

Do Specification Characteristics Correlate with Delivered Software Size?

Richard D. Stutzke (SAIC)

and

**Terry L. Palmer
(Davidson Technology, Inc.)**

21 October 2003

**Presented at the 18th COCOMO and Software Cost Modeling
Forum, Los Angeles, 21-22 October 2003.**

Richard.D.Stutzke@saic.com

Motivation

- **Requirements determine total product size**
- **A written specification usually provides the first “good” requirements**
- **Does the delivered size (SLOC) correlate with information in the specification?**

Approach

- 1. Collect data from completed projects**
 - **Specification (Microsoft Word)**
 - **Total size of the delivered code (PSLOC)**

- 2. Measure information in specification**
 - **Microsoft Word (style checker)**
 - **Automated Requirements Measurement (ARM) tool***

- 3. Look for correlations**
 - **Size**
 - **Document quality**

***from the NASA GSFC Software Assurance Technology Center.**

See <http://satc.nasa.gov/library>

“Good” Strings

- Imperative – word and phrases that command that something be provided
 - “Shall” normally dictates a functional capability
 - “Must”/”must not” establishes performance requirements/constraints
 - “Are applicable” includes, by reference, standards and other documents
 - “Will” indicates something provided from outside the specified product
 - An explicit specification usually has large numbers of imperative words
- Continuances – phrases that follow an imperative and precede low level requirements
 - Examples: “following:”, “in particular:”, and “below:”
 - Continuances indicate how well requirements are organized and structured
 - Many continuances may indicate multiple, overly complex requirements
- Directives – point to illustrative information to make requirements more understandable
 - Examples: “Figure”, “Table”, and “for example”
 - Many directives may indicate more precision in the requirements

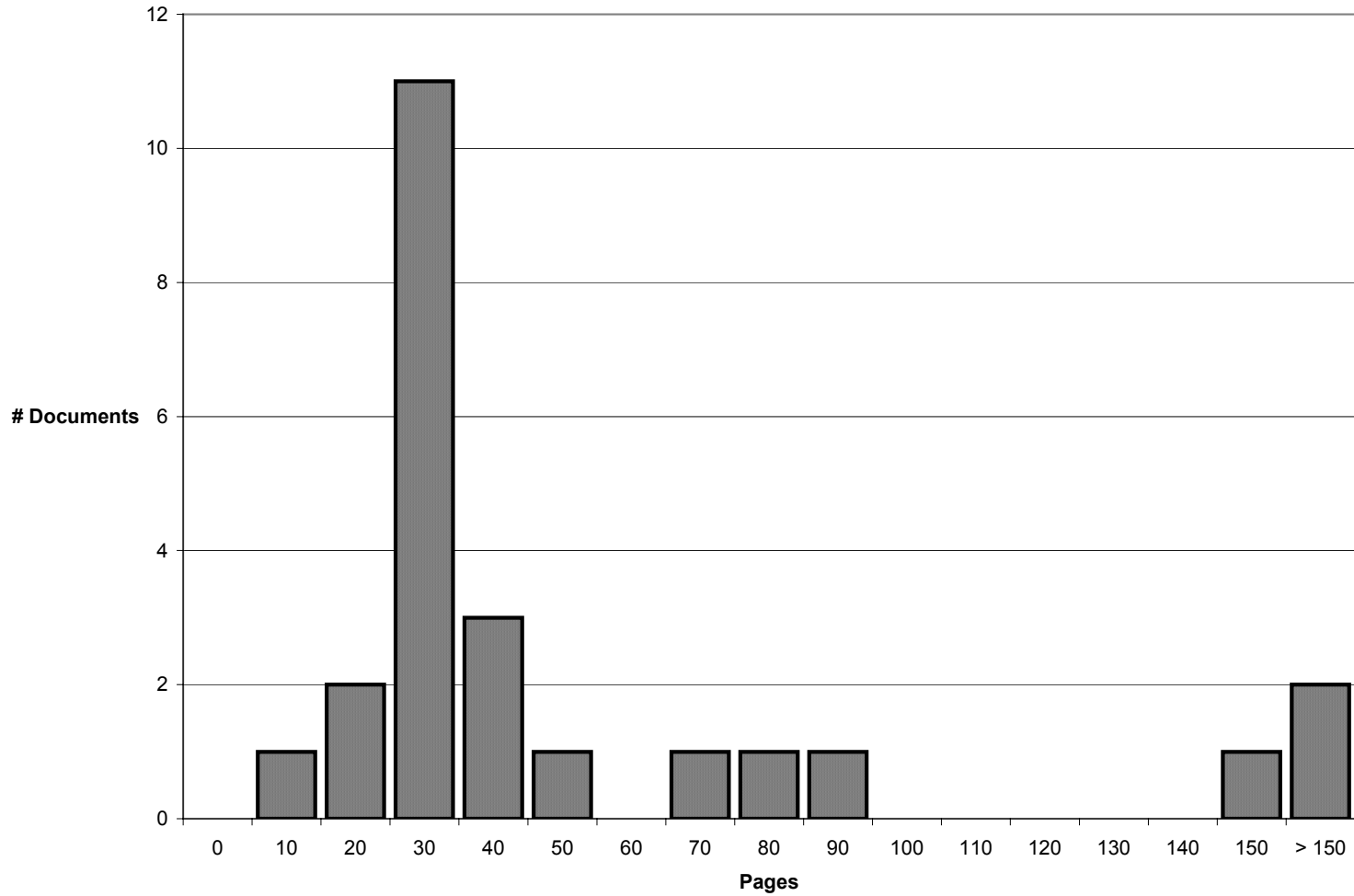
“Bad” Strings

- Weak phrases – clauses that are open to multiple interpretations
 - Examples are “adequate” and “as appropriate”
 - May indicate that the requirement is defined elsewhere
 - May indicate that the requirement is open to subjective interpretation
- Options – words that give the developer latitude in implementation of the requirement
 - Examples are “can”, “may”, and “optionally”
 - Options loosen the specification, reducing control over the produce
- Incomplete – words that clearly indicate missing information
 - Examples are “TBD” (To Be Determined) and “TBS” (To Be Supplied)
 - There indicate the document may not contain all the requirements

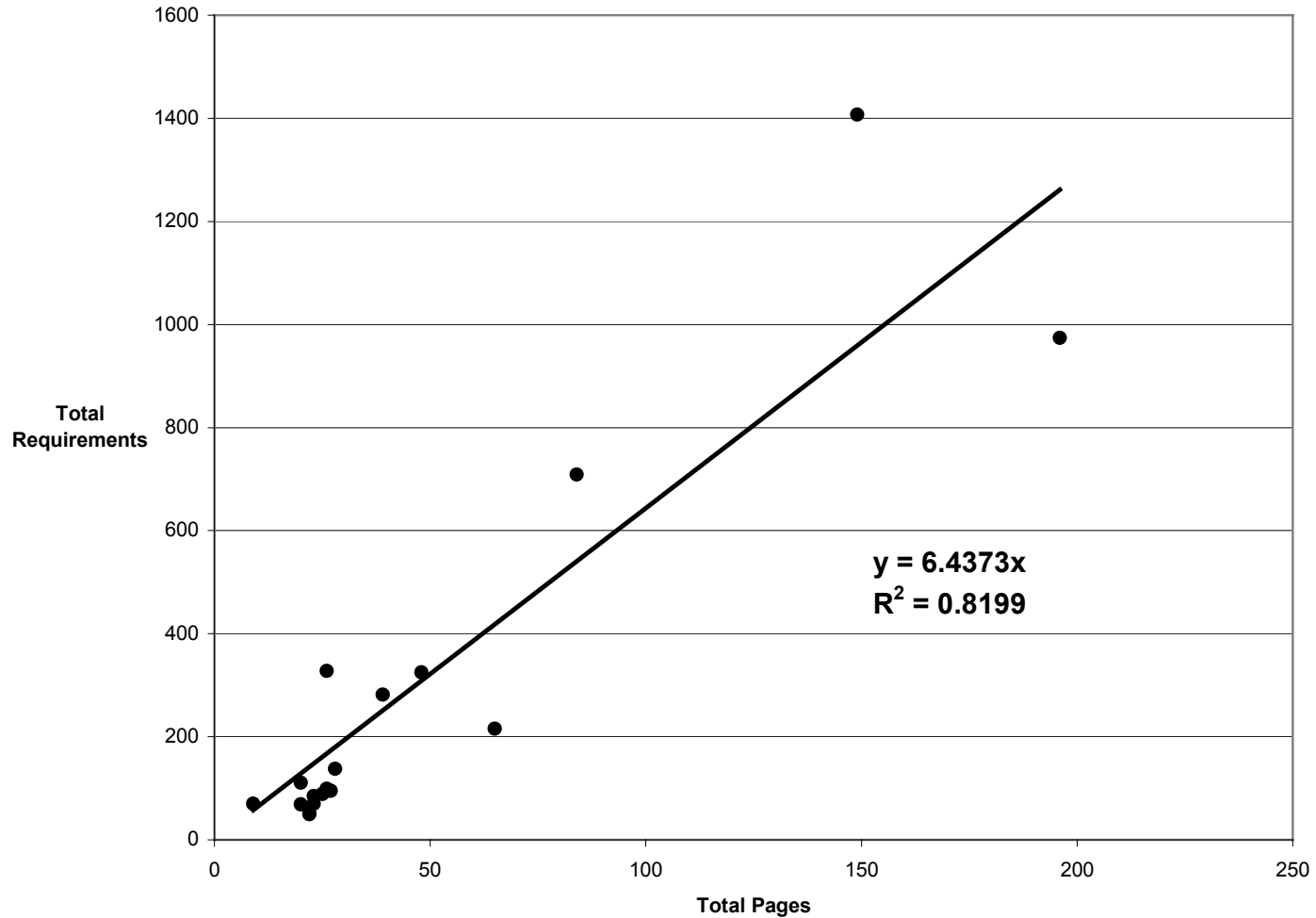
Other Measures

- Lines of text – the number of individual lines read by the program from the source file
 - Provides a size measure to normalize the data
- Structure depth – count of the numbered statements at each
 - Indicates the document's organization, consistency, and level of detail
 - High-level specifications are usually a structural depth of four or less
 - Detailed documents may have depth up to nine or so
 - A well structured document with consistent levels of detail has pyramidal shape
- Specification depth – the number of imperatives at each level of the document
 - Includes count of low level list items that are introduced at a higher level by an imperative followed by a continuance
 - It reflects the structure of the actual requirements
- Comments
 - Differences between the shape of the numbering and specification structures indicate the amount of introductory or background information in the document
 - The ratio of the total for the specification structure (the sum for all levels) to the total lines of text indicates how concise the document is

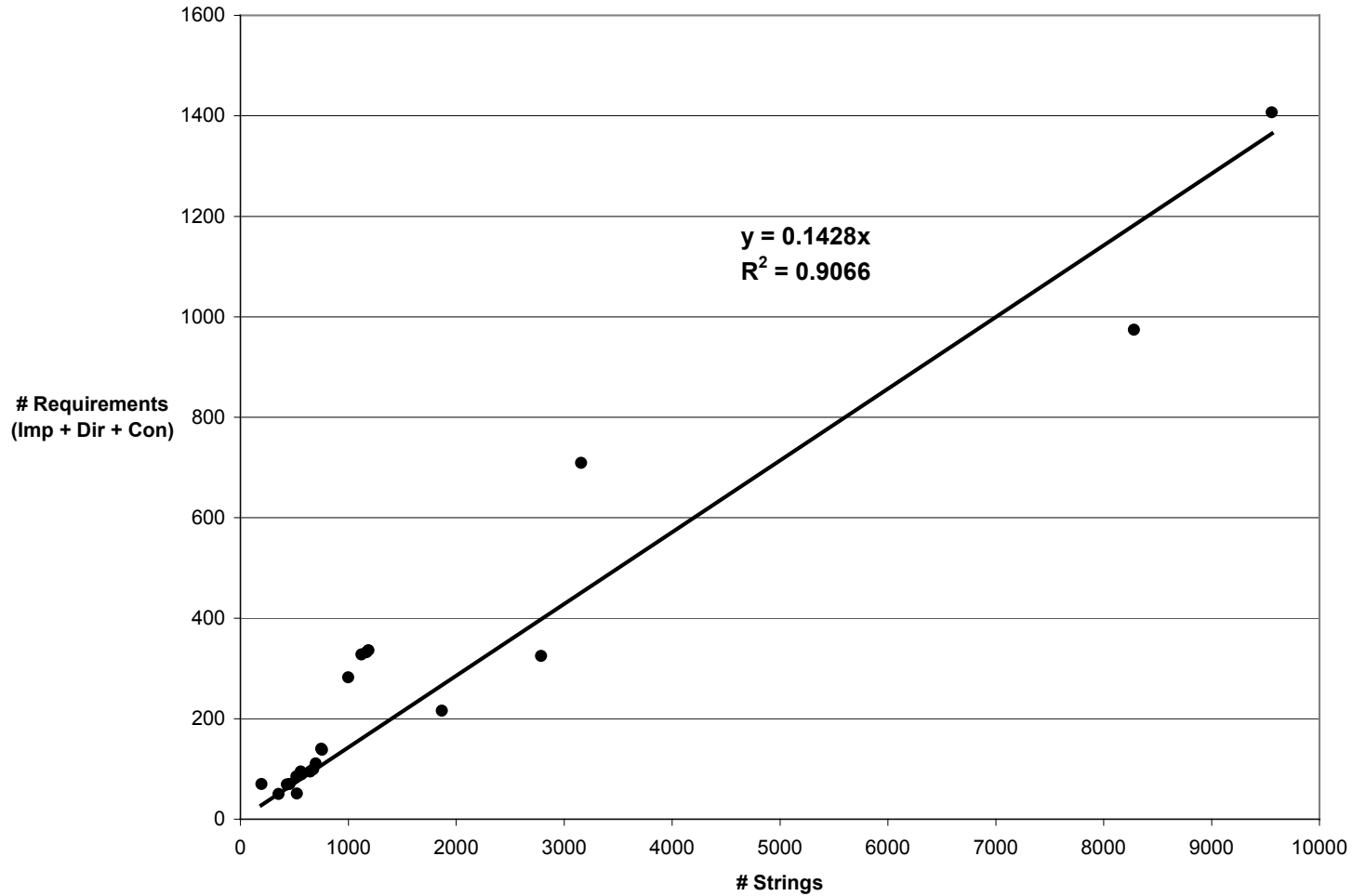
Document Size Distribution



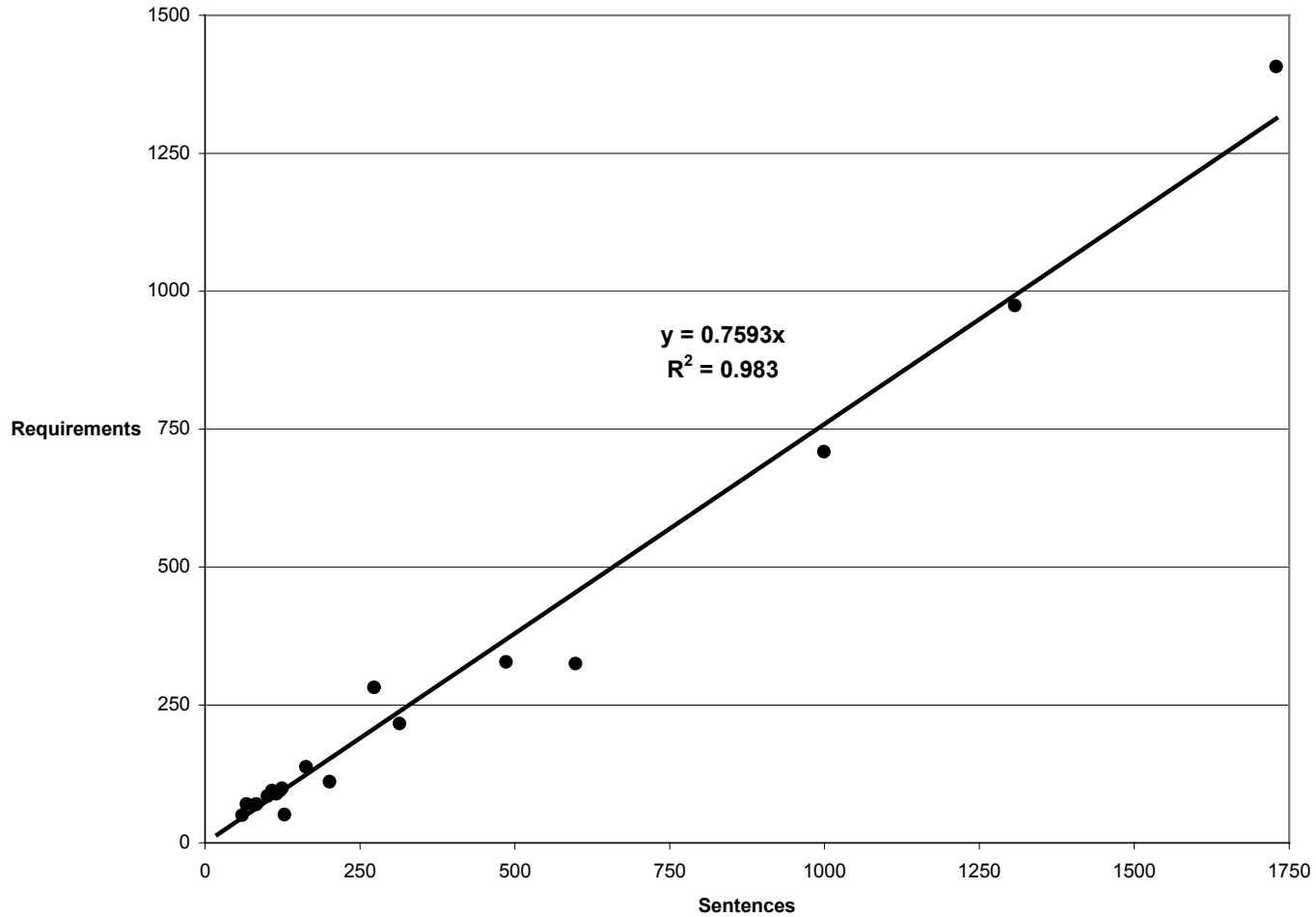
Requirements Per Page



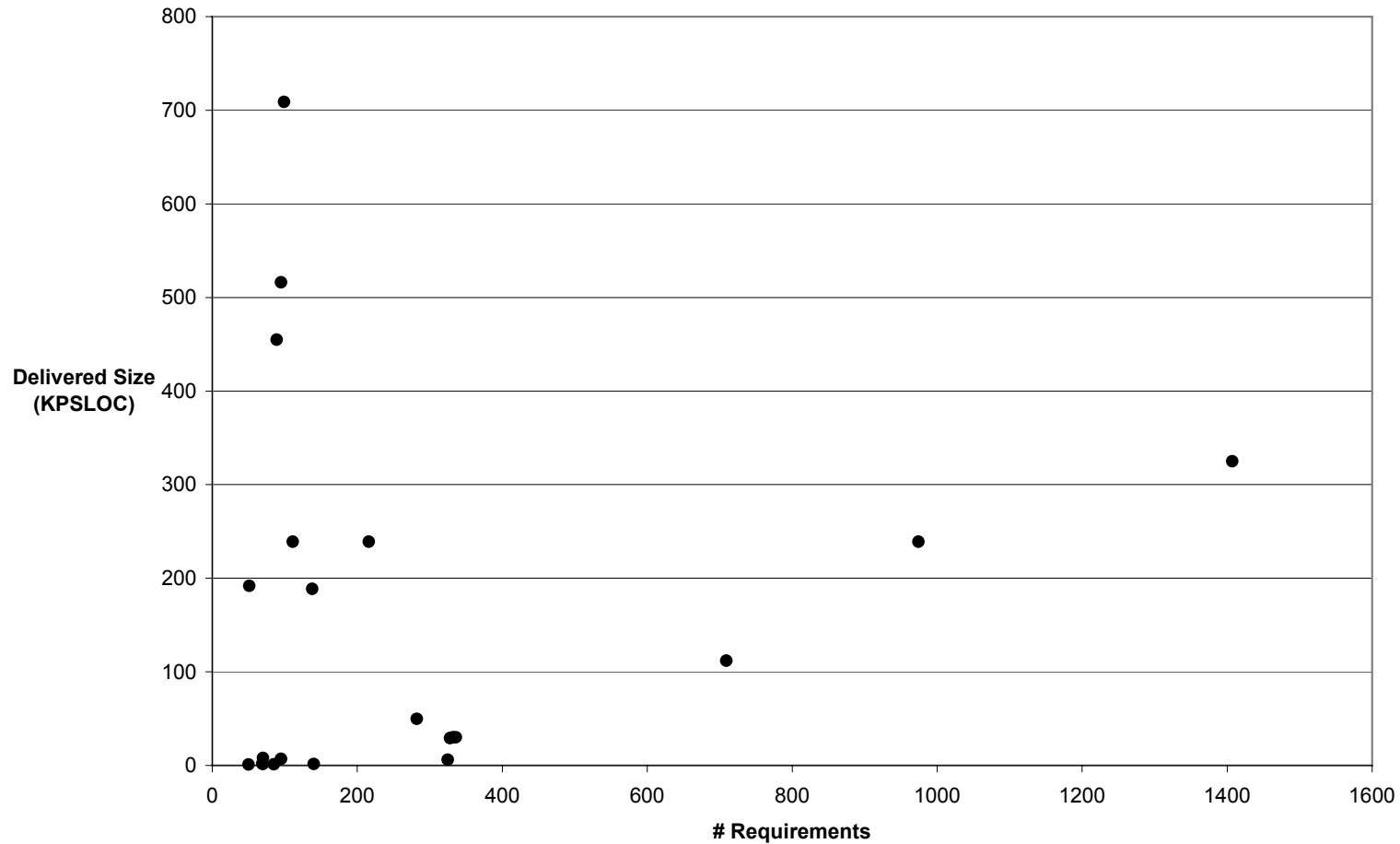
Requirements versus Strings



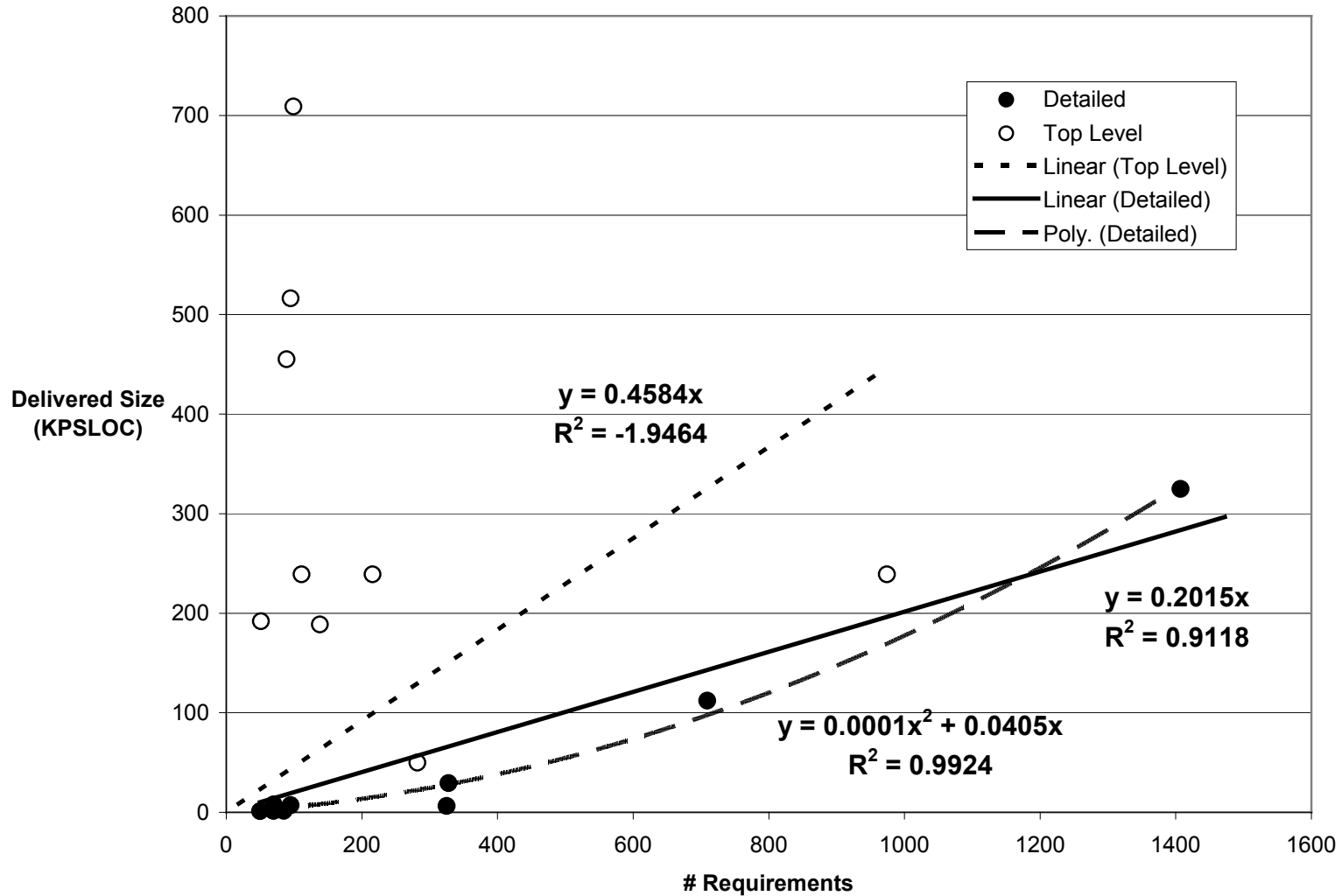
Requirements versus Sentences



Delivered Size versus Requirements



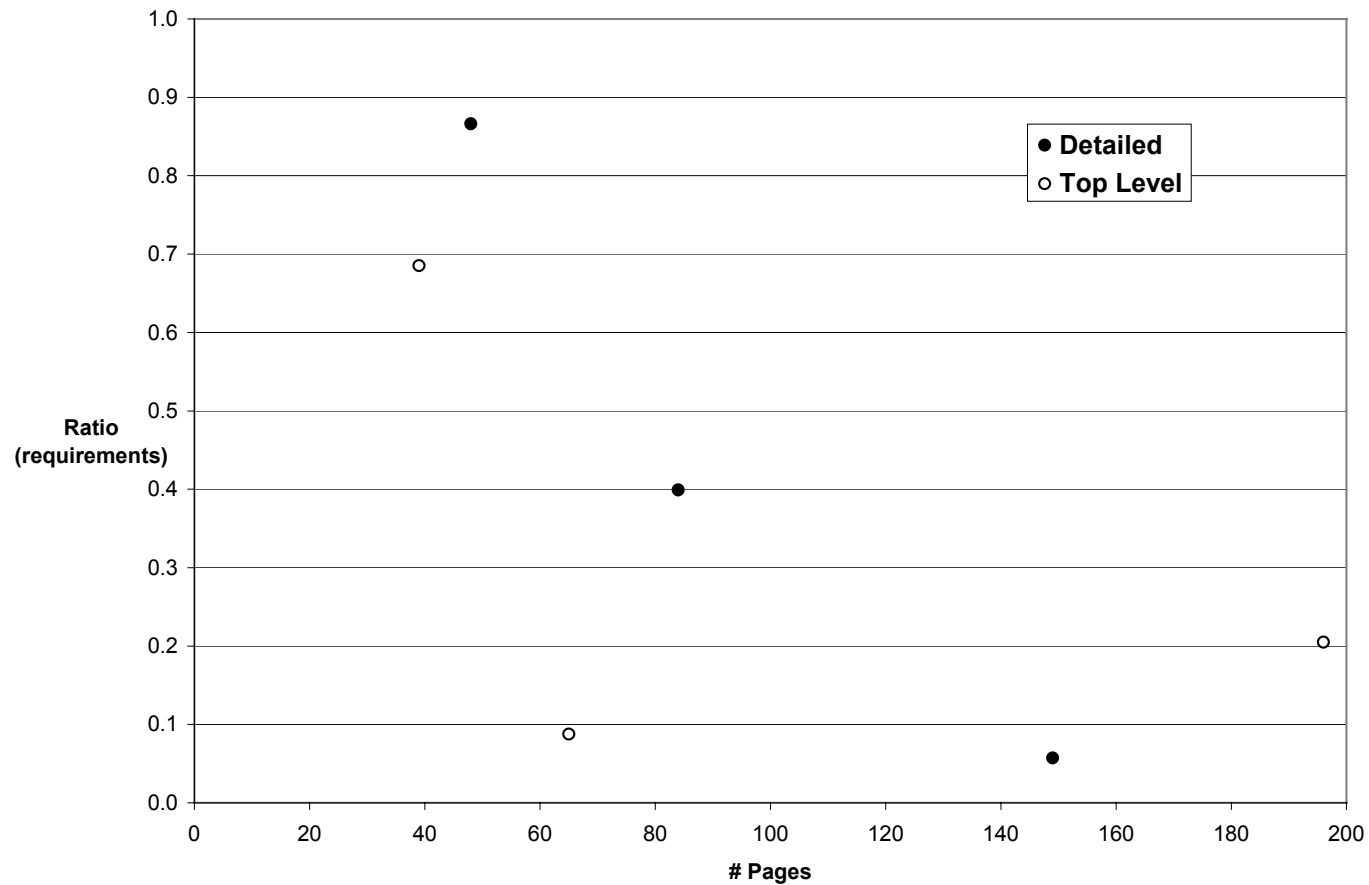
Size versus Requirements



Prediction Accuracy (Detailed Spec)

				a =	0.2015	
					Error	MRE
Project	Type	# Reqmts	Size (del)	Size(est)	(KSLOC)	Error(%)
D2	ND	70	1.5	14.1	13	840
D3	ND	328	29.2	66.1	37	126
D4	ND	95	7.0	19.1	12	173
D5	ND	70	8.0	14.1	6	76
D6	ND	50	1.0	10.1	9	908
D27	SU	85	1.4	17.1	16	1123
D28	SU	69	1.8	13.9	12	672
D32	SU	1407	325.0	283.5	-41	13
D39	SU	325	6.2	65.5	59	956
D42	SU	709	112.0	142.9	31	28
					MMRE =	492
					PRED(25) = 0.1	
					RMS = 29 KSLOC	

Requirements Ratio versus Pages



Other Factors

- **Application Complexity**
- **Choice of Programming Language**
 - Six HOLs used in our sample
 - Some products written in multiple HOLs
- **Choice of Platform**
 - Operating System services
 - COTS Middleware

Conclusions

- **Data suggests software size may be predictable from number of requirements**
 - **Our requirements include growth**
 - **Effect of document quality not studied**
- **Level of detail in specification needs to be better defined**
- **Will need to adjust size for:**
 - **Programming language**
 - **Platform**
 - **Growth**
 - **Volatility**