The Influence of Organizational Structure on Software Quality: An Empirical Case Study

Nachiappan Nagappan
Microsoft Research
Redmond, WA, USA
nachin@microsoft.com

Brendan Murphy
Microsoft Research
Cambridge, UK
bmurphy@microsoft.com

Victor R. Basili
University of Maryland
College Park, MD, USA
basili@cs.umd.edu

Presentation by Etan Kissling
18.05.2010

SOFTWARE METRICS

Code Churn

- Software change history
- Large / recent changes

- Total added, modified and deleted LOC
- Number of times that a binary was edited
- Number of consecutive edits

Code Complexity

- Gathered from code itself
- Multiple complexity values

- Cyclomatic complexity
- Fan-In / Fan-Out of functions
- Lines of Code
- Weighted methods per class
- Depth of Inheritance
- Coupling between objects
- Number of subclasses
- Total global variables

Dependencies

- Components that a class uses
- Both data and call dependencies

- Incoming / outgoing direct / indirect dependencies to a binary
- Layer information: Distance of a binary from the system kernel

Code coverage

- Degree to which the source code is tested

```
foo (x: INTEGER; y: INTEGER): INTEGER
        local
                c: INTEGER
        do
                c := y
                if x > 5 and y > 0 then
                        c := x
                end
                Result := x * c
        end
```

Statement coverage

- Has each node in the program been executed?

```
foo (x: INTEGER; y: INTEGER): INTEGER
        local
                c: INTEGER
        do
Statement 1     c := y
Statement 2     if x > 5 and y > 0 then
Statement 3             c := x
                end
Statement 4     Result := x * c
        end
```

A testing suite which includes `foo(7, 1)` would cover all statements of this code.

Branch coverage

- Has each control structure been evaluated both to true and false?

```
foo (x: INTEGER; y: INTEGER): INTEGER
        local
                c: INTEGER
        do
                c := y
                if x > 5 and y > 0 then
                        c := x
                end
                Result := x * c
        end
```

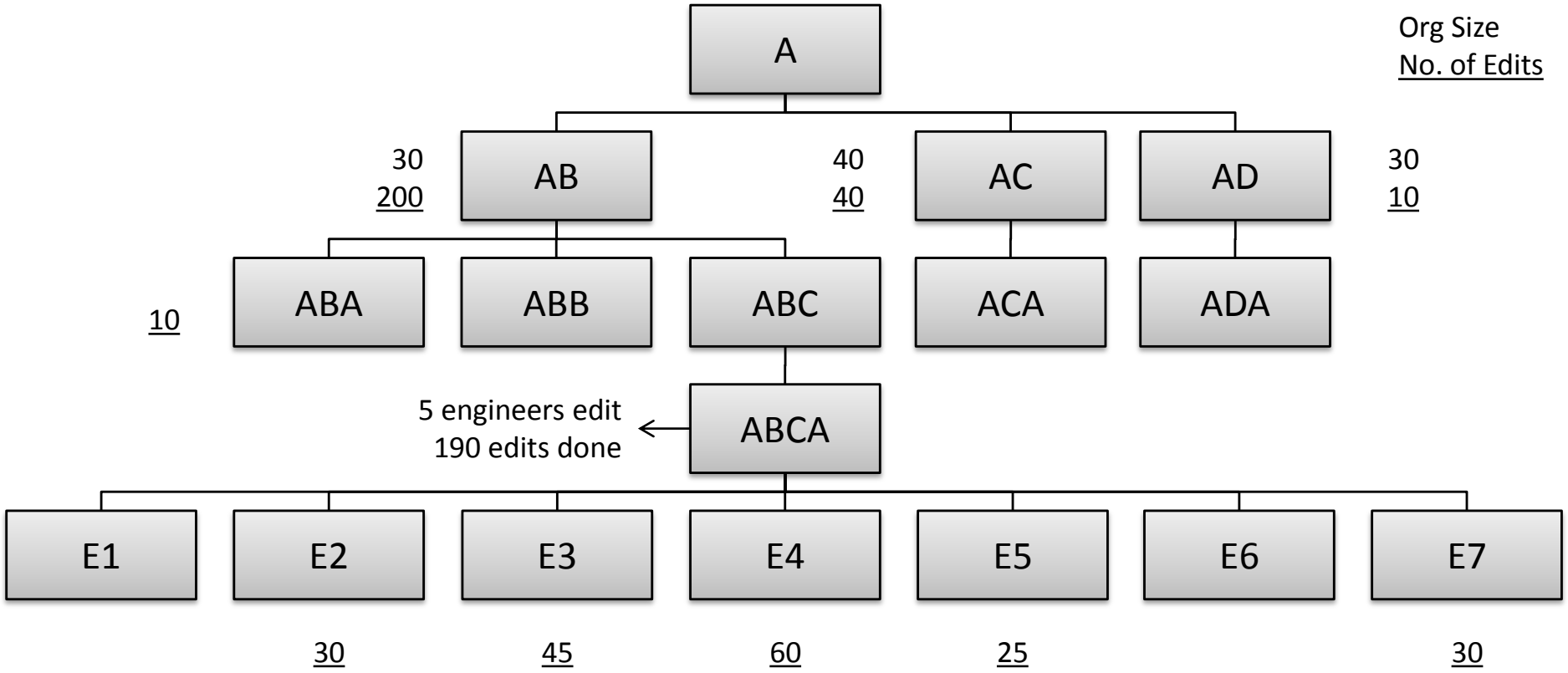
`foo(7, 1)` and `foo(7, 0)` together would cover this branch completely

Pre-release defects

- Number of pre-release bugs found in a binary
- Strong relationship between development defects per module and field defects per module

ORGANIZATIONAL METRICS

Organizational Structure



Total edits = 250

Total Engineers Editing = 32

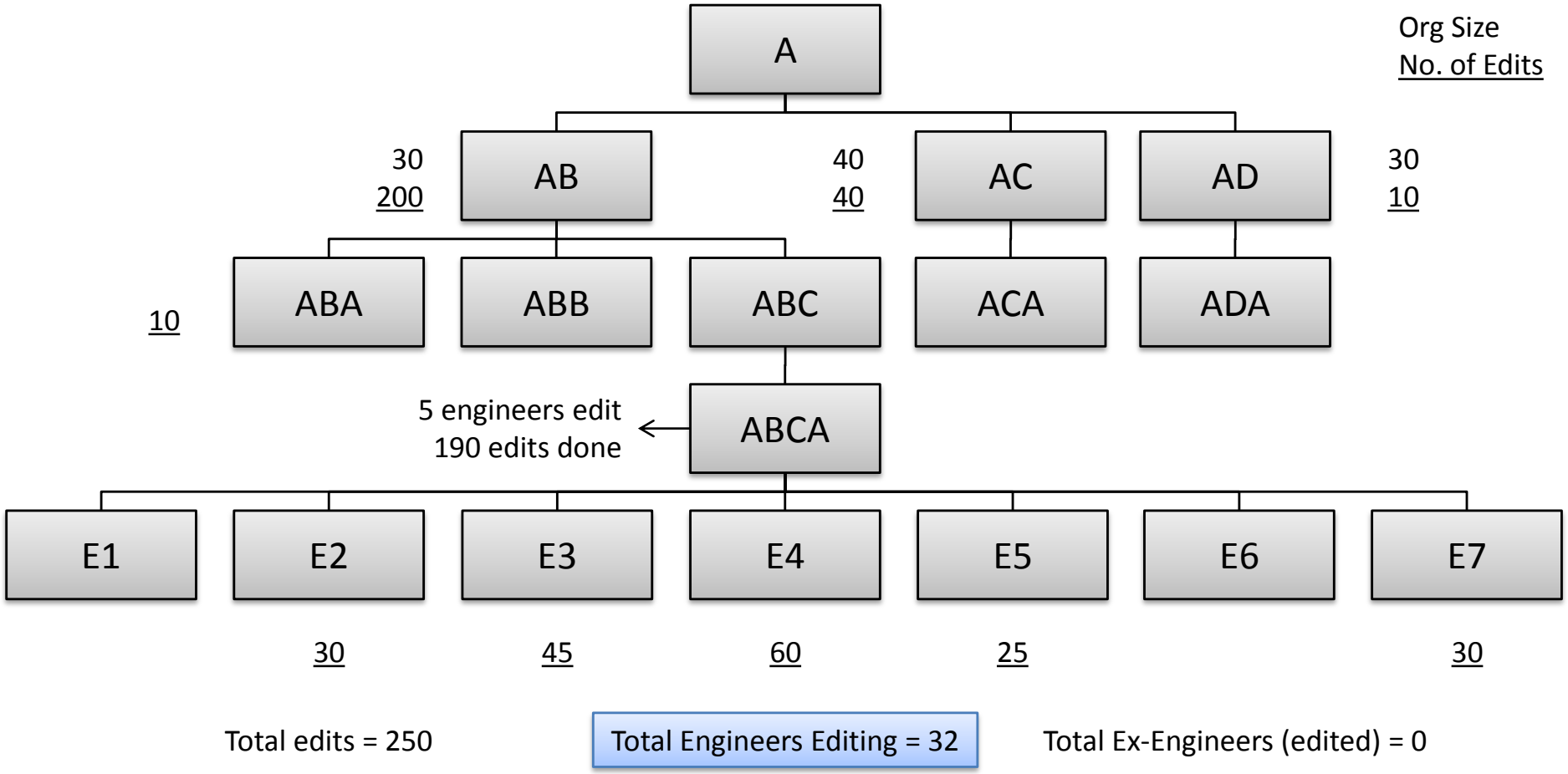
Total Ex-Engineers (edited) = 0

Number of Engineers

- Touched a binary
- Still employed by the company

› The more people who touch the code the lower the quality

Organizational Structure

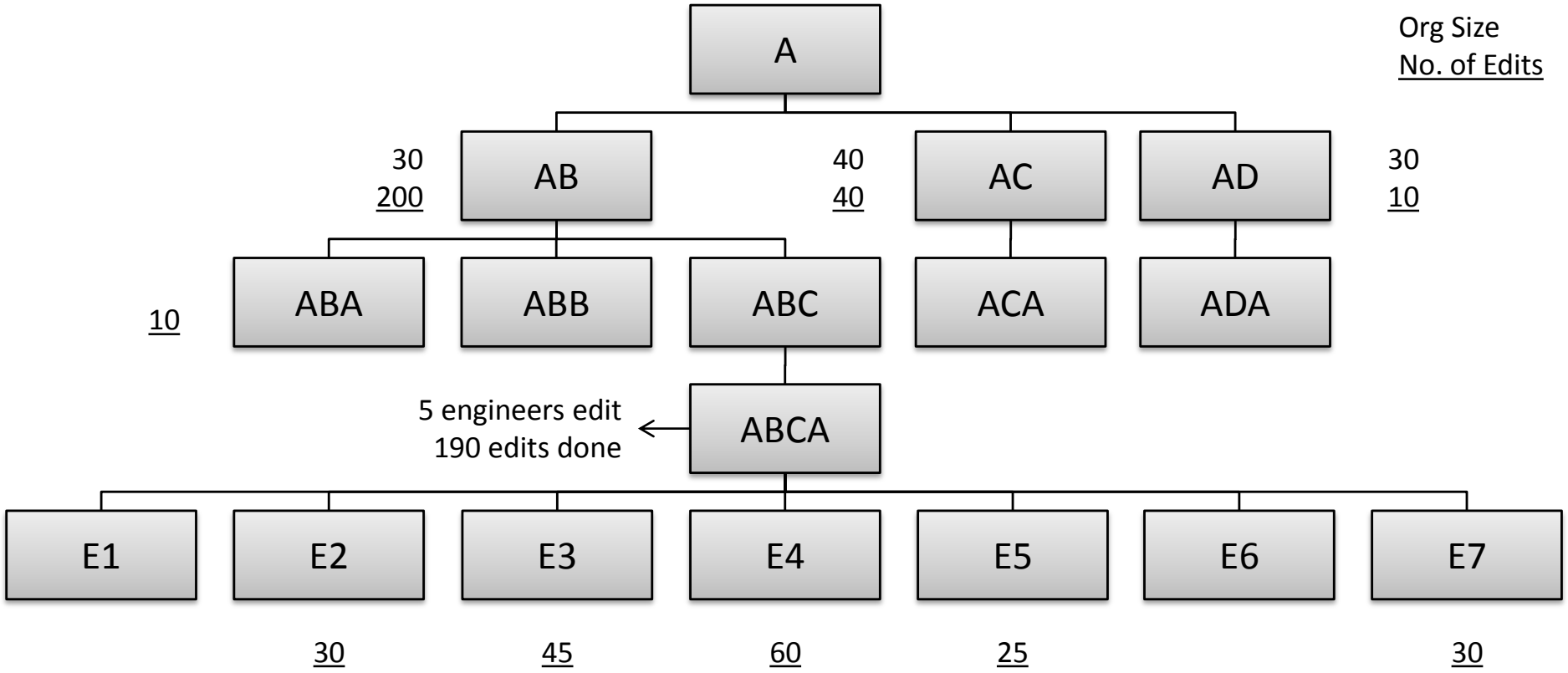


Number of Ex-Engineers

- Touched a binary
- Left the company

› A large loss of team members affects the knowledge retention and thus quality

Organizational Structure



Total edits = 250

Total Engineers Editing = 32

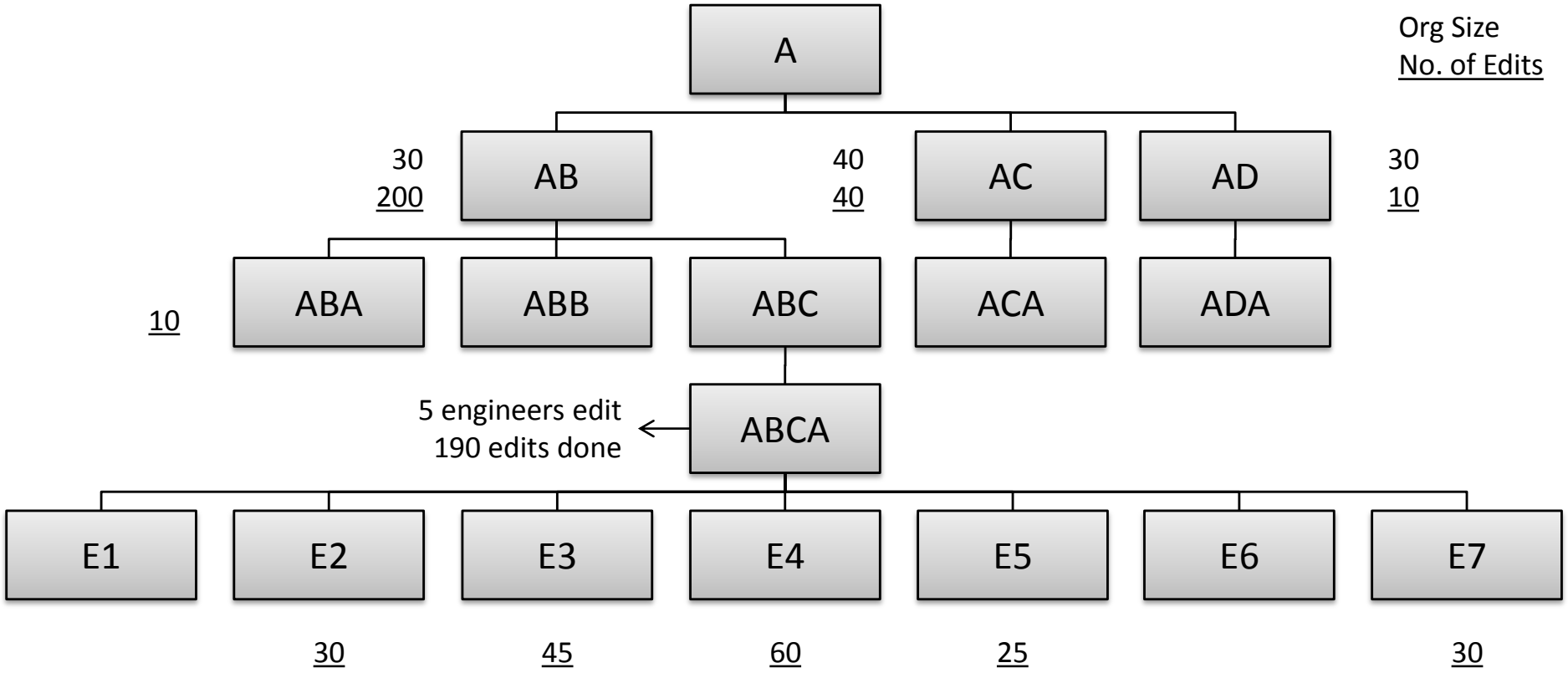
Total Ex-Engineers (edited) = 0

Edit Frequency

- Number of edits

› The more edits to components the higher the instability and lower the quality

Organizational Structure



Total edits = 250

Total Engineers Editing = 32

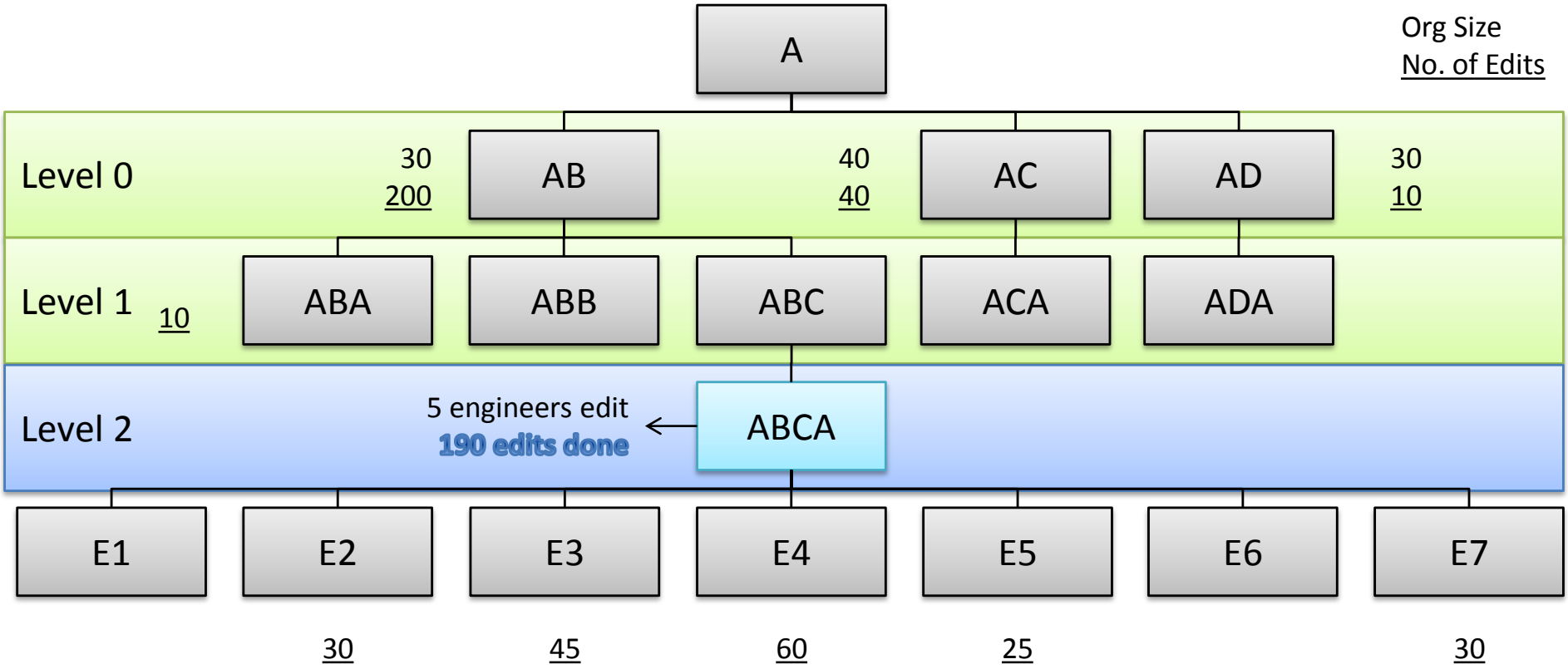
Total Ex-Engineers (edited) = 0

Depth of Master Ownership (DMO)

- Level of ownership
- More than 75% of the edits done by engineers which report to the owner

› The lower level is the ownership the better is the quality

Organizational Structure



Total edits = 250

Total Engineers Editing = 32

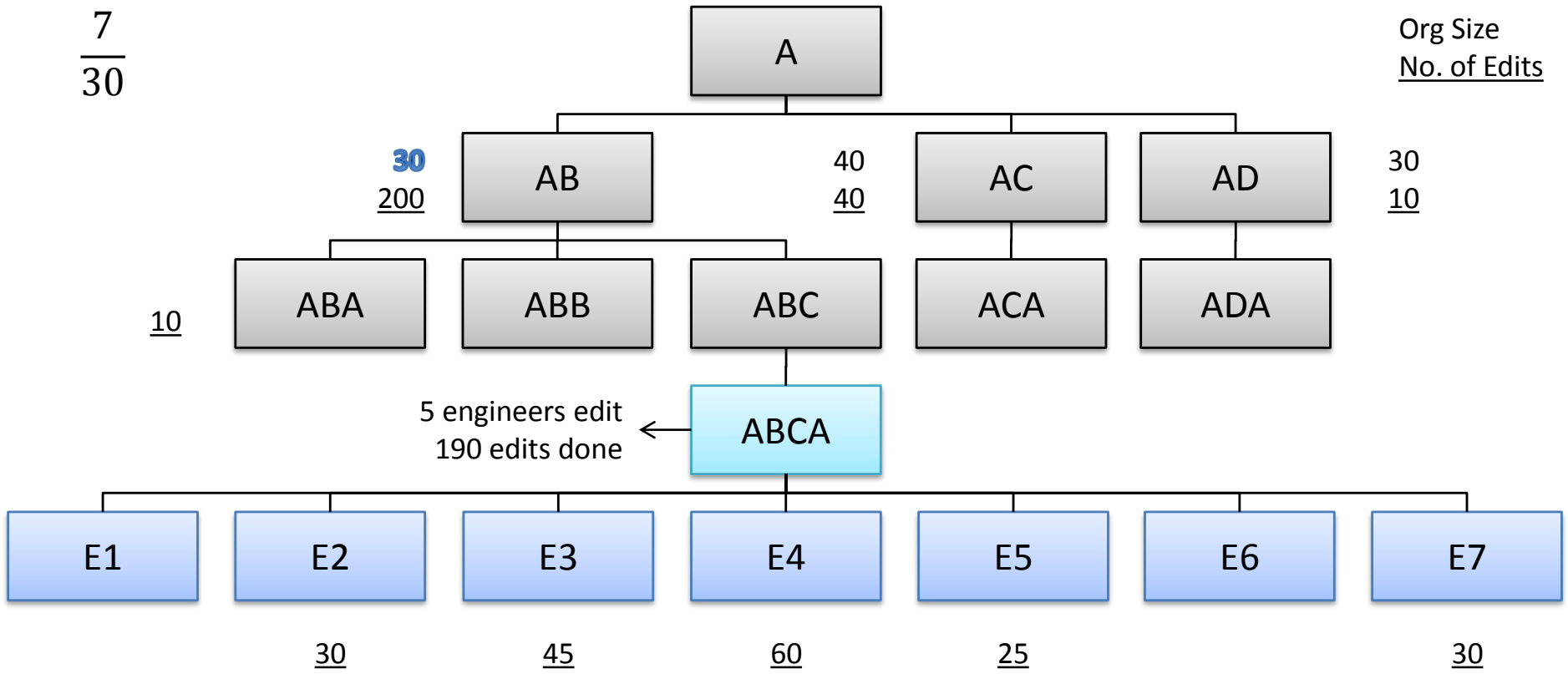
Total Ex-Engineers (edited) = 0

Percentage of Org contributing to development

- $$\frac{Number\ of\ people\ reporting\ at\ the\ DMO\ level}{Master\ owner\ org\ size}$$

› The more cohesive are the contributors (organizationally) the higher is the quality

Organizational Structure



Total edits = 250

Total Engineers Editing = 32

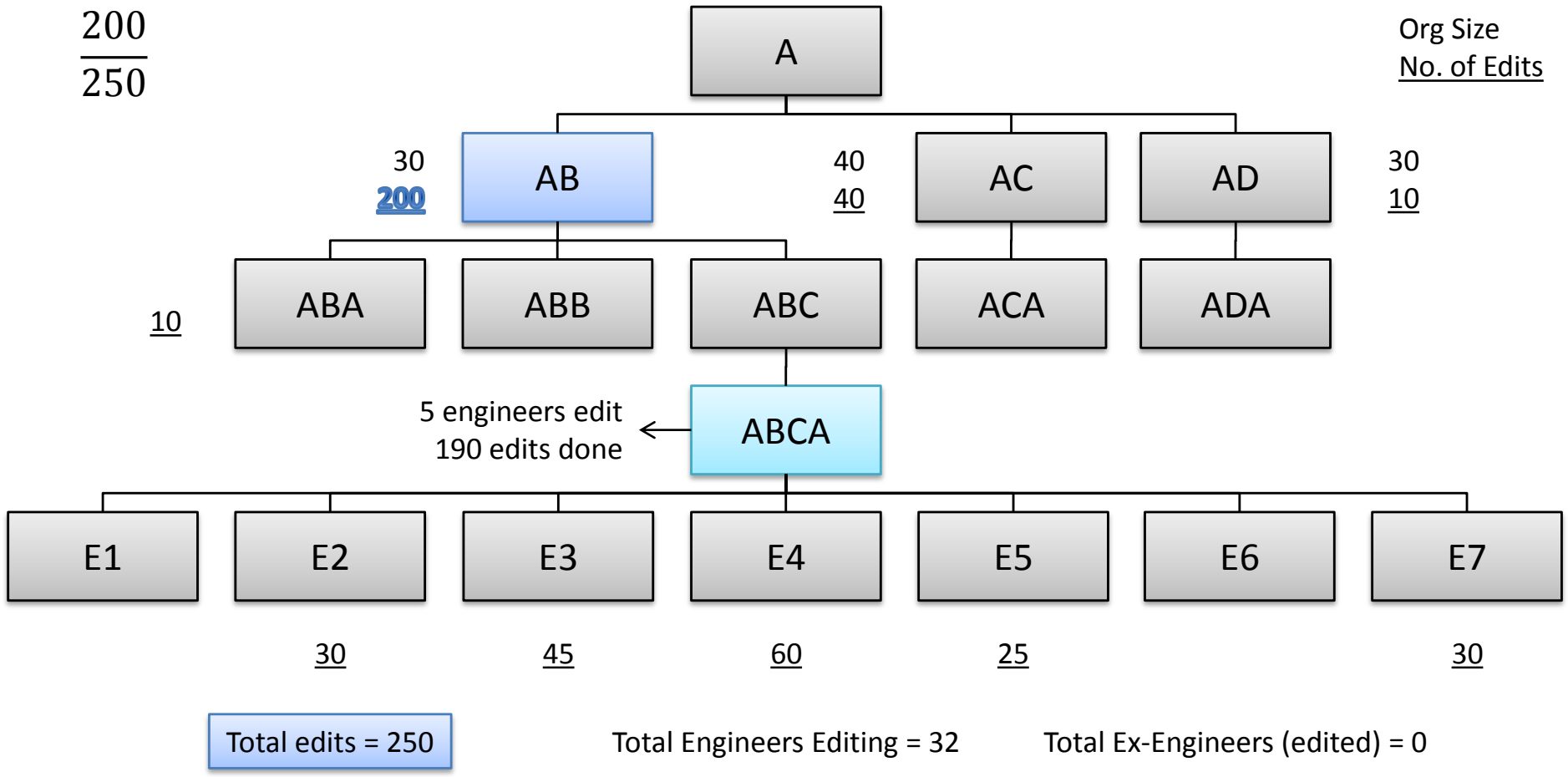
Total Ex-Engineers (edited) = 0

Level of Organizational Code Ownership

- If there is an owner:
  Percent of edits from the owner's organization
- If there is no owner:
  Percent of edits from the organization which made the majority of edits

› The more cohesive are the contributions (edits) the higher is the quality

Organizational Structure

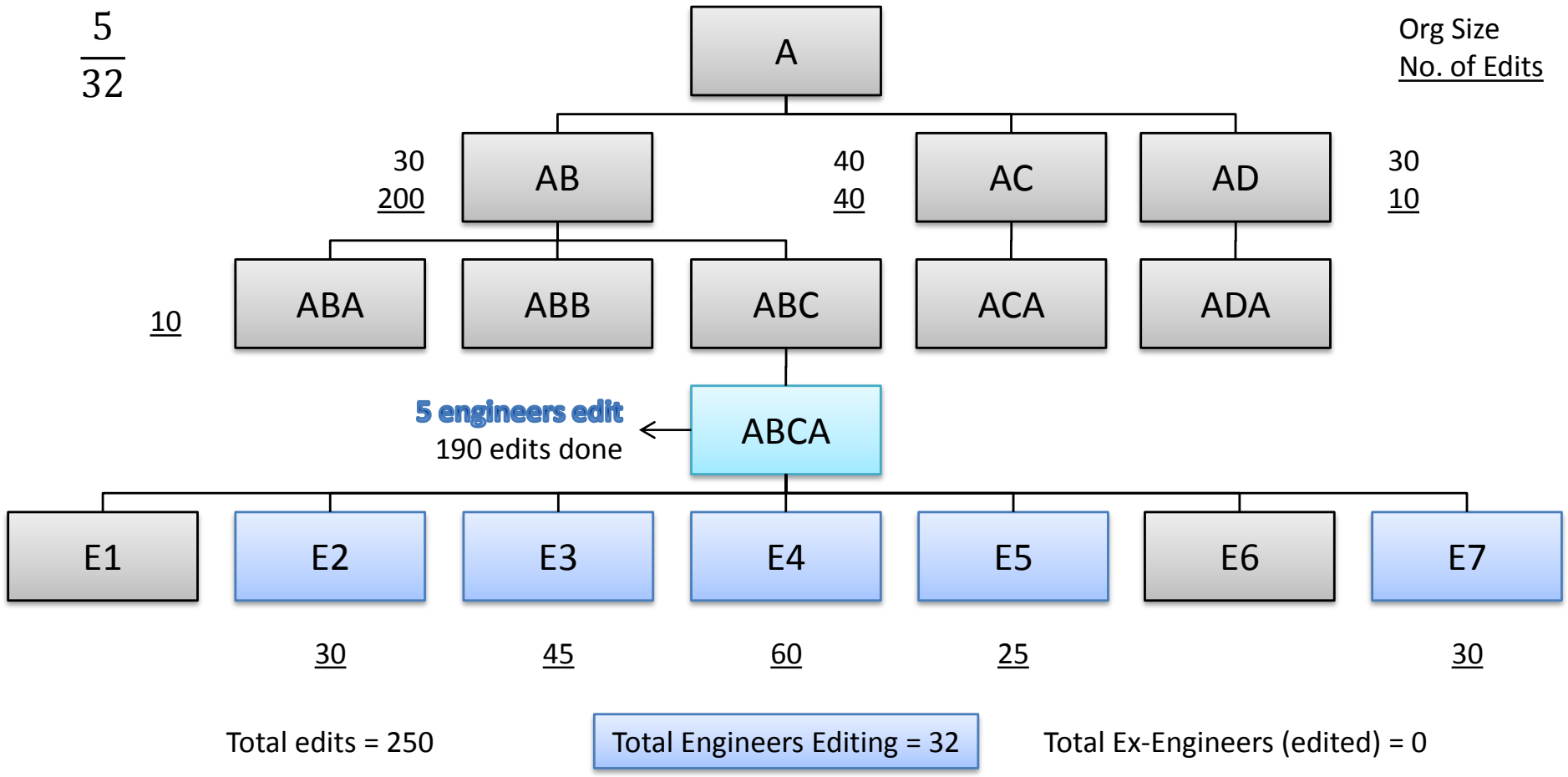


Overall Organization Ownership

- $$\frac{Number\ of\ people\ at\ the\ DMO\ level\ making\ edits}{Total\ Engineers\ Editing}$$

› The more the diffused contribution to a binary the lower is the quality

Organizational Structure

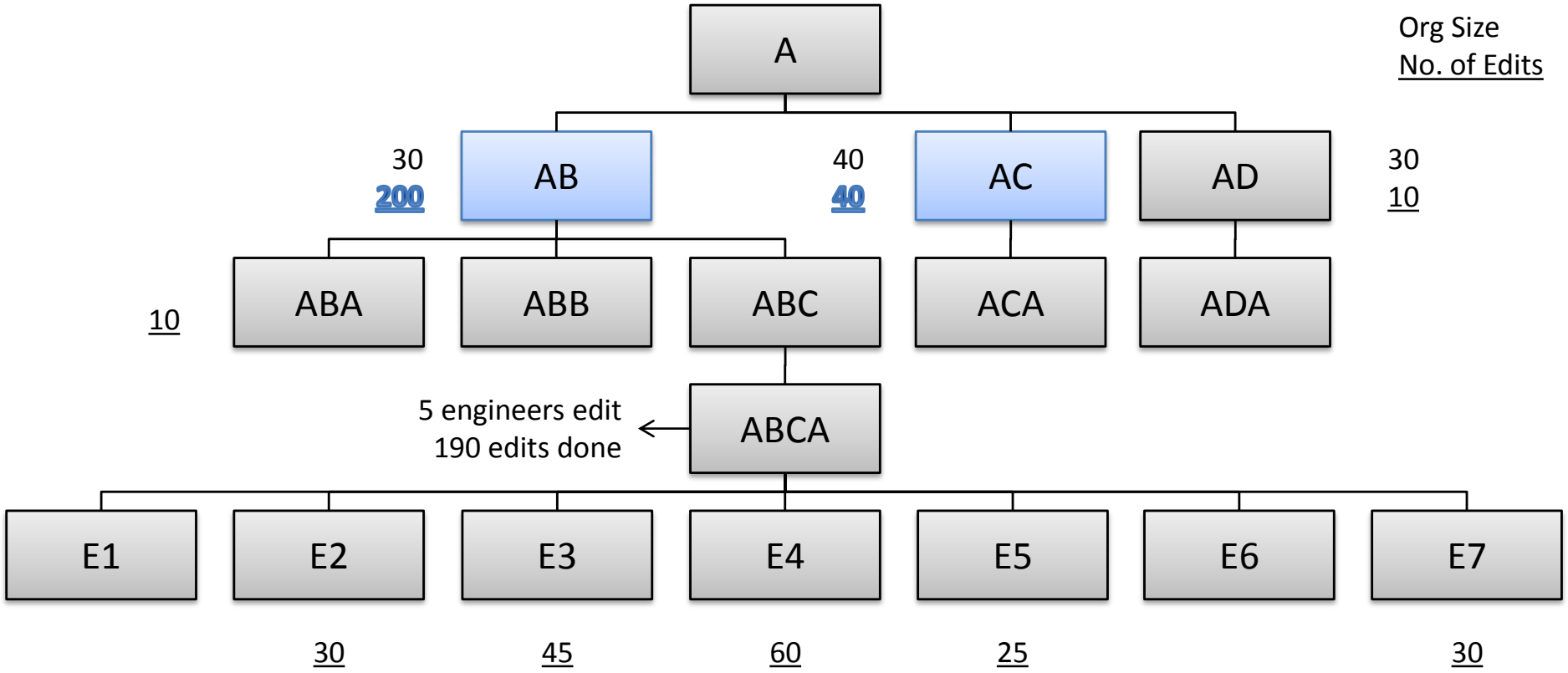


Organization Intersection Factor

- Number of different organizations that contribute greater than 10% of edits

› The more diffused the different organizations contributing code, the lower is the quality

Organizational Structure



Total edits = 250

Total Engineers Editing = 32

Total Ex-Engineers (edited) = 0

CASE STUDY

Case study

- Windows Vista:
  3404 binaries
  50+ Million LOC
- Access to people management software to build tree maps for organizational metrics
- 50 random splits:
  2/3 to build prediction model
  1/3 to verify prediction accuracy

Precision and recall

		Predicted	
		Not Failure-prone	Failure-prone
Actual	Not failure-prone	**A**	**B**
	Failure-prone	**C**	**D**

$$Precision = \frac{d}{b + d}$$

Percentage of correct failure-prone predictions

$$Recall = \frac{d}{c + d}$$

Percentage of correctly identified failure-prone binaries

Comparization

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

Threats to validity

- Internal validity:
  Influence of study to Windows
- Construct validity:
  Errors in measurement
- External validity:
  All data from one software system