



# Exploring the Influence of Belbin's Roles in Software Measurement: A Controlled Experiment with Students

Raúl A. Aguilar<sup>(✉)</sup>, Julio C. Díaz, Juan P. Ucán,  
and Yasbedh O. Quiñones

Universidad Autónoma de Yucatán, Mérida 97000, Yucatán, Mexico  
{avera, julio.diaz, juan.ucan}@correo.uady.mx,  
sky\_lab\_beto@hotmail.com

**Abstract.** The article presents a controlled experiment in which the convenience of using the Belbin Role Theory for the integration of work teams with the task of measuring the software is explored. The study is developed in an academic environment with students of the Software Engineering degree and analyzes the differences between the metrics obtained with the Function Point Technique, by teams integrated with the Belbin Theory, and those obtained by teams integrated with students selected randomly. The results obtained provide evidence regarding the significant differences in terms of the time spent for the task; it was observed that teams integrated with the Belbin theory take more time. Regarding the five metrics obtained to measure the functionality of the software, differences were found only in the functionality linked to the external outputs.

**Keywords:** Software testing education  
Computer-supported collaborative learning  
Computer-supported cooperative work  
Experimentation in software engineering

## 1 Introduction

The body of knowledge of Software Engineering (SE) began to be integrated since the late sixties in response to a situation known as “the software crisis”. This situation generated dissatisfaction in customers mainly for three particular situations linked to the software process: (1) cost overruns in the software process, (2) non-compliance with the delivery deadlines, and (3) requirements for the delivered software product are not met. After just half a century, Software Engineering (SE) has a body of knowledge accepted by professionals and researchers of the discipline [1], which integrates a set of areas linked to development processes (Requirements, Design, Construction, Testing and Maintenance) and management processes (Quality, Configuration, Software Process) associated to the aforementioned areas. There is also a set of methods, techniques, tools and good practices that have been integrated into the body of knowledge of the SE with the intention of improving both the processes as the products generated through said discipline.

The improvement of the software process has been studied from several angles, one of which has had singular interest in recent years, studies the importance of the human factor in the software development process; this factor acquires singular relevance due to the fact that several of the processes linked to Software Engineering are carried out in the context of work teams, and therefore, the human factor can affect the success or failure of a project of this kind. In this sense, De Marco comments in [2] that the main problems or causes of the failure of the projects are not of a technological nature, but rather are due to factors of a sociological nature; for his part, Humphrey in [3] states that the process of forming and building a software development team does not happen by accident, the team needs to establish working relationships, agree on objectives and determine roles for group members.

The purpose of the study described in this article is to explore whether the use of role theory proposed by Belbin [4] for the formation of effective teams has any influence on the measurements made by development teams trained with this theory, in comparison with the measurements made by traditional development groups, integrated randomly. It is worth mentioning that the controlled experiment was developed within the framework of a course on Design of Experiments given to students of a Bachelor in Software Engineering.

The remaining part of this document is organized as follows: Sect. 2 presents a general description of related work. Section 3 briefly presents the software measurement technique known as Function Points. Section 4 presents the experiment performed. Section 5 presents the results and findings. Finally, the conclusions are presented in Sect. 6.

## 2 Related Work

Some researchers [4–6] claim to have identified roles that describe the behavior of individuals in work teams—team roles—and although there is no evidence that they are particularly associated with any type of activity, their absence or presence is said to have significant influence on the work and the achievements of the team [7].

Among the proposals on team roles, Belbin's work [4] is surely the most used among consultants and trainers, its popularity is that, it not only offers a categorization of roles, but also describes a series of recommendations for the integration of work teams, which are known as the Belbin Role Theory. In relation to the roles of Belbin, Johansen in [8] proposed to validate the inventory of self-perception proposed by Belbin through the observation of the behavior of work teams. As part of his conclusions, he indicated that it is worthwhile to use said instrument as a tool to evaluate the composition and possible performance of the team.

Pollock in [9] explores whether diversity of roles and personalities among the members of a team of information systems students can improve the effectiveness of the team. He concludes that diversity does not have a significant influence on the effectiveness of the team; however, he says that the presence of certain roles, such as the Sharper, Chairman and Completer-Finisher, can increase effectiveness. In this same sense, Henry and Stevens [10] reported a controlled experiment with students in which they explore the improvement in the effectiveness of software development teams,

based on the set of roles proposed by Belbin; these authors analyzed the general usefulness of Belbin's roles—particularly the roles that the leader can play—in terms of two aspects: performance and viability; they conclude that teams that contain only one leader role present better performance than those that do not include it or those that include more than one leader; in a similar study, Estrada and Peña [11] reported that some roles have a greater contribution with certain activities, as was the case of the Implementer role in the coding task. Aguilar in [12] reports that the collaboration skills presented by groups formed on the basis of role theory (GBF), is significantly greater than that presented by the groups formed solely taking into account functional roles (GMF); reports that GBFs spend more time than GMFs in their work sessions. In a previous work [13] we compared, in an academic environment, the quality of the readability of the code generated by teams integrated with the Belbin Theory (EB), in contrast to randomly-formed teams (ET); among the results obtained, it was reported that EB teams show significantly better results than ET teams.

### 3 Function Point Analysis

The Function Point Analysis (FPA), introduced by Allan Albrech in 1979, is an accepted standard for measuring the logical or functional size of software projects or applications, based on the functional requirements agreed upon with the user [14]. The general process for measuring the size of a software system can be summarized in the following steps:

1. *Determine the type of Function Point count to be conducted:* This technique allows to measure the functionality at the beginning of the development process, in maintenance time or when a system is already in operation time.
2. *Identify the application boundary.* Identification of the scope is established in the start phase of the project, it is usually documented in the form of software requirements or use cases.
3. *Identify all data functions and their complexity.* There are two types of functionality linked with logical files.
  - *Internal Logical File (ILF):* is a group of logically related data or control information maintained through an elementary process of the application within the boundary of the software system.
  - *External interface file (EIF):* is a group of logically related data or control information referenced by the application but maintained within the boundary of different application.
4. *Identify all transactional functions and their complexity.* It is possible to identify three types of transactions:
  - *External input (EI):* is an elementary process of the application which processes data or control information that enters from outside the boundary of the software system.
  - *External Output (EO):* is an elementary process of the application which processes data or control information that exits the boundary of the software system.

- *External Inquiries (EQ)*: is an elementary process of the software system which is made up of an input-output combination that results in data retrieval.
5. *Determine the Unadjusted Function Point (UFP) count.* According to IFPUG [15], the complexity of each of the five types of functions can be determined using a complexity table (see Table 1).

**Table 1.** Unadjusted function point counting weights.

Type	Low	Average	High	Total
EI	__ x 3	__ x 4	__ x 6	
EO	__ x 4	__ x 5	__ x 7	
EQ	__ x 3	__ x 4	__ x 6	
ILF	__ x 7	__ x 10	__ x 15	
EIF	__ x 5	__ x 7	__ x 10	

*Total UFP*

6. *Determine the Value Adjustment Factor (VAF).* An adjustment factor is calculated that considers fourteen general aspects related to the system (i.e. reusability, ease of change, range of transactions).
7. *Calculate the final Adjusted Function Point (AFP) count.* VAF is applied to the Unadjusted Function Point count.

## 4 Methodology

A controlled experiment was designed to explore the influence of the use of the Belbin Role Theory in the integration of software development teams. In our case, the influence was evaluated using the result—product—of applying the technique of function points. In particular, the five metrics linked to the functionality of the files (ILF, EIF) and transactions (EI, EO, EQ) declared in an Software Requirements Specification were compared. Additionally, the time used by the teams to carry out the task was used as a dependent variable—a metric process.

### 4.1 Objective, Hypothesis and Variables

With the aim of exploring whether the integrated work teams based on the Belbin Role Theory—which we will call *Compatibles*—generate measurements significantly different from those obtained by the integrated teams without using any particular criteria—which we will call *Traditional*—five pairs of hypotheses—a pair for each of the metrics obtained with the function point technique—were proposed, such as the following:

- H0<sub>1</sub>: The average of the ILF obtained with the Function Point Technique by the Traditional Teams is equal to the average of the ILF obtained by the Belbin Teams.

H1<sub>1</sub>: The average of the EIF obtained with the Function Point Technique by Traditional Teams differs from the average of the EIF obtained by the Belbin Teams

Likewise, with the objective of identifying whether the time spent by the integrated software development teams based on the Belbin Role Theory, in the aforementioned software measurement task, differs from the time that the integrated teams invest randomly, we proposed the sixth hypotheses:

H0<sub>6</sub>: The average time recorded by the traditional teams in the measurement task is equal to that reported by the Belbin Teams.

H1<sub>6</sub>: The average time recorded by traditional teams in the measurement task differs from that reported by the Belbin Teams

The factor to be controlled in this experiment is the integration mechanism of the software development teams, which has two alternatives: (1) Belbin Teams, and (2) Traditional Teams. On the other hand, the six response variables (ILF, EIF, EI, EO, EQ, Time) obtained by applying the technique of Function Points in the ERS, will be recorded in the instrument that the work teams deliver at the end of the activity. Aspects such as the complexity of the problem to be solved, the time available for the task, as well as the degree of experience of the participants, are considered parameters that do not affect or skew the results of the study, because they are homogeneous parameters for all the teams of development; in the case of the degree of experience, teams are composed of students who are in their training process and, at the time of the experiment, still unaware of the technique, are also homogeneous.

## 4.2 Participants/Subjects

The participants in the experiment were twenty seven students of a Bachelor in Software Engineering from the Autonomous University of Yucatan, who were studying the Software Engineering Experiments Design course in the August-December 2016 semester, subject located in the fifth semester.

With the students enrolled in the course, nine software development teams of three members each were formed; we used the information obtained—primary roles in students—after the administration of the self-perception inventory of Belbin, and we integrate five teams with compatible roles (Belbin Teams: BT) and four additional teams with students assigned in a random way (Traditional Teams: TT). Given that the measurements would be obtained on the products generated by the development teams, the experimental subjects in this case were the nine work teams integrated by the researchers. The conformation of the five teams (with three members) based on the Belbin Role Theory are described in Table 2.

## 4.3 Experimental Design

A Factorial Design with “one variation at a time” was used, the independent variable corresponds to the way of integrating the work teams, in the study there are two

**Table 2.** Integrated teams with Belbin Theory (Compatible teams).

Teams	Integrating roles
I	(1) Implementer, (2) Monitor-Evaluator, (3) Coordinator
II, IV	(1) Implementer, (2) Implementer, (3) Completer
III, V	(1) Team Worker, (2) Team Worker, (3) Plant

**Table 3.** Experimental subjects by treatment.

Treatment	Teams
Belbin Teams (BT)	I, II, III, IV, V
Traditional Teams (TT)	VI, VII, VIII, IX

experimental treatment factor levels: BT and TT. Table 3 illustrates the assignment of the nine teams to each of the treatments.

#### 4.4 Execution of the Study

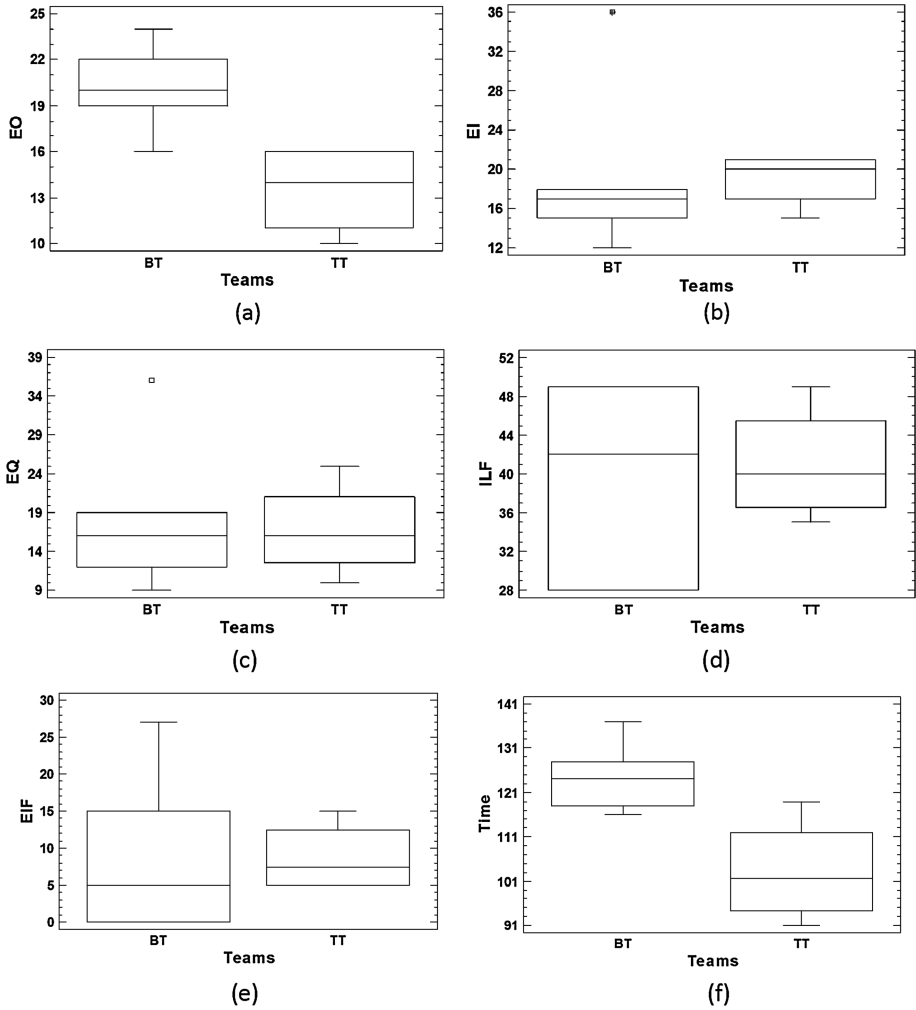
The study was carried out in three work sessions; in the first one, the self-perception study was administered to the students, this session was carried out in the last half hour of a class session of the subject. Subsequently, in a second session that lasted two hours, participants received instruction on the technique function point analysis. In the third session, the experiment was executed.

As the first activity in the experimental session, the nine teams were integrated based on what was described in Sect. 5.2, and they were provided with the ERS of a case study, as well as a report sheet with the scheme of Table 2; brief description of the activity was described and they were asked to identify the team, as well as record the start and end time of the activity.

## 5 Analysis and Results

This section presents both the descriptive statistical analysis of the measurements collected and the inferential statistical analysis.

For the descriptive analysis, due to the limitations of space in the publication, six simultaneous boxplot were generated (see Fig. 1); we selected that type of graph because they represent a descriptive way of comparing treatments. In Fig. 1 we can see that only the graphics (a) and (f) do not present overlaps in the treatments, so it is very likely that they present differences between them; also, in the first case, a measure obtained for EO, the BT average was closer to the real value. We can also observe that in the two measurements linked with data functions, BT present greater dispersion and asymmetry than TT; As for the other two measurements related to transactions, no differences were observed in the dispersion of the treatments, however the BT have outliers in both cases.



**Fig. 1.** Boxplot for each of the six comparative analyzes.

In order to evaluate the differences observed in the EO variables—as a product metric—and Time—related to the process—with the descriptive analysis and determine if they are significant from the statistical perspective, the following statistical hypotheses were raised:

$$H0_{EO} : \mu_{BT} = \mu_{TT}; H1_{EO} : \mu_{BT} < > \mu_{TT} \quad (1)$$

$$H0_{Time} : \mu_{BT} = \mu_{TT}; H1_{Time} : \mu_{BT} < > \mu_{TT} \quad (2)$$

We proceeded to use the one-way ANOVA; the associated linear statistical model is the following:

$$Y_{ij} = \mu + \beta_i + \varepsilon_{ij};$$

where  $Y_{ij}$  is the  $ij$ -th observation (value of the  $j$ -th replica under treatment  $i$ ),  $\mu$  is a parameter common to all treatments called general or global mean,  $\beta_i$  is a parameter associated with the  $i$ -th treatment called effect of the  $i$ -th treatment and  $\varepsilon_{ij}$  is the random component of the error. Tables 4 and 5 present the results of these analyzes.

**Table 4.** ANOVA for EO by teams.

Source	SS	Df	MS	F	P-Value
Between groups	99.7556	1	99.7556	10.94	0.0130
Within groups	63.8	7	9.11		
Total (Corr.)	163.556	8			

**Table 5.** ANOVA for time by teams.

Source	SS	Df	MS	F	P-Value
Between groups	1012.94	1	1012.94	9.96	0.0160
Within groups	711.95	7	101.707		
Total (Corr.)	1724.89	8			

The ANOVA table decomposes the variance of the variable under study into two components: an inter-group component and an in-group component. Since the P-value of the F-test is less than 0.05 in both study variables (EO:0.013 & Time:0.016), we can reject the null hypothesis and affirm in both cases that there is a statistically significant difference between the mean of the variable under study between a level of Treatment and other, with a 5% level of significance.

$$H1_{EO} : \mu_{BT} < > \mu_{TT} \ \& \ H1_{Time} : \mu_{BT} < > \mu_{TT}$$

The ANOVA Model has associated three assumptions that it is necessary to validate before using the information it offers us; the assumptions of the model are: (1) The experimental errors of your data are normally distributed, (2) Equal variances between treatments (Homoscedasticity) and (3) Independence of samples.

To validate the first assumption, we will use the normal probability graph. It is a graphical technique for assessing whether or not a data set is approximately normally distributed. As can be seen in the graph of Fig. 2, the points, in both graphs, do not show deviations from the diagonal, so it is possible to assume that the residuals have a normal distribution in both cases.

In the case of Homoscedasticity, we generate a residuals vs. fitted plot and we observed if it is possible to detect that the size of the residuals increases or decreases systematically as it increases the predicted values; as we can see in the two graphs of



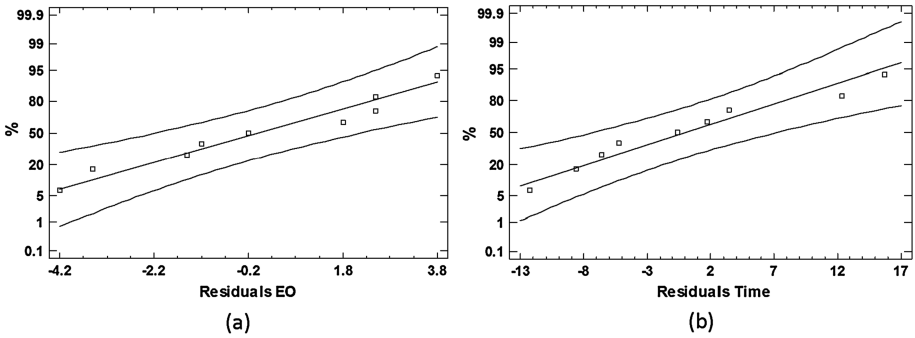


Fig. 2. Normal probability plot for (a) EO and (b) Time.

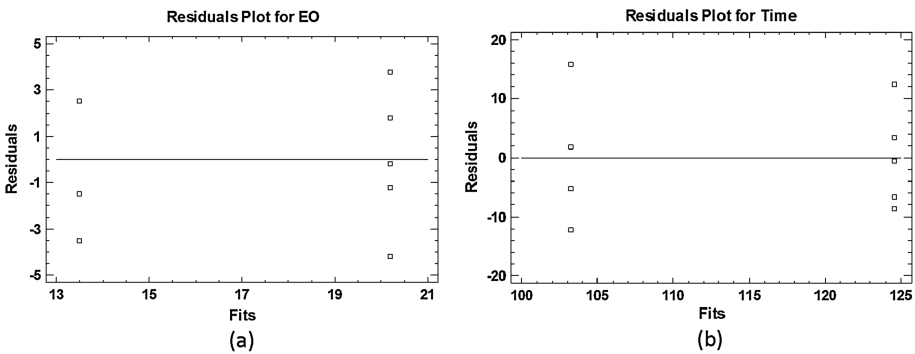


Fig. 3. Residuals vs. Fitted plot for (a) EO and (b) Time.

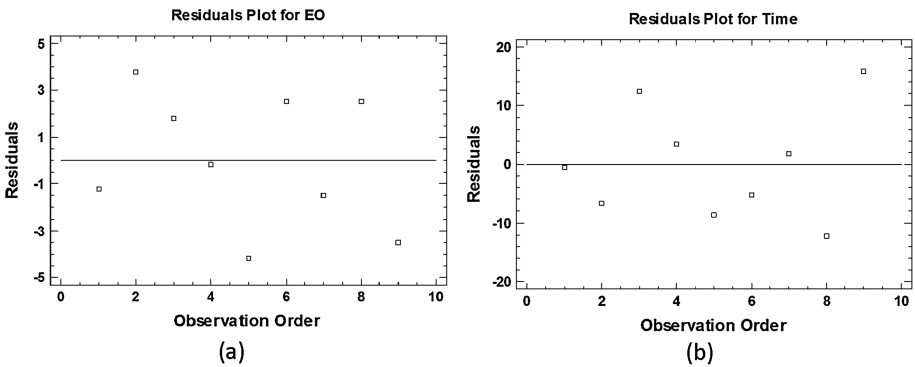


Fig. 4. Residuals vs. Order of the data plot for (a) EO and (b) Time.

Fig. 3, no pattern is observed by which we can accept that the constant variance hypothesis of residuals is met.

Finally, to validate the assumption of data independence, we generate a residuals vs. Order of the Data plot; in this case, we observe if it is possible to detect any

tendency to have gusts with positive and negative residuals; in the case of our analysis, we can see in Fig. 4 that in both cases no trend is identified, so it is possible to assume that the data come from independent populations.

## 6 Conclusions

In this paper, we present a controlled experiment in which we compare the performance—metrics linked to the product and the process—of work teams in tasks related to software development, particularly, software measurement using the Function Point Technique. The two treatments to be compared, were linked with the way of integrating the work teams; firstly, the traditional way of randomly assigning its members, and on the other, the proposal to integrate teams using the Belbin Role Theory. The results regarding the process metrics showed significant differences only in the EO metric; however, no prior information is available that explains in any way the observed result; that is, it is not clear that the type of activity carried out for the measurement of the EO was particularly different than that used in the other metrics, such that this could be influenced by the type of conformation of the team; what could be observed, is that although the average value in both types of teams was far from the real value, the BT averaged a value closer to the real, which coincides with that reported in [13]. Regarding the metric associated with the process, in this case, the time that the teams used to carry out the task, the results showed significant differences; that is, the BT use more time than the TT, which coincides with one of the experiments reported in [12], in the sense that the teams formed according to the Belbin Role Theory develop in their work sessions a greater degree of interaction, which means that they invest more time in trying to reach a consensus in the decisions of the team.

## References

1. Bourque, P., Fairley, R.: Guide to The Software Engineering Body of Knowledge (SWEBOK V3.0). IEEE Computer Society (2014)
2. DeMarco, Y., Lister, T.: Peopleware Productive Projects and Teams, 2nd edn. Dorset House Publishing Co., New York (1999)
3. Humphrey, W.: Introduction to the Team Software Processes. Addison Wesley Longman Inc., Reading (2000)
4. Belbin, M.: Team Roles at Work. Elsevier Butterworth Heinemann, Oxford (1993)
5. Mumma, F.S.: Team-Work & Team-Roles: What Makes Your Team Tick?. HRDQ, King of Prussia (2005)
6. Margerison, C.J., McCann, D.J.: Team management profiles: their use in managerial development. *J. Manag. Dev.* **4**(2), 34–37 (1985)
7. Senior, B.: Team roles and team performance: is there ‘really’ a link? *J. Occup. Organ. Psychol.* **70**, 85–94 (1997)
8. Johansen, T.: Predicting a Team’s Behaviour by Using Belbin’s Team Role Self Perception Inventory. Dissertation at Department of Management & Organisation. University of Stirling (2003)

9. Pollock, M.: Investigating the relationship between team role diversity and team performance in information systems teams. *J. Inf. Technol. Manag.* **20**(1), 42–55 (2009)
10. Henry, S., Stevens, K.: Using Belbin's leadership role to improve team effectiveness. *J. Syst. Softw.* **44**(3), 241–250 (1999)
11. Estrada, E., Peña, A.: Influencia de los roles de equipo en las actividades del desarrollador de software. *ReCIBE* **2**(1), 1–19 (2013)
12. Aguilar, R.: Una Estrategia Asistida por Entornos Virtuales Inteligentes. Doctoral Thesis. Polytechnic University of Madrid (2008)
13. Aguilera, A. Ucán, J., Aguilar, R.: Explorando la influencia de los roles de Belbin en la calidad del código generado por estudiantes en un curso de ingeniería de software. *Revista Educación en Ingeniería* **12**(23), 93–100 (2017)
14. Garmus, A., Herron, D.: *Measuring the Software Process*. Prentice Hall, Upper Saddle River (1996)
15. IFPUG: *Function Point Counting Practices Manual Release 4.1.1*. International Function Point Users Group (2010)