

**Full citation:** Licorish, S., Philpott, A., & MacDonell, S.G. (2009) Supporting agile team composition: a prototype tool for identifying personality (in)compatibilities, in Proceedings of the ICSE Workshop on Cooperative and Human Aspects of Software Engineering (CHASE). Vancouver BC, Canada, IEEE Computer Society Press, pp.66-73.  
<http://dx.doi.org/10.1109/CHASE.2009.5071413>

## **Supporting Agile Team Composition: A Prototype Tool for Identifying Personality (In)Compatibilities**

Sherlock Licorish, Anne Philpott, and Stephen G. MacDonell

*SERL, Auckland University of Technology  
Private Bag 92006, Auckland 1142, New Zealand  
{sherlock.licorish, anne.philpott, stephen.macdonell}@aut.ac.nz*

### **Abstract**

*Extensive work in the behavioral sciences tells us that team composition is a complex activity in many disciplines, given the variations inherent across individuals' personalities. The composition of teams to undertake software development is subject to this same complexity. Furthermore, the building of a team to undertake agile software development may be particularly challenging, given the inclusive yet fluid nature of teams in this context. We describe here the development and preliminary evaluation of a prototype tool intended to assist software engineers and project managers in forming agile teams, utilizing information concerning members' personalities as input to this process. Initial assessment of the tool's capabilities by agile development practitioners suggests that it would be of value in supporting the team composition activity in real projects.*

### **1. INTRODUCTION**

In any software provision model apart from that of the sole contractor, the scale and complexity of many software development activities mean that they are necessarily undertaken by teams rather than individuals. Interactions among the various stakeholders involved have the potential to make or break a project, a portfolio of work, and even perhaps the software organization itself. Incompatibilities among the members of the team in terms of their

personalities may compromise team effectiveness; similarly, a team made up of highly compatible individuals may function extremely well as a group. Therefore, efforts to consider the optimum mix of personality types in a software development team have the potential to bring benefits in terms of maximizing productivity and improving the likelihood that delivery will occur in a timely and cost-effective manner, to the satisfaction of all stakeholders.

In the context of agile software development the need to consider personality types in team formation may be even more pronounced. This is due to the fact that agile teams tend to explicitly include user and/or customer representatives as well as a project facilitator in addition to the core development personnel. Furthermore, team structures tend to be rather more fluid in agile projects, forming, growing, shrinking and disbanding as particular skill needs arise or are dealt with. The higher number and greater diversity of individuals involved, combined with the less rigid nature of their involvement, may increase the incidence of personnel incompatibilities and, therefore, the potential for conflict.

With the above in mind it would seem beneficial to provide project managers with a form of systematic support for assessing the personalities, strengths and weaknesses of the population of individuals from whom their teams are constructed. To that end we have designed and embedded into a project management tool a set of functions to enable the collection, assessment and reporting of the personality traits of intended team members.

In the next section of the paper we consider the impact of personalities on the risks that can arise in agile software projects. We then describe the development and early evaluation of our prototype tool. Finally we draw conclusions from our work to date and describe our ongoing research in this domain.

## 2. BACKGROUND RESEARCH AND RELATED WORK

In this section we briefly consider background research addressing the psychology of personalities and teamwork and then consider this in relation to the development of software systems.

### 2.1. Personality types

The discipline of psychology seeks in part to understand personality type and its influence on individuals' strengths and qualities, and their ability to communicate and form/sustain relationships in teams [1-3]. Additionally, some studies are concerned with the effect of the *mix* of personality types in groups, how personalities interact to influence team performance, and the impact of behavioral differences on teamwork; see, for example, the work of Belbin [4] and McCrae and Costa [5].

These ideas have manifested themselves in a variety of personality assessment instruments. Myers and McCaulley [3] used Jung's theories [1] to develop the well-known Myers-Briggs Type Indicator (MBTI) psychometric instrument, intended to measure and reflect an individual's basic preferences. Their studies suggest that individuals are expected to possess one of four preferences in their behavior. For a person's energetic preference: they are either extrovert or introvert; for what they perceive: sensing or intuition; for their decisions: thinking or feeling; and their lifestyle: judging or perceiving. Myers and McCaulley [3] describe extroverts as often favoring interacting with people while introverts are complete by being alone. Sensing individuals prefer evidence that is factual or concrete whereas intuitive individuals favor exploration and alternative explanation. Thinking individuals are rational and favor formal methods of reasoning whereas feeling individuals focus their judgment on subjective values and their views. Judgers like pre-planning and order whereas perceivers live through adoption and spontaneous decision making. Several studies have demonstrated that these preferences influence the ways in which individuals interact in groups, and the validity and reliability of the MBTI instrument have been verified in studies reported

by Carlson [6], Johnson [7] and McCarley and Carskadon [8].

Similarly, Keirsey [9] used the work of Carl Jung and Isabel Briggs Myers to develop a psychometric instrument to identify an individual's most dominant personality traits. Keirsey suggests that human personality belongs to one of sixteen variants (see [www.keirsey.com](http://www.keirsey.com) for more information). Like Myers and McCaulley [3], Keirsey asserts that an individual's dominant or preferred personality exists among one of four major types. However, there are additional levels within each of these types in Keirsey's approach which further classify individuals to one of sixteen variants.

### 2.2. Role theories and teamwork

An extension of the personality type research in the psychology literature is found in Belbin's work [4]. Belbin conducted many studies in several countries to ascertain the particular human behaviors or attitudes that are essential for team success. Following several years of observation, Belbin found that most successful teams possess eight different functional roles: company worker, chairman, shaper, plant, resource investigator, monitor-evaluator, team worker, and completer-finisher (see Table 1 for details).

Belbin asserts that in successful teams, the eight roles described in Table 1 are performed by the team members. Though Belbin's findings differ slightly from previous role theories which linked individuals to one personality trait, his work built on the MBTI ideas, and also agrees that a person's interactions in a group are influenced by their personality trait. Among his findings he also reported that individuals who possess premium quality in one respect are often lacking in others, and that combining individuals with similar personality traits reduces performance. Thus, successful teams tend to be heterogeneous; normally possessing a balance of team members occupying all roles. Individuals can possess more than one personality preference, having a primary preference and other secondary preferences. Individuals are most comfortable when they are functioning in roles that match their natural preference. Interaction between different personalities without understanding and managing their differences can be a source of conflict.

During his studies on team roles Belbin developed a psychometric test called the Self-perception Inventory (SPI). The SPI is a questionnaire used as an indicator for determining an individual's personality preference. The test consists of seven sections with eight questions for each section. For each section individuals allocate a total of ten points based on how they feel about the

questions. There exist several studies in the software development context that have considered the validity of Belbin's work; Beranek, Zuser and Grechenig [10], Jones [11] and Rajendran [12] have all lent confirmation to Belbin's findings.

### 2.3. Personalities and roles in SE

The consideration of personalities, team formation and group dynamics in relation to software development, some of it assessing the role theories just described, has received ongoing [10-12] and recent attention [13-16].

Gorla and Lam [16] used the MBTI model to study 92 software developers from 20 software companies in Hong Kong, to verify whether personality type is linked to team performance, and to assess the effect of heterogeneity of personalities on team performance. Their findings confirm those of Myers and McCaulley [3] and Belbin [4]. They found that personality preference influences the way team members communicate and also has a significant impact on team output. Additionally, they found that the higher the heterogeneity of software teams, the higher their productivity levels are likely to be.

Rajendran [12] employed the Belbin model for personality assessment when observing three software development teams. He found that both positive and negative qualities are associated with personality preferences. Rajendran also found that, through personality assessment, teams can be constructed to possess members with a mix of compatible personality traits, thereby reducing the risk of personality conflicts and enhancing the likelihood of success in team development tasks.

Previous studies, by Bradley and Hebert [17] and Faraj and Sproul [18], also substantiate the Myers and McCaulley findings [3]. In a case study of two software teams, Bradley and Hebert found that team composition based on personality type influences team performance. They found that a united team is an important variable for team effectiveness and team success. Additionally, their findings indicated that the team with a balance of personality types performed better than the team that was homogenous. In a study involving 69 software development teams, Faraj and Sproul [18] also reported that managing personality traits has a significant positive impact on software development performance.

Studies by Beranek *et al.* [10] and Jones [11] in student developer settings also lend support for linking personalities to roles. Using Belbin's model, Beranek *et al.* reported that personalities associated with

negative attitudes, such as a 'reluctance to share tasks' or 'being overly critical' created a negative impact on success. Beranek *et al.* found that such individuals pose a significant threat to the success of the team and other team members. Additionally, they observed that an individual's willingness to participate in group settings also plays an important role in successful team work. These findings are consistent with those of Jones [11] who reported that personality traits are significantly correlated with team cohesion, and team cohesion is a necessary ingredient for team success.

**Table 1.** Belbin team roles [4]

<b>Role</b>	<b>Positive Qualities</b>	<b>Allowable Weaknesses</b>
Company Worker	Organizing ability, practical common sense, hard-working, self-discipline.	Lack of flexibility, unresponsive to unproven ideas.
Chairman	A capacity for treating and welcoming all potential contributors on their merits and without prejudice. A strong sense of objectives.	No more than ordinary in terms of intellect or creative ability.
Shaper	Drive and a readiness to challenge inertia, ineffectiveness, complacency or self-deception.	Proneness to provocation, irritation and impatience.
Plant	Genius, imagination, intellect, knowledge.	Up in the clouds, inclined to disregard practical details or protocol.
Resource Investigator	A capacity for contacting people and exploring anything new. An ability to respond to challenge.	Liable to lose interest once the initial fascination has passed.
Monitor-Evaluator	Judgment, discretion, hard-headedness.	Lacks inspiration or the ability to motivate others.
Team Worker	An ability to respond to people and to situations, and to promote team spirit.	Indecisive-ness at moments of crisis.
Completer-Finisher	A capacity for follow-through, perfectionism.	A tendency to worry about small things. A reluctance to let go.

## 2.4. Risks, roles and agile development

Among the principles and activities recommended by the proponents of agile methods, communication and collaboration are heavily emphasized [19]. Given that agile methods are being increasingly adopted by software teams [20] it would seem important that the effectiveness of collaboration among team members is maximized, taking into account each member's personality traits, strengths and weaknesses [15].

Chin [21] asserts that agile software development methodologies, especially XP, rely on practices that require experienced and versatile team players. Chin's work implies that any deviation from this – say, teams including too many of Belbin's Company Workers who lack versatility – introduces risks to software quality and delivery. In addition, documentation of the software development process in agile approaches is often replaced by close and constant collaboration among team members. However, De Souza *et al.* [22] and Nord and Tomayko [23] state that in large-scale projects documentation becomes necessary for project success, and that in such scenarios human interaction cannot entirely replace it. Similarly, Boehm and Turner [24] say that the loss of team players in agile teams (which could occur due to personality clashes) may result in the loss of tacit knowledge, which introduces further project risks.

Employee turnover is an inherent risk factor in all software projects. According to Highsmith [25], the impact of turnover can be reduced through training and appropriate amounts of documentation. Williams and Kessler [26] assert that the agile practice of coupled teams facilitates knowledge sharing, which is likely to reduce the impact of employee turnover. The coupled team practice employed in agile project environments may also induce extensive tacit knowledge sharing, which may also reduce risks.

Highsmith [25] further contends that the risk of poor software development productivity arises from three sources: having an inefficient team, having a team that is not cohesive, and having a team with poor morale – all of which can arise due to clashes in personality or poor role assignment choices [17]. Several agile practices may potentially mitigate these risks. Highsmith explains that agile approaches of having the right people on the team, coaching the team and promoting team development all help to offset the risk of poor productivity. Further, some agile methodologies such as XP and SCRUM focus on short iterations, which Chin [21] suggests may also mitigate risks associated with poor productivity.

Personality clashes and poorly informed role assignments in agile projects threaten team cohesion and therefore represent a risk to project success [27]. Thus, assessing and managing team members' personalities should provide an opportunity to reduce such risk. The intent is to help managers to assemble software teams with a balance of personalities in terms of the types described briefly in the prior subsections. For example: examining Belbin's team roles shown in Table 1, a project manager allocating staff to a novel project for a particularly innovative client might assign individuals possessing the *Plant* personality to the primary role of Programmer, whereas individuals possessing a dominant *Completer-Finisher* personality could function most effectively as Quality Assurance Specialists (QASs). The rationale for these allocations could be stated as follows:

- Since the *Plant* role is linked to 'genius, imagination, intellect, and knowledge' personality traits, individuals occupying this role might be capable of generating rapid and innovative solutions to software problems, making them most suitable as programmers in this particular project.
- On the other hand, the *Completer-Finisher* role is associated with 'a capacity to follow-through, perfectionism'. Thus, individuals occupying this role may pay attention to detail, unearthing errors, and ensuring software quality, functioning effectively as QASs.

Of course, another project with different contextual parameters may demand a very different set of role assignments, and in agile projects re-assignment may need to take place during the development process. For instance, if progress becomes problematic and morale is threatened the project manager may call on those *Team Workers* in the group to take a more prominent leadership role.

In order for a project manager to make such decisions, however, a means of assessing individual personality traits is needed. To be used effectively in agile projects, any solution should be unobtrusive and should support the emergence and management of projects on an ongoing, rather than static, basis [28].

## 3. TOOL DEVELOPMENT AND EVALUATION

In light of the above discussion we set out to develop a prototype project management tool component that could be utilized to assist those involved in agile software development to assess team members' personalities in a lightweight yet comprehensive manner. Ideally assessment would

occur before team formation. The project manager then has the opportunity to assemble a team with the best mix of personality types for a given project.

### 3.1. Design, implementation and operation

While the use of tools in agile projects is not universally supported (as per the Agile Manifesto's principle of "Individual and interaction over processes and tools"), *appropriate* use of *appropriate* tools is acknowledged as adding value in terms of enhancing traceability of decision-making and ensuring that important information is not lost [20, 29].

Note that we initially evaluated six existing software project management tools – both proprietary and open-source – that had been designed to support both planned and agile projects (ExtremePlanner, VersionOne, DevPlanner, Rally, AceProject and MS Project). None of the six incorporated any sort of capability in terms of assessing individuals' personalities.

We therefore designed and constructed a web-based tool called the Agile Social Risk Mitigation Tool, or ASRMT, which incorporated a component focused on personality trait assessment. Development followed an iterative, Feature-Driven process with early and ongoing prioritization of functionality. The tool was to be web-based in order to support the collaborative work of distributed agile teams. Our intent to rapidly produce a prototype in the first instance led to the selection of ASP.NET as the development framework and use of the Visual Basic language. Some JavaScript was also written in order to support user validation and an SQL database provided back-end functionality. Both the database and ASRMT front-end application were installed on a local server and were accessible via a web browser. Temporary user permissions were configured so that the scenario-based evaluation (described next) could be performed.

The tool is intended to support project portfolio and personnel management at the organizational level. Therefore whenever a new stakeholder is identified across the organization their profile details would be entered into ASRMT, irrespective of whether they are likely to initially fulfill developer, project manager, QAS or tester roles, or whether they are team members from the provider (software) or consumer (client) context. When each individual logs in for the first time they must undertake some housekeeping (i.e. confirm their details and change their password) and also complete the personality assessment.

The personality assessment component of the tool enables each individual to self-determine their

dominant personality preferences, in this case using an early version of a popular psychometric assessment instrument (although any such instrument could be used). To complete the instrument individuals are required to answer a number of questions. When the survey is submitted, the user's two main personality preferences are fed back to them, along with an indication of their positive aspects. Perceived negative aspects of the personality types are not shown. The intent of this restriction is to discourage stakeholders from manipulating their answers in an effort to avoid appearing to have certain weaknesses.

Information regarding individuals' personality preferences is stored to be leveraged in project management decision making. Through assessment summaries such as that shown in Figure 1, project managers are afforded the opportunity to assemble a balanced team with the right mix of personalities, taking into account team members' major strengths and weaknesses. In addition, through continued observation, project managers can also compare actual performance against the summary provided by ASRMT to make expert judgment regarding individuals' effectiveness in their given roles. This may lead to re-allocation of team members to roles during projects as needed, and in the longer term may inform decisions regarding subsequent professional development and/or training for team members.

Note that while personality assessment is the core functionality of interest here, ASRMT incorporates other capabilities intended to support agile software development. Through a remote interface, customers are able to initiate and track the development of system features, developers can use the tool to ensure that feature status is both up-to-date and visible to all involved, and managers can interrogate the system to review progress on their portfolio of projects – irrespective of stakeholder location.

### 3.2. Preliminary evaluation of ASRMT

While ultimately it would be beneficial to formally evaluate the ASRMT components in live project settings, this would be an appropriate exercise when the tool is fully developed. In its prototype form a more informal evaluation by knowledgeable and representative users would be more suitable. Such an approach would ideally provide insights into enhancements that could be made before the tool was deployed in live projects.

To that end the tool was informally tested by seven software engineers, specifically agile software developers with varying levels of development and

project management experience. Each of the seven evaluators, who were all known to one of the authors but not the others, carried out a two-part scenario-based assessment. After being introduced to the tool and its aims, each evaluator tested ASRMT's functionality by following a number of steps, split roughly evenly across the roles of project manager, developer, and customer. In the second part of the evaluation each of the seven participants was asked to give feedback regarding their impressions of the tool and its likely usefulness in live projects. This was supported by the use of a previously validated usability instrument designed and tested by Lewis [30].

This part of the evaluation comprised 11 questions and used two sub-scales. Seven closed questions, each conforming to a Likert scale, were used to enable each evaluator to comment on ASRMT's stability and their learning experience (the first sub-scale). Two further closed questions conforming to a Likert scale and two open-ended questions were used in order to assess ASRMT's likely usefulness, to determine whether ASRMT addressed the stated aims of the tool, and to

solicit overall impressions and recommendations for improvement (the second sub-scale).

The possible answers to the questions conforming to a Likert scale were on discrete bi-polar continua and included *strongly agree*, *agree*, *disagree*, and *strongly disagree* options. They were linearly scaled from one to four (where a *strongly agree* choice was represented by one and four represented *strongly disagree*), offering respondents no neutral choice such as *neither agree nor disagree*. This approach was deliberately selected to force respondents to express an opinion. While there may be threats to reliability for evaluations employing such an approach, given that the targeted evaluators were domain experts we believe this to be a minimal threat [31].

Given the scale of the ASRMT assessment at just seven evaluators, responses to closed questions were aggregated to determine the number that favored a particular choice – *strongly agree* and *agree* were taken to be positive responses, while *strongly disagree* and *disagree* were aggregated as negative responses. Open-ended responses were analyzed using basic content analysis and the responses summarized.

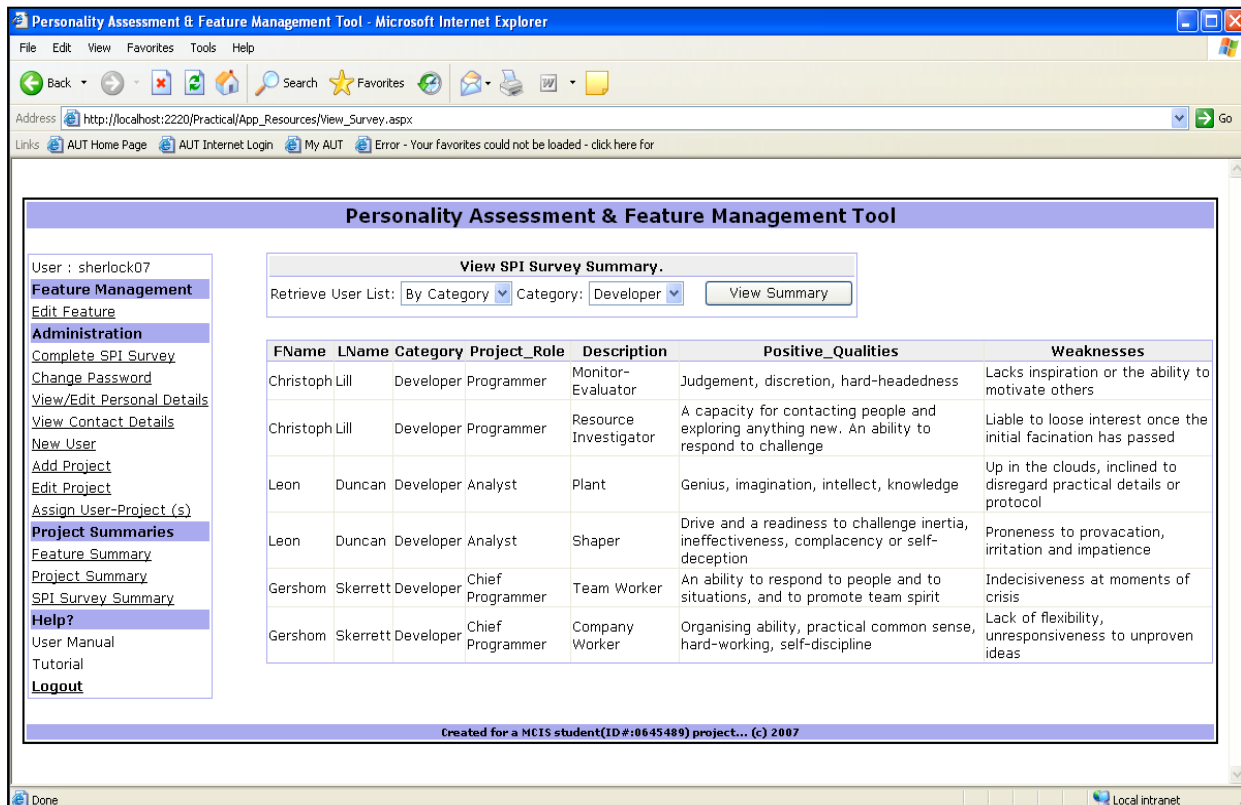


Figure 1. Personality assessment summary screen

All seven evaluators completed the assessment exercise in full. Respondents felt that the concept of supporting team composition informed by personality assessment was a good one, lending informal but encouraging general support for the intended functionality. Six of the evaluators thought that the ASRMT components were easy to use. Of the seven respondents, three reported one or two bugs whilst using ASRMT. All respondents reported that they were able to successfully complete the various scenarios, including the completion of the personality assessment survey in the roles of developer team member and customer team member, and the reporting of the survey summary in the role of project manager. It was also the opinion of all seven evaluators that ASRMT was easy to learn to use.

Five of the seven evaluators reported that ASRMT was simple and satisfying to use, while four believed that ASRMT would be useful if used in live projects. The three respondents who did not agree that ASRMT would be useful if used in live projects felt that the tool needed usability improvement before it would be suitable for implementation in live settings. All of the respondents believed, however, that ASRMT offered functionality to address the features in keeping with its purpose. In terms of evaluators' overall impressions of the tool, all believed that providing a degree of automated support for personnel capability management in a process tool would deliver benefits to project management. Regarding the tool's ease of use, five evaluators believed that ASRMT's simplicity and ease of use would enhance project management.

Among the recommendations for improvement, two respondents suggested that alternative personality assessment instruments should be investigated. They also suggested that perhaps recommendations could be made (semi-)automatically regarding the roles individuals would be most suited to in a project team, based on the analyses. While we can see the appeal of such assistance this would be a step too far for the current tool in that such recommendations may not take into account the specific characteristics of a project, characteristics that would likely influence role assignments. That said, if the tool were augmented to collect data on these characteristics e.g. project novelty, client maturity, utilization of new technologies, then some form of role recommendation functionality could be possible. For example, on a particularly novel project it may be advisable to load the team with more creative developers than those who rely on structure and consistency in approach. Finally, five evaluators believed that a few of the tool's user interfaces could be improved; and one respondent

suggested that the performance of ASRMT would need enhancement if it were to be implemented in large projects.

## 4. CONCLUSIONS

We set out to consider the potential impact of personality incompatibilities on software development projects, specifically agile developments in which team cohesion is crucial. It was evident in the relevant literature, derived from the fields of psychology and management as well as from software engineering, that such issues have the potential to derail a project or to challenge it significantly, whether through risk of schedule overrun, reduced productivity and the like. We also found that several project management tools did not consider such risks, tending instead to focus on the handling of projects' technical challenges.

We therefore built a prototype software tool component that we believed might assist software project managers directly, and entire teams indirectly, by providing lightweight support for personality assessment. The tool has been evaluated positively by a small number of agile software development practitioners, with strong support shown for the idea of part-automated personality assessment, although some enhancements and further work are required. To that end we are considering other personality assessment instruments and also looking at enhancing the functionality of the tool in terms of perhaps recommending particular role allocations and team structures, taking into account both aspects of team members' personalities and project characteristics. It is then our intent to have the tool used in live project settings and evaluated more extensively, incorporating the views not only of developers but also user and customer representatives as well as project/portfolio managers.

## 5. REFERENCES

- [1] Jung, C., *Psychological Types* (Vol. 6), Princeton University Press, New Jersey, 1971.
- [2] Montgomery, S., *People Patterns: A Modern Guide to the Four Temperaments*. Archer, California, 2002.
- [3] Myers, I., and M. McCaulley, *Manual: A Guide to the Development and Use of the Myers-Briggs Type Indicator*, Consulting Psychologists Press, California, 1985.
- [4] Belbin, R.M., *Management Teams: why they succeed or fail*. Butterworth-Heinemann, Woburn, UK, 2002.

- [5] McCrae, R.R., and P.T. Costa, *Personality in Adulthood*, The Guildford Press, New York, 1990.
- [6] J.G. Carlson, "Recent Assessments of the Myers-Briggs Type Indicator", *Jnl Personality Assessment* 49(4), 1985, pp. 356-365.
- [7] D.A. Johnson, "Test-retest Reliabilities of the Myers-Briggs Type Indicator and the Type Differentiation Indicator Over a 30 Month Period", *Jnl Psychological Type* 24, 1992, pp. 54-61.
- [8] N. McCarley, and T. Carskadon, "Test-retest Reliabilities of Scales and Subscales of the Myers-Briggs Type Indicator and of Criteria for Clinical Interpretive Hypotheses Involving Them", *Research in Psychological Type* 6, 1983, pp. 24-36.
- [9] Keirse, D., *Please Understand Me II: Temperament Character Intelligence*. Prometheus Nemesis Book Company, California, 1998.
- [10] G. Beranek, W. Zuser, and T. Grechenig, Functional Group Roles in Software Engineering Teams. Presented at the *Workshop on Human and Social Factors of Software Engineering*, HSSE'05, St. Louis, Missouri, 2005.
- [11] A. Jones, "Experience of Profile-based Group Composition", *Comp Sci Educ* 9(3), 1999, pp. 242-255.
- [12] M. Rajendran, "Analysis of Team Effectiveness in Software Development Teams Working on Hardware and Software Environments Using Belbin Self-Perception Inventory", *Jnl Management Development* 24(8), 2005, pp. 738-753.
- [13] S.T. Acuna, and N. Juristo, "Assigning People to Roles in Software Projects", *Software-Practice and Experience* 34, 2004, pp. 675-696.
- [14] S.T. Acuna, N. Juristo, and A.M. Moreno, "Emphasizing Human Capabilities in Software Development", *IEEE Software* 23(2), 2006, pp. 94-101.
- [15] K.S. Choi, F.P. Deek, and I. Im, "Exploring the Underlying Aspects of Pair Programming: The Impact of Personality", *Info and Software Technology* 50, 2008, pp. 1114-1126.
- [16] N. Gorla, and Y.W. Lam, "Who Should Work With Whom?", *Comm of the ACM* 47(6), 2004, pp. 79-82.
- [17] J.H. Bradley, and F.J. Hebert, "The Effect of Personality Type on Team Performance", *Jnl Management Development* 16(5), 1997, pp. 337-353.
- [18] S. Faraj, and L. Sproul, "Coordinating Expertise in Software Development Teams", *Jnl Management Science* 46(12), 2000, pp. 1554-1568.
- [19] A. Cockburn, and J. Highsmith, "Agile Software Development, the People Factor", *Computer*, 34(11), 2001, pp. 131-133.
- [20] P. Behrens, Agile project management (APM) tooling survey results. Retrieved January 5, 2007, from <http://www.trailridgeconsulting.com>
- [21] Chin, G., *Agile Project Management: How to Succeed in the Face of Changing Project Requirements*, American Management Association, New York, 2004.
- [22] S. De Souza, N. Anquetil, and K. De Oliveira, A Study of the Documentation Essential to Software Maintenance. Presented at the *23rd Annual International Conference on Design of Communication*, Coventry, UK 2005.
- [23] R.L. Nord, and J.E. Tomayko, "Software Architecture-Centric Methods and Agile Development", *IEEE Software* 23(2), 2006, pp. 47-54.
- [24] B. Boehm, and R. Turner, "Using Risk to Balance Agile and Plan-driven Methods", *Computer* 36(6), 2003, pp. 57-66.
- [25] Highsmith, J., *Agile Project Management: Creating Innovative Products*, Pearson, Boston MA, 2004.
- [26] Williams, L., and R. Kessler, *Pair Programming Illuminated*, Addison-Wesley, Boston MA, 2003.
- [27] S.P. Grisham, and E.D. Perry, Customer Relationships and Extreme Programming. Presented at the *Workshop on Human and Social Factors of Software Engineering*, HSSE'05, St. Louis, Missouri, 2005.
- [28] T. Williams, "Assessing and Moving On from the Dominant Project Management Discourse in the Light of Project Overruns", *IEEE Transactions on Engineering Management* 52(4), 2005, pp. 497-508.
- [29] J. Kaariainen, J. Koskela, P. Abrahamsson, and J. Takalo, *Improving Requirements Management in Extreme Programming with Tool Support - an Improvement Attempt That Failed*. Presented at the *Euromicro Conference*, 2004.
- [30] J.R. Lewis, "IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use", *International Journal of Human-Computer Interaction* 7(1), 1995, pp. 57-78.
- [31] Kirakowski, J. *Questionnaires in usability engineering: a list of frequently asked questions*. Human Factors Research Group, Cork, Ireland, 2000.