**Professional experience**

LU6: Teamworking and Project Management

How theoretical ideas of **project management** & **business organisation** integrate into **my team**, and how **control** and **monitoring** of work-based projects are undertaken.

David Norton

01/04/2014

Contents

[Introduction 1](#_Toc381216178)

[Origins & Theory of Team-Working 2](#_Toc381216179)

[IBM Management & Teamwork Structure 7](#_Toc381216180)

[Teamwork in Undertaken Projects 9](#_Toc381216181)

[IBM’s Team-Working Compared 13](#_Toc381216182)

[Conclusion - IBM’s Workplace Compared to Theory 14](#_Toc381216183)

[Study Plan Calendar 15](#_Toc381216184)

[Reflective Logbook 16](#_Toc381216185)

[References 18](#_Toc381216186)

# Introduction

## AIM

My aim for this Learning Unit is to gain theoretical knowledge of how teams work and compare this to what I have encountered in the workplace. I will also contact other students and see how teamwork compares and differs to IBM.

## Objectives

1. The origins and theory of good team working.
2. Identify the management structure and style, as well as the roles of project team members.
3. Review projects that I have been involved in, and how they helped me achieve my goals.
4. COMPARATIVE ELEMENT: IBM’s Team-working compared to other companies.
5. Evaluate how the workplace runs compared to that of theory.

# Origins & Theory of Team-Working

The origins and theory of good team working.

## projects & the need for teams

Teamwork is defined as "a joint action by a group of people, in which each person subordinates his or her individual interests and opinions to the unity and efficiency of the group."[[1]](#footnote-2) Each individual is still important, but effective and efficient teamwork goes beyond individual accomplishments.

2

[[2]](#footnote-3)Companies use teams because:

* **Better Outcomes**: Teamwork can lead to better outcomes as the team can bring more to a challenge and there is more oversight to reduce risk of poor individual contributions.
* **Efficiency**: Teams are able to accomplish more than its individual members can alone.
* **Better Ideas**: A good team is made up of diverse members. When different skills are applied to the same problem, a more effective solution is often created.
* **Mutual support**: When teams work well together, they are supportive of one another. Mutual support can encourage people to achieve goals they may not have realized alone.
* **Sense of Accomplishment**: When working to achieve specific goals, there is often a greater sense of accomplishment when in a group than when working alone. [[3]](#footnote-4)

The most effective teamwork is produced when all the individuals involved equally contribute and work towards a common goal.

## teamworking and theories behind it

A key theory of teamwork is the Belbin Team Roles theory. [[4]](#footnote-5) They state nine key roles to identify people's behavioral strengths and weaknesses in the workplace. This information can be used to:

* Build productive working relationships
* Select and develop high-performing teams
* Raise self-awareness and personal effectiveness
* Build mutual trust and understanding
* Aid recruitment processes

The nine roles are as follows 4:

**People**

* **Resource Investigator**: Extroverted, enthusiastic, and communicative. Explores opportunities and develops contacts who can help the project.
* **Team Worker**: Cooperative, mild, perceptive and diplomatic. Listens, builds, averts friction and calms the waters.
* **Coordinator**: Mature, confident and a good team leader. Clarifies goals, promotes decision-making and delegates well.

**Action/Task**

* **Shaper**: Challenging, dynamic, thrives on pressure. Drive and courage to overcome obstacles.
* **Implementer**: Disciplined, reliable, conservative and efficient. Turns ideas into practical actions.
* **Completer-Finisher**: Painstaking, conscientious. Seeks errors and omissions. Delivers on time.

**Thinking**

* **Plant**: Creative, imaginative, unorthodox. Able to solve difficult problems.
* **Specialist**: Single-minded, self-starting, dedicated. Provides knowledge and skills in rare supply.
* **Monitor-Evaluator**: Sober, strategic, discerning. Sees and evaluates options. Judges accurately.

4

Below are the key stages of Group Development, as defined by Bruce Tuckman[[5]](#footnote-6):

|  |  |  |
| --- | --- | --- |
|  | **Group Structure** *The pattern of interpersonal relationships; the way members act and relate to one another.* | **Task Activity** *The content of interaction as related to the task at hand.* |
| Forming:  *Orientation, testing & dependence.* | Testing and dependence. | Orientation to the task. |
| Storming:  *Resistance to group influence & task requirements.* | Intragroup conflict. | Emotional response to task demands. |
| Norming:  *Openness to other group members.* | In-group feeling and cohesiveness develop; new standards evolve and new roles are adopted. | Open exchange of relevant interpretations; intimate, personal opinions are expressed. |
| Performing:  *Constructive action.* | Roles become flexible and functional; structural issues have been resolved; structure can support task performance. | Interpersonal structure becomes the tool of task activities; group energy is channelled into the task; solutions can emerge. |
| Adjourning:  *Disengagement.* | Anxiety about separation and termination; sadness; feelings toward leader and group members. | Self-evaluation. |

[[6]](#footnote-7)

I believe this is a good summary, but in the real world, groups are always reforming and changing. Each time that happens, they can move to a different Tuckman Stage, so the process is not linear.

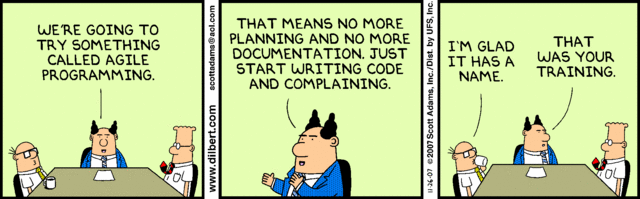
A group might be at the performing stage, but a new member might force them back into the storming stage. Team leaders should be able to deal with this and get the group back to performing as quickly as possible.

There are very few arguments against teamwork, but the few I was able to find include[[7]](#footnote-8):

* Misdirected work efforts; more time is spent on developing the team than creating the product.
* Loss of individual contributions and creativity.
* Lack of accountability and control
* Group judgment tends to be more extreme than the judgments of individuals
* Certain individuals will always work better alone than in teams.

## project management theory

### Agile Development



8

Agile is an approach to project management used in software development. It helps teams respond to the unpredictability of constructing software. It uses incremental, iterative work sequences, known as sprints[[8]](#footnote-9).

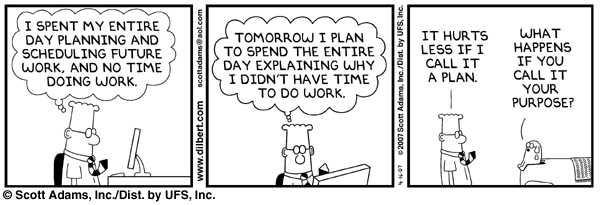
A sprint is a set period of time allocated for a particular phase of a project. When a sprint is complete development is either integrated, extended, or discarded and remaining iterations continue with their respective time frames and set work.

Agile ensures that value is optimized throughout the development process. The use of iterative planning and feedback results in teams producing working software frequently, which reflects the desired needs of a client. It easily adapts to changing requirements by measuring and evaluating the status of a project. The measuring and evaluating allows accurate and early visibility into the progress of each project.

Agile helps companies build the right product. Instead of trying to market software before it is written, Agile helps teams optimize the release during development. It is easier to produce software that only meets key customer demands, than it is to develop more general purpose software with the same functionality.

However, projects must be based around people who are motivated. With the proper environment and support they should be trusted to be self-organized and get their work done. Face-to-face communication is key to share info and at regular intervals, the team should reflect on how to become more efficient.

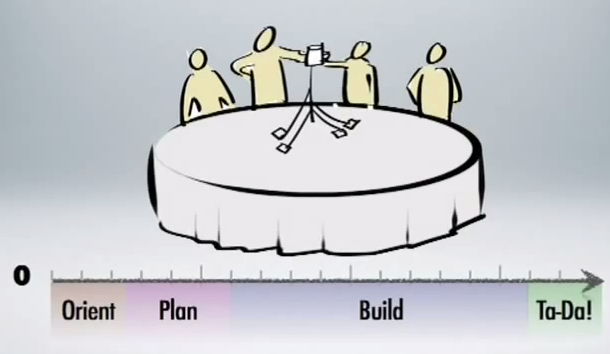
IBM is known for using Agile, and believe its use means that significant organizational changes can take place. They believe that many Agile software development teams will increase their chances of success by partnering with a trusted IBM employee to help them implement the Agile method.



8

### The Marshmallow Challenge[[9]](#footnote-10)

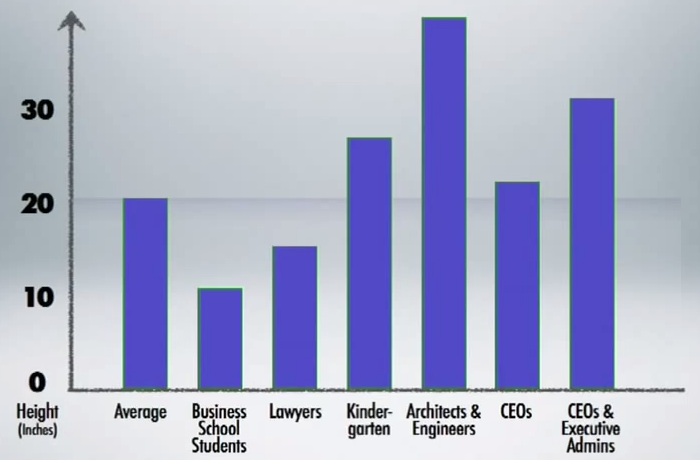
There is a great teambuilding challenge called the Marshmallow challenge. Team members have to create the tallest tower using spaghetti, tape, string and a marshmallow.

More or less every group has the same basic tactic to tackle the challenge. First the team must get to know each other, then they can discuss how they are going to attempt the challenge.

9

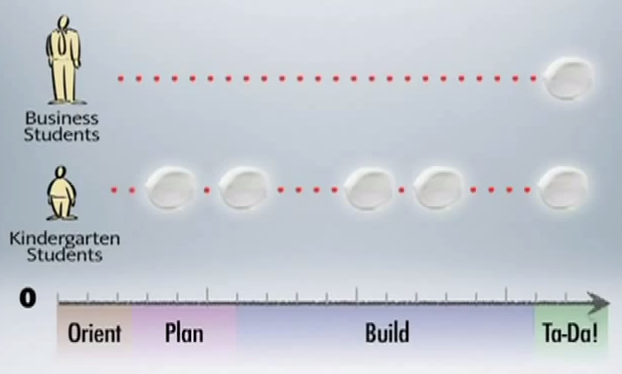
With a plan decided on, the team spend the time building the structure until the end, where they have created the finished result.

Although most teams followed this structure, it was found that certain groups tended to do better, and other groups would do worse.

It was found that business students and layers would do worse that average, normally because they would spend too much time on the orient and planning stages, with too many strong personalities to be able to reach an agreement to actually begin work.

9

Architects and engineers did the best as expected, as they knew and agreed on the standard for the strongest structures (triangles). CEOs did slightly better than average due to their creativity and ability to lead a projects design.



9

However, when Executive admins were added to a group of CEO’s they performed much better, and it was the same reason why kindergartens did so well. Both use the method of trial and error, with admins also able to assign roles and coordinate the team.

So rather than reaching the end with a collapsed tower, the team learns what works and what doesn’t along the way, leading to a better structure.

# IBM Management & Teamwork Structure

Identify the management structure and style, as well as the roles of project team members.

## Management structure

Within my office the main program development is done within the Tivoli department. It comprises of several different products, including OMNIbus, OMNIbus Web GUI, ITNM and Impact. All the products are designed to work together to provide a solution for Event Management.

Each product has a group of around 20 developers, with a task manager to lead the team on its product and feed in further required work from management.

The task managers have their own lead manager for the office. There are then several layers of further management for Cloud & Smarter Infrastructure, the wider group that Tivoli falls under

10

As well as my task manager, I also have a Personal Development Manager, who falls under a completely independent structure. My development manager focuses on helping with personal and work related issues, which could be hard to discuss with my task manager.[[10]](#footnote-11)

## Team Member Roles

There are several different roles in my team. The main developers work on the current product and develop the next update in the line. They also rotate onto L3, the third level of customer support.

L1 is a general support line, L2 is a support team who are familiar with the product, but are not familiar with the code. L3 is only contacted on code specific tasks, often when a company has coded their own feature for the product and needs help to attach it correctly.

There is also a User Experience (UX) Designer spread between the teams, to ensure newly developed code looks consistent with IBM’s guidelines. This person also works as a usability tester, going through usability tests with clients and internal to find potential issues and possible improvements.

Finally, there is one member of the team who works on accessibility, and a small Quality Assurance (QA) team who test the software for bugs or other possible issues. I am currently working as a developer and will be moving on to QA during my time with IBM.

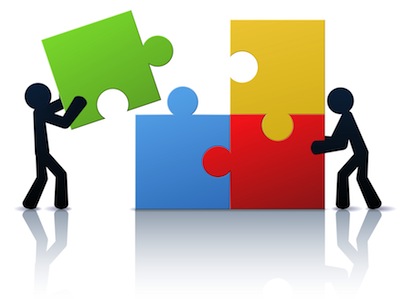
## Team Effectiveness

I think the general concept of our team is relatively solid. Team members rotate between different roles, allowing them to understand all areas, so they are able to rotate into any position when required. It also makes the job more interesting in my opinion, rather than always working on the same thing.

Team members still focus their expertise in certain areas, so there’s always an expert to ask when faced with a problem in a certain area of code. However, I do feel there is a lack of UX designers and usability testers in the team. This means that the software may be stable, but may not be as well suited to the customer as it could be.

I have also noticed a large separation between teams. As all teams are working on products that work together, there seems to be very little communication between the two, generally resulting in the same work being developed in different teams wastefully and problems arising when work needs to be merged together.

The managers do try to get team members to communicate, and set up regular meetings, but team members seem reluctant. This could well be due to the majority of team members working at home, making teamwork and forming relationships more difficult compared to working in the same office.[[11]](#footnote-12)



11

# Teamwork in Undertaken Projects

Review projects that I have been involved in and how they helped me achieve my goals.

## Team Projects involved in

The first project I worked solo on was creating an IBM Support Assistant Data Collector, reporting to my mentor with my progress. The software effectively asks the users a series of questions what problem they are having with the software, then will use this data to collect the correct files from their installation and email it to customer support, saving the user having to do it manually.

My second project was adding icons into the main interface of OMNIbus, a new feature requested by a large number of customers. Again, I worked on this alone, reporting to my mentor. However, I did talk to other team members when I required help and had a couple of stakeholder review meetings to demo my work to get feedback from my team.

In my latest project I worked on fixing all accessibility issues within the main project in a team of three. This is the project I will expand upon below.

## How Project was Planned & Monitored[[12]](#footnote-13)

12

First our team lead discussed the task we would have to achieve and the deadline we had to complete it in. We all then did some research into the kinds of accessibility defects that existed and discussed our findings as a team.

We then split out the sections of the product between us and used the required tools to test the software for defects and recorded them in a private blog. With this done, we could size each defect and split it between us equally so as to all finish by the deadline set.

Each defect was created in RTC/Clearcase so our progress could be monitored throughout the three week iteration. We also monitored our progress via daily scrums, to ensure none of us had any blocking issues.

## 

13

## [[13]](#footnote-14)Team Roles

Using the Belbin team roles theory discussed above, my team leader took on the Resource Investigator and Coordinator roles, assigning us new work when we completed what we were working on and investigating issues we raised so we could continue with development.

The other team member and I worked as Implementer’s and Team Worker’s, completing the work required each day and logging our notes ready for our code to be reviewed before integrating it. I don’t believe any other types of team members were required for our task, as there is little imagination required when fixing accessibility bugs, as there is a standard documented for how to fix them. [[14]](#footnote-15)

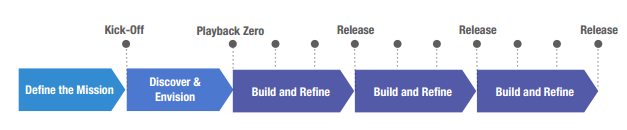
14

The only issue was although having daily scrums, our team leader worked at home for most of the three weeks and fixed far less issues, I believe due to lack of communication, as we frequently discussed how to fix different issues in the office.

## How team-working theories were realised[[15]](#footnote-16)

15

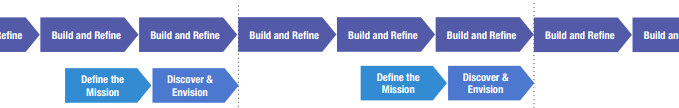
I believe we followed the Tuckman stages reasonably closely. We ‘**Formed**’, discussing the best way to accomplish the required work and running initial tests. We then moved on to ‘**Storming**’, regarding an issue about whether or not it was our responsibility to fix framework issues, then onto the ‘**Norming**’ phase after solving the issue by contacting the main IBM accessibility team. Lastly, by the end we all reached the ‘**Performing**’ stage to complete the task by the deadline by working together.

Below is a step by step diagram of how each project at IBM is supposed to follow:

15

15

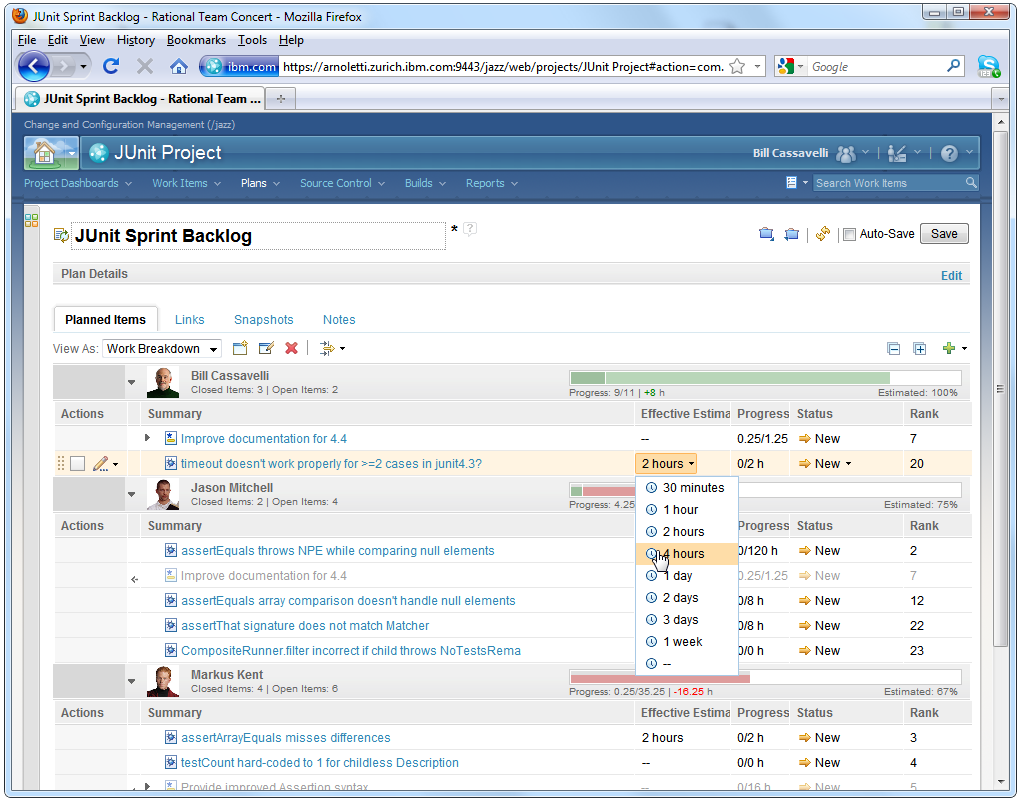
Our project was too small to use this diagram from start to finish, but it has largely been used within the overall product. Our project would fall within a ‘build and refine’ stage, as the product is improved before the next release coming in the next few months. To make this more efficient, the below is used:

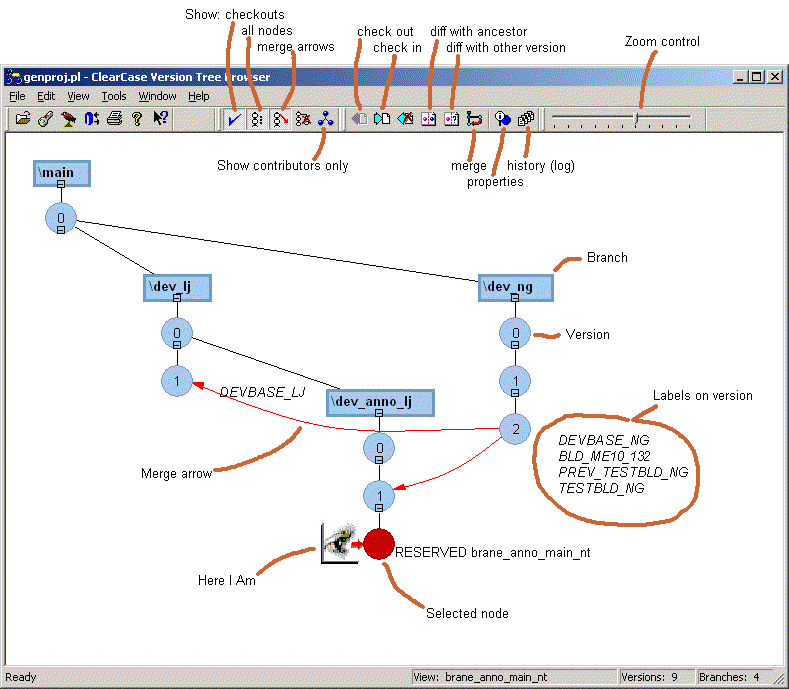
This small change means that the first two stages can be discussed between management and key members of the team, without effecting development time of other team members.

15

## Tools, techniques and methods of project management involved

During the process we used Rational Team Concert linked with Clearcase[[16]](#footnote-17):



RTC is a collaborative project environment providing source control, work item tracking, build management and reporting capabilities, as well as supporting agile planning. As you can see above users create work items for each iteration, with time estimate, rank, priority etc. If this information is kept updated, it is easy to see if someone is overloaded or who has time to complete any new critical items.

Clearcase is used for version control and parallel development, allowing new code to enter the integration team. When editing code, you must check the file out under an RTC activity, then check it back in when complete. Someone else must then code review your work before you are able to integrate it. At any point a role back can be made on an activity if it is found to crash the latest build.[[17]](#footnote-18)

# IBM’s Team-Working Compared

IBM’s Team-working compared to other companies.

### http://t0.gstatic.com/images?q=tbn:ANd9GcTCZebP1uDQbi8SsYoQYeV7ilkaj-AQZxd0FB2opJqDYBDuNvg5jgNokia Solutions and Networks – Matt Finch

### Description

### Matt works for Nokia, on a development team of a large product called One-NDS (Network Directory Server). It is a subscriber database for mobile networks - serving around 1.5bn subscribers worldwide!

### Similarities

They run an agile development pattern, with small teams working on the product. Each team has a scrum master who plans work for each sprint and monitors progress throughout the sprint. This sounds similar to IBM, with task managers and scrum masters planning work for the next iteration.

They use Engineering Release Manager which is a tool used to keep track of each team’s scrumboard and backlog, as well as automating the build process, which looks similar to IBM’s RTC.

They also have a line manager who monitors progress and performance, which sounds similar to our personal development manager, but slightly more focused towards work progress instead of personal development.

### Contrasts

His main role is support, rather than development, for the systems used at his site. This is all done externally at IBM at Hursley and the US.

They have product management teams are customer facing, acting as a layer of protection between the developers and customers. They dictate what new features and bug ﬁxes make the next release. IBM has a large number of customer facing roles, but they never interact with developers and have no say about what new features go into the product. They are simply there to sell the product to customers. There is however one manager who will contact key customers directly when required to help with certain issues.

### Conclusion

As expected from a company using agile development, the basis of how software is developed seems relatively similar. The key difference seems to be that they are far more connected with their customers with product management teams, something that my team is only starting to get into with usability tests.

# Conclusion - IBM’s Workplace Compared to Theory

*Evaluate how the workplace runs compared to that of theory.*

Simularities

IBM is definitely hot on using Agile Development for project management. I believe it helps more code get developed than in any other way, and it helps get the key defects and features raised by customers implemented.

The use of small teams is also far more efficient than working individually, as although no one can work on the same thing, as it would cause clashes in code when trying to integrate code, the knowledge shared when working together is invaluable as it saves a large amounts of time on research and shared ideas create the most efficient solutions.

Differences

The lack of face-to-face communication is the main thing that surprises me. Often there is only one or two members of my team in the office. From my experience of working at home, you can at times be more efficient, but you often run into issues you need to discuss. In these situations it is easy to either submit the code anyway and hope for the best, resulting in buggy code, or put it off until you can talk to someone, resulting in late code.

I am also surprised that usability tests are only now being run. The key aspect of Agile is that the code being developed is what the customer wants. There is no better way of testing what you have developed is what they want than usability tests. Hopefully now that they have begun, customers will be happier with the end result.

Team Effectiveness

I think agile has helped drive my team forward to develop many new features not initially planned but requested by customers. However, I have also seen it be extremely pressuring and demoralizing to other teams who are assigned too many tasks in too short periods. Strangely these are the teams that are largely based in the office, rather than at home. Perhaps there is a needed balance?

**Word Count:** 2977

# Study Plan Calendar

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UWE-FET UFCFE6-15-3 Professional Experience Module Calendar 2013/14 for** *[David Norton]* | | | | | |
| **UWE wk no**. | **Module wk. no.** | **Wk. comm.** | **Study Period (SP)** | **Learning Unit (LU) to be tackled** | **Notes**  **Milestones** |
| 9 | 1 | 23-Sep | Registration leeway period | |  |
| 10 | 2 | 30-Sep |
| 11 | 3 | 07-Oct | **SP1**  **(3 weeks)** |  | Due to course delays and tutors not being assigned, I will spend this time planning the required work **Ends Thu 24 Oct** |
| 12 | 4 | 14-Oct |
| 13 | 5 | 21-Oct |
| 14 | 6 | 28-Oct | **SP2**  **(3 weeks)** | *LU1* | I will complete LU1 and have it pre-marked by my tutor to avoid major mistakes, as I know I have time free over this time. **Ends Thu 14 Nov** |
| 15 | 7 | 04-Nov |
| 16 | 8 | 11-Nov |
| 17 | 9 | 18-Nov | **SP3**  **(3 weeks)** |  | I will likely be too busy with work this SP, but I will plan for LU5 and make contact for collaborative work. **Ends Thu 05 Dec** |
| 18 | 10 | 25-Nov |
| 19 | 11 | 02-Dec |
| 20 | 12 | 09-Dec | **SP4**  **(3 term weeks + 3 optional vacation weeks)** | *LU5* | This SP gives twice the time of the others, so it gives me the best advantage to complete another LU. I have chosen LU5, as it gives me more time to get the collaborative work needed for LU6. **Ends Thu 16 Jan** |
| 21 | UWE vacation | 16-Dec |
| 22 | 23-Dec |
| 23 | 30-Dec |
| 24 | 13 | 06-Jan |
| 25 | 14 | 13-Jan |
| 26 | 15 | 20-Jan | **SP5**  **(3 weeks)** |  | I will keep this SP free for a midway review, to ensure I am keeping up with my targets and deadlines, and adjust my plans if work isn’t being completed as expected .E**nds Thu 06 Feb** |
| 27 | 16 | 27-Jan |
| 28 | 17 | 03-Feb |
| 39 | 18 | 10-Feb | **SP6**  **(3 weeks)** | *LU6* | I will have had plenty of time to get the collaborative work by this SP, so is a good time to complete SP7. **Ends Thu 27 Feb** |
| 30 | 19 | 18-Feb |
| 31 | 20 | 24-Feb |
| 32 | 21 | 03-Mar | **SP7**  **(3 weeks)** |  | I will leave this SP free to plan as I have a holiday booked in this time and don’t want conflictions. **Ends** **Thu 20 Mar** |
| 33 | 22 | 10-Mar |
| 34 | 23 | 17-Mar |
| 35 | 24 | 24-Mar | **SP8**  **(3 weeks)** | *LU9* | Iwill complete the final LU in this SP, as it gives me full advantage of the time made available in the year. **Final deadline Thu 10 Apr 2pm** |
| 36 | 25 | 31-Mar |
| 37 | 26 | 07-Apr |

# Reflective Logbook

|  |  |  |
| --- | --- | --- |
| **UWE-FET UFCFE6-15-3 Professional Experience 2013-14 Reflective Logbook template** | | |
| **PROLUGUE** | **Student name**: David Norton | **SP #:** SP6 |
| **Employer:** IBM | **Date ending:** 27th February |
| **Tutor name:** Stewart Green | **LU #:** LU6 |
| **Overview**: I have covered Teamworking and Project Management in this report. It is a subject discussed at length in the office between the ITs and graduates, as due to the large number of people working at home and people’s reluctance to communicate between teams, project plans often seem wasteful or a mess. The graduates have more of a reference for this, as they have worked at Hursley before where office attendance was not an issue and project management was run more efficiently. | |
| **PLAN** | My plan for this LU is to first research the theories on teamwork, with a focus on agile development as it is key to IBM. I can then discuss what I have experienced with IBM and apply these theories to that, as well additional IBM theories found on the IBM Design website.  I can then compare everything I’ve researched with another company as part of the collaborative element, then finally conclude with a broad comparison on how IBM’s teamwork compares to theory, with its advantages and disadvantages. | |
| **ACTION** | I was able to create my document template relatively quickly and created an XML file that let me reference websites far easier, saving time spent formatting that I did at length in previous LUs. I also included footnotes to make references easier to locate.  Initial research of team theories went ok, but after completing that section I took on additional projects at work and home, including an Ethical Hacking day that will be run in March, set to be the biggest labs event for over a year, as well as teaching programming at a primary school in Clapham.  After completing several side projects, I was able to return to the rest of the assignment with around a week before the deadline. I considered looking for an extension, but managed to find time each day after work and use some holiday time to complete the project to a standard I was happy with. | |
| **REVIEW** | I think looking back on my last LUs, this one’s layout is a lot cleaner and easier to modify for the final LU. I think I started strongly on the project, but the end may be weaker as I felt I was very constrained by the word count and was unable to expand on areas such as how agile works in IBM and how development is different in Southbank compared to Hursley and how different teams operate in different ways.  I found the theory work done by IBM was very interesting and would have liked to include more, but once again my word limit capped the input. It was also not easy to find up to date information, as a lot of the intranet is hard to search and find recent results. | |
| **REFLECTION** | Looking back I believe I could have cut down more on the general theory behind teams, and included more specific theory that is relevant to the teams I’ve worked in. I could have also possibly used an extension to give my work several run through, cutting down certain areas and moving certain topics around.  However, I am now relatively pleased with all the information I have included, with links to information I found very interesting (I plan to use the Marshmallow Challenge in the future!). I was also pleased to see another company used a similar project management technique to us, and we weren’t using a completely inefficient version of agile. | |
| **EPILOGUE** | Since the last LUs, I have become a much better team worker, as I’m now communicating more with my team and taking on more roles where teamwork is required. As mentioned I am running an Ethical Hacking event and assembled a team to help me. I have also organized helping at a local school teaching under 10s and helping lead a project moving our product to the cloud.  This LU has helped reinforce in my mind the need for communication with customers when developing software, as they are who will be using it. It has also made me consider the advantages of agile, as before I related strongly to the Dilbert pictures included.  Several times I’ve spent a day writing out my plan, then half my next day changing it to make it achievable, as I’ve wasted so much time planning. I can now see the advantages, being more flexible and client focused. | |

# References

Belbin. (C 2012-2014) *Belbin Team Roles* [Online]. Available from: <http://www.belbin.com/> [Accessed 11 February 2014].

Boundless. (2012) *Some Benefits of Teamwork* [Online]. Available from: <https://www.boundless.com/management/groups-teams-and-teamwork/defining-teams-and-teamwork/some-benefits-of-teamwork/> [Accessed 09 February 2014].

Christensen, B.C. (2003) *The "Four Stages Theory" of Project Management* [Online]. Available from: <http://www.simpleprojectmanagement.com/think/4stages.html> [Accessed 16 February 2014].

IBM. (2013) *Rational Team Concert* [Online]. Available from: <https://jazz.net/products/rational-team-concert/features/planning> [Accessed 26 February 2014].

Linch SEO. (2012) *The Agile Method Explained: Beginners Guide & Summary of Benefits* [Online]. Available from: <http://www.linchpinseo.com/the-agile-method> [Accessed 26 February 2014].

MI9. (2010) *Office and Teamwork* [Online]. Available from: <http://www.mi9.com/office-and-teamwork_81929.html> [Accessed 11 February 2014].

NDT. (C2001-2012) *Teamwork in the Classroom* [Online]. Available from: <http://www.ndt-ed.org/TeachingResources/ClassroomTips/Teamwork.htm> [Accessed 9 February 2014].

Organizational Development Portal. (2011) *The Need For Teams* [Online]. Available from: <http://www.odportal.com/teams/effective/chapter2.htm> [Accessed 11 February 2014].

Realising Ambition. (2013) *Team Effectiveness* [Online]. Available from: <http://www.realisingambition.com/team-effectiveness> [Accessed 23 February 2014].

Scoop. (2013) *PowerPoint Presentation Tools and Resources* [Online]. Available from: <http://www.scoop.it/t/powerpoint-presentation-tools-and-resources> [Accessed 16 February 2014].

SVN. (2011) *asf - Revision 1571946* [Online]. Available from: <http://svn.apache.org/repos/asf/subversion/developer-resources/guis/pics/> [Accessed 26 February 2014].

Tuckman, B. (1965) Special Issue on Group Development. *Psychological Bulletin*, 63(6), pp.384-99.

Youtube / TED. (2010) *Tom Wujec: Build a tower, build a team* [Online]. Available from: <http://www.youtube.com/watch?v=H0_yKBitO8M> [Accessed 21 February 2014].

1. <http://www.ndt-ed.org/TeachingResources/ClassroomTips/Teamwork.htm> (NDT, C2001-2012) [↑](#footnote-ref-2)
2. <http://www.mi9.com/office-and-teamwork_81929.html> (MI9, 2010) [↑](#footnote-ref-3)
3. <https://www.boundless.com/management/groups-teams-and-teamwork/defining-teams-and-teamwork/some-benefits-of-teamwork/> (Boundless, 2012) [↑](#footnote-ref-4)
4. <http://www.belbin.com/> (Belbin, C 2012-2014) [↑](#footnote-ref-5)
5. Tuckman - Developmental Sequence in Small Groups (Tuckman, 1965) [↑](#footnote-ref-6)
6. <http://www.scoop.it/t/powerpoint-presentation-tools-and-resources> (Scoop, 2013) [↑](#footnote-ref-7)
7. <http://www.odportal.com/teams/effective/chapter2.htm> (Organizational Development Portal, 2011) [↑](#footnote-ref-8)
8. <http://www.linchpinseo.com/the-agile-method> (Linch SEO, 2012) [↑](#footnote-ref-9)
9. <http://www.youtube.com/watch?v=H0_yKBitO8M> (Youtube / TED, 2010) [↑](#footnote-ref-10)
10. <https://design.ibm.com/thinking.html> [↑](#footnote-ref-11)
11. <http://www.realisingambition.com/team-effectiveness> (Realising Ambition, 2013) [↑](#footnote-ref-12)
12. <https://design.ibm.com/thinking.html> [↑](#footnote-ref-13)
13. https://design.ibm.com/thinking.html [↑](#footnote-ref-14)
14. <http://www.belbin.com/> (Belbin, C 2012-2014) [↑](#footnote-ref-15)
15. https://design.ibm.com/thinking.html [↑](#footnote-ref-16)
16. https://jazz.net/products/rational-team-concert/features/planning (IBM, 2013) [↑](#footnote-ref-17)
17. <http://svn.apache.org/repos/asf/subversion/developer-resources/guis/pics/> (SVN, 2011) [↑](#footnote-ref-18)