

Product-centric Approach

Ohlsson and Alberg, 1996: metrics derived from design documents used for prediction
Basili *et al.*, 1996: Chidamber Kermer suite of OO design metrics useful in predicting class fault-proneness, confirmed by Subramanyam and Krishnan, 2003
Zimmerman *et al.*, 2007: mapped defects from Eclipse's bug database to source code locations
Menzies *et al.*, 2007, Nagappan *et al.*, 2006: no *best set* of code metrics for defect prediction

Knab *et al.*, 2006: static code attributes + metrics derived from change history
Ratzinger *et al.*, 2007: various size metrics, change history, other process-related metrics. "... *not size and complexity measures dominate defect-proneness, but many people-related issues are important.*"

Combine

Process-centric Approach

Nagappan and Ball, 2005: found that *absolute code churn* is a poor predictor
Hassan and Holt, 2005: Hit-rate caching
Weyuker *et al.*, 2007: developer info helped improve their prediction model
Bell *et al.*, 2006: used lines of code, file age, change history, type of language; this improved prediction accuracy significantly

This paper

Comparative analysis of all three approaches