, p 108) describe a scenario in which a project is gathering together members for the project team, and it becomes clear that there are not sufficient people available with the necessary expertise to staff the project, and that there are not enough people available for critical mass. To solve this problem, the Apprenticeship pattern suggests that new and less experienced members be brought into the team early and work beside a more experienced member. This will enable the project to maintain the expertise level that it requires.

In the project that is the subject of this paper new team members were introduced as the project went on and were, in general, able to build the level of expertise that was needed for the project. This was a case of being the only way in which the project could proceed, in retrospect it was an application of the Apprenticeship pattern, and proved to be one of the more successful aspects of the project.

4.3 Smoke-Filled Room

In describing the Smoke-Filled Room pattern Coplien and Harrison (2005, p 245) explain that there is a desire to have everyone involved in the decision making process that affects a project, but that a problem can arise when there can be circumstances where team members do not have the authority and/or experience to make key business decisions. The Smoke-Filled Room pattern suggests that in

this type of situation that the decision should be made amongst power brokers as in the “storied smoke-filled rooms stereotypically associated with tycoon businessmen”.

During the course of the project there were regular meetings of the project sponsors, with one of the main functions of these meetings being to make key decisions about the project

4.4 Stand-Up Meeting

Coplien and Harrison (2005, p 247) describe the Stand-Up Meeting pattern as being particularly useful in times of fast change or high stress, so that all members of the team receive the same information. This pattern suggests that in circumstances such as this that short daily meetings should be held with all of the team members to exchange critical information, update progress, and assign tasks, with these meetings taking no more than 15 minutes and happening early in the day.

In the project under review this pattern was not applied or present, with the main reason being that all of the team were located in the same workspace, which enabled communication within the team to be effective. However if team members had been geographically dispersed, following this pattern would have been beneficial.

4.5 Lock ‘em Up Together

The Lock ‘em Up Together pattern that Coplien and Harrison (2005, p 243) describe relates to a situation where a team of diverse people must come up with a single, coherent architecture. While this pattern relates specifically to software development, where many people working on architecture will produce components that do not necessarily work well together, with this being related to the idea that “designs by committee usually look in cohesive and inelegant. The pattern suggests that domain experts should be gathered together in the same room to work out the architecture (or other strategic issue).

In the project that is the subject of this paper, the application of this pattern would relate to the design of the overall look and feel of the content. There was a “Lock ‘em Up Together” meeting early on in the project where rules were set as to how things should be structured, with these rules being followed to the end of the project.

4.6 Community of Trust

The Community of Trust pattern that Coplien and Harrison (2005, p 34) outline is based on the idea that it is vitally important that the people in the team trust each other, otherwise it will be difficult to get anything done. The pattern suggests that there should be activities or processes that explicitly demonstrate trust, for example leaders or managers making it obvious that they are facilitating the achievement of goals rather than being in a controlling role.

In the project under review, there was a high degree of trust between the members of the development team that enabled the team to work together and focus on achieving the common goal. At times of stress during the project, this

was one of the aspects that ensured that the project continued to make progress.

4.7 Engage Quality Assurance

Coplien and Harrison (2005, p 168) describe the Engage Quality Assurance pattern and explain how a good dose of customer reality helps the perspective that development of perfect software impossible, with the pattern suggesting that quality assurance should be a central role and that it should be tightly coupled with development. Applying this pattern to the project under review suggests that the content being developed should have been tested with people fitting the persona of typical students in the CIC programme.

It was only in the later stages of development that people fitting the persona of typical students in the CIC programme tested the content. One of the main reasons for this was the nature of the platform that the content was being developed on was such that a typical CIC student would have found difficulty using. It is however recognised that it would have been very useful to have testing of this nature done much earlier in the development process, however, this would have required the deployment platform to be available at this stage

5 Analysis and Discussion

Many of the things that were learnt during the content development project was the importance of communication between various stakeholders and sponsors of the project. The content development team found that developing an e-Learning platform and content is vastly under rated. The most important lesson that was learnt by the development team was: when creating any form of distance learning material, NEVER take anything for granted, as to do so the students may miss out on a unique way to learn.

Belbin Team Roles would have been useful at the inception of the project because of:

- A team of optimal size may well have been more efficient, than one that needed staffing which could only be filled by workers on a rotational basis. This would have meant that the team could form at an early stage, and the dynamics of the team would not have been changing to accommodate newer members.
- It would have been practically useful to have at least a person with a resource investigator role in the team who could develop contacts and explore opportunities with ease.
- Having a team with appropriate Belbin Team Roles would mean that the existing team would not have been required to take on other roles to the extent that they did. Much effort was put into generating important team role types that were not natural to the team.

Some of organisational patterns confirm that aspects of the process were organised in an appropriate way. In particular the following patterns existed within the project and were of significant benefit to the overall project:

- Apprenticeship
- Smoke Filled Room
- Lock 'em Up Together
- Community of Trust

The Patron pattern was not in evidence early in the project, but was in evidence in the later stages of the project as described above and had a direct and positive impact on the smooth functioning of the the project.

While there were technical reasons for the Engage Quality Assurance pattern not being applied in the project, it would have been extremely useful for the project if it could have been applied. This suggests the importance of being able to provide the deployment environment for any project early on in a project so that this pattern can be applied.

The Stand Up Meeting pattern was not applied during the project due to the members of the team being located in the same workspace. However, it would have been very useful if the members of the team had been geographically dispersed.

This analysis of how these organisational patterns could have been applied to the project indicates that a formal application of them at the initial planning stages of the project would have resulted in some aspects of the project proceeding in the same way that they did, with others suggesting alternative approaches that could be used if the circumstances surrounding them not being used had been different.

The overall success of this project will not be able to be determined until sufficient students have completed the courses so that the completion rates and success rates can be compared with those students who have completed the same courses in face-face mode.

6 Conclusion

By using the Belbin Team Roles the content development team would have been able to take the multidisciplinary skills as dictated by the development process, and combine the many strengths that lay within the team. This would then have allowed the team to perform within an optimal environment.

Documentation was not readily produced, this impacted in two areas. The first being that additional requirements could easily be requested even though they had not been agreed to prior to development. The second impacted area was estimating the duration of tasks was virtually impossible. As the team had never been given the possibility to create or maintain a steady environment due to this ever changing discipline, an approach had to be taken that was both frustrating and time consuming. If the Team Role theory had been applied, each person's strengths could have been used to generate the most favourable outcome.

The use of organisational patterns would have made a difference to the structure of the development. If a Patron pattern had been in place it would have had a positive impact on the smooth functioning of he team. The Engage Quality Assurance pattern would also have been extremely useful if applied. As testing at an early stage on a group of

students that fit the CIC persona would have given early feedback for development purposes.

The application of both Belbin Team Roles and organisational patterns to a small team developing eLearning content can make a significant difference to its overall functioning and smooth running and both should be considered in the early planning stages of such a project.

Further areas to be looked at in direct relation to this project include the comparison of results for these students with students who complete the same courses in other modes.

7 References

- Belbin, R.M. (2004) Belbin Team Roles Retrieved 04 March 2006 from <http://www.belbin.com/belbin-team-roles.htm>
- Belbin, R.M. (1981) Management Teams – Why They Succeed or Fail. Oxford. Butterworth-Heinemann
- Coplien, J. and Harrison, B. (2005) Organisational Patterns of Agile Software Development. New Jersey: Prentice Hall
- McKenna, P. and Laycock, B. (2004) Constructivist or instructivist: pedagogical concepts practically applied to a computer learning environment. Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education ITiCSE '04
- Vrasidas, C. (2004) Engineering e-learning systems (ELS): Issues of pedagogy and design in e-learning systems. 2004 ACM Symposium on Applied Computing

Table 1 - Belbin Team-Roles (Belbin, 2004).

Team-Role Type	Contributions	Allowable Weaknesses
Plant	Creative, imaginative, unorthodox. Solves difficult problems.	Ignores incidentals. Too pre-occupied to communicate effectively.
Co-Ordinator	Mature, confident, a good chairperson. Clarifies goals, promotes decision-making, delegates well.	Can often be seen as manipulative. Off loads personal work.
Monitor Evaluator	Sober, strategic and discerning. Sees all options. Judges accurately.	Lacks drive and ability to inspire others.
Implementer	Disciplined, reliable, conservative and efficient. Turns ideas into practical actions.	Somewhat inflexible. Slow to respond to new possibilities.
Completer Finisher	Painstaking, conscientious, anxious. Searches out errors and omissions. Delivers on time.	Inclined to worry unduly. Reluctant to delegate.
Resource Investigator	Extrovert, enthusiastic, communicative. Explores opportunities. Develops contacts.	Over-optimistic. Loses interest once initial enthusiasm has passed.
Shaper	Challenging, dynamic, thrives on pressure. The drive and courage to overcome obstacles.	Prone to provocation. Offends people's feelings.
Team worker	Co-operative, mild, perceptive and diplomatic. Listens, builds, averts friction.	Indecisive in crunch situations.
Specialist	Single-minded, self-starting, dedicated. Provides knowledge and skills in rare supply.	Contributes only on a narrow front. Dwells on technicalities.