SIGMIS-CPR '10: Proceedings of the 2010 Special Interest Group on Management Information System's. 48th annual conference on Computer personnel research on Computer personnel research. New York: ACM, 2010. 43-47

**DOI:** http://dx.doi.org/10.1145/1796900.1796921

# Software Engineering Group Work – Personality, Patterns and Performance

David Bell
Brunel University
Uxbridge
UK

david.bell@brunel.ac.uk

Tracy Hall
Brunel University
Uxbridge
UK

tracy.Hall@brunel.ac.uk

Jo Erskine Hannay
University of Oslo
Oslo
Norway

johannay@simula.no

Dietmar Pfahl
University of Oslo
Oslo
Norway

dietmar.pfahl@gmail.com

Silvia Teresita Acuna
Universidad Autonoma de Madrid
Madrid
Spain

silvia.acuna@uam.es

## ABSTRACT
Software Engineering has been a fundamental part of many computing undergraduate courses for a number of years. Although many of the tools and techniques used to undertake software engineering have changed, the assessment has typically stayed the same. Students are commonly tasked with producing a number of software artefacts, for example designs using the Unified Modelling Language (UML). We recently attempted to extend the software engineering experience for a group of second year students with them participating in groups that attempt to replicate industrial practice. This paper reports our investigation into the correlation between the personality of group members, their approach with respect to using design patterns and their learning achievements.

## Categories and Subject Descriptors
D.2 [**Software Engineering**]: Management – *Programming Teams.*

## General Terms
Management, Design.

## Keywords
Software Engineering (SE), Personality, Design Patterns, Unified Modelling Language (UML)

## 1. INTRODUCTION
This paper presents an exploratory study into the correlations between the personality of software engineering group members, their approach with respect to design patterns and their learning achievements. Software engineering in this context is the process by which a specific business problem is solved using computer software – including requirements gathering, design, coding and testing. The study reported is based on the analysis of a coursework undertaken by 128 students (the majority UK nationals) on an undergraduate second year software engineering module. The work is motivated by Acuna et al's (2009) study of the relationship between personality, team processes, task characteristics, product quality and satisfaction. Our aim is to extend Acuna et al's (2009) findings by analysing UK data.

The aim of a typical undergraduate software engineering module is to equip students with the knowledge and skills necessary for the design and implementation of software systems using recognised methodologies, tools and technologies. Modules of this type provide an introduction to software engineering and will usually follow a development process from requirement and design ready for implementation. In many cases, this is the finishing point with the student creating a number of software artefacts, e.g. design diagrams, project plans, a range of program code and test scripts. However over fifteen years ago Schlimmer et al. (1994) highlighted the need to include group working that addresses real world problems.

In this paper we describe how we introduced group work into our 'typical' software engineering undergraduate module. Groups of students work together to assess technologies, generate designs and manage the allocation of more detailed design and implementation tasks. This complex mix of personalities, tools, techniques and ideas provides an insight into the socio-technical aspects of software engineering as students progress through their group working journey. An important aspect of software engineering group work is the ability of groups to form effectively and progress in a synergistic manner. The research described in this paper explores how personality affects the choice of tools and patterns (as part of a coursework design). More importantly, the study allows for analysis of the personality profiles across the cohort with respect to performance.

The paper starts by outlining commonly used personality indicators and design pattern definitions. The research method is then described. The personality and group surveys and resulting correlations are then presented and the paper concludes with a reflective summary of the work (with recommendation for future practice).

## 2. BACKGROUND

One of our aims is to extend the findings of Acuna et al,'s (2009) study. Our study is a smaller scale study analysing only a sub-set of the data that Acuna et al. (2009) analysed. For example we do not currently analyse team and task data, we concentrate on individual personality data and product quality. However, unlike Acuna et al. (2009) we also consider the use of design patterns.

### 2.1 Personality

Personality is typically classified using either Myers-Briggs type (Myers & McCaulley 1985) or NEO indicators (Costa Jr, McCrae 2008). Both approaches categorise personality based on answering a number of questions (60 in the case of shorter NEO-FFI and 126 in Myers-Briggs). The NEO-FFI survey has been chosen for this report primarily because it is used in Acuna et al,'s (2009) study. Adopting the same personality profiling mechanisms will allow for subsequent comparative analysis. Analysing the questionnaire responses results in an individual score against each of the categories presented in Table 1. Table 1 presents both personality traits demonstrated by high and low scorers and the associated boundary scores – upper boundary for low scoring and lower boundary for high scoring. The boundary values are provided in order to indicate when particular personality dimensions can be applied (using combined adult normative data taken from the NEO five factor profiling). For example, Neuroticism (N) scores of 23 and above can be classified as high scorers and thus may demonstrate some of the dimensions indicated. Scoring below 15 for N indicates a low scorer. College student data was not used in this preliminary analysis as it is the raw scores that are being analysed. Consequently, the boundary scores are used as sample indicators.

**Table 1. Combined Personality Dimensions (The Guardian, 2009) and NEO Scoring**

| **Dimension** | **High Scorers are… (High Boundary)** | **Low Scorers are… (Low Boundary)** |
|---|---|---|
| Neuroticism/Emotional Stability N | Prone to stress, worry and negative emotions (23) | Emotionally stable, but can take unnecessary risks. (15) |
| Extroversion E | Outgoing, enthusiastic and active; you seek novelty and excitement (31) | Aloof, quite independent; you are cautious and enjoy time alone (25) |
| Openness (To New Experience) O | Creative, imaginative, eccentric and open to new experiences (38) | Practical, conventional, sceptical and rational (31) |
| Agreeableness A | Trusting, empathetic and compliant, you are slow to anger (36) | Uncooperative and hostile, you find it hard to empathise with others (30) |
| Conscientious -ness C | Organised, self directed and successful, but controlling (30) | Spontaneous, careless; can be prone to addiction (24) |

### 2.2 Group Work and Personality

Personality analysis of software engineering teams has been undertaken for a number of years. Preslak (2006) investigated personality relationships to team processes (such as effort, role, leadership, conflict amongst others) and group outcome. Personality was shown to improve team atmosphere, group cohesion, team roles and communication to handle conflict (as part of team performance). Variation in the personality of members has also been investigated; contrasting outcomes highlight it as a cause for conflict (Pihulyk 2003) and team optimisation (Rutherfoord 2001). Although outside of the aims of this research, the work opens up questions of team selection based on personality (highlighted already by (Rutherfoord 2001)) and analysis of the group as whole.

Research results presented in literature are somewhat conflicting. Preslak (2006) found no correlation between personality and team processes. In contrast, Acuna et al. (2009) who investigated task conflict, personal conflict and cohesion and their relationship to personality factors found a strong correlation between the personality characteristic extraversion and software product quality (and between job satisfaction and personality factors agreeableness and conscientiousness). A recent systematic literature review of software engineer motivation (Beecham et al. 2007) provides comprehensive coverage of the area and highlights that little is really known about software engineering motivation (characteristics and benefits) - with domain changes rapidly making research obsolete. The following hypothesis emerges from the group work literature:

<u>Hypothesis 1</u>: Personality characteristics have a significant impact on individual performance within a team environment.

### 2.3 Patterns

Patterns (often called software or design patterns) are often defined as general solutions to software problems. Patterns originated in work by Alexander, in town planning (Alexander et al. 1977), and made a successful transition to computing (largely with the publication of the "Gang of four" book (Gamma et al. 1995)). A number of design patterns were covered in both lectures and labs – more specifically, the data oriented patterns called Table-Data-Gateway and Row-data-gateway (Fowler 2003). These patterns separate business and application logic from database access programming code. The student may choose to use patterns in their individual design work (if they wish to).

Limited current literature on pattern adoption and group work exists. Consequently, the following speculative hypothesis is made:

Hypothesis 2: Using design patterns improves individual performance when working within a team environment.

# 3. METHODOLOGY

Group working is part of the curriculum for a second year software engineering module with a cohort of around 180 (the majority UK nationals). The module interweaves theory coverage with practical labs and concludes with a written report as summative assessment. The coursework requires the students to participate, as part of a group, in the design and implementation of a small software system. The software engineering project attempts to mimic, where possible, a commercial project. A narrative business problem is described within a short case study and used as the requirement for the design and implementation of a system using UML and Java. The coursework requires that students fully participate in the group (typically of 5 people) – and included in this is the design and implementation of individual parts of the final software system by each group member. The groups are formed centrally by the module leader (as opposed to self-selection by students). This approach aims to extend the student's social experience – moving out of their smaller social group.

Over the course of two terms the students are required to: (a) Allocate tasks amongst the group and hold team meetings, (b) evaluate and present technologies for use by the group (presenting a Technology Assessment to the group and receiving a feedback form from the group members, (c) produce a draft UML design as a group, (d) design specific parts of the overall system and (e) develop Java software that fulfils their design. Throughout the process the group will meet to discuss and plan their work.

The summative assessment is in the form of a written report and includes both design/coding work and reflection on both technical working and technical experience. The report is assessed against the criteria summarised in Table 2. The marking scheme evaluated team working, design and programming artefacts. Team working is demonstrated in both the web meeting place communication (e.g. meeting minutes) and the student's personal team reflections and analysis documented in their report. UML design work and Java coding is also presented in the student report. Each student produces an individual report that is marked independently of other group members. Consequently, each team member can (and often did) receive different assessment marks. The interweaving of group and individual working is achieved through the group's decomposition of their system into parts for individual design and implementation work (following group design work that had an emphasis on requirement modelling using Use-cases). Individual tasks followed typical commercial allocations around user interface, business logic and database access design and coding. In summary, the achievement is measured through a single summative assessment – a report that brings together a number of UML diagrams (types being chosen by the student), snippets of Java code generated in part from the UML class diagrams (using the Netbeans development environment) and reflective assessment of the group and the software engineering process.


*ACM SIGMIS-CPR'10*, May 20–22, 2010, Vancouver, British Columbia., Canada.


**Table 2. Assessment Criteria**

| **Criteria** | **Explanation** |
|---|---|
| Team Working and professionalism | Software is typically large and complex; it is inevitably developed by project teams. While teams bring increased human resource, they also bring communication and collaboration problems. This criterion assesses your ability to work within a team and your interpretation skills. |
| Technology Assessment | All software projects use a number of software engineering tools to produce designs, software, test script etc. This criterion assesses your ability to carry out such an assessment in the context of application software development. The criteria support both learning outcomes as an understanding of the tools and techniques is then able to support you design and implementation. |
| Software Development | Strong software development skills are critical to producing good software. Such skills can only be developed through practice. This criterion assesses your ability to design and then implement software. |
| Written Communication | Many design and development issues and decisions will need to be communicated during a software project. Miscommunication leads to delays, errors and generally poor quality software. This criterion assesses your ability to communicate in an effective way, demonstrating both learning outcomes. |

## 3.1 Research Process

The personality survey was carried out early in the module, during the second lab session. The NEO-FFI personality survey was used (described earlier). Sixty pre-existing personality questions were used – with additional ethics related question added upfront. A number of other surveys were used – one of which analyses group aversion (this work is not included in this report). A university portal and associated tools were used to carry out the survey; the aim being to automate the processing of our survey results. Survey responses were analysed using weighted scoring for each question. Groups of question scores were then added together in order to calculate specific personality dimension scores.

The groups on the whole worked well once they overcame their initial worries about working with people they did not know, they started to enjoy working together (clearly highlighted in their final report reflections). Although all groups completed their assignment, a number of students did not engage (or were never seen by their group), leaving some groups feeling that they were under staffed. The results for the module were good – with an average pass mark of 62%.

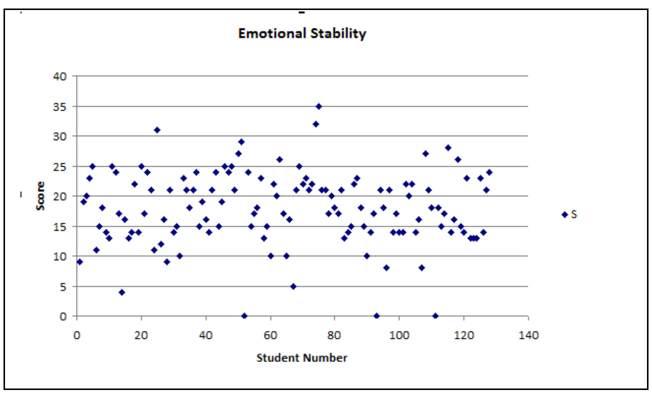
## 3.2 Study Limitations

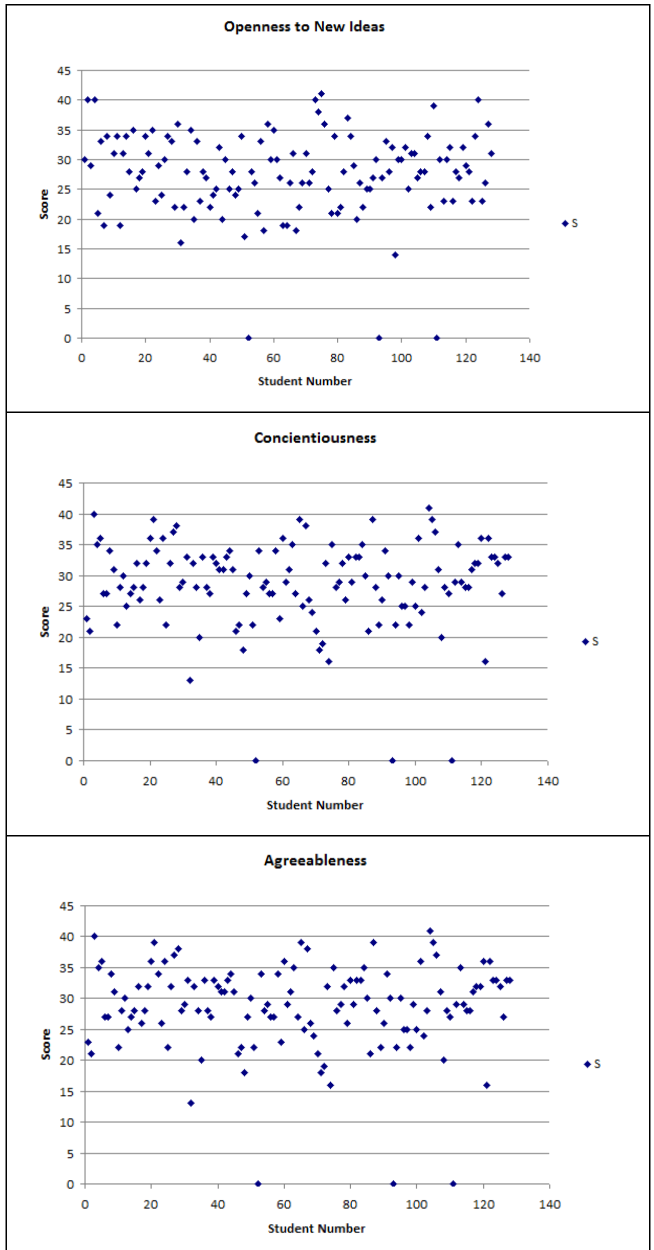
Empirical studies are notoriously difficult to conduct in software engineering. As a result we, along with every other study performed, must report a number of limitations and threats to the validity of our findings:

- Student participants. In this study we used only student developers. Consequently we cannot generalise our findings to professional software developers.
- We did not collect data describing the software development experience/ability of students before they participated in this study. Consequently we are not able to identify the impact of previous experience/ability of product quality.
- Currently we only report individual personality data. We do not report any analysis of group/team factors. We have collected this data but have not yet analysed it. Consequently we cannot comment on the impact of team composition on product quality.
-

# 4. RESULTS

The personality analysis resulting from survey data (taken from 128 respondents) is presented in Figure 1:




Arithmetic means (for each NEO characteristic) for the cohort being studied:

- Neuroticism/Emotional Stability N=18
- Extroversion E=30
- Openness (New Experience) O=28
- Agreeableness A=29
- Conscientiousness C=32

**Figure 1. Personality Analysis**

The raw individual student data presented above shows that the cohort are generally concentrated around the NEO average score (with the arithmetic mean within or just outside the average banding) – giving a level of credibility to the adult combined scoring system for a student cohort. Mean conscientiousness just makes the high category and mean openness and agreeableness just drop into the low category. Although a number of outliers exist, they have been left in as they are few in number.

The correlation between personality, performance and pattern use variables is presented in Table 3 – calculated using Pearson product-moment correlation coefficients (using Excel 2007) - after each of the variable pairs were investigated independently. The NEO dimensions were analyzed, along with achievement (Ach) and the use of patterns (Pat). The results show no strong correlations in the data (disproving both hypothesis 1 and 2). Some interesting weaker correlations warrant some further analysis (0.19 and -0.12) as they suggest a possible relationship between Emotional Stability (N) and Pattern Usage; and achievement and Openness to New Experience (O).

**Table 3: Correlation Results**

| | N | E | O | A | C | Ach | Pat |
|---|---|---|---|---|---|---|---|
| **N** | 1.00 | | | | | | |
| **E** | -0.44 | 1.00 | | | | | |
| **O** | 0.02 | 0.14 | 1.00 | | | | |
| **A** | -0.18 | 0.14 | 0.15 | 1.00 | | | |
| **C** | -0.32 | 0.23 | 0.20 | 0.20 | 1.00 | | |
| **Ach** | -0.03 | 0.07 | 0.19 | 0.03 | 0.06 | 1.00 | |
| **Pat** | -0.12 | 0.06 | -0.05 | 0.08 | -0.01 | -0.04 | 1.00 |

Further analysis of the data is presented in Figure 2. The pattern usage-N result is interesting suggesting that students with lower N scores are more likely to adopt design patterns. However, it is difficult to draw any usable conclusion from such a weak correlation. Consequently, further research is warranted into why this happens, investigating why the patterns were chosen and maybe using a more qualitative approach such as interview and grounded theory. The weak correlation between achievement and O indicates that more openness to new experience may improve performance. The students are experiencing groups, technologies and techniques for the first time and an added level of openness may support the complex environment they find themselves.

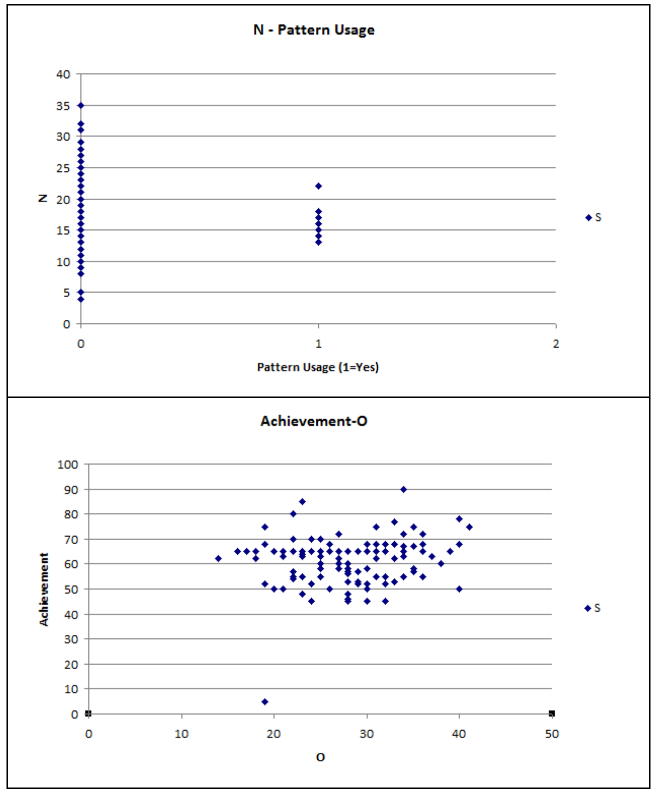



**Figure 2. Correlation Investigations**

## 5. CONCLUSION

This paper presents the results of replicating industrial design-programming team practice in a software engineering undergraduate module. Two hypotheses are tested: (H1) Personality characteristics have a significant impact on individual performance within a team environment and (H2) Using design patterns improves individual performance when working within a team environment.

No strong correlations were found to validate these hypotheses. This is in contrast to Acuna et al. (2009) reporting a significant relationship between personality and product quality. We did, however, find a weak correlation between Neuroticism and Pattern Use that warrants further research.

## 6. REFERENCES


[1] Acuña, S.T., Gómez, M. & Juristo, N. 2009, "How do personality, team processes and task characteristics relate to job satisfaction and software quality?", *Information and Software Technology,* vol. 51, no. 3, pp. 627-639.

[2] Alexander, C., Ishikawa, S., Silverstein, M. & Center for Environmental Structure 1977, *A pattern language : towns, buildings, construction,* Oxford University Press, New York.

[3] Beecham, S., Baddoo, N., Hall, T., Robinson, H. & Sharp, H. 2007, "Motivation in Software Engineering: A systematic literature review", *Information and Software Technology*.

[4] Costa Jr, P.T. & McCrae, R.R. 2008, "Revised NEO Personality Inventory (NEOPI-R)", *The Sage handbook of personality theory and assessment: Personality measurement and testing,* , pp. 179.

[5] Fowler, M. 2003, Patterns of enterprise application architecture, Addison-Wesley Professional.

[6] Gamma, E., Helm, R., Johnson, R. & Vlissides, J. 1995, *Design Patterns: Elements of reusable object-oriented software,* Addison Wesley.

[7] Kruchten, P. 2000, *The rational unified process: an introduction,* Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

[8] Myers I B ,& McCaulley M H 1985, Manual: A guide to the development and use of the Myers-Bnggs Type Indicator, Palo Alto Consulting Psychologists Press.

[9] Peslak, A.R. 2006, "The impact of personality on information technology team projects", Proceedings of the 2006 ACM SIGMIS CPR conference on computer personnel research: Forty four years of computer personnel research: achievements, challenges & the future, ACM New York, NY, USA, , pp. 273.

[10] Pihulyk, A. 2003, Understanding the personality mix: bringing out the best in you and your team. Canadian Manager, vol. 28, no. 1, pp. 12-13.

[11] Rutherfoord, R.H. 2001, "Using personality inventories to help form teams for software engineering class projects", *Proceedings of the 6th annual conference on Innovation and technology in computer science education,* ACM New York, NY, USA, pp. 73.

[12] Schlimmer, J., Fletcher, J. & Hermens, L. 1994, "Team-oriented software practicum", *IEEE Transactions on education,* vol. 37, no. 2, pp. 212-220