COCOMO III Workshop

Brad Clark, Moderator
USC Center for Systems and Software Engineering

29th International Forum on COCOMO and Systems/Software Cost Modeling

October 23, 2014

COCOMOIII Collaboration

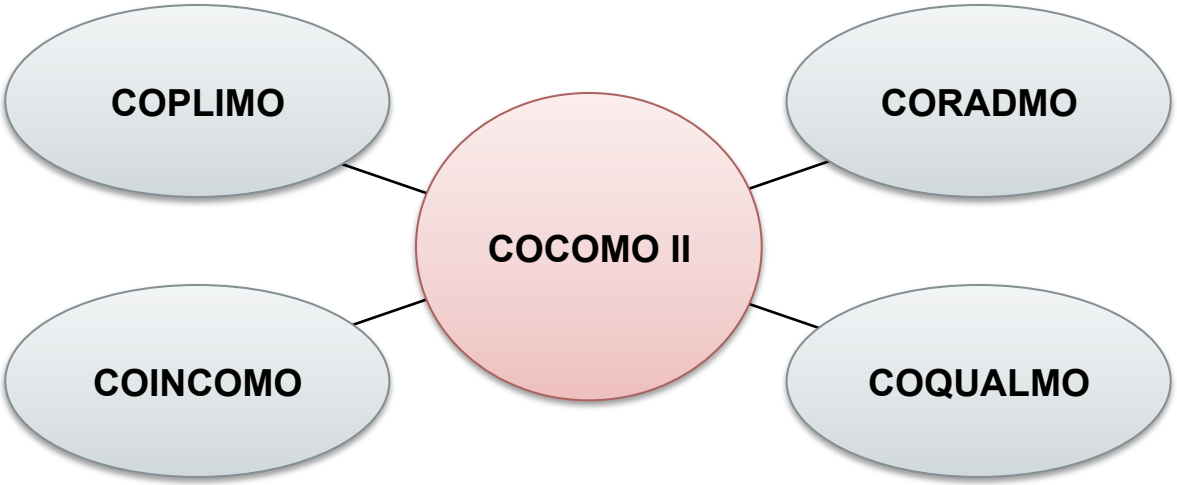
- CSSE Website based on UCC site
  http://csse.usc.edu/ucc_wp/
- Sign-up for site
  - Email notification list for status, updates, decisions, news
  - Email for newsletter
  - Download content according to access level
- Website forum to discuss issues
  - Start a topic
  - Add discussion to an existing topic

COCOMO III Charter

- Follow a “practical” approach to updating the COCOMO II model
- Objectives
  - Adjust “Nominal” ratings for Scale/Cost (S/C) drivers, if appropriate
  - Review S/C drivers
    - Combine highly correlated drivers
    - Review drivers for relevance
    - Add new S/C drivers for estimating today’s development practices (data permitting)
  - Retire older data & collect new data
  - Account for productivity decline in incremental development

COCOMO III Charter

- Objectives (continued)
  - Reduce subjectivity in S/C driver ratings with better selection description
  - Provide a Rosetta stone to map back to previous COCOMO models
  - Explore “reasonableness” of incorporation of other COCOMO-suite models (additional S/C drivers)



COCOMO III Charter

- Objectives (continued)
  - Investigate “alternative” size inputs
    - Theoretically, any consistently applied sizing method should work with S/C drivers
    - Only multiplicative and scale driver constants need to be calibrated
  - Review relevance of the Adaptation Adjustment Multiplier
    - AAF = (0.4 x DM) + (0.3 x CM) + (0.3 x IM)
      - % Design Modified (DM)
      - % Code & Unit Test Modified (CM)
      - % Integration & Test Modified (IM)
    - Software Understanding (SU)
    - Assessment and Assimilation (AA)
    - Programmer Unfamiliarity (UNFM)

COCOMO III Charter

- Objectives (continued)
  - Strategy for maintaining past COCOMO models
  - Address Maintenance / Sustainment costs
  - Develop a set of ~10 Use Cases for COCOMO III
    - Useful in making choices regarding models, etc.
    - Used by Novice Users or Advanced Users?
  - Time-boxing releases
  - Solve for one unknown given other inputs
  - Good for the next ~14 years
  - Needs to useful for the commercial world
  - Needs to be locally calibratable
  - Multi-Driver rating interactions
    - Fences to prevent invalid inputs

COCOMO III Charter

- Objectives (continued)
  - New model should be “consistent” with previous model, e.g. no surprises, no big radical change, upwardly compatible
  - Support incremental or spiral development
  - Produce distributions of estimates

Workshop Purpose

- Consider incorporating a Segmentation Strategy
- Discuss additional model forms
- Review current set of cost and scale drivers
- Collect enough feedback to create a data collection form for COCOMO III

Segmentation Strategy

COCOMO II 2000 Effort Prediction Accuracy

COCOMO II model accuracy improved when segmented by Organization

Prediction Accuracy[1]	Before Stratification by Organization	After Stratification by Organization
PRED(20)	63%	70%
PRED(25)	68%	76%
PRED(30)	75%	80%

Note:

1. PRED(X) = Y% means that Y% of the predicted values fall within X% of the actual values

Globbing Study – Oct 2007

- Create "pre-sets" for Scale/Cost driver ratings for different application domains
  - By selecting an application domain, driver ratings are selected based on the average ratings from data in the domain
  - Allows estimators to generate estimates quickly
  - Permits estimators to create a knowledge base on the basis of application domain characteristics

- Create calibration groups within the data
  - Hypothesis: projects in the database are so diverse that "local" calibration within a group should improve model accuracy and precision

Globbing Study – Driver Selection

- Business-driven drivers (these are independent of the application)
  - Analyst capability
  - Programmer capability
  - Personnel continuity
  - Application experience
  - Platform experience
  - Language and tool experience
  - Use of software tools
  - Multi-site development
  - Required development schedule
  - All Scale Drivers
- Application-driven drivers
  - Required software reliability
  - Database size
  - Product complexity
  - Developed for Reuse (maybe not)
  - Documentation match to life-cycle needs
  - Execution time constraint
  - Main storage constraint
  - Platform volatility

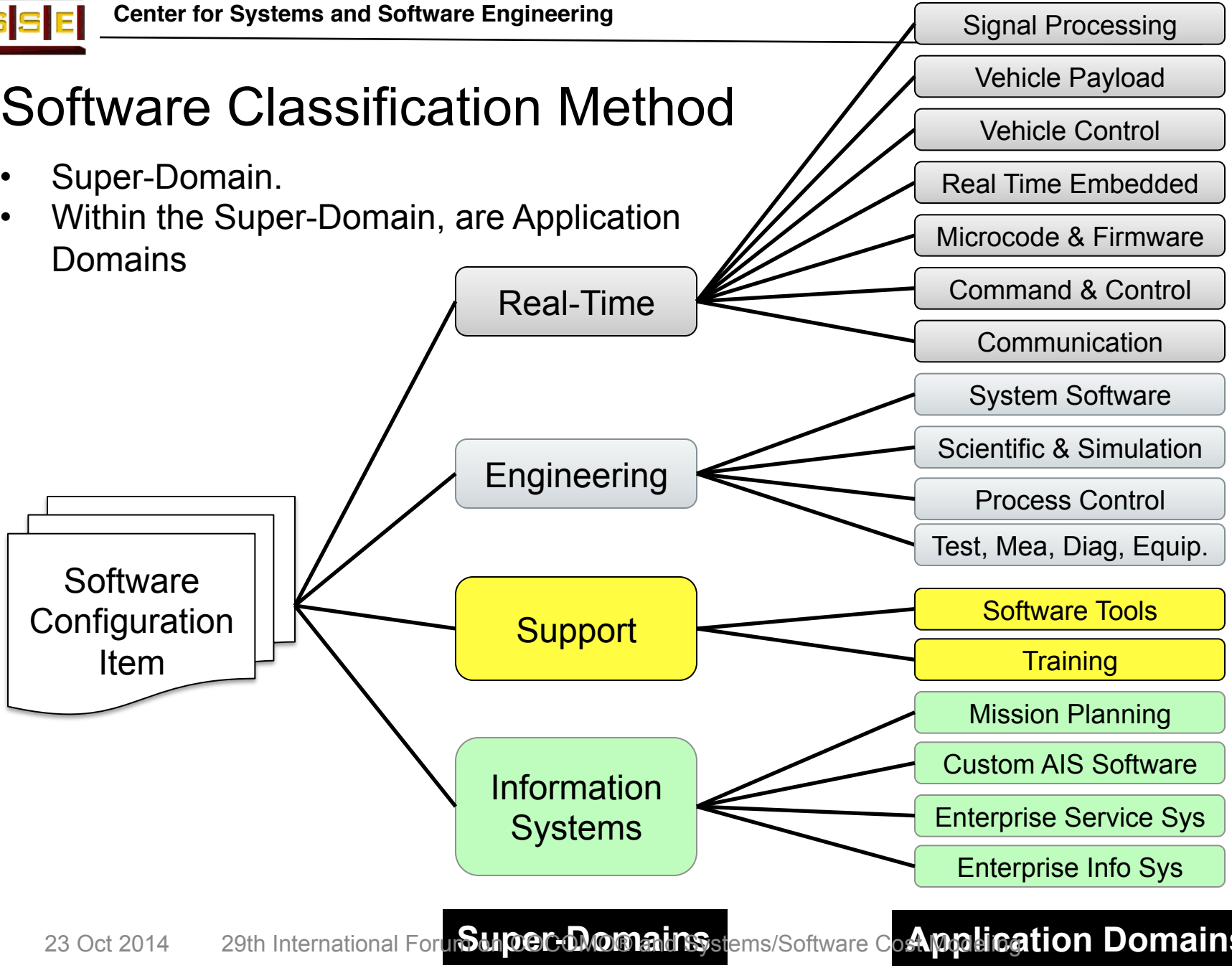
Globbing Study – Pre-set Rating Example

Globbing around application-type and size has been successfully demonstrated in other cost models.

	Glob-1	Glob-2	Glob-3	Glob-4
REL Y	VH	H	H	N
DATA	N	H	L	N
CPLX	H	N	H	L
RUSE	N	N	N	N
DOCU	N	N	N	N
TIME	VH	H	N	N
STOR	VH	H	N	N
PVOL	H	N	N	N

Software Classification Method

- Super-Domain.
- Within the Super-Domain, are Application Domains



Real-Time (RT) Super-Domain

<table>
<tr><td>Definition</td><td colspan="4">Real-Time is the most constrained type of software. These are specific solutions limited by system characteristics such as memory size, performance, or battery life. These projects take the most time and effort due to constraints, e.g.,
• May have guaranteed task execution requirements, i.e., missed deadline means catastrophic result
• May have to be compact and efficient due to limited storage capacity and high throughput requirements
• Could have very high reliability requirements (life critical, manned mission)
• Might have tightly coupled interfaces
• Program code may be imprinted on hardware devices
• May process sensor inputs and directs actuator outputs.
• Sometimes executed on special-purpose processors</td></tr>
<tr><td rowspan="2">Relevant Application Domains</td><td>Microcode & Firmware</td><td>Signal Processing</td><td>Vehicle Control</td><td>Vehicle Payload</td></tr>
<tr><td>Other Real-Time Embedded</td><td>Command & Control</td><td>Communication</td><td></td></tr>
</table>

Engineering (ENG) Super-Domain

Definition	Engineering software operates under less severe constraints than real-time software. This software may take the outputs of real-time software and further process them to provide human consumable information or automated control of devices. Or the software may perform transformation and aggregation / distribution of data. These projects take more time and effort due to multiple factors, e.g., • May have a fast response time requirement • May have more storage capacity • Might need to be highly reliable but not life critical • May have multiple interfaces with other systems • May implement complex algorithms, models or protocols • Program code can be modified and uploaded • Executes on general purpose processors that may be embedded in special purpose hardware			
Relevant Application Domains	System	Process Control	Scientific and Simulation	Test, Measurement, Diagnostic and Evaluation

Support (SUP) Super-Domain

Definition	Support software assists with operator training and software testing. This software has few constraints, e.g., • Has to have an acceptable response time most of the time • Less limited by storage or throughput • Less stringent reliability requirement • Software restarts are acceptable • Fewer interfaces • Relatively low complexity algorithms, models or protocols • Program code can be modified and uploaded • Executes on general purpose processors on general purpose computer boards			
Relevant Application Domains	Training	Software Tools		

Automated Information Systems (AIS) Super-Domain

Definition	Automated information system software provides information processing services to humans or software applications. These applications allow the designated authority to exercise control and have access to typical business / intelligence processes and other types of information access. These systems also includes software that facilitates the interface and control among multiple COTS / GOTS software applications. This software has few constraints, e.g., • Must have acceptable response time • Fewer storage or throughput constraints • Must be reliable enough to prevent data loss • May consist of a single COTS / GOTS solution or multiple products coordinated with customer software • Algorithms, models and protocols are well understood • Code may not be available for modification • Software restarts are acceptable • Executes on commercial processing hardware			
Relevant Application Domains	Mission Planning	Enterprise Service Systems	Custom AIS Software	Enterprise Information Systems

Workshop Discussion

- Identify the interaction between "applicable" COCOMO II Scale/Cost Drivers and each Super-Domain

	RT	ENG	SUP	AIS
PREC				
FLEX				
RESL				
TEAM				
PMAT				
RELY				
CPLX				
RUSE				
DATA				
…				

Model Form

COCOMO II Models

- Three models comprise COCOMO II
  - Application Composition
    - Object Points (screens, reports and third-generation language modules)
  - Early Design: Size + 7 predictor variables

$$PM = A \cdot Size^{E} \times \prod_{i=1}^{7} EM_i + PM_{Auto}$$

  - Post-Architecture: Size +22 predictor variables

$$PM = A \cdot Size^{E} \times \prod_{i=1}^{17} EM_i + PM_{Auto}$$

$$E = B + 0.01 \times \sum_{j=1}^{5} SF_j$$

$$PM_{Auto} = \frac{\text{Adapted SLOC} \times \left(AT / 100 \right)}{ATPROD}$$

Model Calibration

$$PM = A \times Size^{E} \times \prod EM$$

$$\text{where } E = B + 0.01 \times \sum SF$$

A: Multiplicative constant
B: Exponent constant
EM: Effort Multiplier for each Cost Driver
SF: Scale Factor for each Scale Driver

How would a segmentation strategy impact calibration?

- Use pre-sets only?
- Different A and B values?
- Different A, B, EM and SF values?

Should a value for E be derived for each segment?

Globbing Results

In the Globbing study discussed earlier, COCOMO II constants, A and B, were created for each Glob

Size Bucket		Glob-1	Glob-2	Glob-3	Glob-4
2-25 KSLOC	n	6	10	6	10
	A	3.53	4.27	2.88	1.90
	B	0.87	0.75	0.92	1.07
	PRED(25)	83%	80%	100%	60%
	PRED(20)	67%	70%	100%	50%
25-100 KSLOC	n	13	15	15	14
	A	7.14	5.42	6.44	2.05
	B	0.69	0.76	0.72	0.98
	PRED(25)	69%	80%	73%	93%
	PRED(20)	69%	80%	73%	86%
100+ KSLOC	n	18	8	6	9
	A	4.58	9.52	1.20	8.76
	B	0.87	0.73	1.04	0.68
	PRED(25)	72%	75%	50%	100%
	PRED(20)	56%	75%	50%	100%

Size Input

$$PM = A \times Size^{E} \times \prod EM$$

Percentage adjustments to effort-size relationship

Size represents the amount of work, i.e. the problem scope

Should the size input be limited to only source lines of code? Would the model still work with other well defined and consistently measured size types?

- Requirements
- Unadjusted Function Points
- Use Case Points
- Story Points
- COSYSMO size types

SLOC Sizing Model

$$\text{Size} = \left(1 + \frac{\text{REVL}}{100}\right) \times \left(\text{New KSLOC} + \text{Equivalent KSLOC}\right)$$

$$\text{Equivalent KSLOC} = \text{Adapted KSLOC} \times \left(1 - \frac{\text{AT}}{100}\right) \times \text{AAM}$$

$$\text{where AAM} = \begin{cases} \dfrac{\text{AA} + \text{AAF} \times (1 + [0.02 \times \text{SU} \times \text{UNFM}])}{100}, & \text{for AAF} \leq 50 \\ \dfrac{\text{AA} + \text{AAF} + (\text{SU} \times \text{UNFM})}{100}, & \text{for AAF} > 50 \end{cases}$$

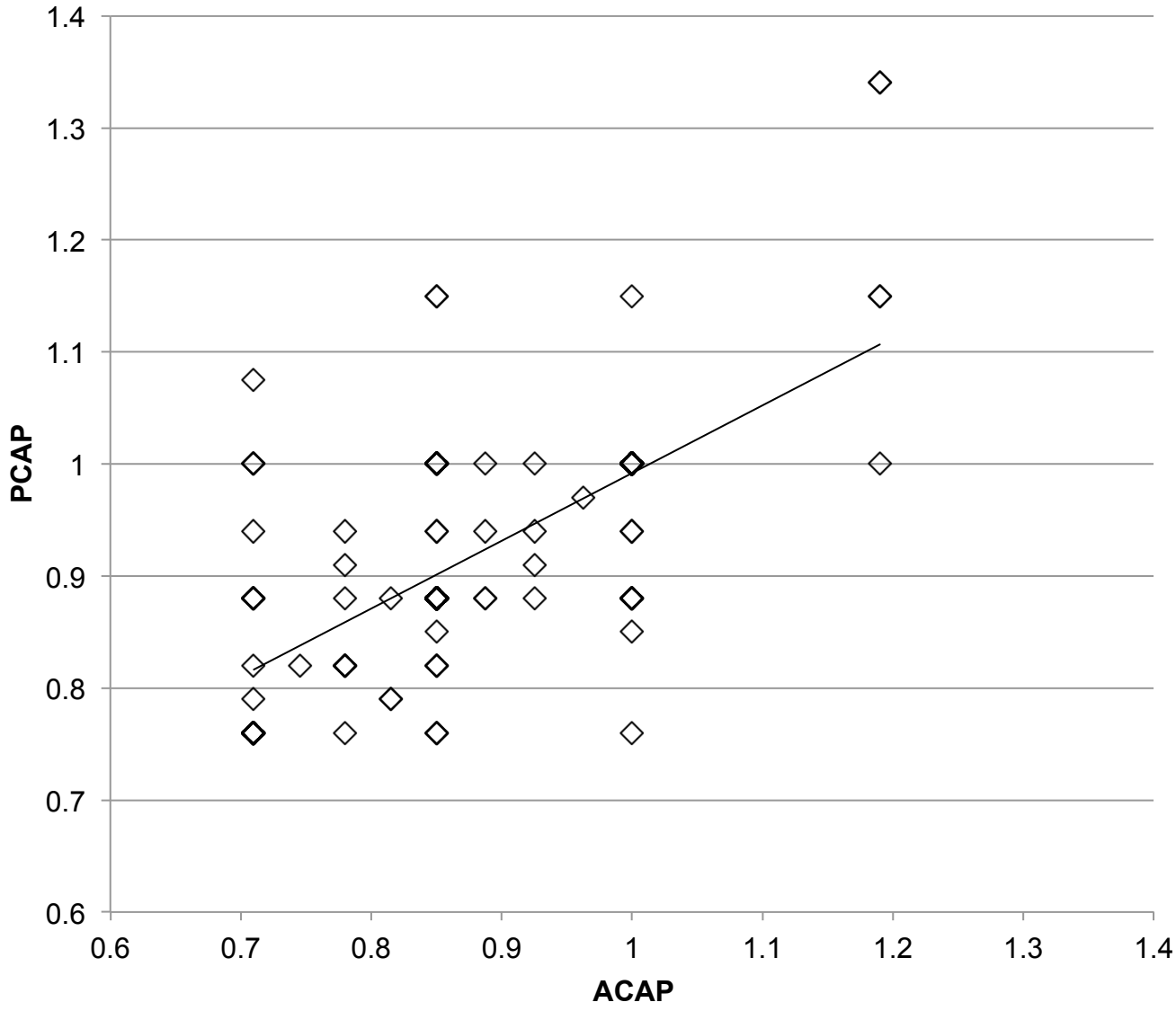
$$\text{AAF} = (0.4 \times \text{DM}) + (0.3 \times \text{CM}) + (0.3 \times \text{IM})$$

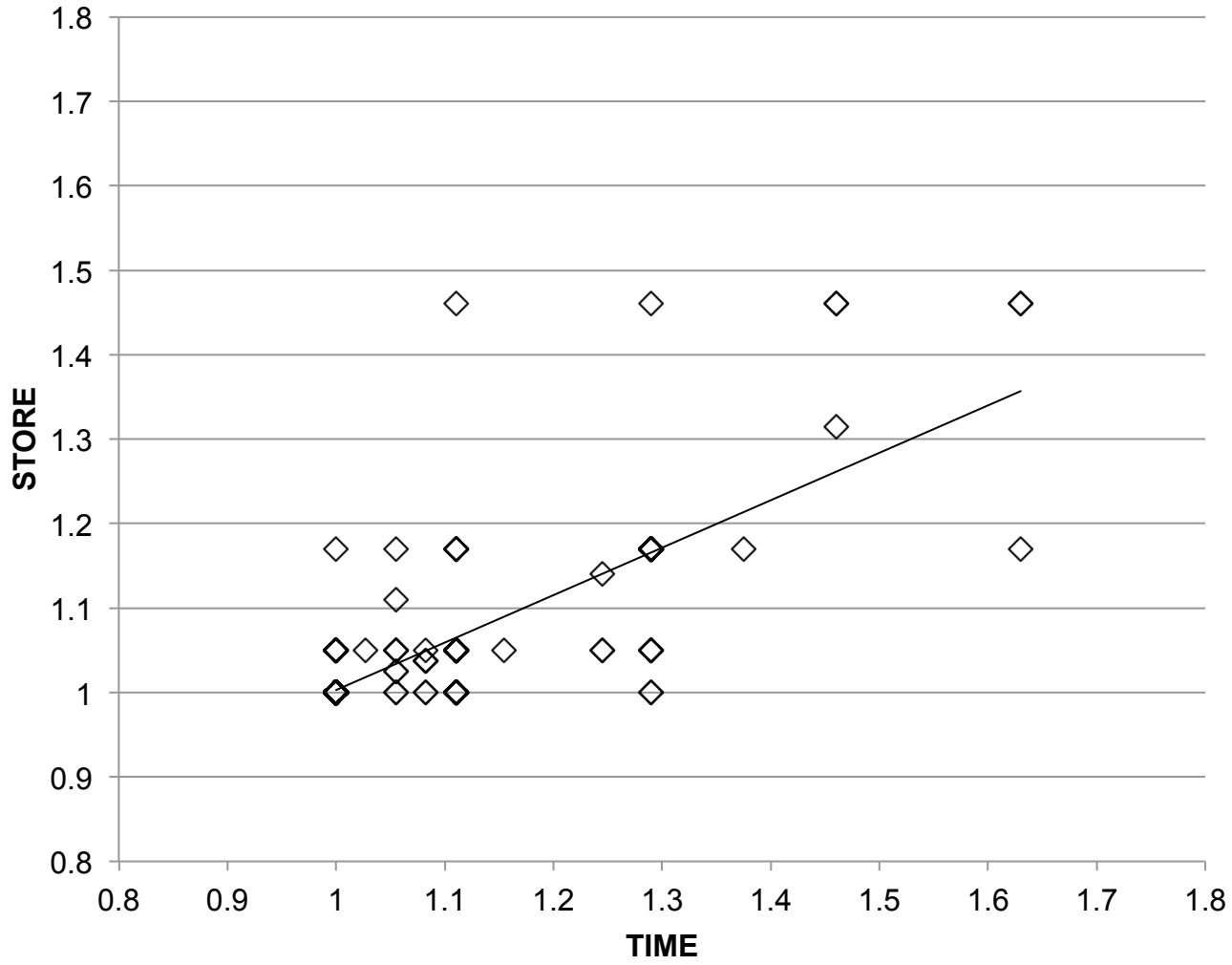
Are all of these factors still relevant?
How are different code types treated (modified, reused, auto-generated)?

Cost & Scale Driver Review

Scale/Cost Driver Interactions

- Pairwise correlation analysis
- Multi-collinearity analysis
- Possible opportunity for consolidation
- COSYSMO parameter harmonization?

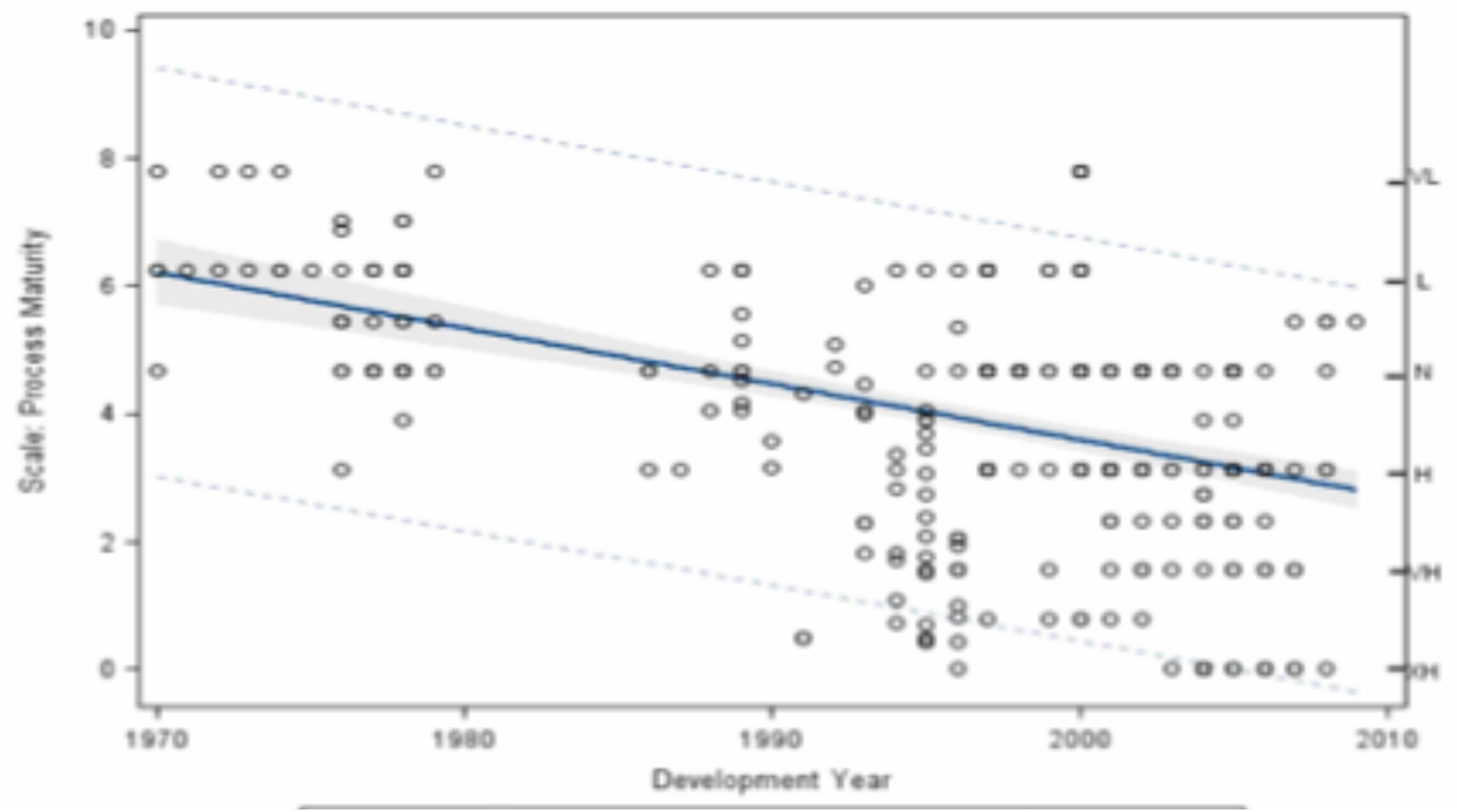

1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
PCAP
0.6
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
ACAP


1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
STORE
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
TIME

Adjusting Scale/Cost Driver Nominal

- Ratings for drivers shifting away from nominal
- See following charts

COCOMO II Data: Process Maturity Trends



Use of Software Tools Rating Trends

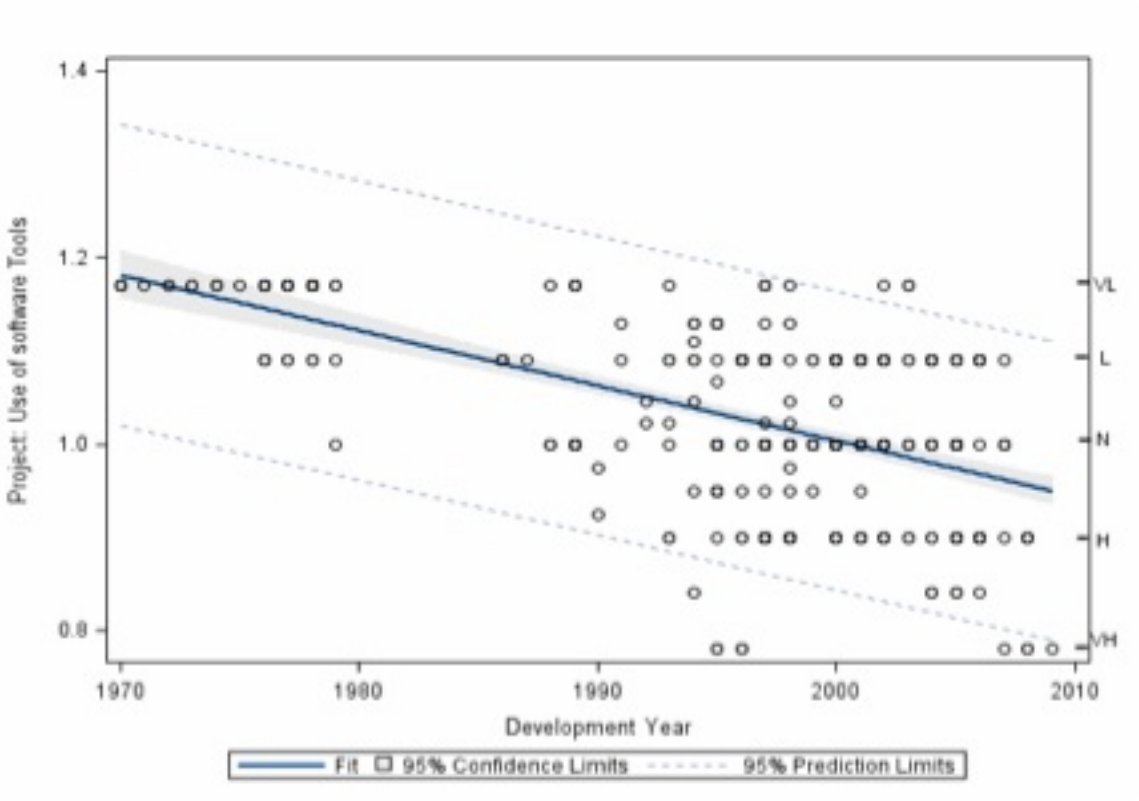


Figure 5. Linear Regression of TOOL on Year (p<0.0001)

Platfom Experience Rating Trends

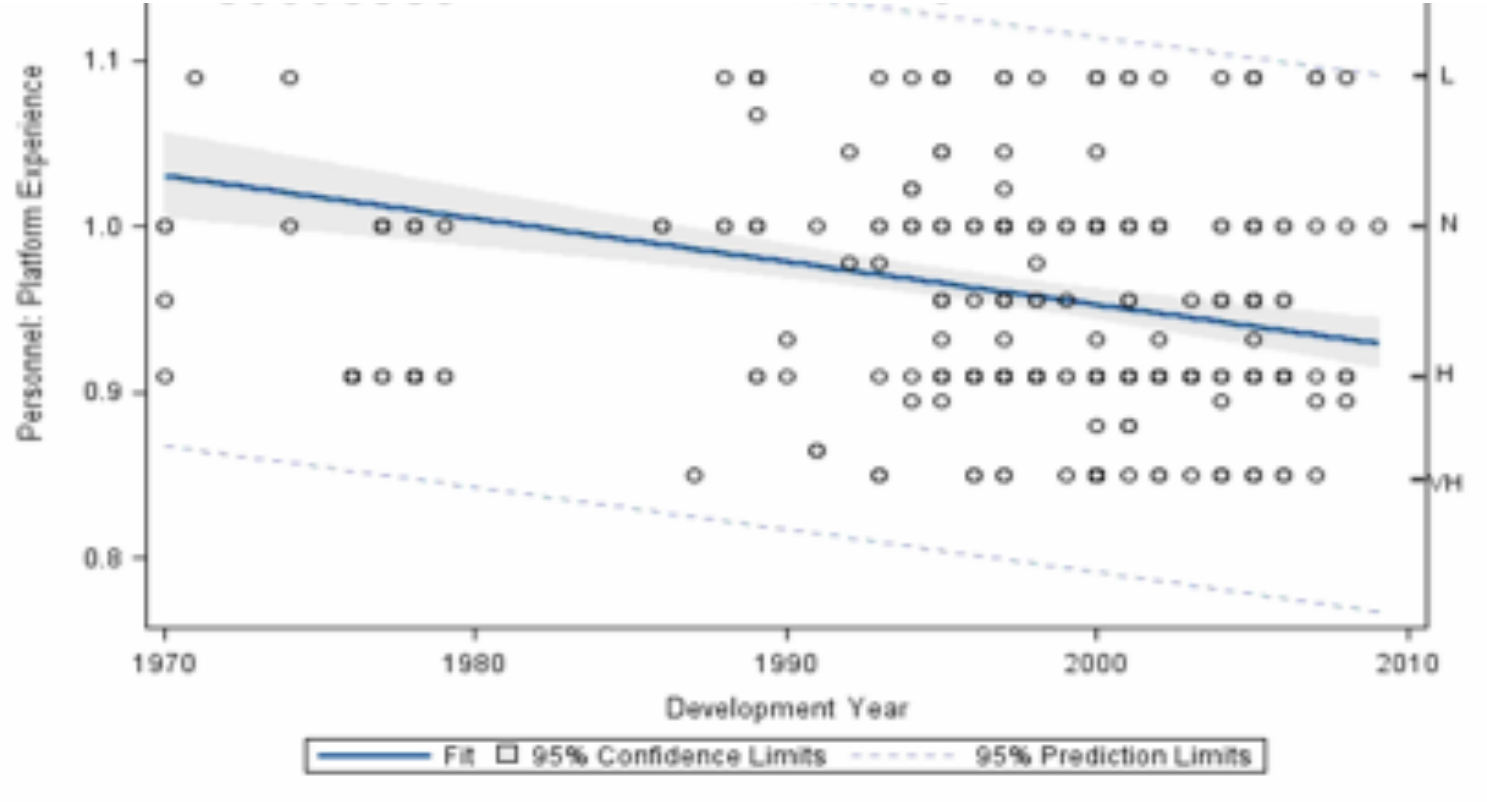
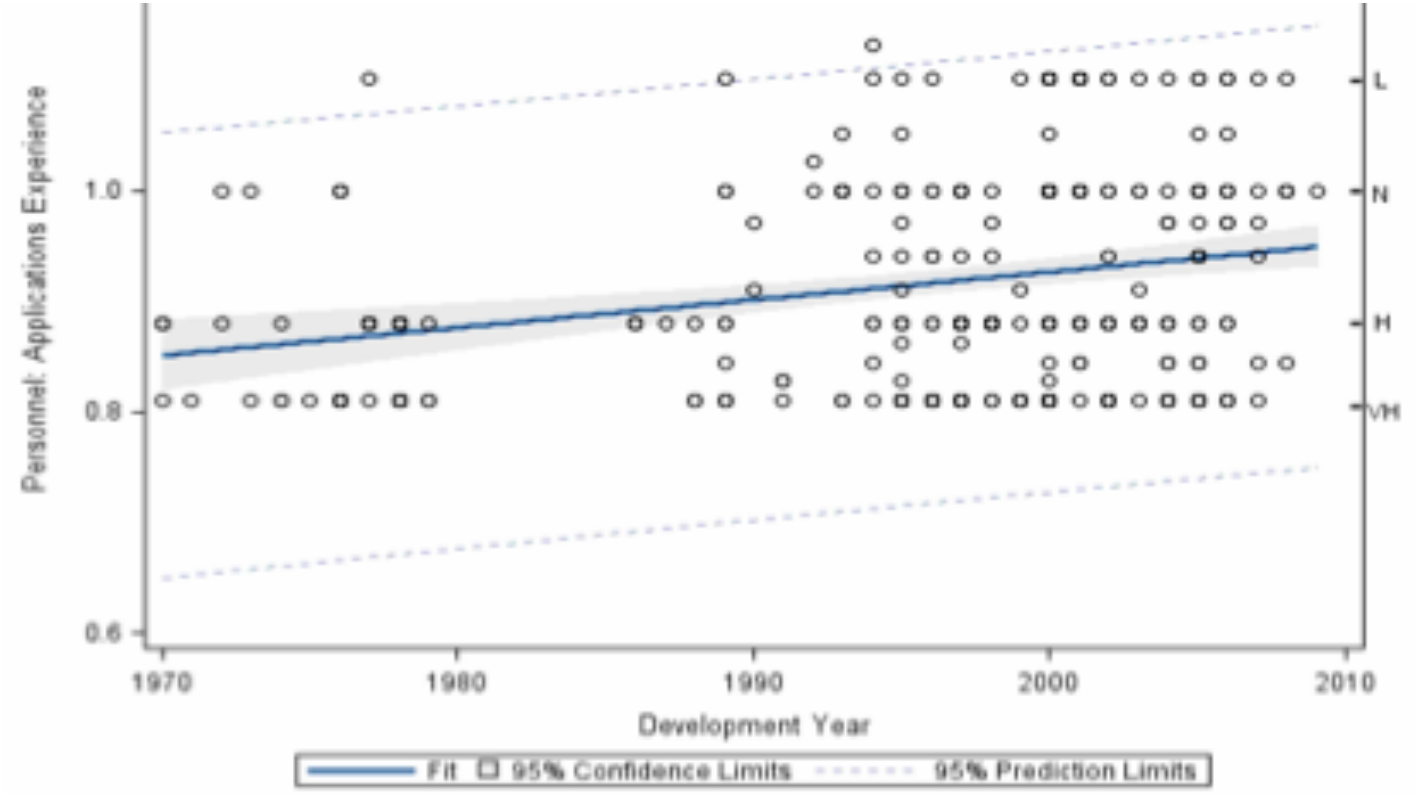


Figure 7. Linear regression of PLEX on Year (p<0.0001)

Application Experience Rating Trends



Language & Tool Experience Rating Trends

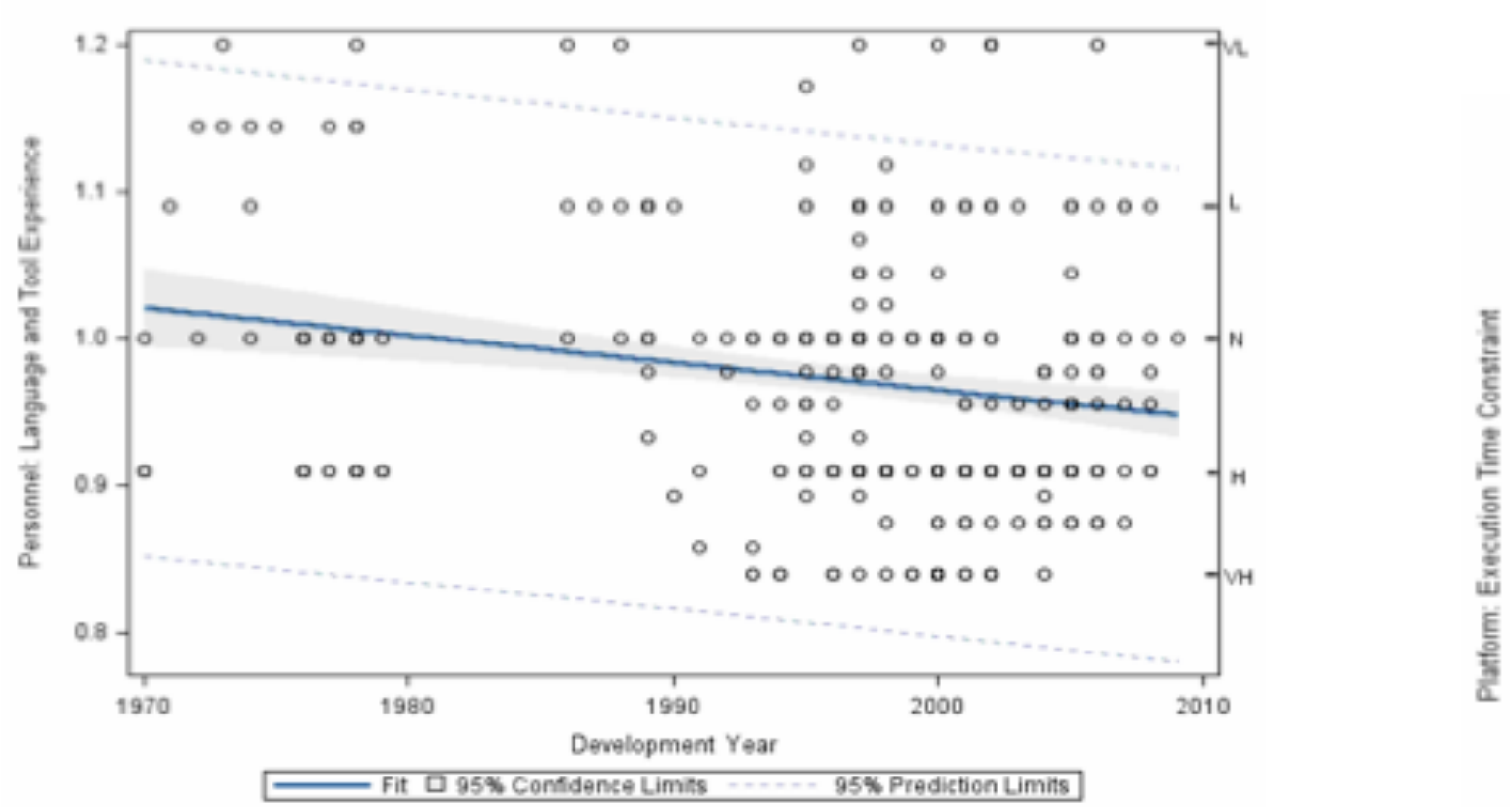


Figure 9. Linear regression of LTEX on Year (p<0.0001)

Storage Constraint Rating Trends

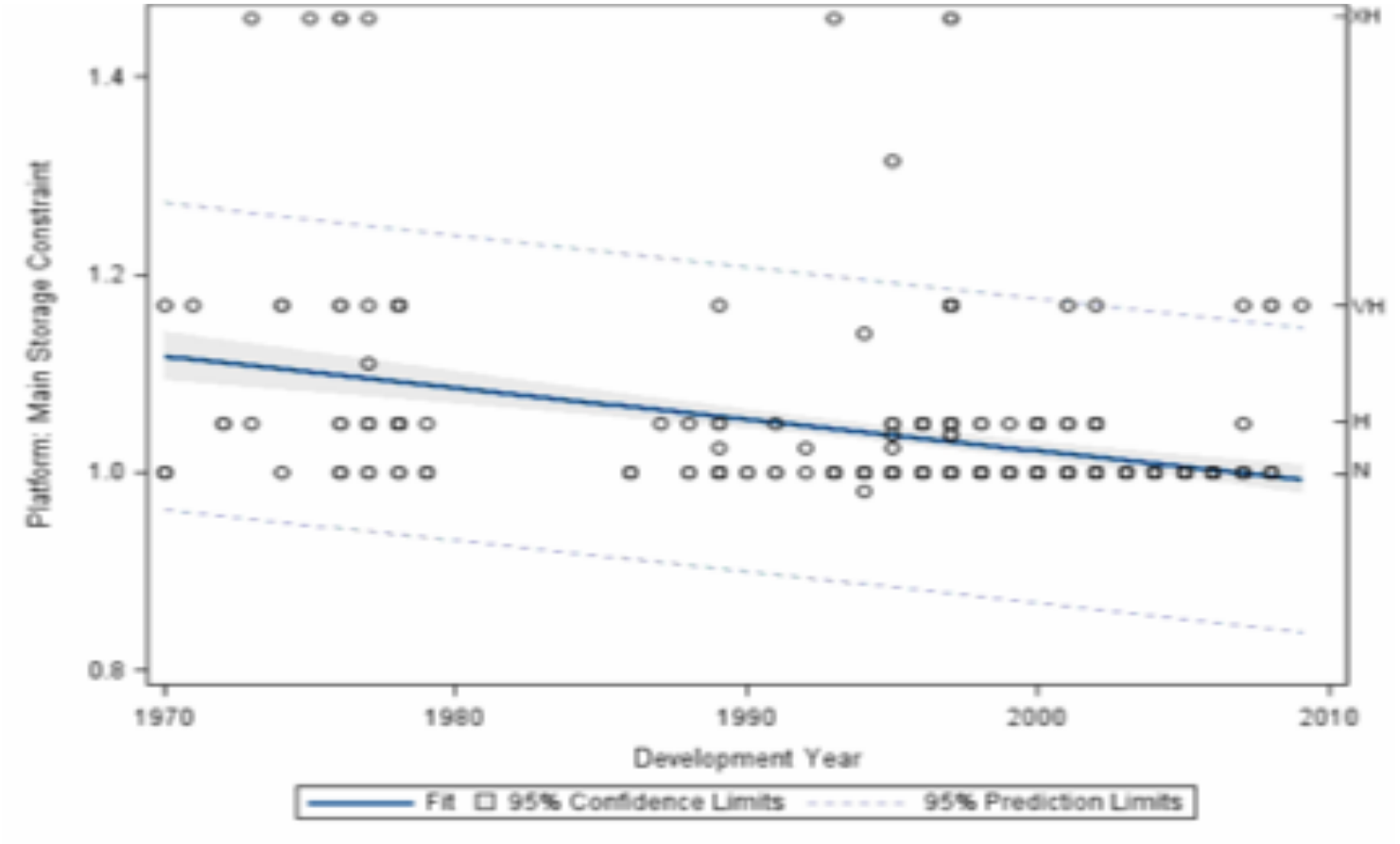


Figure 10. Linear regression of STOR on Year

Execution Time Constraint Rating Trends

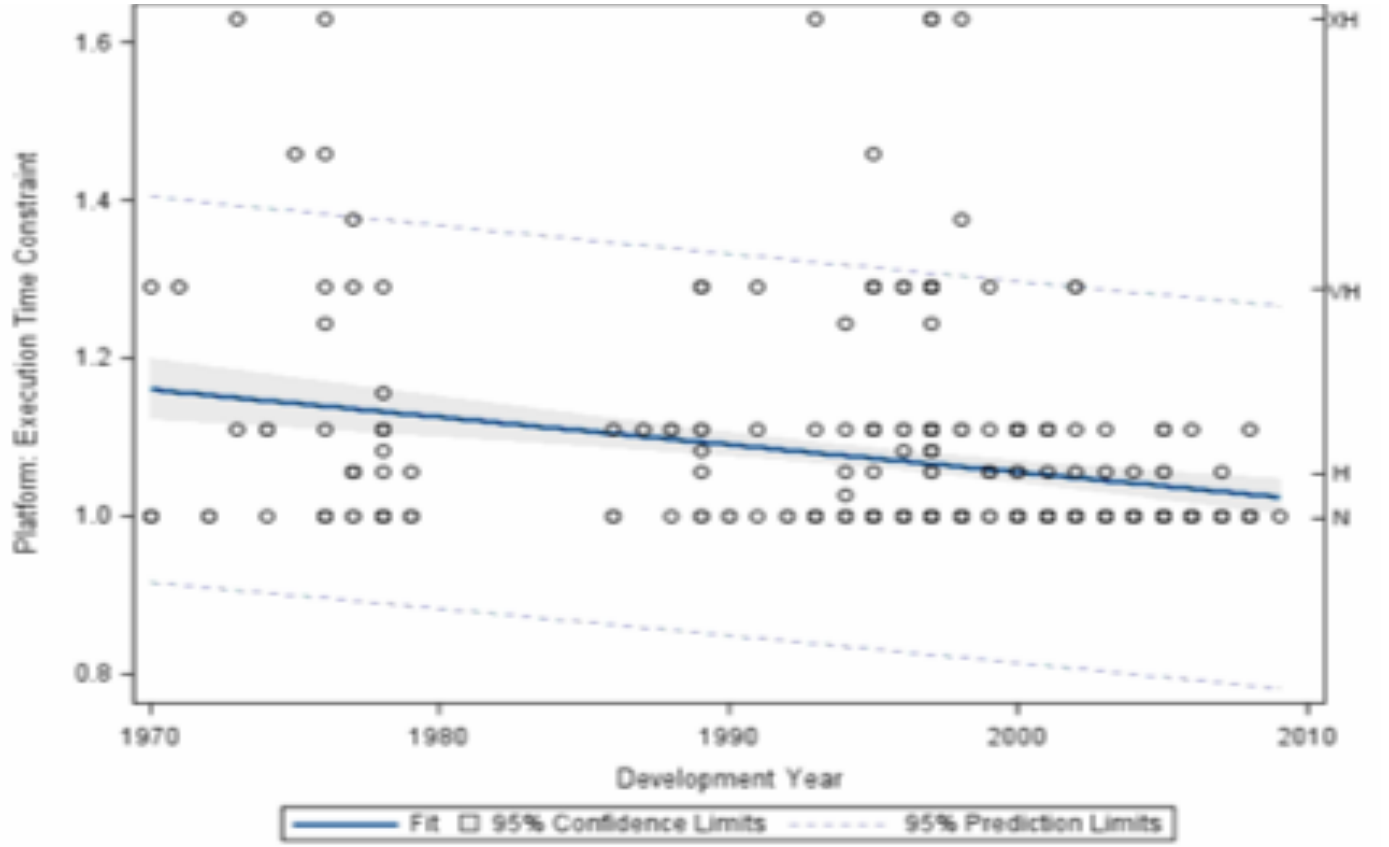


Figure 11. Linear regression of TIME on Year

Shift Definition for PMAT Nominal

Scale Factors	Very Low	Low	Nominal ⇨	High	Very High	Extra High
PMAT	The estimated Equivalent Process Maturity Level (EPML) or					
	SW-CMM Level 1 Lower	SW-CMM Level 1 Upper	SW-CMM Level 2 ⇦	SW-CMM Level 3	SW-CMM Level 4	SW-CMM Level 5

Solution:

- Shift Nominal label over to the High label
- Shift the definition of High over to Nominal

What problems would this shift cause?

New Scale/Cost Drivers

- Parameters from other COCOMO Suite models
  - CORADMO
  - COQUALMO
  - COINCOMO
  - COPLIMO
- Review drivers for relevance
- Add new S/C drivers for estimating today's development practices (data permitting)
  - Process Methodology: indicate the type of processes that are planned for the development, e.g. plan-driven, rapid development, architected agile, formal methods, COTS-Intensive, etc
  - *See other model parameters handout*

COCOMO Suite Parameters

- CORADMO (For smaller/shorter projects)
  - Parameters
    - Very High-level Languages
    - Development Process Reengineering
    - Collaboration Efficiency
    - Prepositioning Assets
- COQUALMO (Impact of rework)
  - Parameter
    - Disciplined Methods
  - Collect defect injection & removal data

COCOMO Suite Parameters

- COPLIMO (relative cost of reuse)
  - Parameters
    - Product-specific fraction of software product's size (PFRAC)
    - Adapted-software fraction (AFRAC)
    - Reused-software fraction (RFRAC)
    - Relative cost of writing for Reuse (RCWR)
- COINCOMO (Incremental development)
  - Collect data by increments, e.g.
    - Start & end dates
    - Increment size
    - Increment effort

New Driver Candidates

- End-user expertise
  - The degree of usability for an effective end-user experience, e.g. how simple does it have to be