https://doi.org/10.48009/2_iis_2015_91-101

*Issues in Information Systems*

*Volume 16, Issue II, pp. 91-101, 2015*

# DO YOU KNOW WHAT YOUR NAME IS? THE QUESTION OF ADDITIVE OR REDUCTIVE PRACTICES

***Gerald Johnson, Independent Consultant, garyjohnson61ad@gmail.com***
***D. Lance Revenaugh, Montana Tech University, Colorado Technical University***
***lrevenaugh@mtech.edu***

## ABSTRACT

*After decades, the propensity for data related project failure (over budget, over time estimate, under-performing) continues. There has been a myriad of methodology changes, all meant to alleviate the problems, but other than specific case successes, all have proven unsuccessful in significantly reducing the failure rate in data related projects. Could it be that methodology is not the problem? If bricks are brittle and prone to breaking, no matter what method the builders use, the wall will prove weak. Throughout many methodology changes, the practice**s** used in requirements gathering have remained constant. This study looks at these requirements gathering practices and how addressing Technological Frames of Reference (TFR) might provide a resolution through change of practice, not methodology*.

**Keywords:** Reductive Practice, Additive Practice, Methodology, Practices, Business Analysis, Requirements Project Failure

## INTRODUCTION

### Knowing the Data

"'Do you know what your name is?" The question seems simple, but the answer may surprise you. By the end of this discussion you may surprise yourself and come to the conclusion that not only do you not know what your name is, but very few other people do either.

On a colloquial basis, of course we each know what our names are. But the question really is 'Do you know WHAT your name is?' Do you know all the components of your name (given, middle, surname)? Do you know the nuances of titles that accompany names both before (Mr., Reverend, Dr., etc.) and after (Jr., DMD., Esq., etc.)? Do you know the proper way to handle hyphenated names, multiple word names, initials, nicknames, etc.? Do you recognize the affix and stem portions of your name?

Why is this important? Most systems keep track of people's names for one reason or another. These systems may include customer names, employee names, vendor representative names, etc. These names might be used in marketing campaigns, legal matters, HR records, etc. Personal names are a common piece of data to include in most data systems. The question then arises – 'Why do we need to know these details about names?' The answer should always be 'because there is a business need'. Perhaps titles are required because formal mailings are part of the business activity. Perhaps middle names are needed because legal filings are one output. Perhaps alias names are needed because the system handles arrests and incarcerations. Whatever the business need, and whatever the data point, it is critical to know and identify the details in order to make the resultant system perform successfully.

### Breadth of Issue

To understand the breadth of the issue, one must look at the typical case, where requirements gathering is done using additive practices, and at least a portion of the overall optimal details regarding personal names are not investigated for the requirements. The data architect then produces the architecture based on the requirement provided, and is followed by the developers assembling logic and interfaces based on the requirements and the architecture. The resulting system is then implemented and soon after is tis discovered that name prefixes were accounted for, but suffixes are required too. Customers notice the issue, changes are requested and the bug is eventually fixed. Meanwhile large sets of data are then imported by a customer, and it is noticed that no logical method of handling people with only a single name (Madonna, Pele, etc.). This is identified as another bug, and customers wait for

another fix process to be completed. This may cycle through a dozen or more bugs and fixes before the name data and processes become settled. Now consider that a personal name is just one of perhaps thousands of data points and their associated logic that comprise the entire system and the end result is a litany of bugs, fixes, and ultimate failure.

The full breadth of the issue is understood when it becomes clear that conversational awareness of data and its requirements is not necessarily sufficient for requirements gathering, and yet while using the standard additive practices it is exactly that personal awareness (or TFR) that is brought into play to gather the requirements from. The many finer points of knowledge, the necessary logic around handling those points, etc. are not in the typical person's repertoire, even if they perform requirements gathering as a profession. The standard additive requirements gathering practices (the gatherer using their own knowledge to question the business partner) will continue the pattern of sub-optimal requirements production. These same people that are unaware of the required information (beyond a conversational level), are continuing to be expected to produce the requirements for the system that uses, manages, and handles this data.

## CURRENT SETTING

### History of Attempted Fixes

Over the last five decades there have been a myriad of project methodologies to mitigate the propensity for software and database project failures and their associated project methodologies. There has been a constant battle to bring software / database development projects in on time, under budget, and with appropriate functionality. Amongst the many methodologies have been the Kepner-Tregoe methods used when requirements gathering is seen as just another quality task [9] Waterfall method where "the work is contracted based on significant milestones in the development process [14]," prototyping methods where "technology plays an important transitional role" and "reduces development time and cost by 30-50 percent" in manufacturing and has been adopted in software development as well [6], Spiral methodologies, which are a "appropriate mixture of specification-, prototype-, simulation-, automatic transformation-oriented approaches to software development [3]." Rapid Application Development, Object Oriented, Top Down, Unified Process, Agile, Scrum, and others [2, 4, 7, 12, 13, 15, 17, 18, 20, 21]. While the resultant problem of project failure (significantly over budget, significantly over time, system under performs or lacks functionality, or the project was completely cancelled) has remained, the studies on methodologies and presentations of new methodologies continues.

Interestingly, while each of these attempts have changed the methodology that surrounds the projects, the basic practices of requirements gathering have not changed. While every methodology is conceived with the intention of breaking out of this pattern of failure, none have led the industry as a whole down a path that significantly reduces this failure rate. In fact, with higher availability of information and quicker communications of issues, the perception is that the problem has not been addressed, and perhaps has even been exacerbated. While many claim success with new methods (typically through very limited case studies), when attempting to make generic claims it is found that "research evidence supporting proponents' claims is somewhat lacking [10]". It should also be noted that, as the individual case studies referenced by Erickson and others [10]" show, each methodology can demonstrate case examples of success. This dichotomy points toward some factor other than the project methodology as being causal to the failures. One of these potential factors that we look at with this study is the practices within those methodologies, and in particular – the potential for reductive practices to have a significant impact.

### Others Using Reductive Practices

Virtually no one is going to select an employee at Jiffy Lube as the best mechanic in town. Typically they employ many college age workers, or beginning to journeyman level mechanics. And yet millions (judging by their business quantity) recognize that they do a very good job of checking and replacing fluids in automobiles, along with many commonly degraded parts such as air and oil filters. They accomplish this not through their agile methodology of using several people in different positions to analyze, notify, and change, but through their practices of using a reductive list of universally defined items and sequences to check a series of components in a very specific sequence. With this practice the individual inexperienced young 'mechanic' can perform with high quality, as they are using the combined knowledge of dozens of top level mechanics totaling hundreds of years of experience. The

US. Military is replete with this type of reductive practices, turning large groups of 18-25 year olds into the strongest military the world has ever seen. Cleaning organizations often hire low The key then is to begin to look at whether, and to what extent, the use of data practices (not methodologies) of this same level, making use of vast experience and global levels of knowledge, may be a key to success in data related development as it is in other industries.

## RESEARCH METHODS

### Points of Research

In order to manage the scope of this initial research, of the 4 clearly delineated sets of data practices (requirements gathering, data architecture, SQL and data retrieval, and data content), only the data requirements gathering portion were examined. And within requirements gathering realm, the current research was limited to information surrounding one specific data point – the aforementioned personal name. With this view, the participants, consisted of people who perform or have performed data requirements gathering, with a wide variety of skill and experience.

### Technological Frames of Reference

This variety in background and experience has best been described in research by Wanda Orlikowski and Elizabeth Davidson in their studies of Technological Frames of Reference (TFR). To understand what reductive practices bring into the formula for success, one must understand the concepts of learning and TFRs. The theory of TFR begins with understanding that most learning done by people is through Naïve Inquiry [10], or experiential learning. This, combined with limited number of experiences actually performing tasks such as system wide requirements gathering, or data architecture, leads to, in many cases, a lower than optimal TFR for a person performing a task [18]. While there is not a definitive rate at which any person learns, and age or actual time does not define a person's knowledge level [5]. Naïve inquiry does contribute to a person's TFR level. The relevance of TFR to data related projects is that, based solely on experience levels, the TFR of many, if not most data related personnel can be expected to be quite low. In studies that track experience levels of typical professional roles, this experiential naïve inquiry learning is seen to be over a relatively short time frame, as Nitin Agarwal and Urvashi Rathod have shown in their studies, often amounting to less than three years in the industry for individual contributor roles such as data requirements gatherers [1] and only 7 or fewer years average for entire teams, even when adding in more heavily experienced management positions. The problem is further compounded, as each succeeding step in the overall process is built on a previous step that was not performed optimally. Less than optimal data requirements are used to produce less than optimal data architecture, which then becomes the base for sub-optimal data access and sub-optimal data content decisions. At each step, the project becomes further limited from its potential for success.

Each participant was asked several questions which were used to quantitatively assign TFRs for three related areas. The relevant TFRs used for this research were (a) Requirements Gathering Knowledge, (b) Data Point Knowledge, and (c) Business Knowledge. While this scheme of breaking required skills (and relevant TFRs) into task, subject, and need is not a defined standard, there is in fact no defined standard, so this part of the research could later be examined with different TFR schemes or assignment logic. The determining questions were based on years of experience and quantity of occurrences to determine ratings of High, Medium, and Low for each of the defined TFRs. While the 'self-rating' system of assigning TFRs based on participant response could be deemed less than optimal, it also is the basic method of using self-reported information from a resume to be a primary filter for who fills these business analysis roles.

### Data Requirements Gathering

As noted, there are several specific areas where data related practices could be relevant, and should be examined for optimization. These include data requirements gathering, data architecture, data retrieval and data content. They should be looked at individually, and the first one sequentially in the project overall plan is the gathering of data related requirements, commonly referred to as 'business analysis'. Most IT/ IS personnel, including data requirements gathers, have only gone through a limited number of system design and architecture endeavors. Most users or business personnel have even less experience going through the process of system development, where their role initially is to provide requirements to the Business Analysts doing the requirements gathering. Having two sides (the requirements gatherer, and the requirements provider) perform a task with this limited experience can't

reasonably be expected to naturally produce optimal results. The predominant practice (additive in nature) is for requirements analysts to ask questions to add to the knowledge base of known needs, wants and wishes for a system. Using their own limited knowledge, background, and experience, a person will not typically perform a task to the level they could if they had universal knowledge, gained from the experience of many [16]. A classroom demonstration of this principle would be to ask a person to list all of some list that (1) they typically would know a portion of, but not all of, and (2) is a reasonable lengthy list, such as a list of North American bird species, or a list of countries. Most people will produce a portion of the overall list, but when presented with a complete list will recognize many of those that they failed to include. This is the principle mentioned above, used in organizations such as Jiffy Lube. Rather than asking employees to check everything on a car they think they should, and expecting optimal results, they provide those employees with a reductive set of knowledge to work from. In their case this universal set of knowledge is in the form of a checklist that the employees use to guarantee they have looked at all the points on a car that need to be examined. The reductive nature of data requirements gathering may differ from that of automobile tune-ups, but the basic concept of the people working reductively from universal sets of knowledge rather than additively from their own personal experience will remain constant.

**Data Requirements Gathering**

In the progression of data related tasks, at any one step, if the step is performed with less than optimal TFR, the successive steps and resultant project has less chance to succeed than if it had been performed optimally. This principle applies for any set of tasks or project. If a building is poorly designed, then it is unlikely the foundation will be put in well. Following that, with a foundation that does not meet the needs of the building, there will be issues structurally. Supporting beams will not be properly anchored or balanced, the stairs will not even from the first floor to the second, etc. The overall finished product stands no chance of being the building the customer expected or needed. The steps shown in figure 1 illustrate the loss of potential success when TFR is not fully taken into account and less than optimal results occur at each stage. While no system or process will ever be perfect, the key is to minimize the loss at each stage, maximizing the proximity to optimal at each subsequent stage and for the overall project.

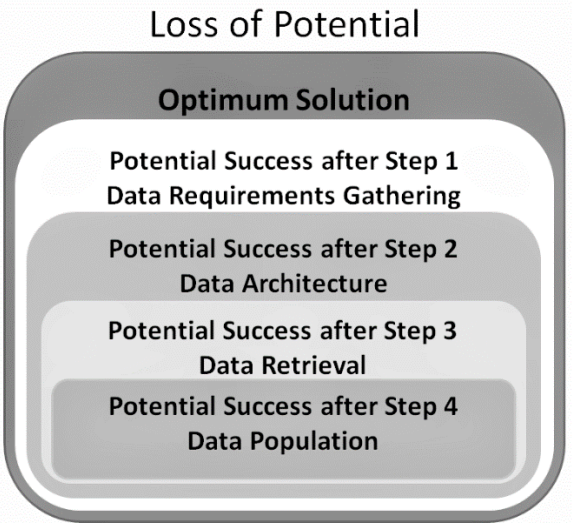


**Figure 1.** Loss of Potential for Success in Data Tasks

**Additive Practices**

By definition, a set of processes and practices is implemented to make a change of state, so no task or practice is required if no change is to take place. When performing these practices, if the practice begins with an empty slate, and is performed with the knowledge and experience of the person performing the task, then it would be considered an additive process. An example of this is the standard data requirements gathering process where a Business Analyst (BA) interviews the user for technical requirements, asking them questions until the BA is satisfied that

enough information has been received to architect and build the system. Within this additive mode, if a question is forgotten and not asked by the BA, then some requirements for that data point will not be covered and passed to the architects and developers. For instance, if the BA is asking about the geographic information needed for a shipping system, they may ask if the county an address is in is relevant and needs to be tracked. Likely for simple shipping functionality it is not, so a Country/State/City model of data may be envisioned. However, the fact that the question is not asked is the critical issue. While some omissions (based on lower than optimal TFR) end up being innocuous, many are not, as they lead to omissions which leave holes in processing ability or assumptions, which often are incorrect and lead to erroneous processing.

**Research Details**

This particular research looked at the differences in results while using reductive and additive practices for requirements gathering for one data point (personal name) for one business scenario (genealogy):

- Data requirements gathering was chosen because it presents the first data related practice in sequence for a standard development lifecycle. It was also chosen because a large number of people are commonly involved in requirements gathering as compared to some other data related roles such as architecture. For data requirements gathering, it is also relatively straightforward to test participant's knowledge through primarily multiple choice questions on a survey.
- Personal Name was chosen as the data point because most people are at least conversationally aware of how names are constructed and what constitutes a name (given the participant audience, this would be an Indo-European name, with first, middle, and last names, prefixes, suffixes, etc.) , However, most people also have limited knowledge about these components beyond basic conversational knowledge.
- A Genealogy scenario (a hypothetical new database/system to do genealogy on-line). This was chosen as a scenario again because most people have at least a cursory knowledge of genealogy – they are at least aware of some of their own family history. However, very few people, including long time professional developers, have actually worked on gathering requirements for a genealogy system.

Each of the participants first answered several questions related to the amount of time and number of occurrences they have experienced in each of these three areas, to determine a TFR for each of the three. The participants provide a wide range of experience and background, and covered all combinations of the three TFRs.

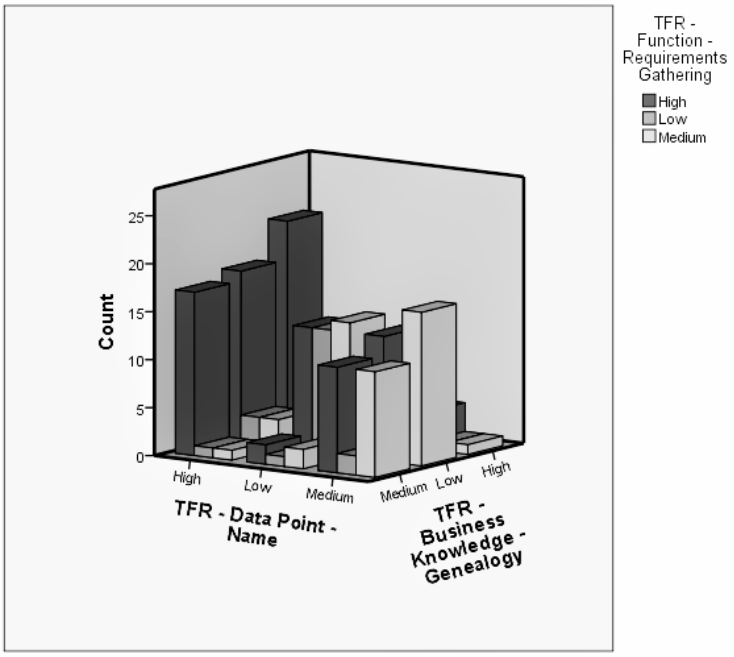


**Figure 2.** Distribution of TFRs Among Participants

As expected, there were some categories that had lower frequency (anecdotally, someone who reported that they had over 15 projects as a requirements gatherer would not be expected to have only one or two of these projects involve people's names), with tendencies toward anecdotal expectations.

Each participant was then given a specific scenario and asked simply to provide the requirements for storing and managing a personal name. This reflects the additive mode of requirements gathering, where the business analyst meets with the business representative and asks questions to glean the requirements. Responses ranged from very simple statements reflecting conversational knowledge ('First name, Middle Name, Last Name) to quite extensive responses that detailed the need for specific sizing, use of various title prefixes and suffixes, nicknames, hyphenated names, etc. These responses were then scored against a set of 22 specific factors relevant to a universal set of requirements for handling a personal name. While subjective, the scoring followed tight rules in order to bring the calculations as close as possible to equivalence to the objective scoring of the reductive responses. As an example, any reference to delineating the 3 major parts of the personal name qualified as a correct response relevant to the factor for handling the name as distinct components (Given Name, Middle Name, and Surname).

These 22 factors are not the comprehensive set of those needed for personal name handling, but a representative set, with some of each type of requirement. As an example, the factors include requirements for familial prefixes (Mr., Mrs., Miss, etc.) but not for professional prefixes (Dr., Pres., etc.) familial suffixes (Jr., II, etc.) or others. This leads to the use of weighted scoring, where the addressing of some factors (such as the aforementioned familial prefix) are weighted heavier than factors that do not represent other unscored factors.

This research is discovery oriented, and is not geared towards answering one specific formulaic situation. The stated parameters were to determine if there were significant and actionable differences in the results from the typical additive requirements gathering and the alternative of reductive requirements gathering. As such a variety of approaches were taken to decipher patterns and significance within the returns. In its simplest form, the weighted total scores for the participants across both sets of practices show a clear significant difference.

With each of the 177 participants scoring significantly higher using the reductive practices, the significance of the practice difference is clear. However, what remains is to clearly identify, weight, and delineate the individual

factors of the participant characteristics and/or the reductive approach characteristics that can optimize that difference and make the significance actionable.

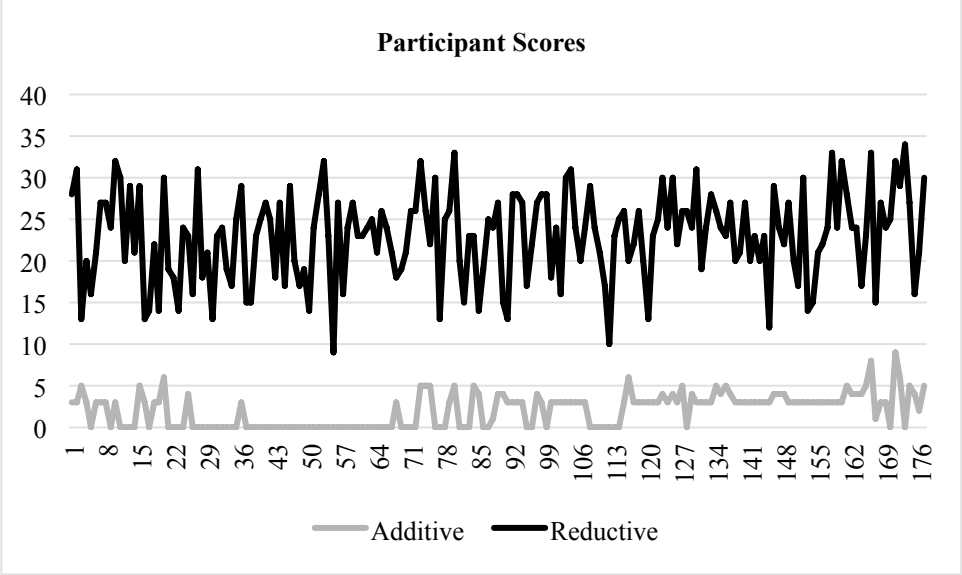


**Figure 3.** Participant Additive and Reductive Scores

When looking at the frequencies of scores, the significance in change of practices becomes even clearer. Please note the anomaly of low quantities of scores of 2 and 3. This is due to the most frequently addressed factor (handling names as split components for first, middle, and last), which was given a weighted score of 3 because it affects architecture, and logic, and is a core decision that other factors such as hyphenated names and multiple names depend on.

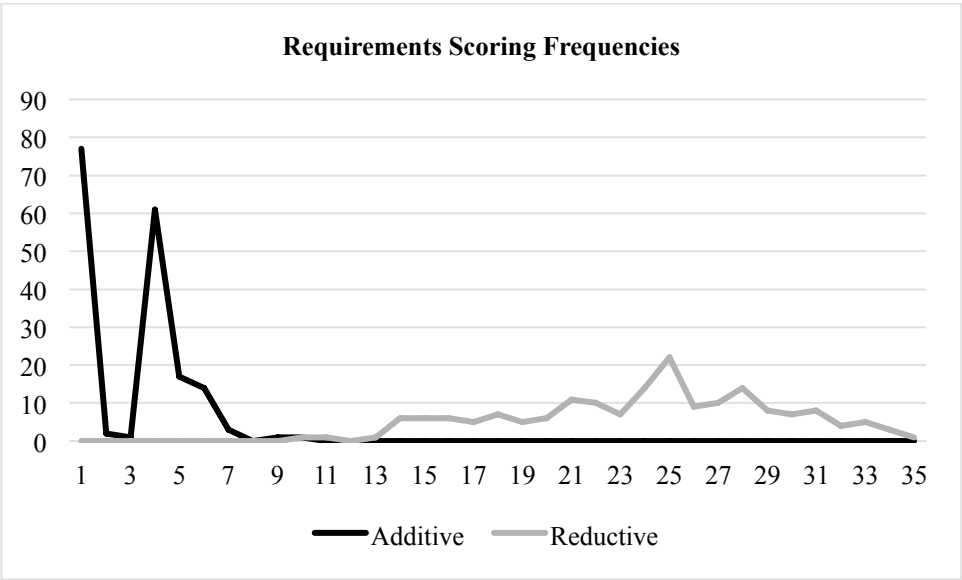


**Figure 4.** Participant Scoring Frequencies

While the reductive responses have a wider variance, they are both quite normal in their form as well as significantly higher than the additive scores by the same participants. In fact, the highest additive score only equaled the lowest reductive score.

| Paired Samples Test | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Paired Differences | | | | | t | df | Sig. (2-tailed) |
| | | Mean | Std. Deviation | Std. Error Mean | 95% Confidence Interval of the Difference | | | | |
| | | | | | Lower | Upper | | | |
| Pair 1 | Additive - Total -Data Point Plus Business TFR - Reductive - Total -Data Point Plus Business TFR | -20.944 | 5.250 | .395 | -21.722 | -20.165 | -53.071 | 176 | .000 |

**Figure 5.** Additive to Reductive Paired Samples

This brings the research back to what factors, other than the specific practices, contributed to the scores, and in particular what significance did the TFR ratings of the participants play.

While it was anticipated that Requirements gathering TFR would have some significant effect on the scoring, it did not on the reductive scores. In fact, of the three individual TFR values, only that for the data point itself even had sequential match with the TFR ratings

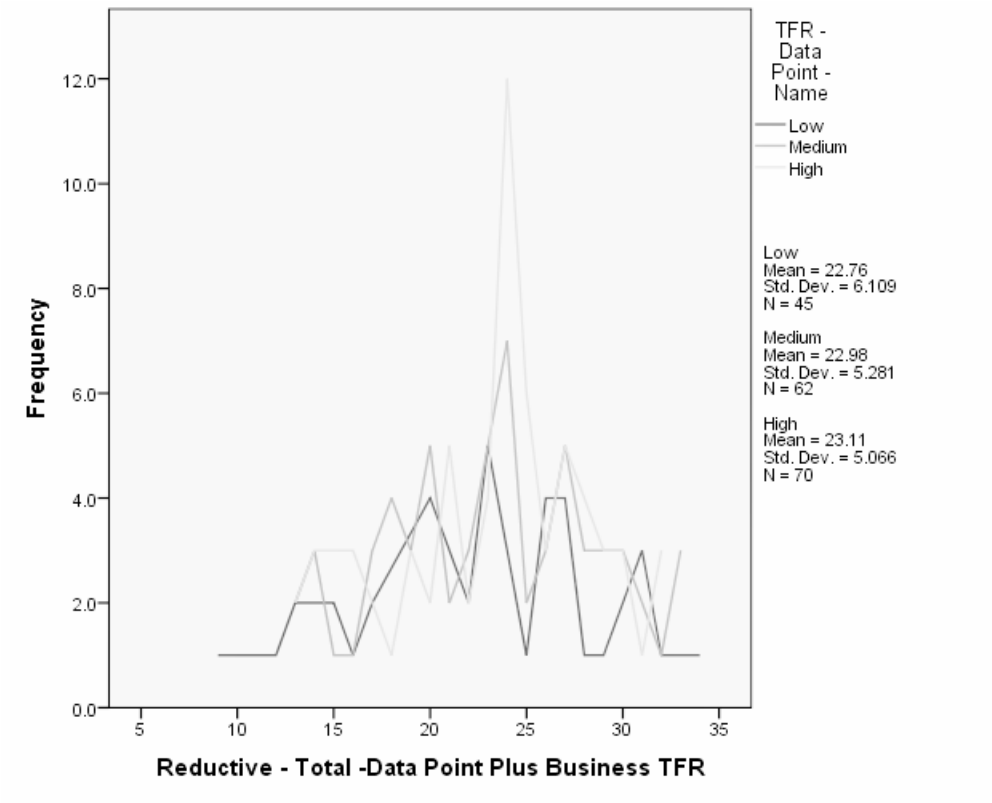


**Figure 6.** Data Point TFR Reductive Scoring Breakdown

**Reductive Practices**

By definition, a non-additive practice is a reductive practice. For this discussion, and the current studies in progress, we are discussing and exploring the feasibility of using a particular type of reductive practices (a Poke-Yoke style ([19]) universal set of prescribed checklists of universal knowledge option sets) to reduce the failure rate in data oriented projects or portions of those projects by ensuring that all options are investigated.

## CONCLUSIONS

Even though the individual TFR ratings do not show significance to predicting either additive or reductive performance, the overall TFR combined rating does correlate strongly with the additive results. In addition the Additive and Reductive performance of individual participants, regardless of TFR, also shows significant correlation. However, perhaps the most valuable factor thus far, is that neither the individual TFRs nor the combined TFR has a correlation to the reductive performance. In short, reductive practices can be used to overcome lack of appropriate TFR related skills to perform more optimally on requirements gathering tasks.

Research will continue up until and beyond the conference, along many related paths. While the current study is aimed specifically at data requirements gathering (and specifically for one data point in one business scenario), the expectation is that related research will expand out to more data points and all data related practices. Ongoing and future anticipated continuing related research includes but is not limited to the following:

- Repeating the research using different audience selection modes to validate original returns
- Conducting equivalent research on additional data points to explore the breadth of the effects
- Conducting equivalent research on additional data area including data architecture, data retrieval, and data population
- Exploration of data point component type effects and variations

There is potential that if results are definitive in the differences between the results for additive and reductive practices, they could be codified into data related products, projects and tasks. These could then be applied to many data related products and services for a variety of markets and industries and potentially result in gains along several fronts:

- Increased efficiency of development teams (to successfully complete system architecture and development projects), and therefore their associated organizations.
- Increased productivity and functionality of developed systems, and therefore the organizations and people that use those systems.
- Products and practices that could move data quality closer to a commodity level service
- Reduced training and ramp-up time for new employees.
- Greater adherence to regulatory demands.
- Increased Data Quality
- Potential new suites of development products oriented around reductive rather than additive practices.

## REFERENCES

1. Agarwal, N. R., Urvashi. (2006). Defining success for software projects: An exploratory revelation. *International Journal of Project Management, 24*, 358-370.
2. Bird, M. S. (2010). *Utilitizing Agile Software Development as an Effective and Efficient Process to Reduce Development Time and Maintain Quality Software Delivery.* (Doctor of Philosophy), Capella University. (3398706)
3. Boehm, B. W. a. P., Philip N. (1988). Understanding and controlling software costs. *IEEE Transactions on Software Engineering, 14*(10), 1462-1477, References Page 1465.
4. Börjesson, A. M., Lars. (2004). Improving software organizations: agility challenges and implications. *Information Technology & People, 18*(4), 24.
   Bytheway, W. (2005). Age-identities and the celebration of birthdays. *Aging and Society, 25*(4), 463-477.
5. Che Chung Wang; Ta-Wei, L. H., Shr-Shiung. (2007). Optimizing the rapid prototyping process by integrating the Taguchi method with the Gray relational analysis. *Rapid Prototyping Journal, 13*(5), 304-315, Referenced Page 305.
6. Chua, A. Y. K. (2009). Exhuming IT Projects From their Graves: An Analysis of Eight Failure Cases and their Risk Factors. *The Journal of Computer Information Systems, 49*(3), 8.
7. Davidson, E. J. (2002). Technology Frames and Framing: A Socio-Cognitive Investigation of Requirements Determination. *MIS Quarterly, 20*(4), 30.
8. de Mast, J. (2013). Diagnostic Quality Problem Solving: A Conceptual Framework and Six Strategies. *The Quality Management Journal, 40*(4), 21-36, References directly 34,35.
9. Dewey, J. (1933). How We think - A Re Statement of the Relation of Reflective Thinking to the Educative Process (Excerpts Only).
10. Erickson, J. L., Kalle; Siau, Keng. (2005). Agile modeling, agile software development, and extreme programming: the state of research. *Database Management, 16*(4), 13.
11. Georgiadou, E. (2003). Software Process and Product Improvement: A Historical Perspective. *Cybernetics and Systems Analysis, 39*(1).
12. Kautz, K. (2011). Investigating the design process: participatory design in agile software development. *Information Technology & People, 24*(3), 19.
13. Lott, C. M. (1997). Breathing new life into the waterfall model. *IEEE Software, 14*(5), 103-105 - Referenced 103.
14. Maletic, J. I. (1995). *The Software Service Bay: A knowledge-based software maintenance methodology.* (Doctor of Philosophy), Wayne State University. (UMI Number: 9530565)
15. Qiang Zhu, Y. T., Calisto Zuzarte. (2005). Optimizing complex queries based on similarities of subqueries. *Knowledge and Information Systems, 8*(1), 350-373.
16. Salo, O. (2007). *Enabling Software Process Improvement in Agile Software Development Teams and Organisations.* University of Oulu.
17. Samra, T. S. (2012). *Software Risk Management: An Exploration of Software Life Cycle methodologies, Best Practices and Tools for Their Application to Medical Device Software Risk management.* (Doctor of Regulatory Science), University of Southern California. (UMI Number: 3514303)

18. Six Sigma US. (2014). *The Fundamentals of Lean Agent Training*: Six Sigma U.S.
19. Tripp, J. F. (2012). *The Impacts of Agile Development Methodology use on Project Success: A Contingency View.* (Doctor of Philosophy), Michigan State University. (UMI Number: 3523854)
20. Warkentin, M. R. S. M., Ernst Bekkering, Allen C. Johnston. (2009). Analysis of Systems Development Project Risks: An Integrative Framework. *The DATA BASE for Advances in Information Systems, 40*(2), 21.