

Predicting Software Quality Using Organizational Metrics: An Empirical Case Study

Presented by: Victor R. Basili
University of Maryland

Authors: Nachiappan Nagappan (MSR Seattle),
Brendan Murphy (MSR Cambridge),
and Victor R. Basili (University of Maryland)

Outline

- Motivation and Related Work
- Organizational Metrics
- Case Study
- Conclusions and Future Work

Motivation and Related Work

Motivation

- **Software engineering** is a complex engineering activity
- **Commercial software development** is done by teams comprising of several individuals ranging from the tens to the thousands, working via an organizational structure reporting to a manager or set of managers
- A common focus has been to **identifying problem prone components early** in the development process using software metrics
- Early indicators of software quality are beneficial for
 - determining the reliability of the system,
 - estimating and prioritizing work items,
 - focusing on areas that require more testing, inspections
 - identifying “problem-spots” to manage for unanticipated situations

Motivation

- Often such estimates are obtained from measures like code churn, code complexity, code coverage, code dependencies, etc.
- But these studies often ignore one of the most influential factors in software development, specifically “people and organizational structure”.
- This interesting fact serves as our main motivation:
 - *How does organizational complexity influence quality?*
 - *Can we identify measures of the organizational structure?*
 - *How well do they do at predicting quality, e.g., do they do a better job of identifying problem components than earlier used metrics?*

Motivation

- **Conway's Law:** “organizations that design systems are constrained to produce systems which are copies of the communication structures of these organizations.”
- **Brooks** argues in the Mythical Man Month that the product quality is strongly affected by that structure.
- With the advent of **global software development** where teams are distributed across the world the impact of organization structure on Conway's law and its implications on quality is significant
- There has been **little empirical evidence** regarding the relationship/ association between organizational structure and direct measures of software quality like failures

Motivation

- We propose a set of **eight measures that quantify organizational complexity** capturing issues such as:
 - organizational distance of the developers;
 - the number of developers working on a component;
 - the amount of multi-tasking developers are doing;
 - the amount of change to a component within the context of that organization etc.
- Using these measures we **empirically evaluate the efficacy of the organizational metrics** to identify failure-prone binaries in Windows Vista.

Related Work

- **Software Organizational Studies**

- Herbsleb and Grinter look at Conway's law from the perspective of global software development from a team organizational context identifying barriers to team coordination
- Herbsleb and Mockus formulate and evaluate an empirical theory of coordination towards understanding engineering decisions from the viewpoint of coordination within software projects. This paper is one of the closest in scale, size and motivation to our study
- Mockus et al. how different individuals across geographical boundaries contribute towards open source projects (Apache and Mozilla).
- Perry et al. discuss the larger development picture, which encompasses organizational and social as well as technological factors

Related Work

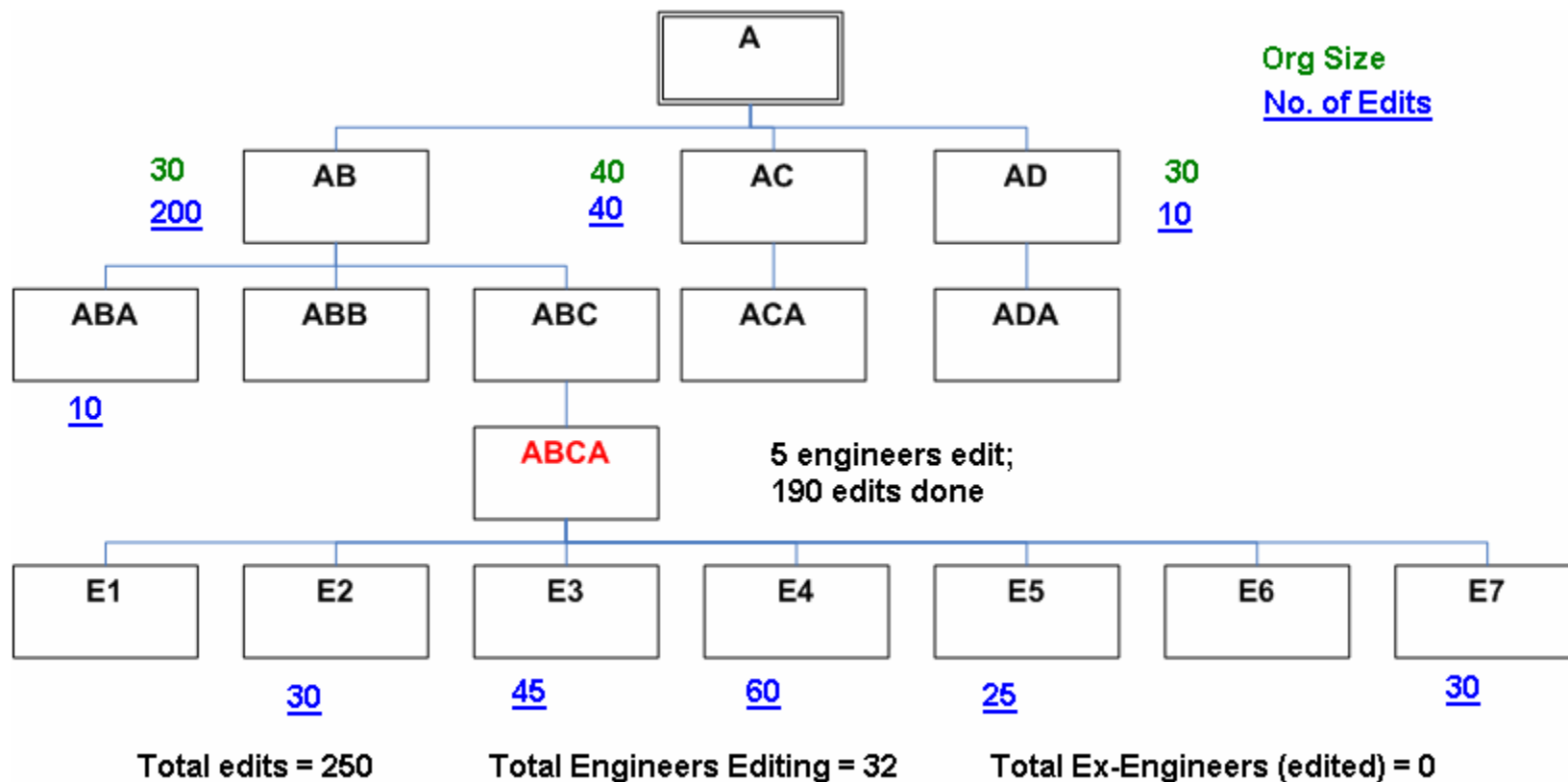
- **Software Metrics and Faults/Failures**
 - **Code Churn**
 - **Code Complexity:**
 - **Code Dependence**
 - **Code Coverage**
 - **Combinations of metrics**
 - **Pre-release bugs**

Contributions of this Study

- An organizational metric suite targeted at the software domain.
- Methodology to systematically build predictors for failure-proneness using organizational metrics
- An investigation of whether organizational metrics are better predictors of failure-proneness compared to traditional metrics
- quantification of institutional knowledge in terms of developer experience on prior versions of Windows to set a baseline for other systems and applications outside of Microsoft.
- One of the largest studies of commercial software—in terms of code size (> 50 Million lines of code), team sizes (several thousand), and software users (several Million).

Organizational Metrics

Example Company XYZ and a particular binary



1. The more people who touch the code the lower the quality (NOE)

- **Number of Engineers (NOE):** This is the absolute number of unique engineers who have touched a binary and are still employed by the company.
- **Implication:** The more people who touch the code, the higher the chances of defective code as there is a higher need for coordination amongst the engineers, the more chance for miscommunication

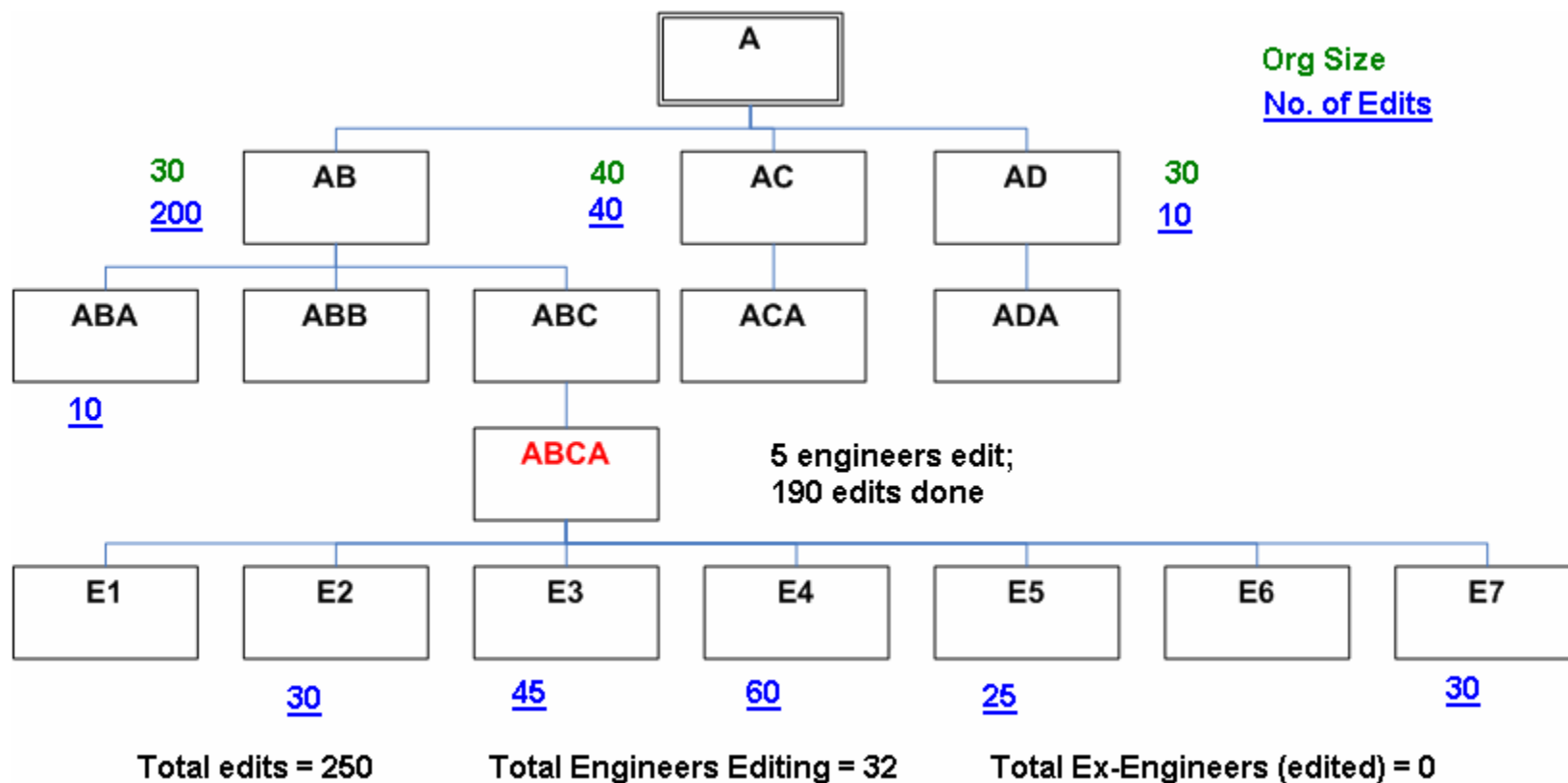
2. A large loss of team members affects the knowledge retention and lowers quality (NOEE)

- **Number of Ex-Engineers (NOEE):** This is the total number of unique engineers who have touched a binary and have left the company as of the release date of the software system
- **Implications:** This measure deals with knowledge transfer. If the employees who worked on a piece of code leaves the company then there is a likelihood that the new person taking over might not be familiar with the design rationale, the reasoning behind certain bug fixes, and information about other stake holders in the code

3. The more edits to components the higher the instability and lower the quality (EF)

- **Edit Frequency (EF):** This is the total number times the source code, that makes up the binary, was edited
- **Note:** An edit is when an engineer checks code out of the VCS, alters it and checks it back in again. This is independent of the number of lines of code altered during the edit
- **Implications:**
 - (1) if a binary had too many edits it could be an indicator of the lack of stability/control in the code from the different perspectives of reliability, performance etc.
 - (2) examined in conjunction with NOE and NOEE, it provides a more complete view of the distribution of the edits: did a single engineer make majority of the edits, or were they widely distributed amongst the engineers

Example Company XYZ and a particular binary



EF = 250 from VCS.

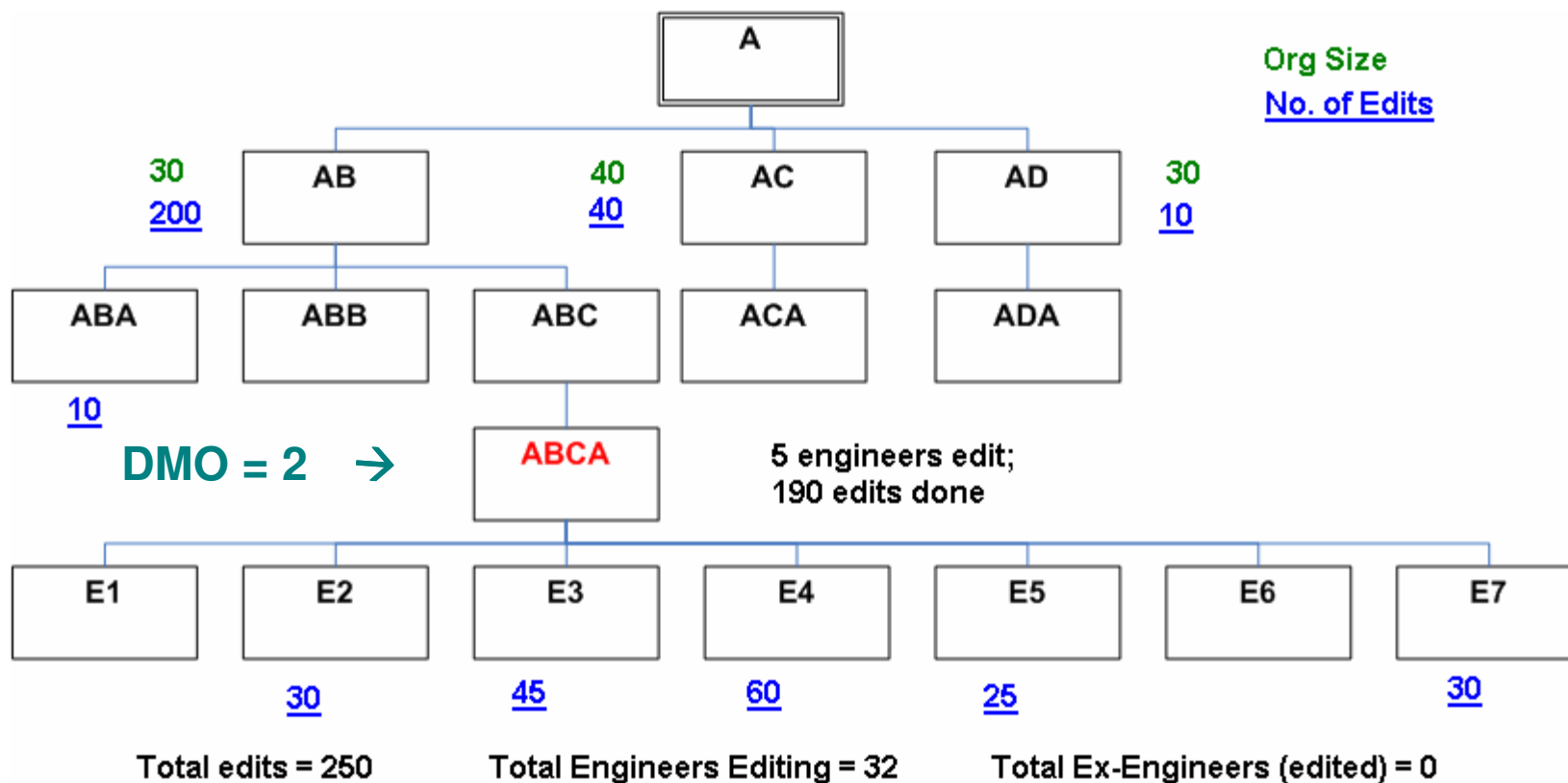
4. The lower level the ownership, the better is the quality (DMO)

- **Depth of Master Ownership (DMO):** The level of ownership of the binary depending on the number of edits done, i.e., the organization level of the person whose reporting engineers perform more than 75% of the rolled up edits is deemed as the DMO. The DMO metric determines the binary owner based on activity on that binary

Note: The choice of 75% is based on prior historical information on Windows to quantify ownership

- **Implications:** The deeper the ownership is in the tree, the more focused the activities, communication, and responsibility. A deeper level of ownership indicates smaller diffusion of activities and a single point of approval/control which should improve intellectual control

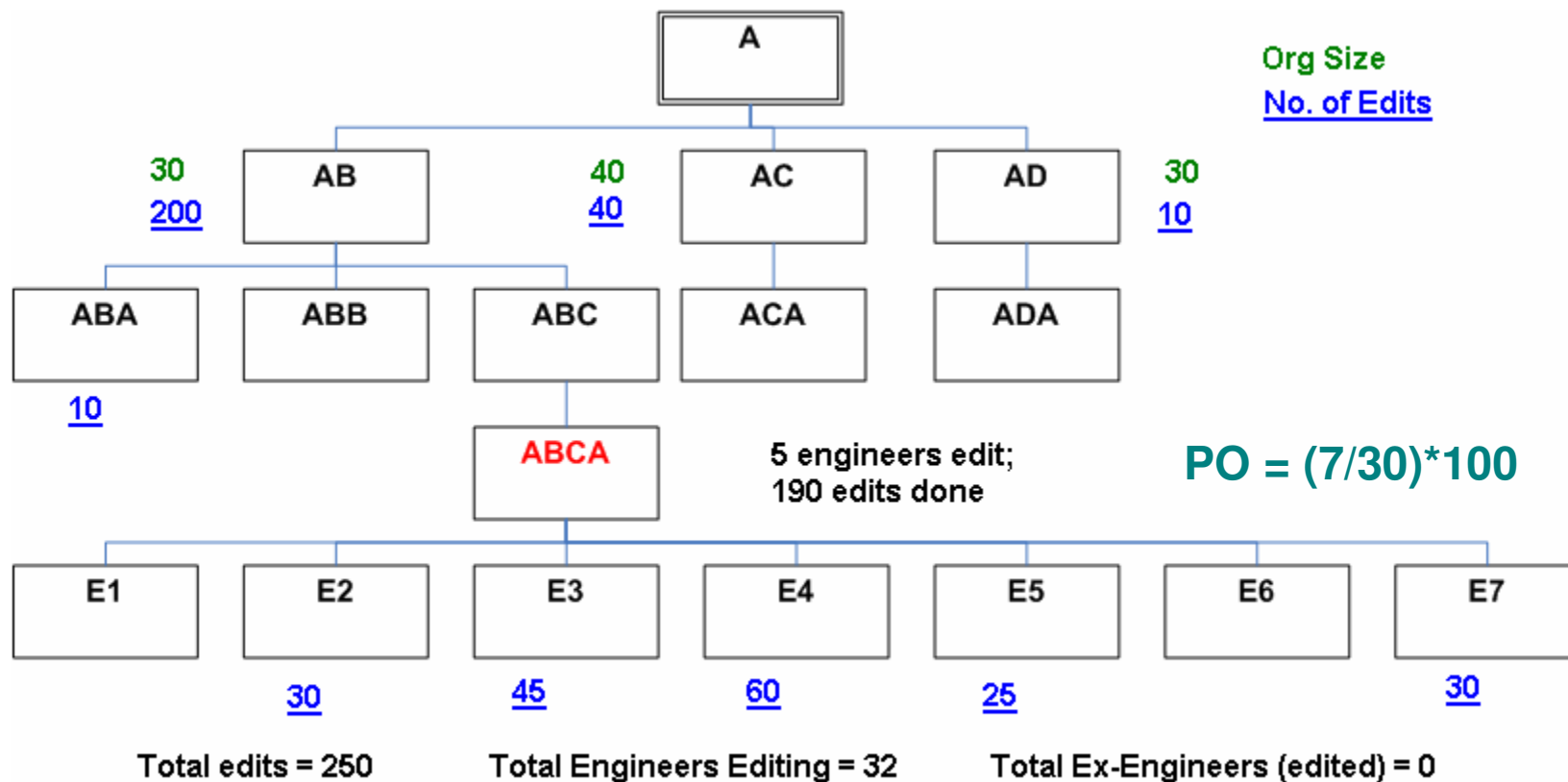
Example Company XYZ and a particular binary



5. The more cohesive the contributors organizationally the higher the quality (PO)

- **Percentage of the Organization contributing to development (PO):** The ratio of the number of people reporting at the DMO level owner relative to the Master owner org size
- **Implications:** The lower the PO the more local is the ownership and contributions to the binary leading to lower coordination/communication overhead across organizations and improved synchronization amongst individuals, better intellectual control and provide a single point of contact.
- This metric minimizes the impact of an unbalanced organization, whereby the DMO may be two levels deep but 90% of the total organization reports into that DMO

Example Company XYZ and a particular binary

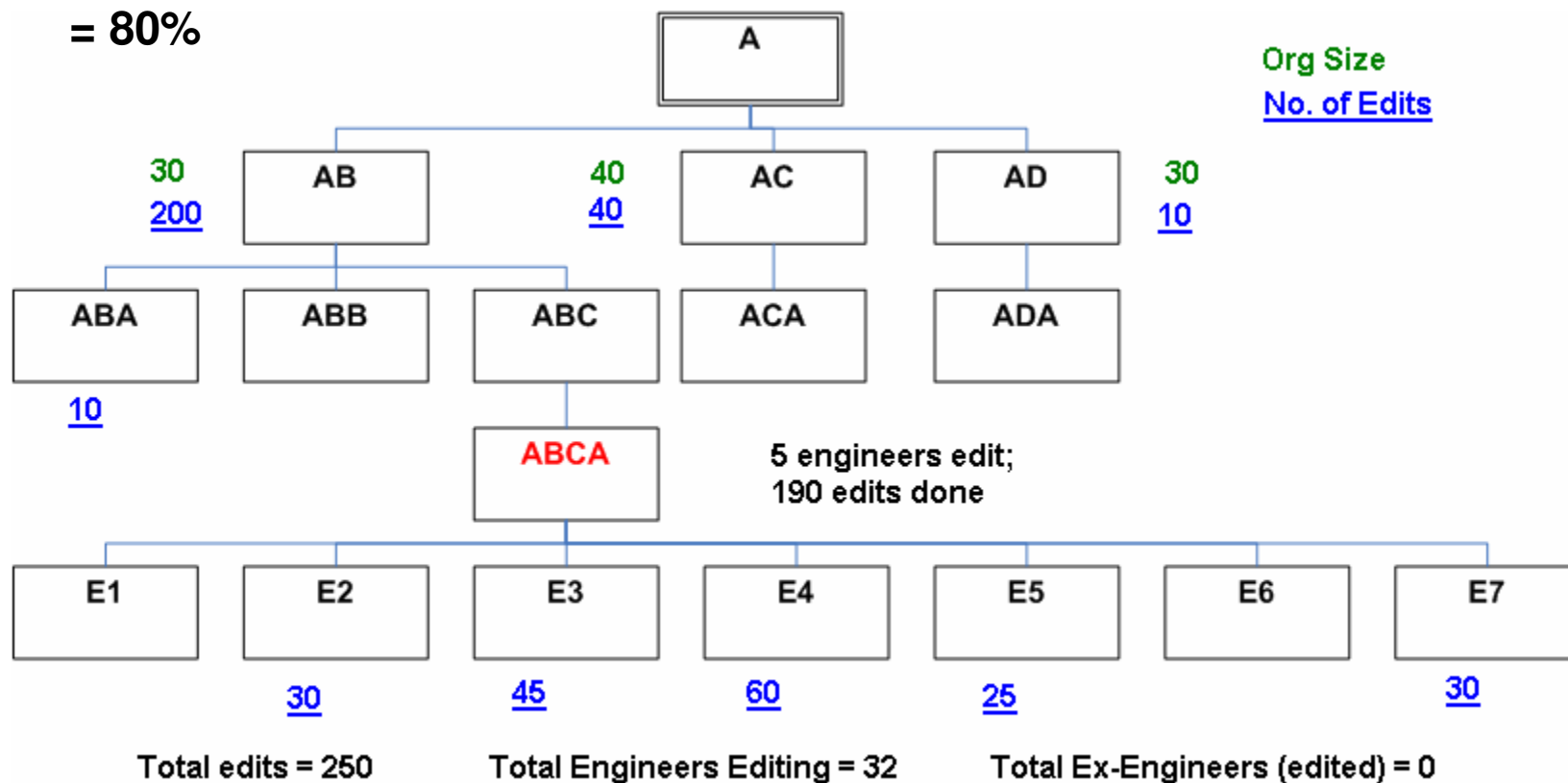


6. The more cohesive the contributions (edits) the higher the quality (OCO)

- **Level of Organizational Code Ownership (OCO):** The % of edits from the organization that contains the binary owner or if there is no owner then the organization that made the majority of the edits to that binary
- **Implications:** The more the development contributions belong to a single organization, the more they share a common culture, etc. The more diverse the contributors, the higher the chances of defective code.
- If a binary has a defined owner then this measure identifies whether the remaining edits to the binary was performed by people in the same organization (common culture). This measure is particularly important when a binary does not have a defined owner, as it provides a measure of how much control any single organization has over the binary. If there is a large PO value due to several of the engineers only having worked on the binary a few times the OCO measure will counter-balance that taking into account the development activities in terms of the edits.

Example Company XYZ and a particular binary

$$\text{OCO} = 200 / (200 + 40 + 10) = 80\%$$

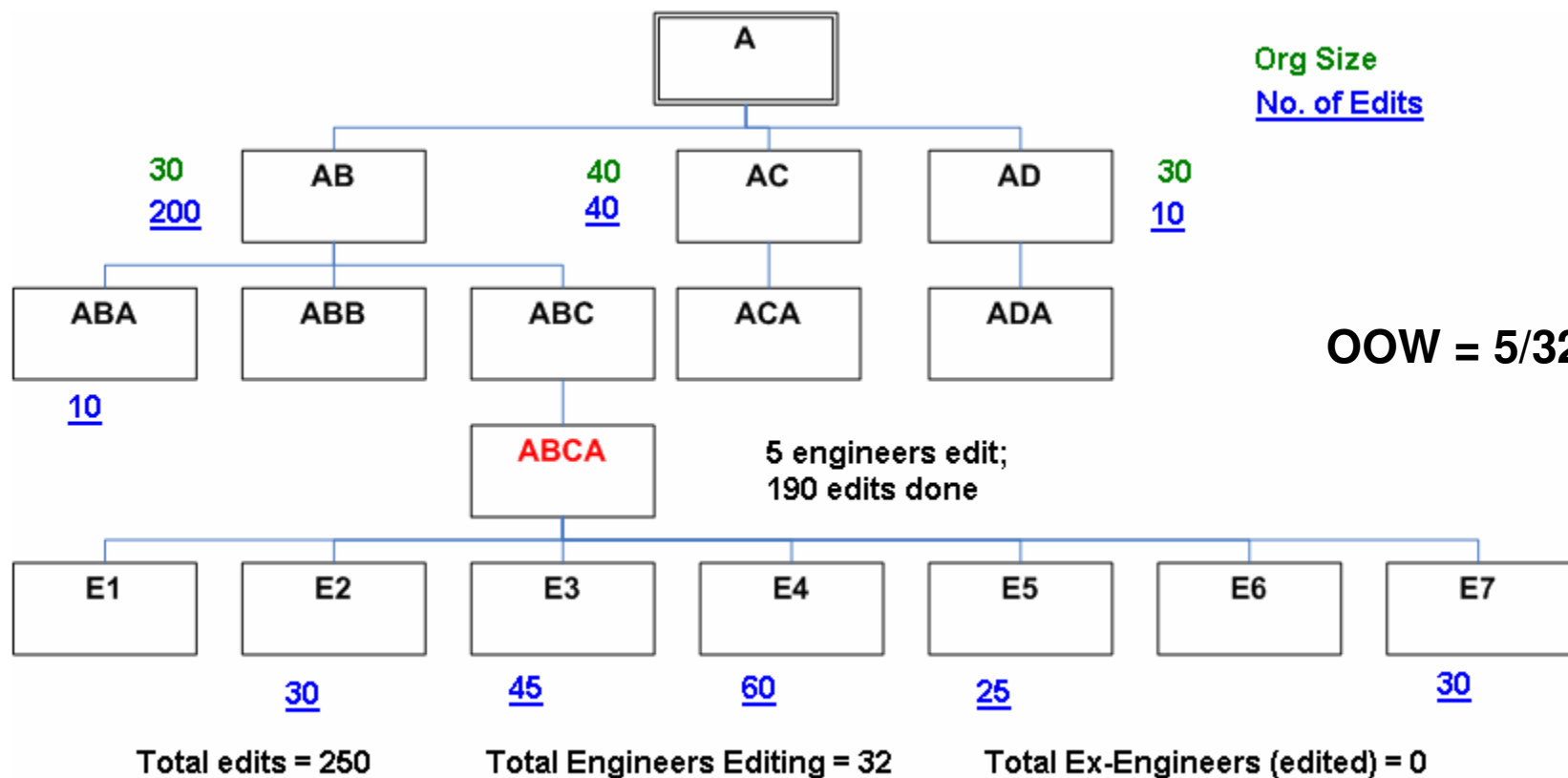


7. The more the diffused the contribution to a binary the lower the quality (OOW)

- **Overall Organization Ownership (OOW):** The ratio of the percentage of people at the DMO level making edits to a binary relative to total engineers editing the binary. A high value is good
- **Implications:** The more the activities belong to one organization, the more they share a common culture, focus, and social cohesion. The bigger the organizational distance the more chance of miscommunication and misunderstanding of goals, focus, etc.

OOW balances OCO and PO to account for “super” engineers who contribute a substantial amount of experience and code to the system. We do not want a few such engineers influencing our measures nor do we want them to be ignored. PO, OCO and OOW account for this type of inter relationship

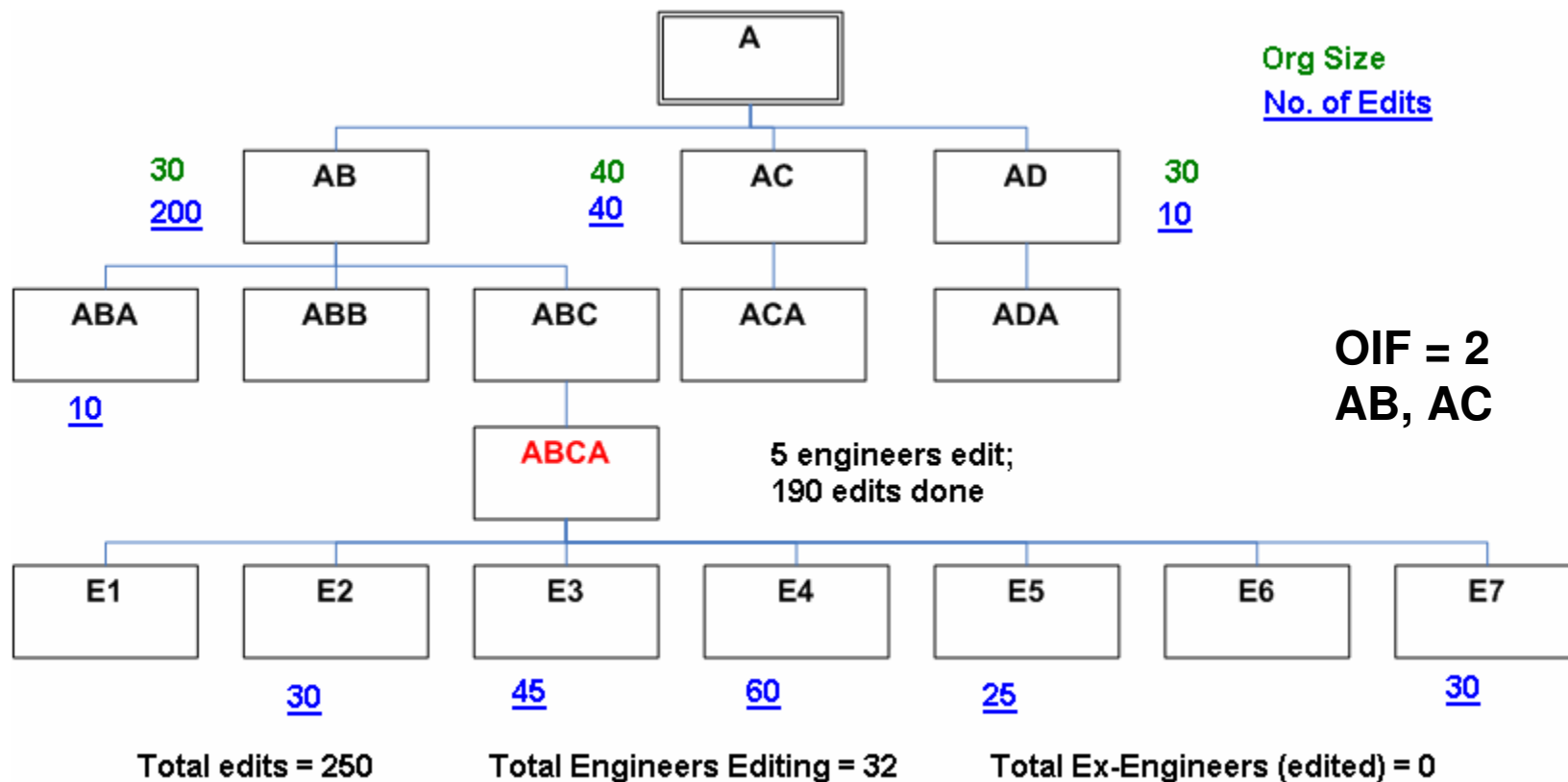
Example Company XYZ and a particular binary



8. The more diffuse the different organizations contributing code, the lower the quality (OIF)

- **Organization Intersection Factor (OIF):** A measure of the number of different organizations that contribute greater than 10% of edits, as measured at the level of the overall org owners
- **Implications:** The greater the OIF the more diffuse the contribution to a binary. This implies a lack of strong ownership from one particular org. Ideally a lower value is considered to be better
- This measure is particularly important when a binary has no owner as it identifies how diffused the ownership is across the total organization.

Example Company XYZ and a particular binary



Assertions

1. The more people who touch the code the lower the quality (NOE)
2. A large loss of team members affects the knowledge retention and lowers quality (NOEE)
3. The more edits to components the higher the instability and lower the quality (EF)
4. The lower level the ownership the better the quality (DMO)
5. The more cohesive the contributors (organizationally) the higher is the quality (PO)
6. The more cohesive the contributions (edits) the higher the quality (OCO)
7. The more the diffused the contribution to a binary, the lower the quality (OOW)
8. The more diffuse the different organizations contributing code, the lower the quality (OIF)

Case Study

Case Study: Windows Vista

- **Size:** > 50 million lines of code
- **Data Sources:** VCS (Version Control System), Microsoft People Management System (e.g., employ id, email alias, start date), a tree map of the organizational structure
- **Context Data:**
 - A few thousand engineers wrote code for Vista
 - 33% of the Vista engineers had previously worked on Windows 2003 or Windows XP
 - 61% of the engineers had managers who had contributed code to Windows 2003 (or XP), and 37% of them had managers of managers who previously contributed code to Windows

Case Study: Windows Vista

- **Quality variable:** post release failures measured for the first 6 months
- To test the **interrelationship** among the organizational metrics we used Spearman rank correlation
 - EF ↔ NOE (0.860)
 - EF ↔ NOEE (0.758)
 - OCO ↔ DMO (0.816)
 - OCO ↔ PO (0.530)
 - OOW ↔ PO (0.793)
 - OOW ↔ OCO (0.816)
 - all statistically significant at 99% confidence

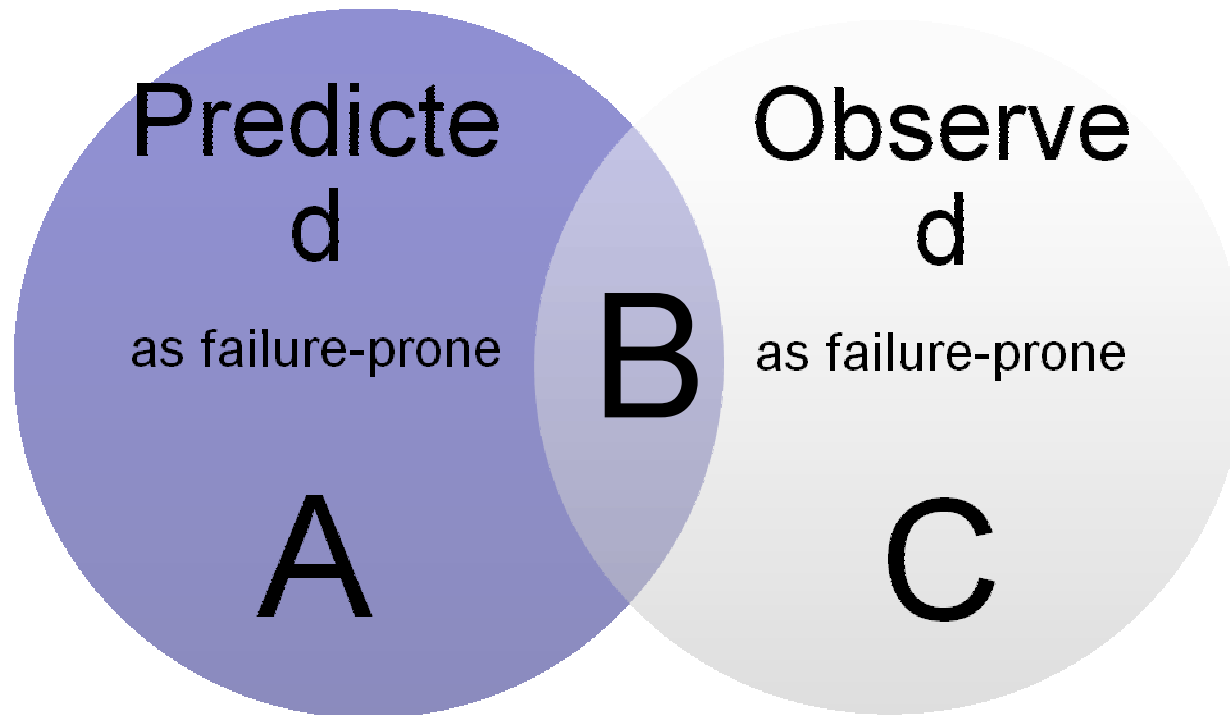
Research Questions

- Are organizational metrics are effective predictors of code quality?
- Are these organizational metrics all required ?
- Are the organization metrics better predictors of quality than the traditional metrics used so far ?
- Can we quantify the organizational domain/institutional knowledge of Windows?

Are organizational metrics are effective predictors of code quality?

- **Failure-proneness** is the probability that a particular software element (such as a binary) will fail in operation in the field
- The higher the failure-proneness, the higher the probability of experiencing a post-release failure
- We classified binaries as failure-prone or not using statistical correlation and model-fitting techniques (using transformations)
- Probability (π) =
$$\frac{e^{(c+a_1X_1+a_2X_2+\dots)}}{1 + e^{(c+a_1X_1+a_2X_2+\dots)}}$$

How to measure success



$$\text{Precision} = B/A$$

Precision = the proportion of predicted failure-prone binaries that were correct

$$\text{Recall} = B/C$$

Recall = the proportion of failure-prone binaries correctly identified

How to measure success

Predicted

as failure-prone

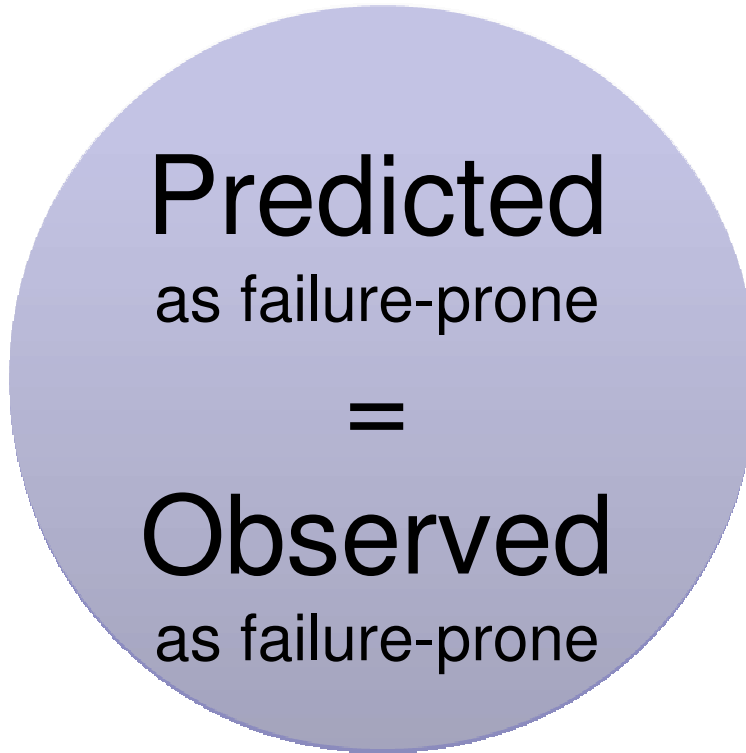
Observed

as failure-prone

WORST!

Precision=Recall=0

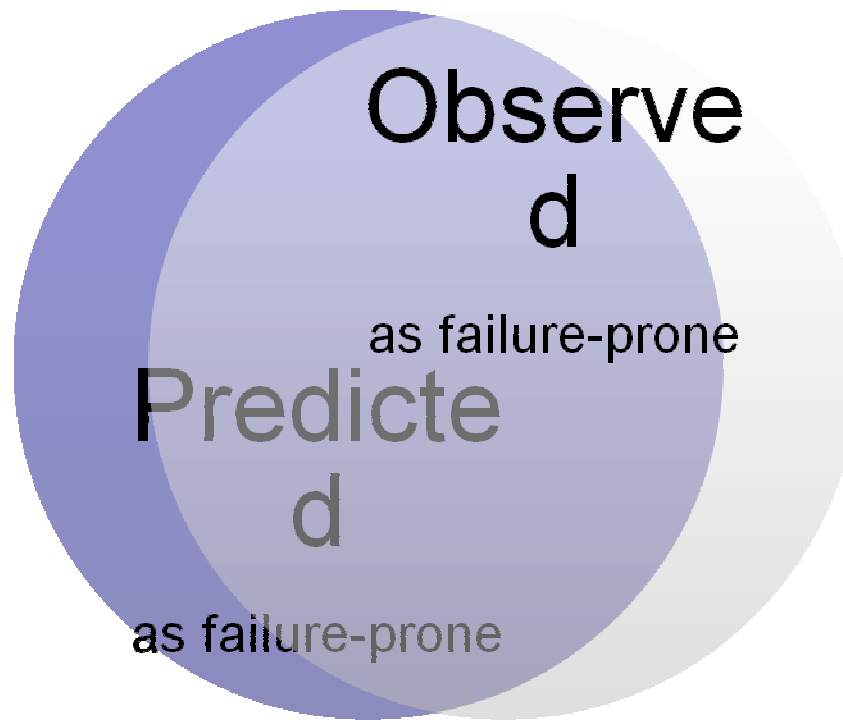
How to measure success



BEST!

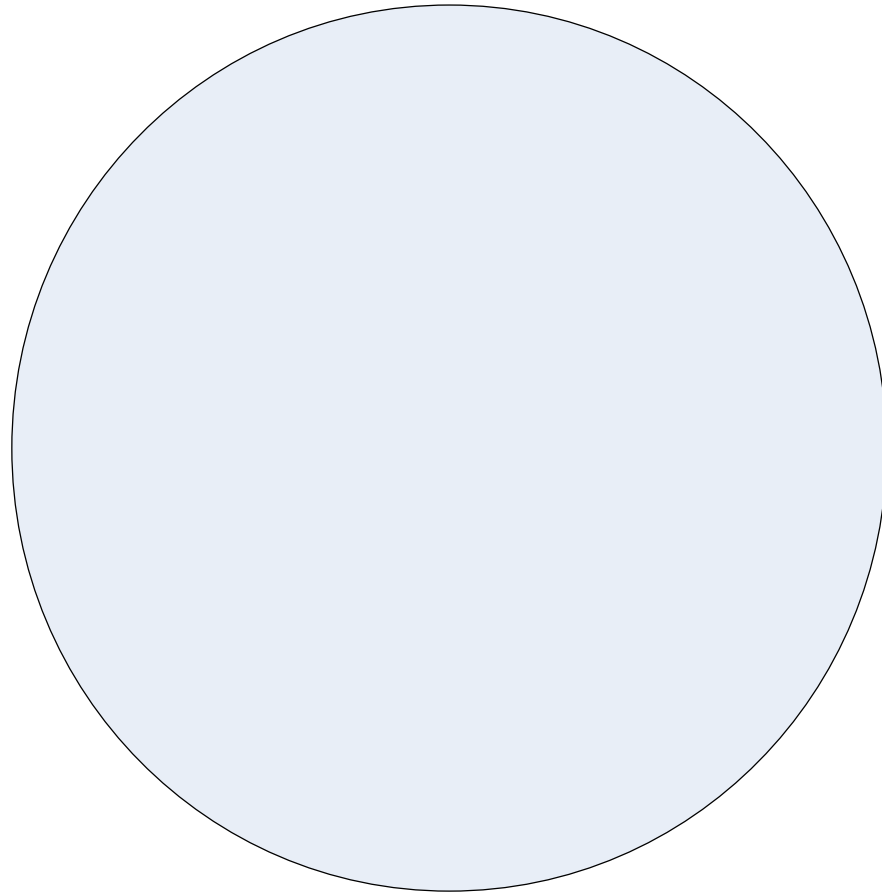
Precision=Recall=1

How to measure success



REALITY!

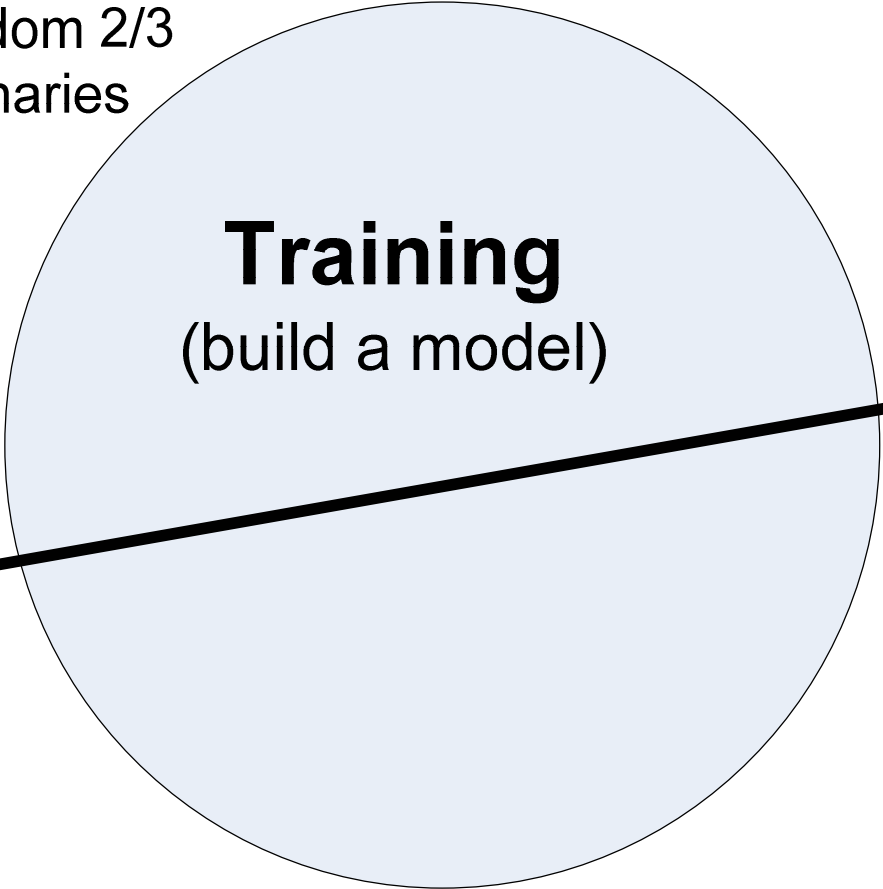
Random splits



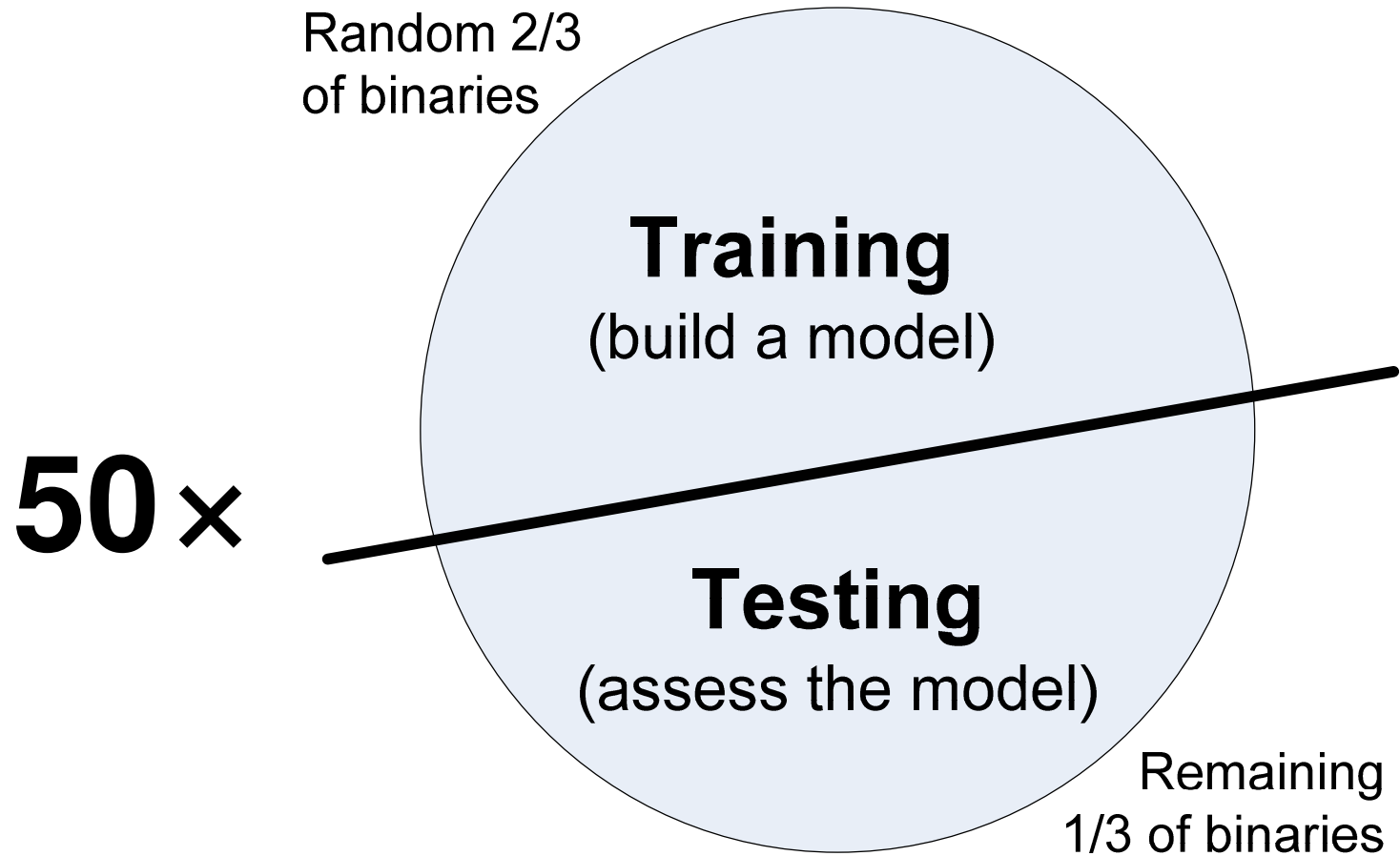
Random splits

Random 2/3
of binaries

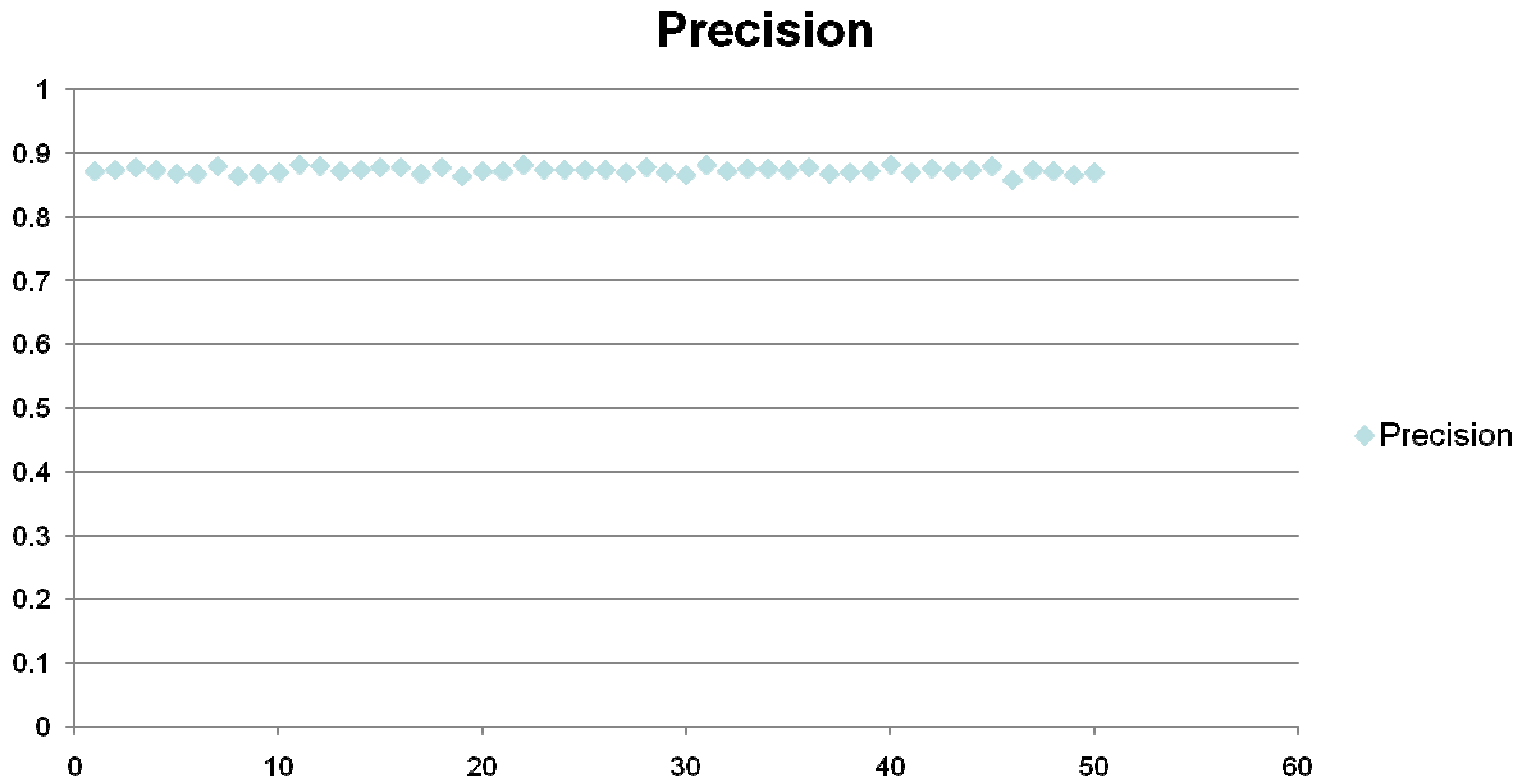
Training
(build a model)



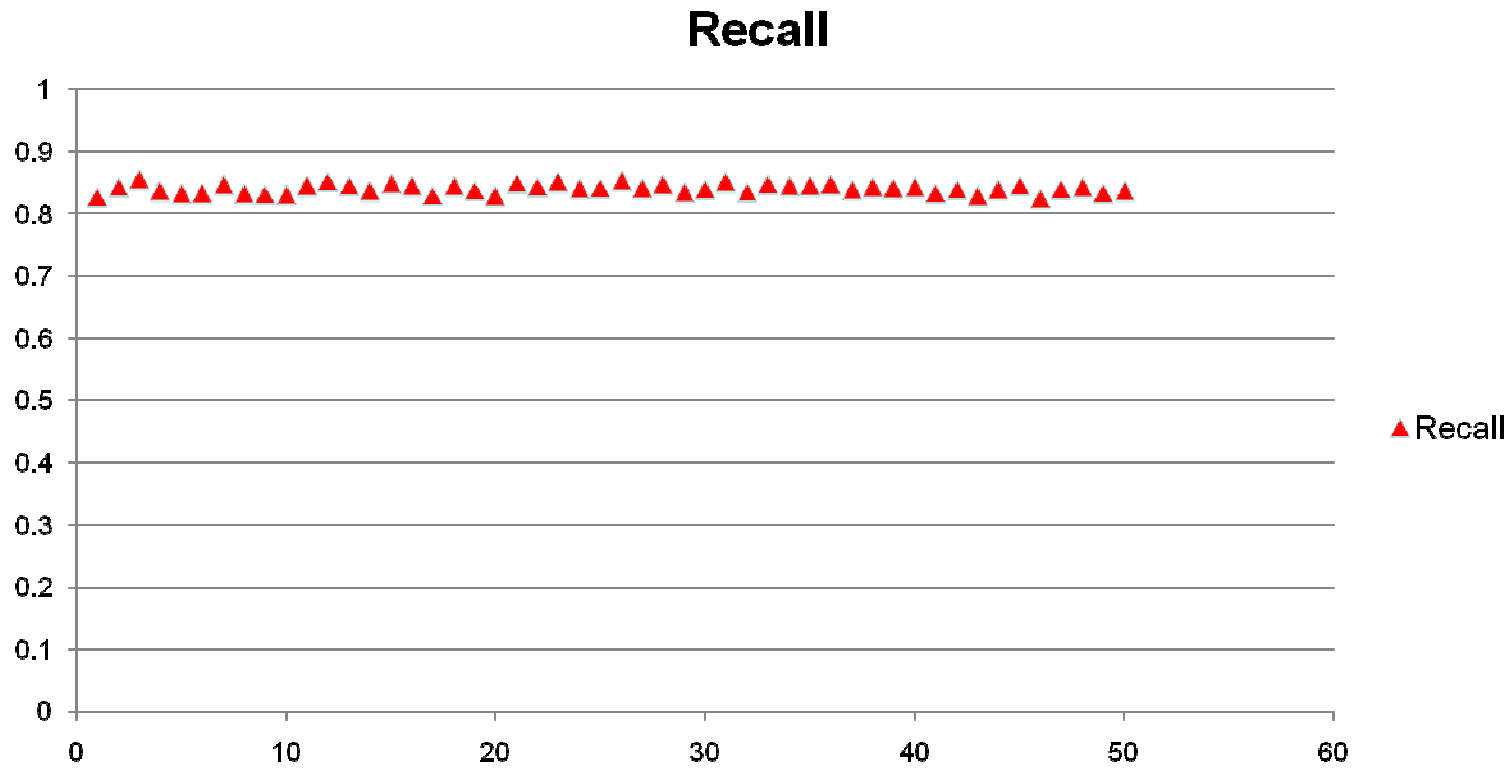
Random splits



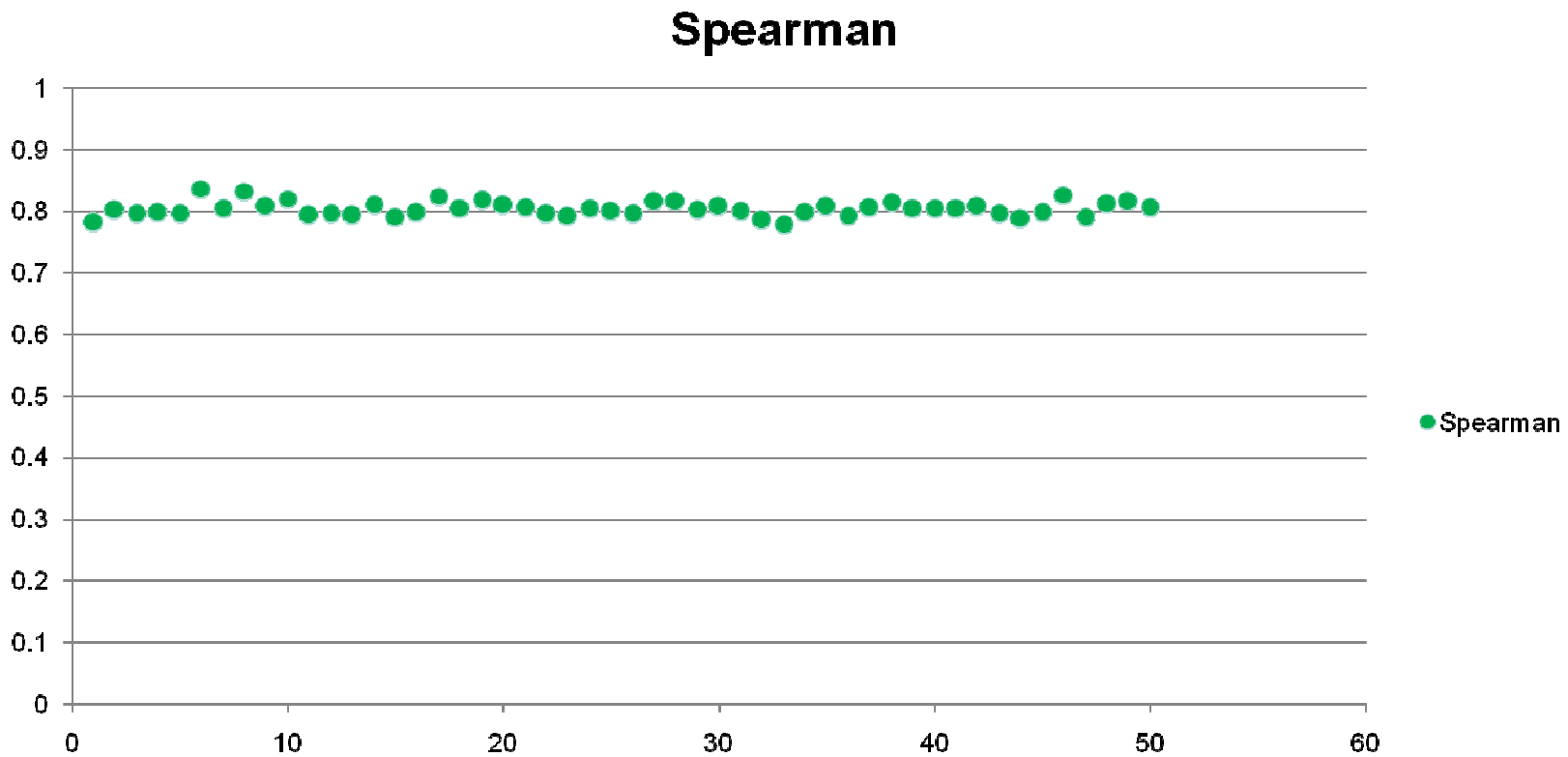
Average Precision = .87



Average Recall = .84



Sensitivity of Failure probability vs. Failures



as predicted failure-proneness increased, actual field failures increased

Are these organizational metrics all required?

- Step-wise regression is the robust technique of these methods
- The initial model consists of the predictor having the single largest correlation with the dependent variable
- Subsequently, new predictors are selected for addition into the model based on their partial correlation with the predictors already in the model
- With each new set of predictors, the model is re-evaluated

Are these organizational metrics all required?

- Predictors that do not significantly contribute towards statistical significance in terms of the F-ratio are removed
- Eventually the best set of predictors explaining the maximum possible variance is left.
- All measures were retained. None were removed from the step-wise regression.

Are the organizational metrics better predictors of quality than the traditional metrics?

Model	Precision	Recall
Organization Structure	86.2%	84.0%
Churn	78.6%	79.9%
Complexity	79.3%	66.0%
Dependencies	74.4%	69.9%
Coverage	83.8%	54.4%
Pre-Release Bugs	73.8%	62.9%

Classification results

Can we quantify the organizational domain/institutional knowledge of Windows?

- This is done by repeating the entire analysis against Windows XP and Windows Server 2003

Threats to Validity

- **Internal validity.** causal issues of our results
 - engineers had no knowledge the study was being performed
 - Data collection and analysis were independent
- **Construct validity.** errors in measurement.
 - metrics, failures and VCS all automated.
 - cross checks performed to identify abnormal values
 - large size and diversity of our dataset.
- **External validity.** data is from one very large software system and other software systems may not be of comparable size.
 - the analysis replicated on a reduced set to determine the type and size of organization structure required
 - results indicate **team of size 30 engineers and 3 levels of depth** should be sufficient to collect the organizational metrics to predict failure-proneness

CONCLUSIONS AND FUTURE WORK

Conclusions

- Study empirically supports Brooks' argument about organization structure and impact on quality.
- We have presented a new metric schema for quantifying organizational complexity
- It implies people should consider organizational structure, especially in the context of global development
- The organizational metrics are better predictors of quality (in our context)

Future work

- Replicate this study in other organizations within and outside of Microsoft
- Investigate this line of research from the context of open source teams where virtual organizations exist to quantify appropriate organizational measures and study their effect on quality
- Study such organizational metrics in the context of global software development
- Collaborate with cognitive psychologists and organizational behavior researchers to look at social and cognitive aspects of our work by doing observational studies of engineers

About Microsoft Research (MSR)

- **Research arm of Microsoft**
- **About 700 researchers in 55 areas**
- **Six Labs Worldwide**



The Empirical Software Engineering and Measurement Group (ESM) at MSR

- People
 - Nachi Nagappan, Brendan Murphy, Tom Ball
 - <http://research.microsoft.com/esm/>
- External collaborators
 - Al Aho, Vic Basili, Jeff Carver, Audris Mockus, Laurie Williams, Andreas Zeller
- Interns
 - Thomas Zimmermann, Andreas Johansson, Shilpa Bugde, Lucas Layman

Thank You

- **Acknowledgements**
 - **Windows Group Engineers**
 - **Windows Senior Management**
 - **Microsoft Research**

Questions/comments are welcome!

nachin@microsoft.com