

EXHIBIT A

Expert Rebuttal Report Of David P. Garmus

I. Introduction And Summary Of Opinions

As fully documented in this report, I am a universally recognized expert in Function Point Analysis (FPA) and software measurement. Having reviewed the document entitled “Expert Report of Paul C. Pinto” (Pinto Report), I refute any contention that Mr. Pinto applied any recognized form or version of FPA in his report. There is nothing in the Pinto Report that indicated to me that Mr. Pinto understood how Function Points are to be determined, or how they are to be utilized. Mr. Pinto’s “Estimating Approach” and his “Ten-Step Analysis To Determine The Cost Of Development Using Function Point Analysis” did not analyze anything using FPA. That “Approach” and that “Analysis” are not recognized steps in any FPA. Consequently, any Function Point value that he achieved through his “Ten-Step Analysis” is totally contrived and without any cognizable basis. The Function Point values he reported are completely fabricated, and his analysis and process would not be recognized by any expert in the field as being a legitimate means of valuing or assessing anything using FPA.

In the Pinto Report, Mr. Pinto purported to estimate “what it would have cost [the Defendants] to independently develop certain software applications.”¹ If – for the sake of argument – the Defendants actually were to independently develop all four of the Oracle suites of products recited in the Pinto Report to support TomorrowNow’s (TN) customers (as proposed by Mr. Pinto), the newly developed software essentially would have to be an exact replica of the four, recited Oracle suites of products (especially for the purpose of providing most tax updates, bug fixes, etc.). The probability that a software-development project as proposed by Mr. Pinto would result in the creation of four exact replicas of the four Oracle suites of products is essentially zero (i.e., it is essentially impossible). In my opinion, determining the cost for independently developing the four underlying application suites is not appropriate for the case in question.

Further, in my expert opinion, Mr. Pinto should not have attempted to place a value on entire suites of products as TN utilized only a limited percentage of the applications contained in those suites of products.

Mr. Pinto’s derivation of productive hours of effort and his estimated development cost are erroneous and baseless, as documented in this report.

¹ See Pinto Report, page 1.

C. Compensation

My agreed-upon compensation in this litigation is \$300.00/hour. My compensation is not in any way contingent upon the results of my analysis.

D. Prior Testimony

A list of the cases in which I previously have provided testimony as an expert witness is attached as Appendix C.

E. Material Considered

- Pinto Report and Appendices
- PeopleBooks located on TomorrowNow's BU01 Servers: TN-OR02989997, TN (Hard Drive).33 at BU01_G\JDE 36-44\Generic PeopleBooks – JDE; TN-OR02989993, TN (Hard Drive).29 at U01_G\BU01_LOGICAL_IMAGE_43-76\PeopleSoft Enterprise Documentation
- Plaintiff's Fifth Amended and Seventh Supplemental Responses and Objections to Defendant Tomorrow Now, Inc.'s Interrogatory No. 13 (Responses to Interrogatory 13)
- TN-OR06515453
- TN-OR06515454
- TN-OR06515455
- TN-OR07717977
- Appendix L to the Expert Report of Stephen K. Clarke (Clarke Appendix L)
- IFPUG CPM, Releases 4.2, 4.2.1, and 4.3, for each of which I was a principal editor
- IFPUG Website (<http://www.ifpug.org/>) and miscellaneous materials available on the Website (e.g., <http://www.ifpug.org/discus/messages/1751/3800.html>; and www.ifpug.org/discus/messages/1751/4699.html)
- David Consulting Group White Papers, Studies, etc. (attached as Appendices E and F)
- Materials prepared or provided by others, as specified in this report
- Sylvan VI Corporate Fact Sheet

III. My Rebuttal Report

In the following report, I am rebutting the arguments and claims made by Mr. Pinto and his approach taken in deriving a value of the four suites of products referenced in the Pinto Report. In particular, I disagree with what Mr. Pinto called his Function Point Analysis for the reasons identified in this report.

2-3	Low	Avg	High
4+	Avg	High	High

* an EQ must have at least 1 FTR

Note: DETs are equivalent to non-repeated fields or attributes.
RETs are equivalent to mandatory or optional sub-groups.
FTRs are equivalent to ILFs or EIFs referenced by that transaction

The final Function Point calculation yields a single number that represents the total amount of functionality being delivered. Once completed, the Function Point size of an application or a new development project can be communicated in a variety of ways. As a stand-alone value, the Function Point size of a system tells us the size of the overall software deliverable. This could be reflected for a previously developed application or for a development or enhancement project. During the years 2003 to 2007, costs were often estimated for development projects at a value of \$400 to \$1,200 per Function Point; costs varied significantly based upon the development location, which is why outsourcing and off-shore development were so popular then and remain so today.

E. Mr. Pinto's Analysis Of JD Edwards World And Siebel

Mr. Pinto claims that he was able to extrapolate estimates for JD Edwards World and Siebel suites of products from the JD Edwards EnterpriseOne and PeopleSoft estimates, but he does not provide evidence of his methodology for doing so accurately. As I have already discussed, he did not apply FPA to size any application, so he should not have extrapolated from his faulty FPA-derived estimates. Mr. Pinto then purports to provide estimates for all four suites of products using what he calls a COCOMO analysis. Donald J. Reifer, another software-estimation expert engaged by the Defendants, is a recognized COCOMO expert, and, in his expert report, Mr. Reifer comments on Mr. Pinto's COCOMO analysis. I do note that neither Mr. Pinto's curriculum vitae nor the public Sylvan VI Corporate Fact Sheet found on the Internet refers to any experience with COCOMO.

F. Mr. Pinto Inappropriately Sized Applications Not Maintained By TN

Although Mr. Pinto clearly states that cumulative development costs would amount to double counting, his estimations derived from both his so-called FPA and COCOMO analysis significantly over-valued Source Lines of Code carried over from previous versions of each application as well as reused code.⁹ He also sized applications and components within applications that were not utilized by TN in the course of TN's business. This was true for substantial numbers of applications included in Mr. Pinto's analysis; apparently, he did not attempt to eliminate the significant over-counting of Source Lines of Code. TN provided maintenance services for only a limited number of the many applications comprising the four suites of products sized / valued by Mr. Pinto.

⁹ With respect to COCOMO issues, see Expert Report of Donald J. Reifer.

Moreover, in my experience, most users of the PeopleSoft, JD Edwards, and Siebel software do not utilize much of the functionality included in the applications.

I determined by reviewing the spreadsheets of TN's maintenance activities¹⁰ that many of the applications (sometimes referred to as modules) identified in Plaintiff's Fifth Amended and Seventh Supplemental Responses and Objections to Defendant Tomorrow Now, Inc.'s Interrogatory No. 13 as modules contained within a sample of the suites of products allegedly infringed by TN, were in fact not being used by TN, including:

PeopleSoft HRMS:

- Global Payroll for Brazil
- Global Payroll for France
- Global Payroll for Germany
- Global Payroll for India
- Global Payroll for Italy
- Global Payroll for Spain
- Global Payroll for Switzerland
- Global Payroll for The Netherlands

PeopleSoft CRM:

- Analytics - Customer Value
- Analytics - Smart Views
- Call Center
- CRM for Communications
- CRM for Energy
- CRM for Financial Services
- CRM for Government
- CRM for High Technology
- CRM for Insurance
- Data Transformer
- HelpDesk for Employee Self Service
- Configurator
- FieldService
- Multichannel Interactions
- Telemarketing

PeopleSoft Financials, Distribution & Manufacturing and/or Financials & Supply Chain Management:

- Application Fundamentals
- Collaborative Supply Management
- Customer Portal
- Deduction Management
- Engineering
- Enterprise Service Automation
- eRFQ

¹⁰ See TN-OR06515453, TN-OR06515454, Clarke Appendix L, and Responses to Interrogatory 13.

- eStore
- Inventory Policy Planning
- MarketPay
- Mobile Order Management
- Remote Order Entry
- Order Promising
- Pay/Bill Management
- Promotions Management
- Remote Order Entry
- Supplier Portal
- Bank Setup and Processing
- Databridge
- eBid Payment
- Global Options and Reports
- Grants
- Mobile Time and Expense for Palm

PeopleSoft Enterprise Performance Management:

- Activity-Based Management
- CFO and Governmental Portal Solution Key Performance Indicators
- Customer Behavior Modeling
- Customer Scorecard
- ESA Warehouse
- Funds Transfer Pricing
- Inventory Policy Planning
- Investor Portal
- Investor Portal Pack
- Profitability Management for the Financial Services Industry
- Project Portfolio Management
- Risk-Weighted Capital
- Sales Incentive Management
- Sales Incentive Management for High-Tech & Industrial
- Supplier Rating System
- Supply Chain Warehouse

PeopleSoft Student Administration Solutions:

- Community Access
- Community Directory
- Contributor Relations
- Financial Aid
- Gradebook
- Involvement
- Learner Services
- Personal Portfolio
- Student Administration Portal Pack

EnterpriseOne 8.12 ALM:

Adv Real Estate Forecasting
Condition-base Maintenance
Commercial Property Management
Equipment Cost Analysis
Resource Assignments

EnterpriseOne 8.12 CRM:

Call Management
CRM Call Scripting
CRM Case Management
CRM Integrated Solution
CRM Sales Force Automation
CRM Solution Advisor
Extensity Integration
JDE E-Procure Intgr for Ariba
JDE Integrators for Siebel
M&D Complementary Product
Mobile Sales
Promotions Management
Siebel - Consumer Goods

EnterpriseOne 8.12 Financial Management:

Argentina
Asia Pacific Localization
Austria
Belgium
Brazil
Chile
Colombia
Conversion Programs
Czech Republic
Denmark
DREAMwriter Conversion Tool
Ecuador
Electronic Mall
EMEA Localization
Enhanced Accounts Receivable
Expense Reimbursement
FASTR Conversion Tool-Columnar
Finance Report Writer - FASTR
Finland
France
General Back Office
Germany
Greece
Hungary
India

Ireland
Italy
Japan
Latin America Localization
M&D Complementary Product
Mexico
Multi-National Products
Netherlands
Norway
Peru
Plant Manager Dashboard
Poland
Portugal
Rapid Start
Russian Federation
Singapore
South Korea
Spain
Sweden
Switzerland
Taiwan
Turkey
United Kingdom
Venezuela
World Writer Conversion Tool

EnterpriseOne 8.12 HCM:

ADP Integrator
HCM Learning Management Integr
New Zealand
Old Payroll
OneWorld HR Canadian
Payroll SUI

EnterpriseOne 8.12 Project Management:

Advanced Order Configurator
Blend Management
Distribution Contracts
Government Contracting
Homebuilder Management
Promotions Management

EnterpriseOne 8.12 MFG & SCM:

DRP/MPS/MRP
M&D Complementary Product
Plant Manager Dashboard

EnterpriseOne 8.12 SM:

- Grower Pricing & Payments
- Grower Management
- Operational Sourcing
- Purchase Order Receipts/Routing
- Voucher Match and Landed Cost¹¹

G. Mr. Pinto's Inappropriate Claim Of Using FPA

In Section IV of his report, Mr. Pinto states that he analyzed the cost of development for the following “products” using FPA:

- JD Edwards EnterpriseOne, Version 8.12,
- PeopleSoft 8.8 Customer Resource Management (“CRM”),
- PeopleSoft 8.8 Human Resources Management System (“HRMS”),
- PeopleSoft 8.4 Financial Supply Chain Management - rev 1 (“FSCM”),
- PeopleSoft 8.0 Student Administration (“Student Admin”), and
- PeopleSoft 8.8 Enterprise Performance Management - rev 1 (“EPM”)

It is important to recognize first that Mr. Pinto did not analyze anything using FPA, including the suites of products listed above. Consequently, any Function Point value he reported is completely fabricated and without substance.

Second, he should not have attempted to place a value on entire suites of products when TN utilized only a limited percentage of the applications contained in those suites of products.

It is certainly possible to count any of the applications contained in the suites of products listed above or to calculate smaller components of those applications involved in adaptive or corrective maintenance. Mr. Pinto did not properly count any of the applications. In order to demonstrate how a FPA actually is performed, I calculated a Function Point count of Accounts Payable by reviewing the EnterpriseOne 8.0 Accounts Payable PeopleBook and a Function Point count of Payroll by reviewing the PeopleSoft Enterprise Global Payroll for United States 8.9 PeopleBook.¹² If there were a reason to

¹¹ I understand that, in addition, TN also was not using various modules comprising the JD Edwards World and Siebel suites of products, such as Agent Portal, eAdvisor, Release Manager, SOA Enablement, Homebuilder Management, Distributed Data Processing. See TN-OR06515455, TN-OR07717977, Clarke Appendix L and Responses to Interrogatory 13.

¹² See JDE EnterpriseOne 8.0 Accounts Payable PeopleBook Vol. 1, SKU REL8EAP0502V1 May 2002 (TN-OR02989997, TN (Hard Drive).33 at BU01_G\JDE 36-44\Generic PeopleBooks - JDE\OneWorld\8.0\Financial Management\Accounts Payable\EnterpriseOne 8.0 Accounts Payable PeopleBooks\Accounts Payable V1.pdf); JDE EnterpriseOne 8.0 Accounts Payable PeopleBook Vol. 2, SKU REL8EAP0502V2 May 2002 (TN-OR02989997, TN (Hard Drive).33 at BU01_G\JDE 36-44\Generic PeopleBooks - JDE\OneWorld\8.0\Financial Management\Accounts Payable\EnterpriseOne 8.0 Accounts Payable PeopleBooks\Accounts Payable V2.pdf); and PeopleSoft Enterprise Global Payroll for United States 8.9 PeopleBook, SKU HRCS89MP1GPT-B 0405 April 2005 (TN-OR02989993, TN (Hard Drive).29 at BU01_G\BU01_LOGICAL_IMAGE_43-76\PeopleSoft Enterprise Documentation\PeopleSoft

For purposes of demonstrating that a proper FPA could have been conducted with respect to the individual applications that comprise the four suites of products, I completed Application Function Point counts of Accounts Payable and Global Payroll for United States (EnterpriseOne 8.0).

I, as a recognized expert in the area in software measurement and estimation and particularly in FPA, have never encountered his “Ten-Step Analysis To Determine The Cost Of Development Using Function Point Analysis.” I certainly do not agree in any way with its use in determining the cost of development. I also do not agree that the cost of developing entire similar suites of products is appropriate in a situation where an organization is not selling or developing a competing suite of products.

A. Mr. Pinto Improperly Applied FPA

IFPUG does not agree with Mr. Pinto’s Step One: Identify And Group Source Code Components. Mr. Pinto did not apply FPA and incorrectly defined FPA, particularly as it applies to components and Source Code. Instead, Mr. Pinto used backfiring. IFPUG does not concur with the use of backfiring to extrapolate Function Points from underlying Source Code. I also understand that Mr. Reifer addresses in his expert report the issue of whether Mr. Pinto’s Source Lines of Code counts are correct. Irrespective of whether they are incorrect (something I leave to Mr. Reifer to discuss), they in any event should not have been used as the basis for backfiring Function Points. IFPUG also does not agree with sizing based upon individual programs or modules; the scope of a Function Point count is based upon analysis of functional aspects of an application or a component of the application. An application boundary is the border between the application being measured and either external applications or the user domain. IFPUG has defined specific rules for identifying boundaries, and they have never included the grouping proposed by Mr. Pinto.

B. IFPUG Membership Does Not Agree With Mr. Pinto’s Approach

It is not only IFPUG itself that does not agree with Mr. Pinto’s approach. I have retrieved a number of comments below extracted from the IFPUG Website by various members and non-members related to extrapolating Function Points from underlying Source Code. These are not official positions, but they are simply comments submitted by individuals to the IFPUG Bulletin Board on the IFPUG Website:

- Posted on Wednesday, June 13, 2001 - 02:03 pm:

You probably will not get anybody talking about the benefits of backfiring on this board! The position of the software metrics profession is that it simply does not work. Those tables of backfire coefficients that were published about twenty years ago were meant for very high level statistical discussions, never for simulating a FP count on a single application.

The pitfalls of backfiring are that it has a standard deviation of about 1000%. Just think about assigning a project to a new developer fresh out of programming

school, then giving the identical project to a highly experienced programmer as a benchmark. The beginner will use about ten times as many lines of code as the expert.

Couple that with the fact that the backfire coefficients themselves were never derived very accurately. The figure generally used for Cobol is approximately 100 LOC = 1 FP. Backfire proponents (there are still a few of them around) have admitted that the average has turned out to be more like 300 LOC. And they never talk about the standard deviation in public!

I believe you will receive very few replies that disagree with these points. Function Points were developed so we no longer have to try to derive metrics from lines of code, not so we can glorify LOC and keep using them. LOC based metrics do not work, no matter how they are presented.

- Posted on Sunday, June 17, 2001 - 10:02 am:

I was recently part of a team whose task was to count a system that was believed to be 60,000 function points according to a backfire-based estimate provided by a vendor.

After counting the system was found to be slightly less than 10,000 function points.

- Posted on Wednesday, January 21, 2009 - 10:11 am:

I think this is a perfect example of why backfiring is a dangerous way to derive function points and should be done with care and strong caveats. No two organizations are going to have the same results. In fact I have seen extreme differences within the same development organization (4 to 1 difference in SLOC per FP developing similar applications in the same language). It really comes down to some of the difficulties of using SLOC as a sizing measure. For a more in-depth analysis, check out this article in Crosstalk Magazine:

<http://www.stsc.hill.af.mil/crosstalk/2005/04/0504schofield.html>

The conclusion of this article included the following:

"The purpose of this article is clear: Statistically significant variation in LOC counts render those counts undesirable for estimating and planning, and deceptive as an accurate portrayal of product size. To those left pondering, "What is a better approach for measuring software size?" despite criticisms, function point analysis, endorsed by International Organization for Standardization/ International Electrotechnical Commission 20926:2003, is used by thousands of companies worldwide to measure software size."

- Posted on Wednesday, January 21, 2009 - 10:20 am:

Just two cents more to add to this thread, as another view on the reason why relevant deviations could be observed in backfiring applications with a posteriori data: LOCs are a product-level measure of the 'length' of the code, while FP (or another fsu) is a product-level measure of the 'functional size' of a software application.

Thus, different attributes for the software 'product' entity level.
We'd compare apples with oranges...

- Posted on Wednesday, February 23, 2005 - 02:35 am:

In general backfiring is discouraged because the degree of error tends to be so great. A lot has already been said on this board so I'm not going to repeat it. If you decide to go this route, you should not go it alone. A couple of the IFPUG consulting firms have a lot of experience in this area. One is SPR (<http://www.spr.com/>) which was originally Capers Jones company. Capers Jones is person best known for backfiring; however he's also known for his now famous quote "the use of lines of code metrics for productivity and quality studies to be regarded as professional malpractice starting in 1995."

The other company that has a great deal of experience in backfiring is the David Consulting Group (<http://www.davidconsultinggroup.com/>). They have an interesting page which discusses Industry Data and Lines of Code at <http://www.davidconsultinggroup.com/indata.htm>

- Posted on Sunday, April 29, 2007 - 01:36 pm:

a general suggestion is to avoid to use "backfiring", in particular if it is not a "controlled backfiring", done on your own historical data.

Two simple reasons:

(1) you cannot control the way a LOC (whatever the programming language) has been defined

(2) IFPUG FPA has not a particular fit for sizing a COTS such as SAP; in any case, a Functional Size Measurement Method (FSMM) can measure only the "functional" side of a software system, not ALL the system. Thus, also if backfiring is applied within COCOMO, the conversion from LOC (that's a length measure based on code) to FP (that's a functional measure based on requirements) is a practice that should be avoided.

You can take a look also @

http://www.geocities.com/lbu_measure/fpa/fpa.htm#p6a for further info on this point.

- Posted on Monday, November 18, 2002 - 09:10 am:

You will not find very many IFPUG members who encourage the use of LOC/FP conversion tables. The reason that FP were invented is precisely because LOC were a notoriously poor unit of measurement.

The conversion tables were created for academic purposes at a time when there simply were not enough real FP counts available to study. They may be useful for estimating the total number of FP on the entire planet, but I would not trust them for any sample smaller than that.

You have already discovered one of the conversion tables' major flaws: inaccuracy. Even the handful of people who find a place for LOC in their toolset have stated that the ratio for Cobol, for example, is probably 300LOC/FP instead of the 100 in the table.

Another major flaw is imprecision. The standard deviation of "backfired" FP,

as they are called, can be greater than the differences you are trying to measure.

I urge you to be extremely cautious about using backfired FP. Metrics derived from them may be misleading.

- Posted on Tuesday, January 07, 2003 - 06:56 pm:

Hi Anand, I have had the same problem as you have and perhaps is due to the lack of time to perform an accurate FP counting.

I agree with Carol and Gene, backfiring is risky, but I had used the following procedure to reduce these risks:

1.- Try not to use the capper jones factors, instead of that I have found another factors which are more updated, you can find this factors in the following page: <http://www.qsm.com/FPGearing.html>

2.- Because you can't trust only in gearing factors, you could do the following:

2.1 To apply the "the contribution of the ILFs" to the overall application, that means that the you have to perform a counting of the ILFs based on the IFPUG rules and then apply the 22.1% contribution. You can find an example of this in www.isbsg.org.au/html/fpsize.html

2.2 Another way is to perform a "backfiring sampling" and the idea of this is that you determine your own gearing factor. In this case, you will need the entire LOC counting, and select modules with low, average and high functionality.

This modules are counted using the FPA rules, and then the LOC counting has to be performed. As you can imagine, from the FP counting of the functionality you have selected and with the LOC counting of that functionality, you can derive your own gearing factor.

3.- You have to consolidate the sizes you get by the backfiring and the "ILF contribution".

I have applied this steps in practice and I have found differences of less than 10%.

Of course, don't forget to perform the FP Counting as soon as you can. This steps can be used as a way to get a ballpark estimation, later on you will need the FP counting complete, to track the project, to establish the baseline, to perform an enhancement count, etc.

Luca Santillo, another expert in project management and software measurement, stated: "You should avoid converting Lines of Code to Function Points (and vice versa) because it can introduce strong errors in the estimation process."¹⁶

¹⁶ "ESE: Enhanced Software Estimation", *IT Measurement, Practical Advice from the Experts*, IFPUG, Addison-Wesley, April 2002

The consistency of these comments relate to the fact that backfiring Function Points from Source Lines of Code or even using Source Lines of Code as an application measure is inappropriate and misleading.

C. Mr. Pinto Has Exhibited Total Unfamiliarity With FPA

Mr. Pinto's totally incorrect definitions of Internal Logical Files (ILFs), External Interface Files (EIFs), External Inputs (EIs), External Outputs (EOs), and External Inquiries (EQs) are evidence of his unfamiliarity with Function Points and FPA. Incorrect definitions invariably result in incorrect and unreliable Function Point counts. Mr. Pinto apparently did not even use his referenced copy of the IFPUG CPM to obtain the correct definitions.

ILFs

Mr. Pinto's definition of an ILF is that it "holds and maintains information that is stored within the boundaries of a specific program/module. For every piece of data that is stored, updated, maintained, and retrieved from within the database, the system is acknowledged as performing a corresponding Function Point of work."¹⁷

In contrast, the industry-accepted IFPUG definition of an ILF is: "An internal logical file (ILF) is a user identifiable group of logically related data or control information maintained within the boundary of the application. The primary intent of an ILF is to hold data maintained through one or more elementary processes of the application being counted."¹⁸

An ILF does not "maintain data"; it is overall data stored by an application and not "within the boundaries of a specific program/module". There is virtually nothing correct in Mr. Pinto's statement that, "For every piece of data that is stored, updated, maintained, and retrieved from within the database, the system is acknowledged as performing a corresponding Function Point of work." The functional complexity of an ILF is based upon the number of unique Data Element Types (attributes) and the optional or mandatory subgroups of the ILF.¹⁹

EIFs

Mr. Pinto's definition of an EIF is that it "controls information that is passed to other related application programs/modules that are outside the boundaries of the specific program/module. For every piece of data that is passed from the database to another program, the system is acknowledged as performing a corresponding Function Point

¹⁷ Pinto Report, page 18.

¹⁸ See the International Function Point Users Group (IFPUG) Function Point Counting Practices Manual (CPM), Release 4.2.1, Part 1, page 6-3 and Part 4, page G-4

¹⁹ See Appendix D, which contains a copy of the International Function Point Users Group (IFPUG) Counting Practices Manual (CPM) Version 4.2 Function Point Counting Guidelines published by The David Consulting Group and approved by IFPUG.

users of the PeopleSoft, JD Edwards, and Siebel software do not utilize much of the functionality included in their applications.

H. There Was No Need To Translate Vast Quantities of Documentation

In my opinion, if TN were to create any documentation at all, it by and large only would have need for the README Files, CD-ROM Information, Operator Guides, and Maintenance Guides. Accordingly, at most, some – but certainly not all – of the documents referenced by Mr. Pinto might have needed to be translated, so Mr. Pinto's estimates requiring translation of extensive libraries of documentation are significantly overstated. However, Mr. Pinto suggests that every document – irrespective of its use or relevance to the particular applications supported by TN – should be translated into 21 languages. For the vast portion of the documentation, there was no need for translation from English, since TN's personnel presumably all spoke and worked in the English language, and their clients presumably were English speaking, at least with respect to technical matters.

VI. MY RESULTS

A. My Summary Of Analysis

In this report, I have demonstrated that Mr. Pinto did not apply any recognized form or version of FPA in his report. I have documented the IFPUG methodology for sizing in Function Points, and I have shown that Mr. Pinto's "Estimating Approach" and his "Ten-Step Analysis To Determine The Cost Of Development Using Function Point Analysis" did not analyze anything using FPA. The Function Point value he achieved through his Ten-Step Analysis is inconsistent with the IFPUG methodology or any approved sizing methodology. These results are completely fabricated and without substance. In addition to being derived by means of his unrecognized, unorthodox and repudiated approach (and thus suspect), they also seem to me to unsubstantiable and vastly inflated.

Further, in my expert opinion, he should not have attempted to place a value on entire suites of products as TN utilized only a limited percentage of the applications contained in those suites of products.

Moreover, if – for the sake of argument – the Defendants actually were to independently develop all four of the Oracle software application suites recited in the Pinto Report, then, in order for the newly developed software to be completely useful to TN to support its customers, the newly developed software essentially would have to be an exact replica of the four, recited Oracle application suites (especially for the purpose of providing most tax updates, bug fixes, etc.). The probability that the Defendants (or anyone) could create four exact replicas of the four Oracle application suites from scratch is essentially zero (i.e., it is essentially impossible). Accordingly, in my opinion, determining the cost for independently developing the four underlying application suites is not appropriate for the case in question.

Mr. Pinto's derivation of productive hours of effort and his estimated development cost have been developed utilizing faulty logic as documented in this report and should be disregarded.

B. My Function Point Count Of EnterpriseOne 8.0 Accounts Payable

As an exercise to demonstrate how to properly perform an FPA, I analyzed the Accounts Payable module of JD Edwards EnterpriseOne 8.0.³⁵ I performed this analysis not by using Mr. Pinto's unknown and unconventional Ten-Step process, but instead by utilizing the industry-standard IFPUG FPA methodology, namely:

- Determine counting scope & boundary
- Identify functional user requirements
- Measure data functions
- Measure transactional functions
- Calculate the functional size (in Function Points)

Mr. Pinto's "Ten Steps" – apparently created by him from whole cloth – have no commonality with the above-mentioned FPA methodology. To illustrate the extent to which his "Ten Steps" differ from actual, industry-recognized FPA, I list his "Ten Steps" below:

Step One: Identify and Group Source Code Components

Step Two: Count the Number of Source Lines of Code

Step Three: Determine the Amount of Functionality

Step Four: Determine the Number of Pages of Documentation

Step Five: Derive the Productive Hours of Effort

Step Six: Distribute the Effort across the Product Development Life-Cycle

Step Seven: Allocate Productive Hours of Effort to Team Roles

Step Eight: Derive the Cost of Localization and Documentation Translation

Step Nine: Apply Hourly Rates to Determine the Development Costs

Step Ten: Analyze the Estimated Development Costs

As an internationally recognized expert in FPA, and based on my many years of real-world FPA experience, it is my opinion that following Mr. Pinto's "Ten Steps" would not result in any sort of correct, meaningful, useful, or substantiatable Function Point size.

My results for my FPA of the Accounts Payable module of JD Edwards EnterpriseOne 8.0 can be found below.

³⁵ This was one of the modules with respect to which sufficient user documentation was provided by Oracle, from which an FPA could be conducted. See TN-OR02989997, TN (Hard Drive).33 at BU01_G\JDE 36-44\Generic PeopleBooks – JDE.

Type	Low	Avg	High	Total
EI	51 x 3 +	42 x 4 +	16 x 6 =	417
EO	3 x 4 +	34 x 5 +	0 x 7 =	182
EQ	25 x 3 +	35 x 4 +	6 x 6 =	251
ILF	7 x 7 +	5 x 10 +	3 x 15 =	144
EIF	0 x 5 +	0 x 7 +	0 x 10 =	0
	Total			994
	Value Adjustment Factor			1.18
	Total Adjusted Function Points			1173

C. My Function Point Count Of PeopleSoft Enterprise Global Payroll For US 8.9

As an exercise to demonstrate how to properly perform an FPA, I analyzed the Global Payroll for the US module of PeopleSoft HRMS 8.9.³⁶ I performed this analysis not by using Mr. Pinto's unknown and unconventional Ten-Step process, but instead by utilizing the industry-standard IFPUG FPA methodology.

My results for my FPA of the Global Payroll for the US module of PeopleSoft HRMS 8.9 can be found below.

Type	Low	Avg	High	Total
EI	35 x 3 +	78 x 4 +	11 x 6 =	483
EO	2 x 4 +	11 x 5 +	21 x 7 =	210
EQ	31 x 3 +	35 x 4 +	4 x 6 =	257
ILF	5 x 7 +	8 x 10 +	5 x 15 =	190
EIF	2 x 5 +	8 x 7 +	3 x 10 =	96
	Total			1236
	Value Adjustment Factor			1.18
	Total Adjusted Function Points			1458

³⁶ This was one of the modules with respect to which sufficient user documentation was provided by Oracle, from which an FPA could be conducted. See TN-OR02989993, TN (Hard Drive).29 at BU01_G\BU01_LOGICAL_IMAGE_43-76\PeopleSoft Enterprise Documentation.

OPTION TO REVISE AND SIGNATURE

I reserve the right to modify and/or supplement this report and/or the opinions set forth herein if subsequent rulings are made by the Court and/or additional evidence becomes available.

I, David P. Garmus, having conducted the aforementioned analysis and having authored this report, confirm that the opinions contained herein represent a fair and unbiased analysis of the facts presented to me.

Date: March 26, 2010



David P. Garmus