Does Distributed Development Affect Software Quality?

An Empirical Case Study of Windows Vista

PRESENTED BY – SIMRANDEEP SINGH

Outline

1. What is distributed development

2.Difficulties in distributed development

3.Assumptions in the paper

4.Hypotheses

5.Methods and Results

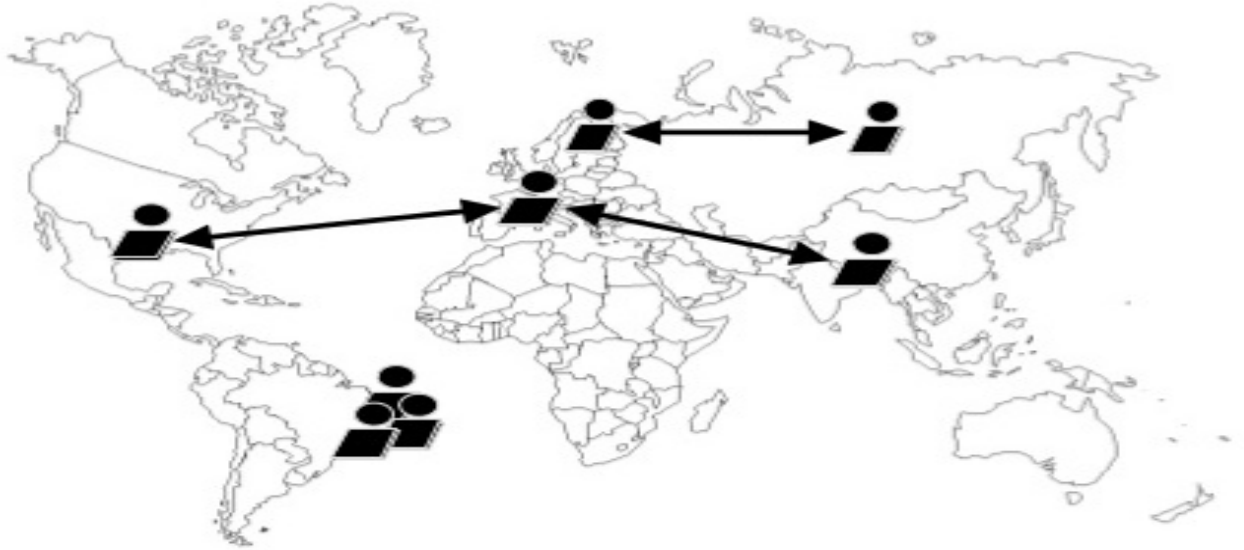
6.Conclusion

What is distributed development

“A Distributed Development project is a research and development project that is done across many business work sites or locations.“

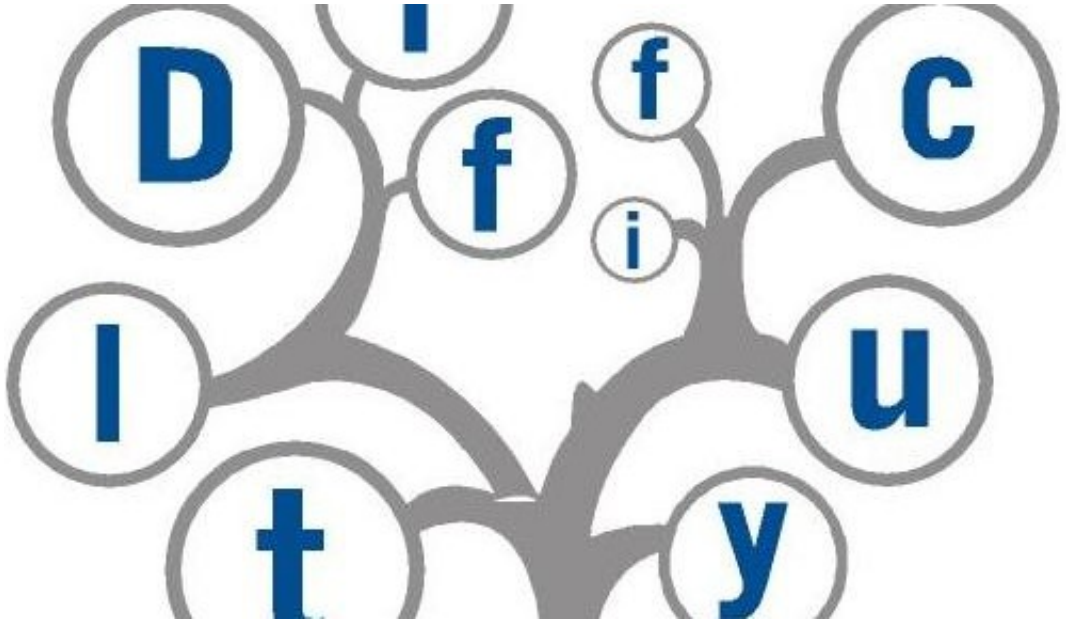
A project can be distributed in many different ways:

Geographically, **Organizationally,** **Temporally**



Difficulties in Distributed Development

- Communication
- Coordination breakdowns
- Diversity in operating environments
- Distance
- Organizational and National Cultural Barriers



Hypothesis

1. Binaries that are developed by teams of engineers that are distributed will have more post-release failures than those developed by collocated engineers.

2. Binaries that are distributed will be less complex, experience less code churn, and have fewer dependencies than collocated binaries.

Method Used - Step-1 Data Collection

Windows Vista involves:

- Nearly 3000 developers
- Over 3300 unique binaries
- Source code bases of over 60 MLOC
- 59 buildings and 21 campuses in Asia, Europe and North America
- Developed in house (no outsourced elements).

Data is focused on three main properties that is -

- Code quality
- Geographical location
- Code ownership

Code Quality

Measure of post-release failures.

Recorded for 6 months following release.

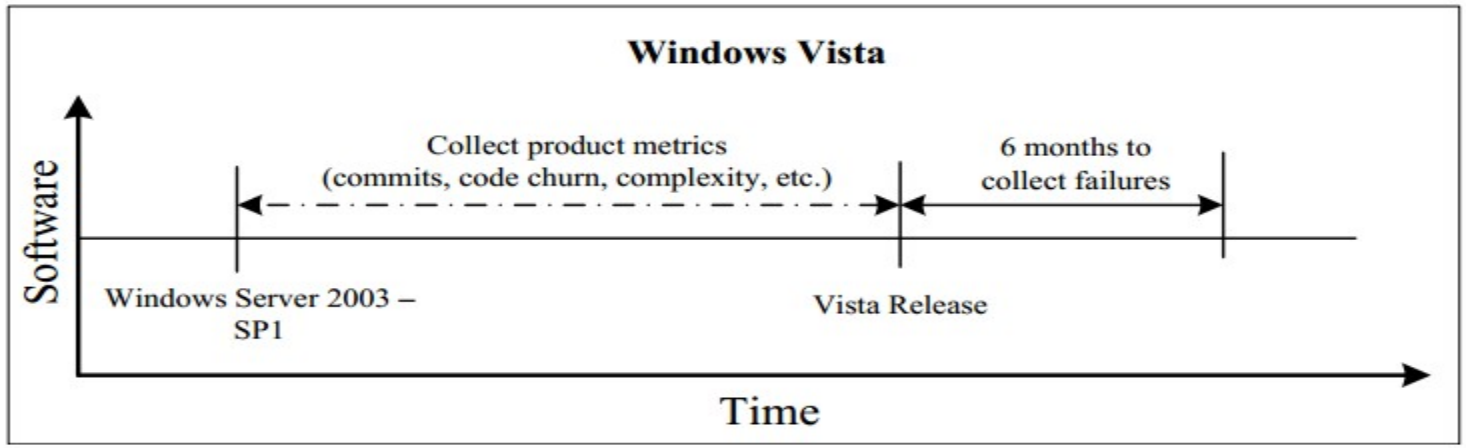


Figure 1. Data collection timeline

Geographical Location

Collected for each SW developer.

Geographic levels

Building

Cafeteria

Campus

Locality

Continent

World

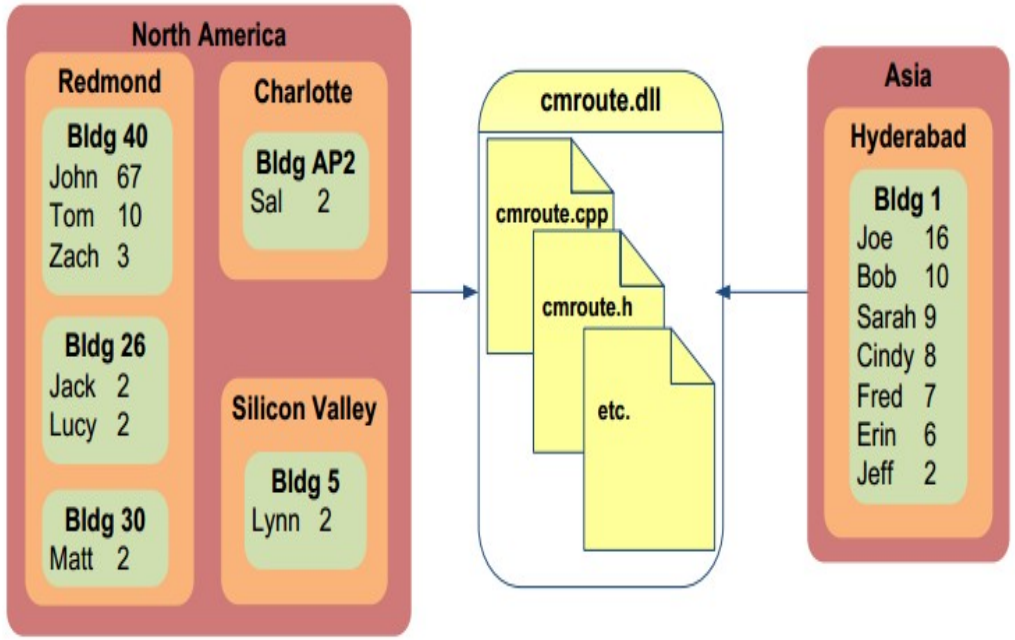


Figure 2. Commits to the library cmroute.dll. For clarity, location of anonymized developers is shown only in terms of continents, regions, and buildings.

What does the paper say

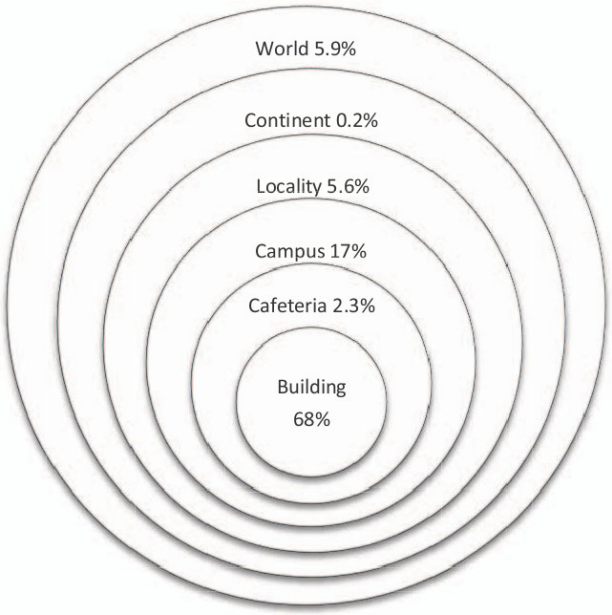
Proportion of binaries that fall in each category.

Each binary is assigned the lowest level in the hierarchy from which at least 75% of the commits were made.

Majority of binaries have over 75% of commits from one building.

Low percentage in continent level: US only country with multiple regions.

Each level of hierarchy, examine no. of binaries AND distribution of failures

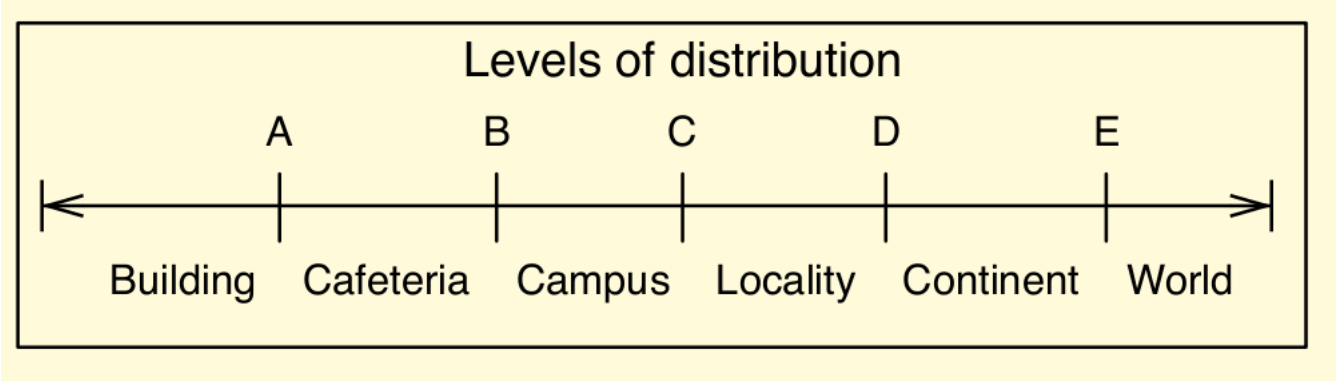


Assumptions in the paper

Divide binaries in 5 different ways.

Example: Split B categorizes building and cafeteria level binaries as collocated and the rest as distributed.

Performed to observe if there is a level of distribution leads to a significant increase in the number of failures.

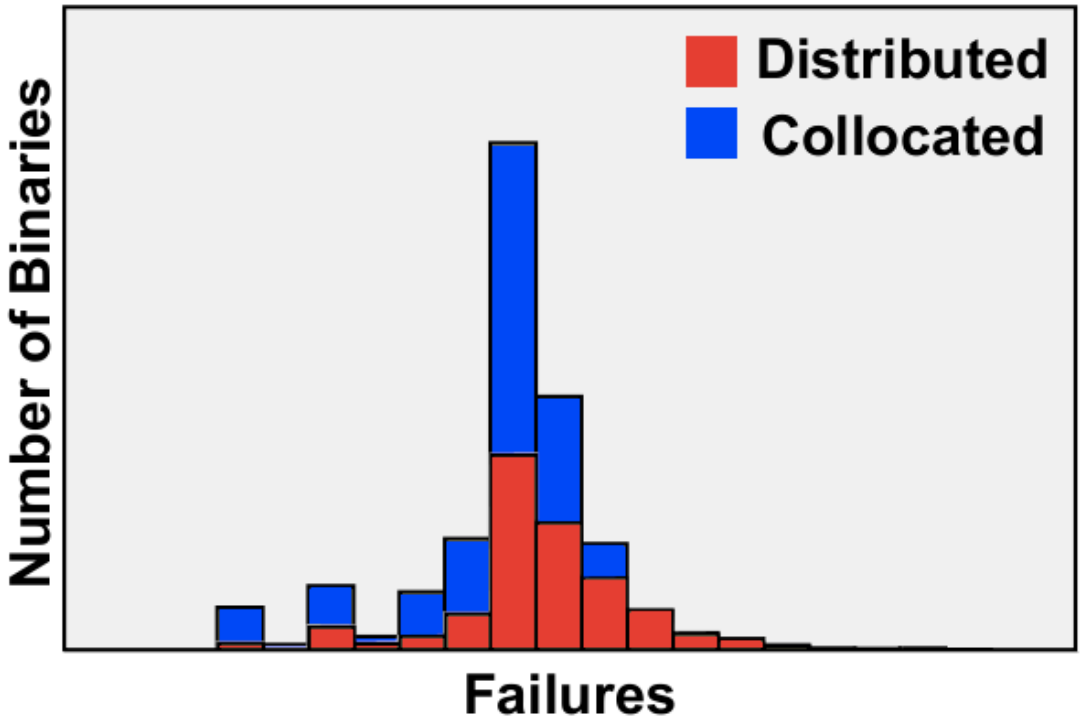


Step Two: Experimental Analysis

Number of bugs for distributed and non distributed binaries.

More binaries classified as non distributed.

Distribution of failures is very similar.



Mann-Whitney test

To quantitatively measure the difference in means because the number of failures was not normally distributed

- Small significant statistical difference.
- Average number of failures per binary is higher when the binary was distributed.
- 8% increase than non distributed.

Linear Regression

Results when splitting binaries at regions level (locality).

distributed: binary variable indicating whether the binary is distributed or not

numdevs: # of developers that worked on the binary

To examine effect of distributed development on number of failures.

Model 1 F Statistic = 12.43, $p < .0005$

Variable	% increase	Std Err.	Significance
(Constant)		0.30	$p < .0005$
distributed	9.2%	0.31	$p < .0005$

Model 2 F Statistic = 720.74, $p < .0005$

Variable	% increase	Std Err.	Significance
(Constant)		0.25	$p < .0005$
distributed	4.6%	0.25	$p = .056$
numdevs		0.00	$p < .0005$

Linear Regression …

To examine effect of the level of distribution on the number of failures of a binary.

diff_buildings is 1: if the binary was distributed among different buildings served by the same cafeteria.

% increase: the increase in failures relative to binaries that are developed in the same building.

Model 3 F Statistic = 25.48, $p < .0005$

Variable	% increase	Std Err.	Significance
(Constant)		0.09	$p < .0005$
diff_buildings	15.1%	0.50	$p < .0005$
diff_cafeterias	16.3%	0.21	$p < .0005$
diff_campuses	12.6%	0.35	$p < .0005$
diff_localities	2.6%	1.47	$p = .824$
diff_continents	-5.1%	0.31	$p = .045$

Model 4 F Statistic = 242.73, $p < .0005$

Variable	% increase	Std Err.	Significance
(Constant)		0.09	$p < .0005$
diff_buildings	2.6%	0.42	$p = .493$
diff_cafeterias	3.9%	0.18	$p = .016$
diff_campuses	6.3%	0.29	$p = .019$
diff_localities	8.3%	1.23	$p = .457$
diff_continents	-3.9%	0.26	$p = .101$
numdevs		0.00	$p < .0005$

Two Main Observations

When controlling for the number of developers:

- Variance explained by the predictor variables for the built models rises from 2% and 4% (models 1 and 3) to 33% (models 2 and 4)
- Not all levels of distribution show a significant effect, but the increase in post-- release failures for those that do is minimal with values at or below 6%.

Teams that were distributed wrote code that had virtually the same number of post--release failures as those that were non distributed.

Step Three: Differences in Binaries

Comparing differences in characteristics between distributed and non distributed binaries.

Code changes

- Size of changes
- Frequency of edits

Size and complexity

- Number of lines
- Number of functions
- Depth of inheritance

Dependencies

- Method Calls, Data Types, Registry Values

Differences in Binaries …

To examine relationship of development metrics with distribution level:

Spearman rank correlation: Even with low p values, most metrics have correlation levels <= 0.1.

Logistic regression

Only differences because of team sizes

No visible difference in the measured metrics between binaries that are distributed and non distributed.

Metric	Avg Value	Correlation	Significance
Functions	895.86	0.114	$p < .0005$
Complexity	4603.20	0.069	$p < .0005$
Churn Size	53430	0.057	$p = .033$
Edits	63.82	0.134	$p < .0005$
Indegree	13.04	-0.024	$p = .363$
Outdegree	9.67	0.100	$p < .0005$
Number of Devs	21.55	0.183	$p < .0005$

Discussion

- It is possible to conduct in-house globalized distributed development without adversely impacting quality.

- Why this occurred and how can the experience be repeated in other projects and contexts?

Positive Factors at Microsoft

Relationship between sites

- No competition between the sites.
- Sites has existed for a long time and worked on software together.
- Salaries and benefits to employees are equivalent at all sites in the company.

Cultural Barriers

- Professionals with lots of experience from different sites working together.
- For example - Microsoft employees moving from US to India.
- Increase trust and communication between the sites.
- Reduce organizational & cultural differences.

Common Schedules

- Fixed deadlines and release dates.

Positive Factors…

Communication

- Communication is the single most referenced problem in globally distributed development.
- Vista developers relied heavily on synchronous daily communication.
- Frequent contact like conference calls,
  - Increases the level of awareness
  - Resolves issues quickly
  - Convey status

Consistent Use of Tools

- Use of one configuration management and build system at all sites.
- All employees familiar with same tools.

Positive Factors…

End to End Ownership

- Common problem - who is responsible for task
- Many binaries committed to from different sites during implementation phase.
- Every developer in charge of particular piece of code of binary at all stages of development.

Organizational Integration

- No top level executive in one country for all to report to.
- Organizational structure spans geographical locations at low levels.
- A common direct manager for multiple sites.
  - More integrated into the company and project
  - Is able to spot problems in poor communication and coordination.

Threats to Validity

Construct Validity

- Automated data collection
- Production level tools used to collect metrics and other data – errors in measurement.

Internal Validity

- Factors to reduce distributed development problems are not empirically evaluated – further study required.
- May not included important characteristics of binary differences

External Validity

- Generalize to other situations – unclear.
- Some other ways and the particular characteristics
  - Outsourced project might behave differently.

Conclusion

- Binaries divided based on the level of geographic dispersion of their commits.
- Studied post-release failures for the Windows Vista - distributed development has little to no effect.
- Organizational differences are much stronger indicators of quality than geography.
- Microsoft might follows some special criteria for distributed projects which reduces the impact of difficulties.

Questions --- kind of suggestions

What do you think are the Limitations of the paper?

Other factors that might affect the distributed development?

Other Geographical separation levels?

What are further studies the authors can conduct?