Technomics
Better Decisions Faster

Natural Language Processing: A New Approach to Simple Function Point Estimation

Presented to ICEAA
May 2023

Agenda

- Problem Statement
- Simple Function Points
  - Definitions
  - The Simplified Software Estimation (SiSE) method
- Natural Language Processing (NLP) definitions and concepts
- Implementation of NLP / SFP
- Example Results
- Conclusions
- Next Steps

Problem Statement

- Cost estimators need a method for estimating software development cost that is:
  - Data-driven – based on data, and not SME judgement
  - Directly tied to requirements
  - Available early in the software development process, possibly before technical design is available

- Better software development estimating will allow:
  - Budget justification, with a defendable basis of estimate
  - Validation / cross check to other estimates
  - Better understanding of costs, and their relationship to requirements
  - Better understanding of cost risk and uncertainty

Simple Function Points (SFP)

Simple Function Points

- Why use Simple Function Points (SFP)?
  - All software estimates need a measure of software size
  - SLOC is outdated and difficult to predict
  - IFPUG function points require detailed functional design documentation
  - SFP count can be created quickly, based on already existing requirements documentation

- What is SiSE?
  - Simplified Software Estimation
  - Developed and used by DHS
  - A proven method for estimating SFP
  - Some DHS-developed tools and methodology already exists

- The following slides show the (manual) SFP / SiSE process

Simple FP (SFP) sizing[1]

- SFP Method developed by Dr. Roberto Meli and Italian researchers, and acquired by IFPUG in 2019[2]
- Simplifies functional sizing into two types of functions:
  - Generic transactional functions (elementary processes)
  - Generic Logical Data Groups
- Sizing process can be performed quickly and early in a program's lifecycle using existing documents. Compatible with IFPUG FP

IFPUG Components	Low	Average	High
External Inputs	3	4	6
External Outputs	4	5	7
External Inquiries	3	4	6
Internal Logical Files	7	10	15
External Interface Files	5	7	10

SFP Components	Weighting Factor
Transactions (Create, Update, Delete, Report, Read)	4.6
Logical Data Groups (Saves)	7

1. Slide adapted from DHS CAD
2. https://www.ifpug.org/ifpug-acquires-the-simple-function-points-method/

How does SiSE work? Example of LBC Requirements and SiSE Verb Keywords

Creates an enterprise-level rules and alerting service that is **networked** to all office-level rules and alerting systems **allowing** awareness and visibility of all agency systems without having to have prior knowledge of each office-level rules and alerting system. **Allows** users to **search**, **discover**, **store**, and **subscribe** to existing agency alerts or **create** new alerts by chaining together a series of existing alerts coming into the system…

Keyword	SFP TX (4.6 SiFP)	SFP Data (7 SiFP)	TOTAL FOR VERB SiFP	*Synonyms*
Allow	2	1	16.2	*grant, accept*
Create	2	1	16.2	*build, conceive, constitute, construct, devise, establish, initiate, invent, make, set up, start, inception, origination*
Detect	1		4.6	*determine, discover*
Search	1		4.6	*hunt, inquire, investigate, research, study, filter, find*
Store	2	1	16.2	*keep, stow*

- Simple FP (SiFP) values are assigned to over 400 verbs & synonyms

Natural Language Processing (NLP)

What is NLP?

- NLP is a methodology allowing a computer to understand, interpret, and manipulate human language through speech or text

- Different industries leverage NLP for different tasks
  - Classifying emails as spam
  - Autocorrect and spellcheck
  - Foreign language translation

NLP allows text to become machine-readable, which analysts can use for statistical analysis

NLP in Industry

Humanities
Analyzing language evolution over time, how meanings of words change across the eras.

Construction
Categorizing construction accidents to detect patterns in similar geographical regions.

Aviation
Classifying incidents at airports to determine what causes delays, malfunctions, and miscommunications.

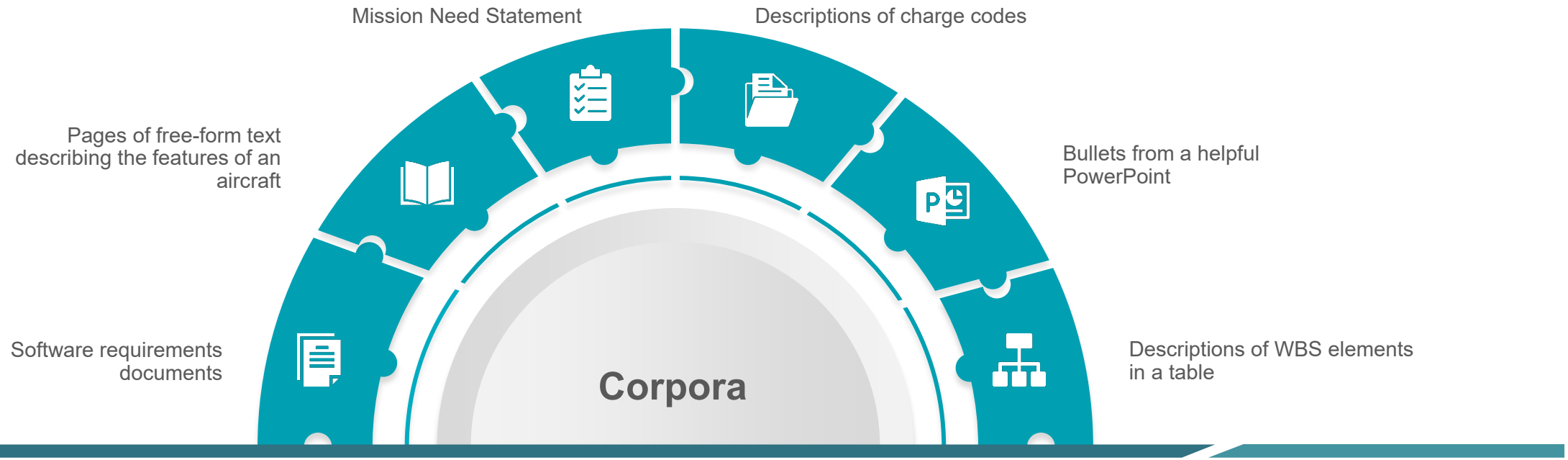
Finance
Using sentiment analysis on message boards and publications to predict market patterns.

Social Media
Refining sentiment analysis classification methodologies using comments and tweets.

Healthcare
Classifying medical diagnoses to create centralized repository, enhancing medical care.

Corpus

- A **corpus** is the body of text(s) you desire to extract information from
- A **corpus** could be a sentence, a few pages, or an entire collection of documents
- Once the **corpus** is identified, the text must be read by a software program, such as R or Python



Data Cleaning for NLP

- Remove **special characters** (e.g., line breaks, strange encodings from a PDF)
- Replace **symbols** (e.g., @, #, $, %)
- Replace **contractions** (e.g., “isn’t” to “is not”)
- Expand **acronyms**
- Remove **punctuation**
- Converting to **lower-case**

Original Sentence

The equipment is used for the Rad-Hard capability for NNSA, including R&D... It’s used to protect nuclear weapons against certain hostile environments.

Cleaned Sentence

the equipment is used for the **radiation hardened** capability for **national nuclear security administration** including **research and development it is** used to protect nuclear weapons against certain hostile environments

Stopwords

What are **stopwords**?
Stopwords are common words in the English language that act as "filler" words, and aren't used for analysis. Examples: "a", "the", "to"

Raw Corpus

"She gave 100% but it wasn't enough," he observed.
"Seemed more like 50 to me tbh" the coach responded.

Cleaned Corpus

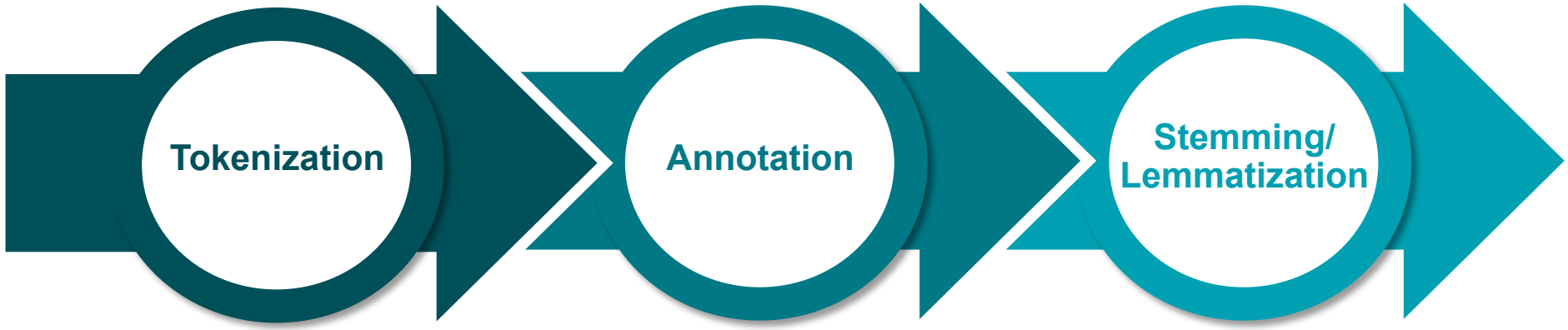
she gave 100 **percent** but it **was not** enough he observed seemed more like 50 to me **to be honest** the coach responded

Removed Stopwords

gave 100 percent not enough observed seemed more 50 honest coach responded

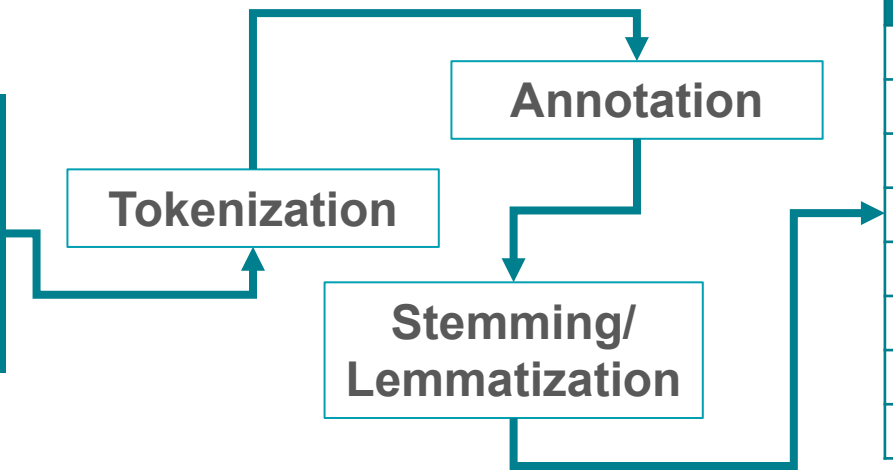
Normalizing Text

- Like all data, text data must be transformed into legible tables for analysis. There are specific processes to wrangle unstructured, raw text
- All these processes seek to make human language machine-readable



I am no bird; no net ensnares me

– *Jayne Eyre*, Charlotte Brontë



word_id	word	stem	lemma	POS
1	i	i	i	pronoun
2	am	am	be	verb
3	no	no	no	determiner
4	bird	bird	bird	noun
5	no	no	no	determiner
6	net	net	net	noun
7	ensnares	ensnar	ensnare	verb
8	me	me	i	pronoun

Application of NLP / SFP method

Implementation of a NLP Tool

- Technomics has an internally-developed NLP tool
  - Written in the R programming language
  - Implements the DHS SiSE algorithm
  - Automatically cleans and tokenizes text using open source NLP libraries in R
- Software requirements
  - R and R studio
  - NLP and general libraries: dplyr, stringr, textclean, tm, udpipe, writexl, readxl, openxlsx
- Key Capabilities:
  - Ability to read multiple LBC's (Lean Business Case) from a single MS Word file
  - Ability to read text from Excel, and save SFP counts to the same excel file
  - Ability to auto-scrape LBC text from Confluence

NLP / SFP Methodology

- Start with a decomposition of program requirements, where each segment is associated with a corpus of requirements documentation.
- Use NLP to identify verbs, convert them to a lemma, and associate these with an established list of verb lemma's and their assumed SFP value. Start with the DHS verb list.
- Use existing cost estimates to measure the calculated SFP count of each program segment against cost. It is assumed that positive correlation validates the method.
- Improve the method by exploring assumptions surrounding the data, verbs, verb weights, and overall approach. Make adjustments where warranted.
- Repeat the process, and re-evaluate the results. Continue the process of data exploration, process modification and assessment.

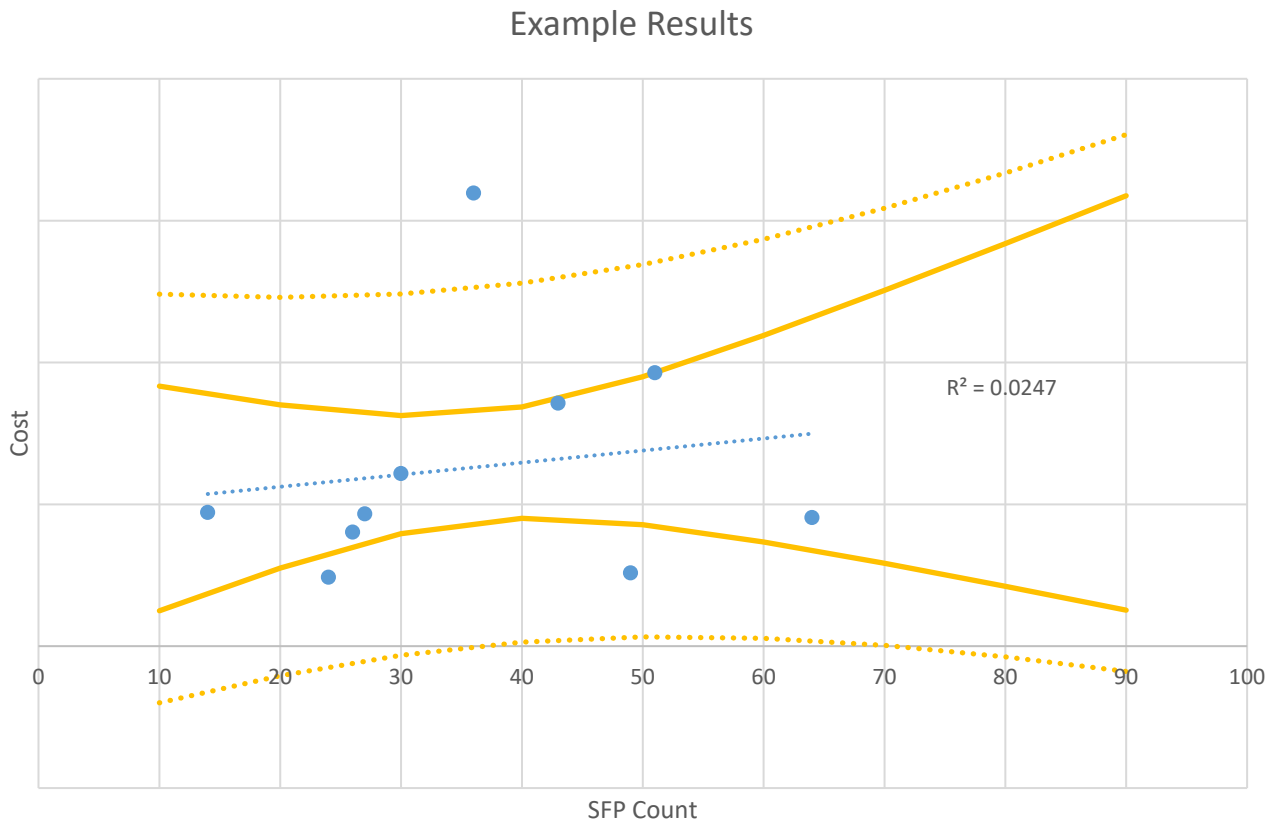
Results from NLP / SFP Analysis

- Results are real, but have been anonymized
- Available Data:
  - Lean Business Case (LBC) text was available, either in JIRA or Confluence
  - Programs can be decomposed into Solution Epics (SE) and Program Epics (PE)
  - Consistent LBC format was used for most PE's
  - Estimated cost data was also available at the SE and PE level
- Initial Findings:
  - Some correlation between SFP and cost was apparent
  - The NLP / SFP method provided a valuable way to assess an existing estimate
  - LBC author, in some cases, seemed to have an impact
  - NLP SFP was an effective method for identifying outliers
    - Prediction intervals / confidence intervals were effective
    - All outliers needed to be examined
  - Standard estimating error stats provided a data-driven way of applying risk and uncertainty

Example Results

- Data has been modified, but is representative
- Solid line = 90% confidence interval
- Dashed line = 90% Prediction Interval
- Results useful for identifying outliers
- Positive correlation is evident
- R-squared of 0.02 is low

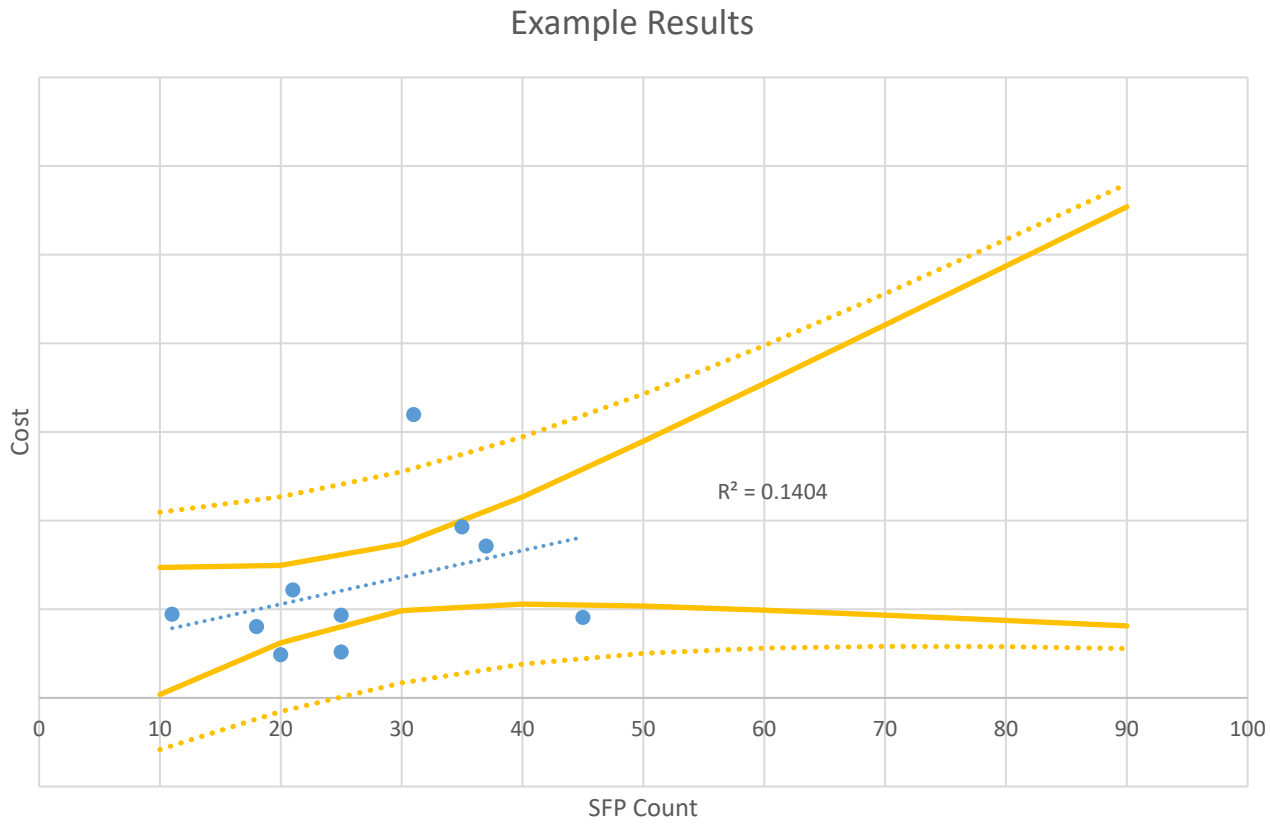
- Analysis of data indicates:
  - DHS SiSE algorithm is not counting all verbs
  - Some omitted verbs appear to be agency-specific



Example Results

- Agency-specific verb list derived from a higher-level requirements document
- Initial verb weights set at 1.0
- R-squared improves to 0.14

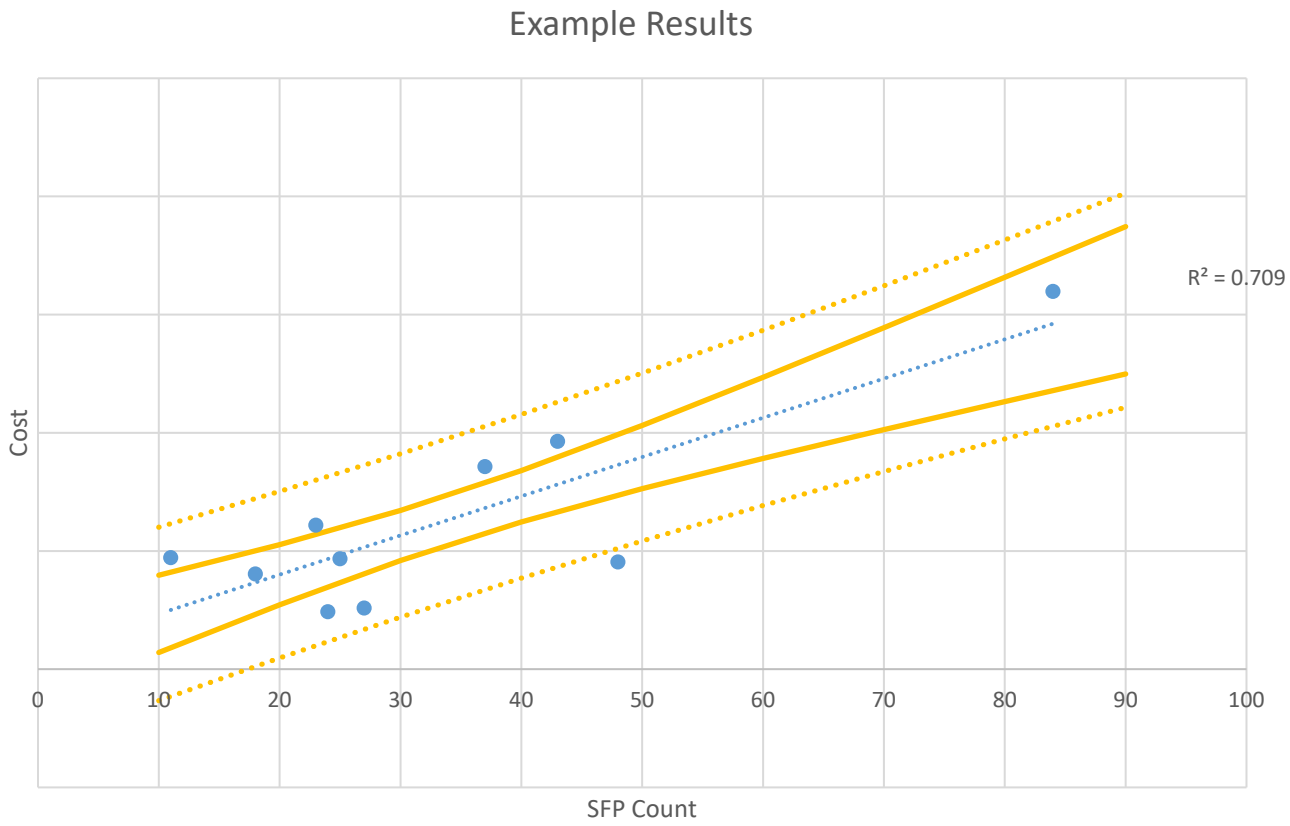
- Analysis of data indicates:
  - Only one outlier outside the PI bands
  - Outlier data point has some unique verbs



Example Results

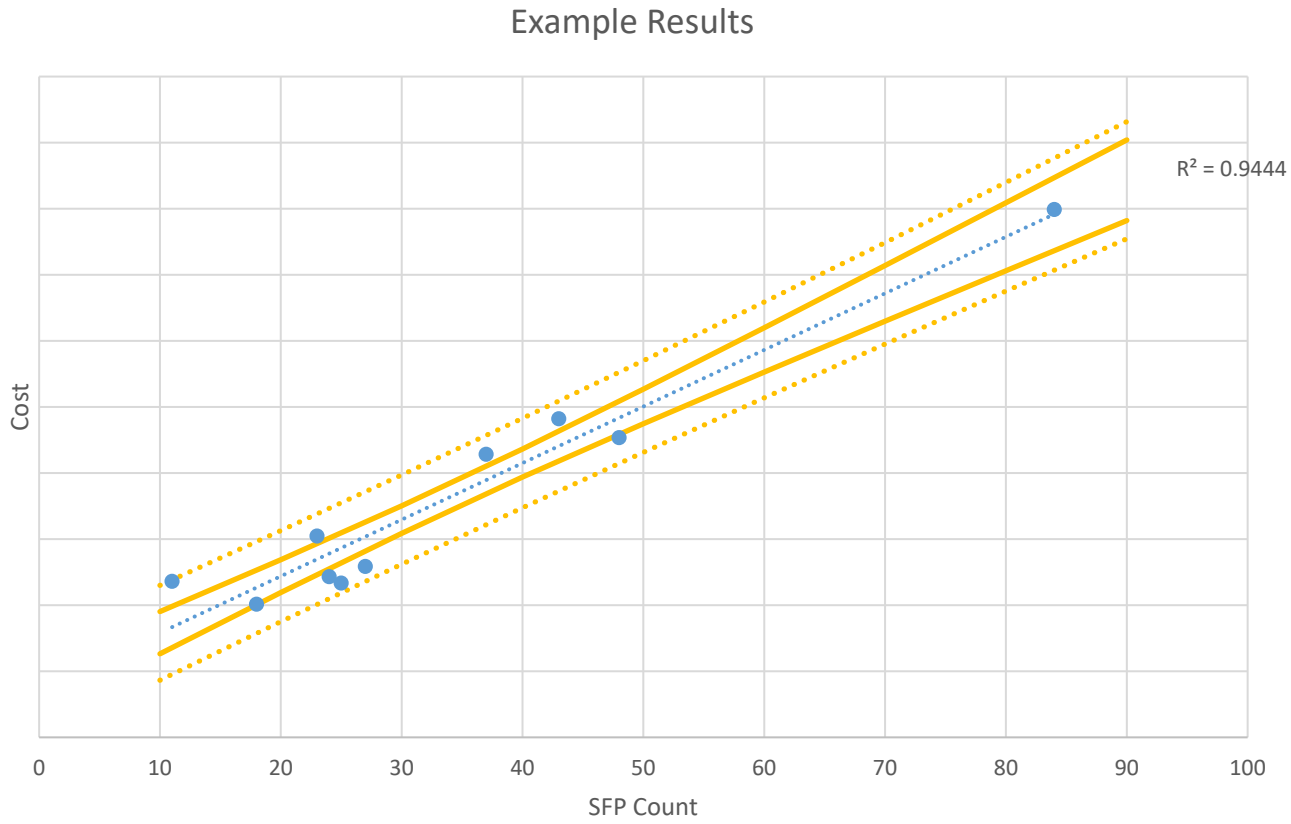
- Some verbs, assumed to be more significant, were adjusted to either 3.0 or 5.0, resulting in a 1-3-5 scale.
- Adjusted verb weights had a big impact on the outlier

- Analysis of data indicates:
  - About 1/3 of results are within the CI, 1/3 above, and 1/3 below
  - Three low-side outliers are all approximately ½ their predicted value
  - Analysis of outliers is consistent with T-shirt sizing used in the cost data
  - For the 3 low-side outliers, doubling the cost was determined appropriate **(this alone was a valuable conclusion)**



Example Results

- After doubling 3 low-side outliers, a strong correlation is shown
- Results are over-stating the true relationship because adjustments were designed to address outliers
- Risk / uncertainty is higher than implied
- Results are not usable as a general CER
- However, results were usable for their intended purpose, as a rough estimate for a specific program



Conclusions

- Current state of this analysis has some significant limitations
  - Use of estimated cost is not ideal
  - Results are more likely yielding relative sizing versus absolute sizing. Therefore not (yet) comparable with other function point sizing methods
  - In spite of high correlations seen in the last charts, the method still carries a high level of risk and uncertainty
- Some correlation between SFP and cost was apparent
- The NLP / SFP method provided a valuable way to assess an existing estimate
- NLP SFP was an effective method for identifying outliers
- The method will improve with additional data and better calibration

Next Steps

- Obtain more/better historical cost data (actuals) or actual effort data, re-evaluate results
- Explore additional calibration of agency-specific verb weights. Solver optimization?
- Determine ratio between function points counted using this method and IFPUG function points
  - Potentially would allow use of databases that are based on IFPUG FP, such as ISBSG
- Validate results with cross checks
  - Full IFPUG function point count
  - Manual SiSE FP count

Technomics
Better Decisions Faster

NLP to Generate SFP

For additional information, please contact:

David H. Brown
dbrown@technomics.net

Abstract

The need for data-driven methods for predicting software size has never been greater. This is especially true for agencies that rely on SME opinion or T-shirt based sizing as the primary method for estimating the size and cost of agile-developed software. Recent research at NSA and DHS provides a promising approach, known as Simplified Software Estimation, which generates a simple function point sizing estimate based on functional requirements. This simple function point sizing method offers significant advantages to expert-based sizing because it is a repeatable process that provides defensibility and traceability to either a high-level requirements document or requirements written as user stories. This presentation details a method for using Natural Language Processing (NLP) to automate the estimation of simple function point counts. The advantages of this approach are (1) it provides complete consistency among multiple counts and (2) it offers a way to quickly generate a count from a large dataset spanning multiple projects. Initial research shows that a NLP-derived simple function point count is an effective approach that offers great potential for improving data-driven estimates for small and large agile software development programs. In particular, the requirements documentation delivered in Lean Business Cases (LBC) at the onset of a go/no-go decision is readily available, standardized across developers/agencies, and an effective source for NLP-generated simple function point counts.

Short Abstract (75 words)

The need for data-driven methods for predicting software size has never been greater. This is especially true for agencies that rely on SME opinion or T-shirt based sizing as the primary method for estimating the size and cost of agile-developed software. This presentation offers a method for using Natural Language Processing to automate the estimation of Simple Function Point counts. Results show an effective method for relative sizing, which facilitates identification and analysis of outliers.

Bio

Mr. Dave Brown is a CCEA certified Subject Matter Expert for Technomics, Inc. He has 30 years of experience providing cost estimating and analysis to DoD and DHS clients. His primary area of expertise is in the area of IT and software estimating, with products such as life cycle cost analysis, applied cost estimating, independent cost assessment, cost research, program management support, modeling and simulation, data analysis, and database development.