Improving Virtual Team Performance: An Empirical Approach

D. Radoiu(1), C. Enachescu(1), H. F. Pop(2)
(1) Department of Mathematics and Computer Science
Petru Maior University, Târgu Mures Romania
(2) Faculty of Mathematics and Computer Science
Babeş-Bolyai University, Cluj-Napoca, Romania

ABSTRACT

Increased performance is the key reason for developing software with virtual teams but the research efforts to date are limited. Difficulties for a formal, generic approach are multiplied by the significant differences between how small and large software organizations react to changing operational conditions and act to increase performance.

The aim of the paper is to provide the guidelines on how to analyze small virtual team software development process performance using the workflow model and meaningful measurable elements which provide more insight into the process. The purpose is to improve small virtual team performance by obtaining knowledge about problem areas and cooperation issues which affect the operational efficiency.

A limited quantitative survey was used to test the conceptual model and hypotheses of the study. The preliminary results indicate that such an approach works, that organizational issues are as important as the technical ones and that the virtual team performance depends critically on communication.

Keywords: Virtual team, empirical approach, operational efficiency.

1. Introduction

Continuing developments in software development have led to new organizational forms that are more flexible and responsive. One of the fundamental components of these new organizational forms is the virtual team.

Virtual teams are usually distributed, both geographically (different countries) and organizationally (different companies). To successfully complete a project, virtual team members need to share/exchange information of all kind. Typically, the Internet is used as an infrastructure and the project is managed using so called workflows and synchronous/asynchronous communication (e.g. e-mail messages, phone, instant messages). The workflow ensures the logical progression of work across the virtual team and can be monitored and visualized using different tools.

The success of the project depends not only on how well each virtual team member/party executes its part of the work but on how the virtual team operates like one. Virtual teams, due to their limited life span and cross-functional or cross-organizational membership, may be unable to quickly reach maximum efficiency and therefore coordinating the work, quickly and accurately pinpointing problem areas becomes more critical.

As a consequence, interest in how to improve collaboration in global virtual teams is growing [Duarte D.L. and Snyder N.T.(1999), Lipnack J. and J. Stamp J. (1997)].

Deriving a generic virtual team model involved in software development is too difficult a task although the advantage would be that individual processes could be obtained by instantiating such a model.

The questions are: “How can we enhance a virtual team performance? How can we identify problem areas? How can we make predictions in an on-going process?”

The paper’s aim is to answer these questions – for small virtual teams - through an empirical approach. The methodology is based on modeling the specific virtual team workflow, identifying the corresponding meaningful set of measurements, and using the numerical scores to identify problem areas.

2. Definitions

Process

The software development community uses *process* as synonym to everything that needs to be done in a successful development. We will use the appellation process to describe the software development activity viewed as the transformation of input to output including technology issues, human resource issues, lifecycle issues as well as issues pertaining to software development phases.

Life-cycle

The term *life-cycle* is used to describe a sequence of phases that cover the extent of the temporal existence what whatever is being described [Henderson-Sellers, 2002]. Thus, a project development life-cycle describes the duration over which the project is conceived and constructed (called together *development*).

Virtual team

The term *virtual team* is used to describe a group of geographically dispersed and organizationally differentiated members brought together temporarily through information and communication technologies, engaged to complete a project.

As one of the primary advantages offered by virtual teams is their flexibility, organizations that employ them have the ability to group the most appropriate set of individuals required to complete a specific task. For this reason, virtual teams are normally assigned to mostly non-routine tasks, atypical and highly specialized projects involving unique information and changing requirements.

Measurements and metrics

The area of software *measurements* is also known as software *metrics* (not considered in the sense of a metric space). We will use the term *measurement* (in the sense of measurement theory) to define any mapping of empirical objects to numerical objects by homomorphism.

Input

In this paper the input in a software development project is a set of requirements (work package functionalities, quality, and time to complete) and a set of delivery acceptance criteria.

Output
The *output* of a software development project is usually a deliverable software component or software system. We assume it can be measured against initial requirements and acceptance criteria by testing.

Acceptance criteria
The success of a project is measured by comparing the output with the input with regard to functionalities, quality and delivery time. A deliverable has required quality if no rework is needed after delivery.

3. Research objectives and methodology

Issues in a virtual team software development process are complex and multidimensional. The aim of the paper is to describe an empirical approach enabling a small virtual team to asses its performance and increase awareness on what mechanisms to activate to improve operational efficiency.

One could argue this could be reached by simply choosing the right skilled team members who agree to put in place a good software development processes and operate at a high CMMI level. We agree that all of the above are essential criteria to *having* a virtual team software development process. Our research is focusing on making it *happen* more efficient.

The steps proposed by our empirical approach are:

- model the specific workflow
- identify critical areas and their measurements
- use the information to understand co-operation issues, to take corrective actions or predict the outcome of the on-going process

Whereas modeling phase was pretty straightforward and eliminated the difficulty (in taking corrective actions) generated by the fact that virtual team parties usually have different views about the process, some difficulty was encountered in identifying critical areas and establishing the effective measurements.

In the second part of the research we asked five organizations engaged in virtual team software development to use the proposed approach and collect data. Finally, we checked if the measurements were relevant to identify problem areas. Preliminary results are presented.

4. Modeling a simple workflow

The key to enabling parties to identify problem areas is having a common agreed activity description; the project workflow. The example presented in this paper refers to a frequently encountered case in a small virtual team consisting of two groups of IT specialists (belonging to two different organizations/companies), physically separated and integrated by adequate networking and communication technology. An example from the real life could be two companies (one operating like an outsourcing broker and leading the project) working together for a third party.

It is important to see that the two parties forming the virtual team have control and insight on different process areas (depicted in Figure 1 as lanes in the project pool using BPMN – Business Process Management notation), therefore the interpretation of project elements is inherently different. It follows parties agreement on the workflow model is essential in making sense of all measurements and interpretation.

We will call the two teams Team1 and Team 2, Team 1 leading the project.

We start by identifying process areas and which party exercises control over it.

The overall process consists of six major areas:

- Authorizing a work package (Team 1)
  - Activities: requirements management, defining the work package and stating acceptance criteria for executed work
- Accepting a work package (Team 2)
  - Activities: clarifying work package definition, acceptance criteria, assessing risk
- Executing a work package (Team 2)
  - Activities: work package planning and execution, reporting progress (status). Activities are strongly influenced by directions and corrections, change requests, from Team 2 and communication quality.
- Assessing progress (Team 1)
  - Activities: monitoring progress, directions and corrections
- Delivering executed work package (Team 2)
  - Activities: check executed work package against acceptance criteria (usually changed according to post-awarding change requests due to poor requirements management) and timely delivery
- Accepting executed work package (Team 1)
  - Activities: receiving executed work package and checking requirements satisfaction

Team 1 has full control over:

- Work package (WP) definition
- Work package (WPS) stability
- Requirements management (area that usually is not well measured/managed problems being transferred to the Team 2)
- Project monitoring (status) and oversight (directions and corrections)
- Requirements satisfaction (by checking deliverables against acceptance criteria)

Team 2 has full control over:

- Internal development processes
- Deliverables quality (e.g. executed work package, intermediate deliverables)

Team 2 and Team 1 share control over:

- Communication effectiveness
- Relationship management
- Contract execution
- Communication channels
- Risk management

Now that we have a common view of the process, next step is to identify meaningful measurements and to collect data. The aim is to gather useful data either to pinpoint problem areas in order to implement corrective actions or to analyze the process performance.

Problem areas in our simple example could span from insufficient definition of the work package, through instability of the requirements, delayed communication, to late major change requirements. The overall goal we are focusing on is securing the best co-operation between the two groups which form the virtual team. The main criteria in choosing what is worth measuring were to facilitate an early identification of problem areas and thus enable the implementation of corrective actions.

The workflow model suggests both a set of relevant measurements and a consistent way to interpret hem.

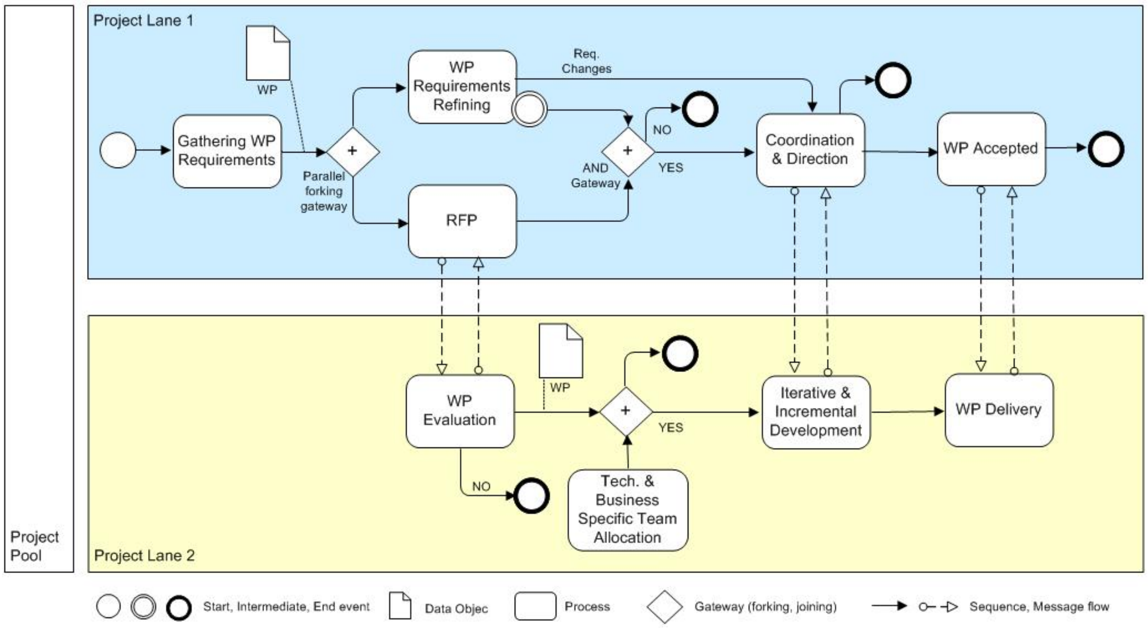

Figure 1. Workflow model for software development with a virtual team

5. Measurements

Literature shows that few measures have successfully survived the initial definition phase and are actually used in industry [Briand L. C. *et all*] due to debatable theoretical and empirical validity. The reasons are:

- Measures are not always defined in the context of some explicit and well-defined measurement goal
- The experimental hypotheses are often not made explicit (e.g., what do you expect to learn from the analysis)
- Measurement definitions do not always take into account the environment or context in which they will be applied
- A reasonable theoretical validation of the measure is often not possible because the attribute that a measure aims to quantify is often not well defined
- A large number of measures have never been subject to an empirical validation

Coming back to our example, we were preoccupied to keep the measurement process simple and relevant and we applied the following criteria for data collection [McGarry J *at all*, 2001, Florac *at all*, 1997] and used some of the measurements proposed by [Radoiu D. and Vajda A., 2004]:

- Data must be relevant for the interaction between parties
- Data must cover the interaction between parties

- Data should be validated by both parties
- Collection process should be economic (i.e. maximum benefits/minimum effort)
- Parties must agree on the interpretation of measurements (before the collection)

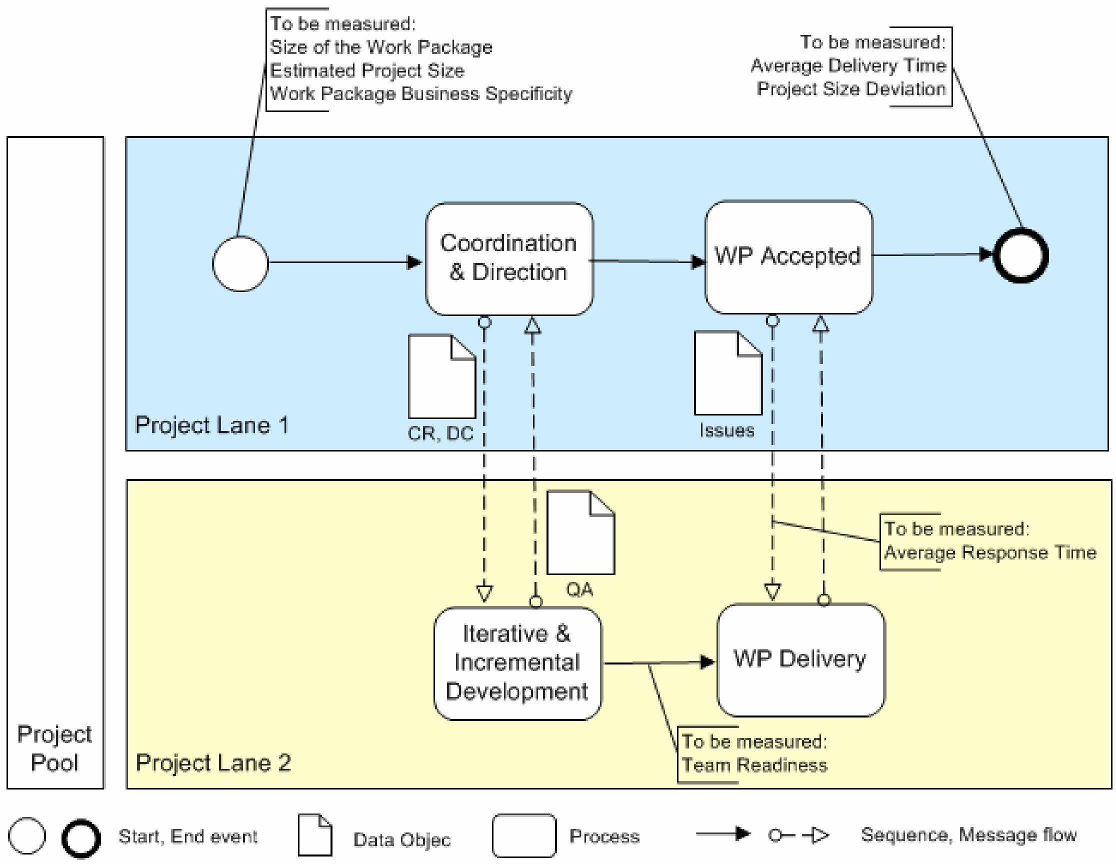

Figure 2. Data collection on the workflow model

5.1 Authorizing and Accepting a WP measurements

The process is initiated by the Team 1 which authorizes a WP and proposes it to the Team 2. A WP could be described in various ways, e.g. WP size (based on Functional Points) and WP Accuracy (WPA, a quantitative estimation of how well the assignment is described through analysis documents, design documents, acceptance criteria).

Communication (e.g. questions and answers, QA) plays a very important role in clarifying WP accuracy related issues. QA could be quantified simply by their number, using the log files and strict counting criteria based on relevance to project. One could argue that the amount of communication tells nothing about the state of the project. This is correct, but in a virtual team communication could reveal in the initial stage work package accuracy and in the following stage work package stability affected by change requests. This is evident when a project starts without good requirements analysis and the problems are transferred to the second team, during the project, via change requests.

But regardless of their number it is important that questions are answered in a reasonable time (quantified by average response time, ART) and answers are specific and complete (quantified by communication quality, CQ).

Based on agreed estimating models (e.g. historical data) the parties could quantify measurements either quantitatively (numerical values, e.g. on a scale from 1 to 5, 1 poor, 5 very good) or qualitatively (e.g. color code levels). For instance, using the largely accepted color code, communication quality could be quantified as good ("Green", score 4-5), average ("Amber", score 3) or poor ("Red", score 1-2). Similarly, WPA could be also described using the same color code or quantitatively by:

WPA = WP/QA

Informal discussions with project managers revealed they felt a difference in the way a virtual team is performing, depending on the work package business specificity (e.g. a project to build a website vs. a project to build a trading application for an exchange) and team experience in the specific field.

To characterize the specific knowledge in the business area of the project we ranked the work package business specificity (WPBS) on a scale from 1 to 5, 1 meaning no specificity at all and 5 representing the highest specificity. For instance, a project involving the development of a trading application for capital markets is considered highly business specific WPBS=5

Similarly, team specific experience (TSE) in the field was quantified on a scale from 1 to 5. Desirably is to assign to a highly business specific project a highly experienced team in that field. As it happens this is not always possible and we decided to characterize team adequacy to the job by team readiness (TR).

Team readiness (TR) is computed as the ration between team experience and project business specificity.

5.2 Executing a WP and Assessing Progress Measurements

This phase is initiated by the Team 2 by accepting a WP and acknowledging the acceptance criteria for the executed WP. Let us call the specific WP accuracy at the moment of awarding as WP definition level.

Although the development process starts by executing the WP at the definition level, frequently the initial request is changed via formal change requests, CR. If accepted by the Team 2, CR also modifies acceptance criteria. Let us consider a first approximation where the WP stability is also a measurement of acceptance criteria stability.

WP Stability (WPS) is a quantitative estimation of change requests – CR - per functional point after the WP has awarded by Team 1 and accepted by Team 2:

WPS = WP/CR

Based on statistics, WPS associated risk could be also color coded.

Ideal stability means no change requests are made after the work package has been accepted by the Team 2.

The lack of face-to-face interaction in virtual teams may make obstacles to effective coordination and thus impair team effectiveness. As there was no data available (e.g. communication logs) and no consistent way to interpret it we quantified communication quality (CQ) on a scale from 1 to 5, 1 poor, 5 very efficient.

5.3 Delivering and Receiving a WP Measurements

As in the previous stages, the focus is on virtual team co-operation issues; therefore measurements will not focus on WP quality, amount of rework, etc., these being well covered by literature.

Average delivery time, defined as:

ADT = delivery time/WP

Could be used to assess delays introduced by different interference factors like change requests, communication quality, corrections and directions.

Measurements	Acronym	Value	Tracked Process/Project Issues
WP size	WP	FP	Product complexity, growth
Estimated effort	EE	Man Day MD	Effort
Required effort	RE	Man Day MD	Effort
Questions and answers	QA	number	Analysis and design quality
WP accuracy	WPA	WPA = WP/QA	Quality of requirements gathering, design quality
WP Business Specificity	WPBS	1 to 5, 1 no specificity, 5 highly specific	Project complexity
Team Specific Experience	TSE	1 to 5, 1 no experience in the specific field, 5 extensive experience in the specific field	Team adequacy for the task
Team Readiness	TR	TR=TSE/WPBS	Team adequacy for the task
Change Requests	CR	number	Quality of requirements gathering
WP stability	WPS	WPS = WP/CR	Sourcing process efficiency, productivity, rework, schedule delays
Communication quality	CQ	1 to 5, 1 poor, 5 excellent	Communication
Early Error Detection	EED		Quality
Average response time	ART	$\text{ART} = \frac{\sum \text{response time}}{\text{QA}}$	Communication
Average Delivery Time	ADT	ADT = delivery time/WP	Productivity

Table 1. Measurements

6. Analysis of the Data Samples

Data was selected to exhibit the following characteristics:

- no reasonable doubt as to data validity;
- virtual teams are made of two parties belonging to different organizations;
- parties involved in the virtual team have:
  - no language or cultural barriers
  - no co-ordination problems
  - no technology infrastructure problems like bandwidth availability or connection reliability
- synchronous/asynchronous communication technologies are available and intensively used;
- parties in the same project pool operate at the same quality standards (ISO 9000 or CMMI);
- good software development processes are agreed and in place for all team parties (e.g. RUP);
- all projects for the same platform.

A data set representing 17 projects involving five organizations meet the above criteria. We plotted work package size vs. effort to see if there is a difference between inside projects and virtual team conducted projects in this respect (Figure 3).

Figure 3 shows a large degree of scatter for large size projects. We represented the data in a logarithmic scale to see if the values are consistent with empirical findings with regard to effort-duration relationship. Figure 4 shows a scatter reduction and a more normal distribution.

The two graphs support the credibility of the selected data showing a homogenous data set.

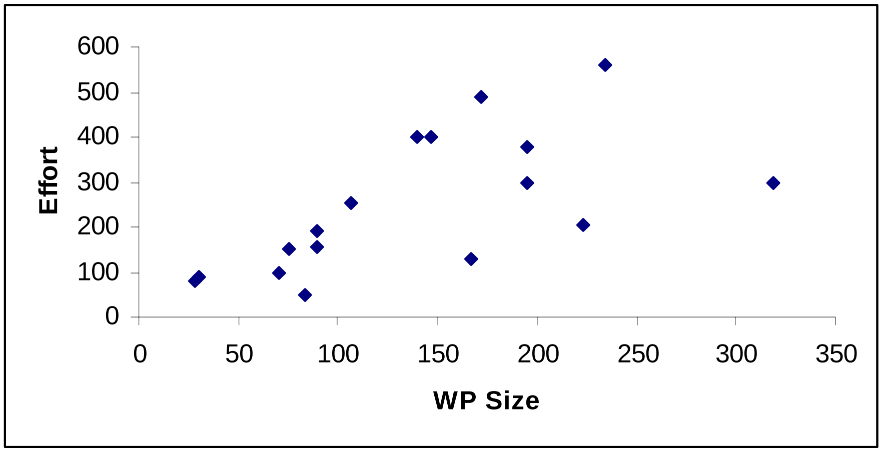

Figure 3. Scattered plot of project effort vs. work package size

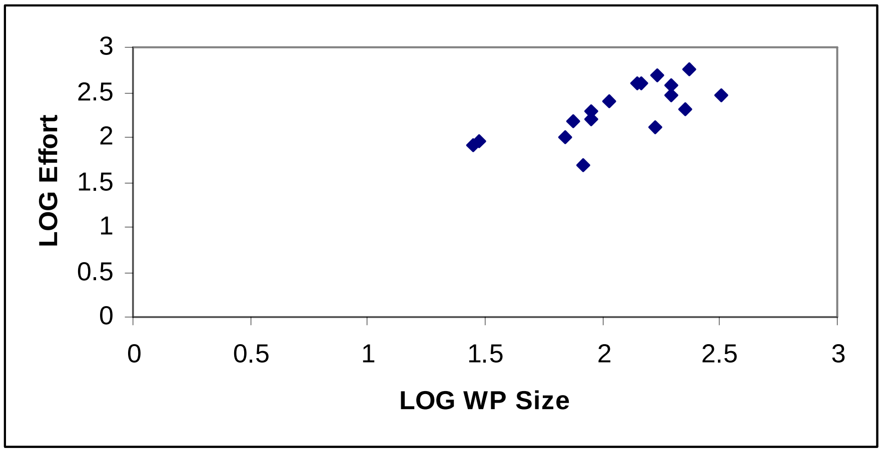

Figure 4. Scattered Plot of Log (project effort) vs. Log (work package size)

Through informal discussions we asked project managers to identify the virtual team organizational characteristics which have most impact on the team effectiveness. Discussions have revealed that, based on experience over a significant large number of projects, they come to the conclusion that some variables act like "centrifugal forces" in the project and strongly affect the effectiveness of the virtual team. In their opinion, the variables could be considered predictors of the effectiveness of virtual team, In the order of their decreasing impact they are:

a) Communication
b) Work package business specificity and team readiness
c) Work package stability

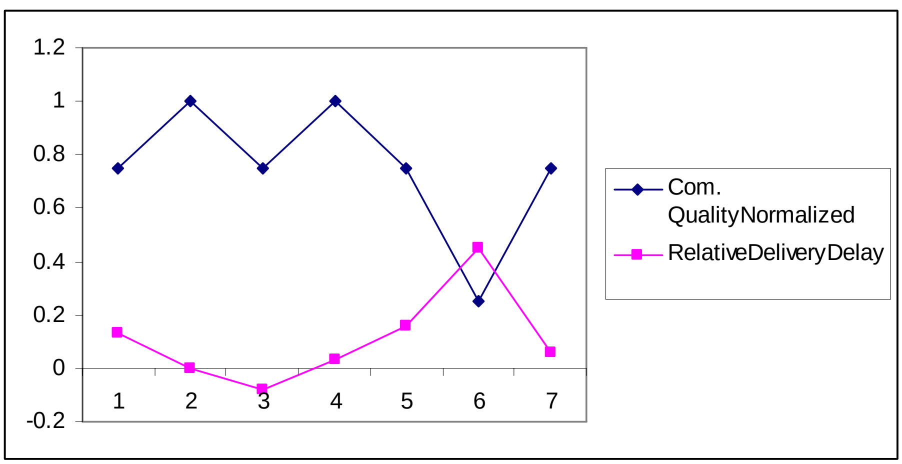

Figure 5. Communication quality and relative delivery delay

Figure 5 plots the data from a single virtual team and represents delivery relative delay and communication quality over a number of seven projects. Delivery relative delay was calculated as the ration between the variations of the project duration ratio (actual duration minus estimated duration) and estimated duration. Communication quality was assessed on a scale from 1 to 5 based on a set of criteria (e.g. average response time, communication effectiveness) and later normalized. Despite the size of the data set, apart from inherent noise we can see there is a relationship between the drop in communication quality in the virtual team and the increase of the duration of the project.

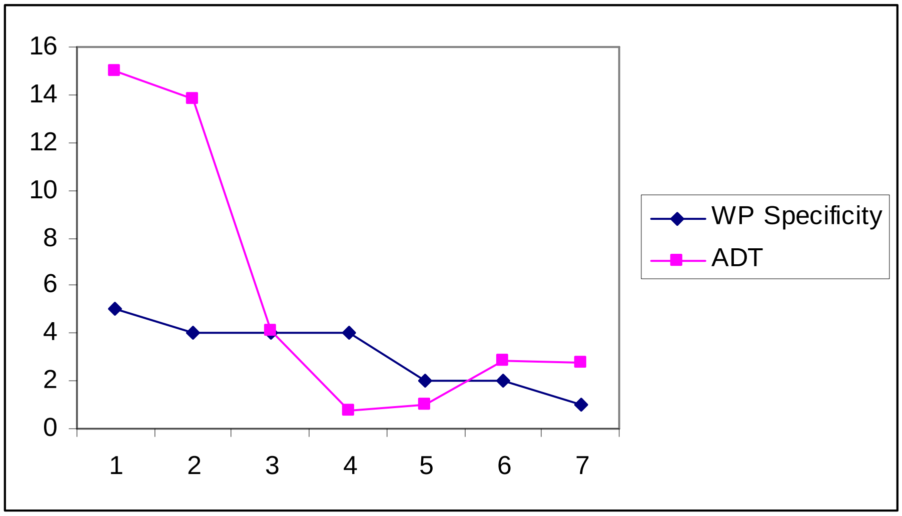

Figure 6. Work package business specificity and average delivery time

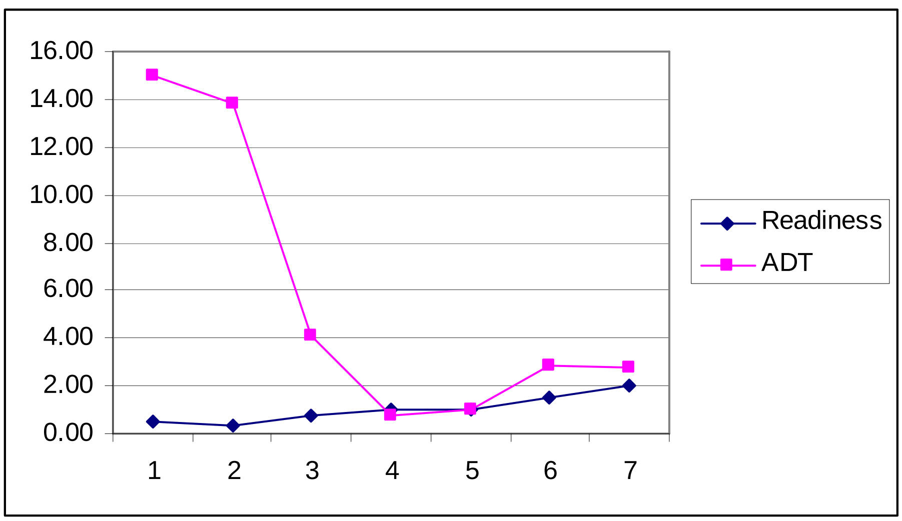

Figure 7. Team readiness and average delivery time

Figure 6 plots in parallel WP business specificity and average delivery time (for a functional point) for the same seven projects.

Figure 7 represent team readiness and average delivery time for the same data set. Both graphs also support the idea of this paper that even an empirical approach could bring valuable information to improve virtual team performance.

Quantitative measurements bring more insight into the overall process. For instance, based on historical data one can infer that a certain CR average score indicates that up to certain percentage of the requirements were not known when the WP was awarded.

7. Data Analysis Using Fuzzy Logic Methods

The theory of fuzzy sets was introduced in 1965 by Lotfi A. Zadeh [Zadeh, 1965] as a natural generalization of the classical set concept. Let X be a data set, composed of n data items characterized by the values of s characteristics. A fuzzy set on X is a mapping $A : X \rightarrow [0, 1]$. The value $A(x)$ represents the membership degree of the data item x from X to the class A. The advantage of this approach is that it allows a data item x to be a member of more classes, with different membership degrees, according to certain similarity criteria.

Principal Components Analysis (PCA) is designed to reduce the number of variables that need to be considered to a small number of axes called the principal components, that are linear combinations of the original variables. The new axes lie along the directions of maximum variance thus containing most of the information. PCA provides an objective way of finding attributes of this type so that the variation in the data can be accounted for as concisely as possible. Moreover, due to this space rotation, PCA is often used as a dimensionality reduction method: very few principal components provide a good coverage of all the original variables. Fuzzy PCA [Pop, 2001] is an essential improvement of the traditional PCA in that it deals with each data item according to its fuzzy membership degree to the set being analysed. As such, it is best suited to deal with outliers and heteroskedasticity in data sets.

FPCA has been applied on the set of project measurements in order to produce a visual projection of the data set onto the two largest data spreading directions. Table 2 describes the reduction coefficients produced by considering the set of student projects, and Table 3 and Figure 8 present the scores corresponding to the first two fuzzy principal components.

No.	Eigenvalue	Successive diff.	Proportion	Cummulative prop.
FPC1	42094.5	39409.9	0.915362	0.915362
FPC2	2684.61	1688.56	0.0583779	0.97374
FPC3	996.046	866.684	0.0216594	0.9954
FPC4	129.362	47.7513	0.00281304	0.998213
FPC5	81.6109	81.0363	0.00177466	0.999988
FPC6	0.574568	0.574568	1.24942e-005	1

Table 2. Reduction coeffcients for the set of student projects
(the remaining eigenvalues have proportions very close to zero, and, thus, are less important)

Item	FPC1	FPC2
1	87.6026	21.9068
2	605.594	-52.6234
3	171.066	-66.1984
4	412.451	200.461
5	75.2339	11.903
6	436.347	113.641
7	432.703	49.5847

Table 3. The scores corresponding to FPC1 and FPC2 for the set of student projects

From an analysis of Figure 8 we remark two groups that separate very visibly from the main group of data items: the group formed with projects 1, 5 and 3, and the group formed with projects 4, 6 and 7. As well, project 2 appears as isolated item.

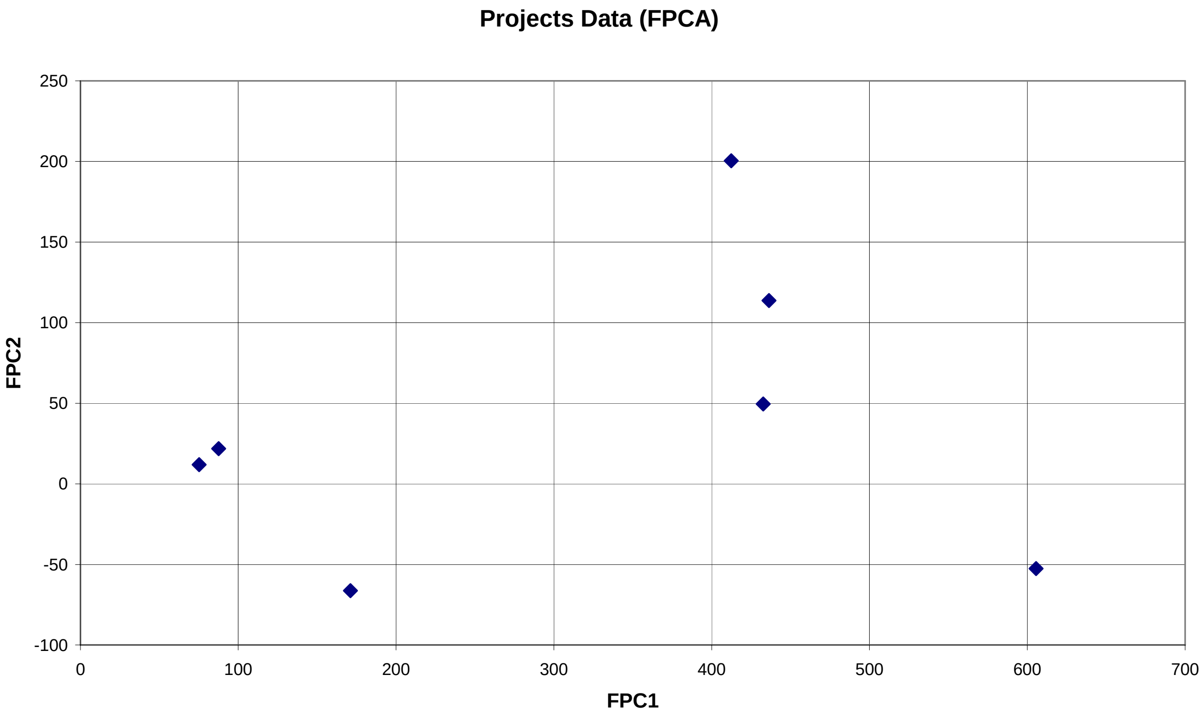

Figure 8. Representation of scores corresponding to FPC1 and FPC2 for the set of student projects

Clustering algorithms based on fuzzy sets have proved their superiority due to their ability to deal with imprecise sets, imprecisely-defined boundaries, isolated points, and other delicate situations. The class of fuzzy clustering algorithms based on fuzzy objective functions [Bezdek, 1981] provides a large share of geometrical prototypes and combinations thereof, to be used according to the data substructure. On the other hand, the **Fuzzy Divisive Hierarchical Clustering** scheme [Dumitreswcu, 1988 and Pop, 1996] provides an in-depth analysis of the data set, by deciding on the optimal subcluster cardinality and the optimal cluster substructure of the data set.

The visual representations in Figure 8 enable us to further analyse the set of projects. The final binary partition produced by FDHC for the set of student projects, using a polarization index of 0.3, are depicted in Table 4. The corresponding fuzzy membership degrees to the classes from the final fuzzy partition are displayed in Table 5. We have preferred a polarization index of 0.3 in order to favorise the obtaining of a more grouped cluster substructure.

Class	**Members**
1.1.1.	1
1.1.2.	5
1.2.	3
2.1.1.1.	7
2.1.1.2.	6
2.1.2.	2
2.2.	4

Table 4. Final fuzzy partition for the set of student projects, with a polarization threshold of 0.3

Prj	1.1.1	1.1.2	1.2	2.1.1.1	2.1.1.2	2.1.2	2.2
1	**0.91157800**	0.00166286	0.05326530	0.00763289	0.00691524	0.00673944	0.01220580
2	0.02728150	0.02619290	0.08020940	0.00000457	0.00000322	**0.77269400**	0.09361400
3	0.00658360	0.00935089	**0.92200900**	0.01535900	0.01251910	0.01336200	0.02081620
4	0.06474440	0.07104880	0.17084200	0.00017638	0.00022056	0.00022410	**0.69274400**
5	0.00212347	**0.92116800**	0.06016290	0.00354705	0.00324658	0.00317903	0.00657297
6	0.00981140	0.00937825	0.02705340	0.00000128	**0.86863700**	0.02393710	0.06118110
7	0.01080040	0.01021570	0.03307930	**0.89768200**	0.00000196	0.03060810	0.01761280

Table 5. Fuzzy membership degrees to the final partition for the set of student projects (boldfaces indicate the membership degree to the major defuzzified class)

By analysing the Table 5, we recognize the same grouping as with FPCA: the projects 1, 3 and 5 have been separated as class 1., and th projects 2, 4, 6, 7 have been separated as class 2. We remark here, on one side, that membership degrees of projects from class 1. to their individual classes are higher that those of projects from class 2. to their individual classes. This shows a more homogeneous class 1. On the other side, as opposed to the FPCA projection, which shows the project 2 as an isolated item, Table 4 shows that project 4 has been separated from the group of three projects (2, 6, 7) as a separate class (2.2.).

The collected data shows two opposite situations: work package for project number 2 got the highest business specificity score (5) and was assigned to a very experienced group in that specific field (score=5). The result was that the group worked almost independently in the virtual team showing a very reduced dependence on communication. Work package for project number 4 was ranked a relative low business specificity (2) and was assigned to a group scoring 2 on experience in the specific field. The project was characterize by a strong interaction between the virtual team groups, a large number of questions and answers, directions and corrections. We need to further analyse the individualities of projects 2 and 4 as compared to the other projects from their group as well as the characteristics of the two classes.

8. Conclusions

The aim of the paper was to present an empirical approach to analyze small virtual team software development process using the workflow model and simple measurements.

The research was limited to virtual teams with members from two organizations but we are aware that multi-organizational virtual teams are more complex and therefore require more attention.

A weak point of the research is that the workflow model we used is very specific and collected data to check the validity of our empirical approach is limited. For many companies there is often a much more complex process before the initial "authorizing a work package" stage let alone the multi-sourcing case.

Yet, the paper is not aiming to provide a generic answer or to define new measures but to show that in the absence of a formal model we could use an empirical approach to make better decisions.

We think that the WP measurements require a tighter definition to generate better statistics. Given the description "how well the assignment is described through documents" WP Accuracy can only really be a qualitative measure.

Also, WPS assumes that each CR is approximately of the same size - what if the CR affects multiple FP's? The number of CR's does not tell if they are lots of small changes to the system or lots of major changes. This is also true of function point analysis as a whole. Unless you can create a common sizing metric the figures can be skewed.

References

Bezdek J.C., *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.

Duarte, D.L. and Snyder, N.T., *Mastering Virtual Teams*, San Francisco: Jossey-Bass, 1999.

Dumitrescu D., *Hierarchical pattern classification*, Fuzzy Sets and Systems 28 (1988), 145-162.

Florac, Park, and Carleton, *Practical Software Measurement: Measuring for Process Management and Improvement*, CMU/SEI-97-HB-003, 1997

Fulk, J., and DeSanctis, G. "Electronic Communication and Changing Organizational Forms," *Organization Science* (6), 1995, pp. 337-349.

Henderson-Sellers, B., Process Metamodelling and Process Construction: Example Using the OPEN Process Framework (OPF), *Annals of Software Engineering*, Kluwer Academic Publishers (14), 2002, pp. 341-362.

Jarvenpaa, S., and Ives, B. "The Global Network Organization of the Future: Information Management Opportunities and Challenges," *Journal of Management Information Systems* (10), 1994, pp. 25-57.

Lipnack J., Stamps J., *Virtual teams*. New York: Wiley, 1997.

Briand L.C., Morasca S., Basili V.R., *An Operational Process for Goal Driven Definition of Measures*, http://www.iese.fhg.de/ISERN/pub/isern_biblio_tech.html

McGarry J., Card D., Jones C., Layman B., Elizabeth Clark, Dean J., Hall F., *Practical Software and Systems Measurement: A Foundation for Objective Project Management*, Guidebook, Addison Wesley Professional, October 2001, http://www.psmsc.com/PSMBook.asp

Pop H.F., *SAADI: Software for fuzzy clustering and related fields*, Studia Univ. Babes-Bolyai, Informatica 41 (1), 1996, pp. 69-80.

Pop H.F., *Principal Components Analysis Based on a Fuzzy Sets Approach*, Studia Univ. Babes-Bolyai, Informatica, 46 (2), 2001, pp. 45-52.

Radoiu D., Vajda A., Process-Oriented Metrics for Application Development Outsourcing (A Practitioner's Approach), *Studia Univ. Babes Bolyai, Informatica*, 2 (49), 2004, pp. 65-74

Zadeh L.A., *Fuzzy sets*, Information and Control 8 (1965), pp. 338-353.