Predicting Upgrade Project Defects based on Enhancement Requirements: An Empirical Study

Lei He, Juan Li, Qing Wang, Ye Yang

Institute of Software, Chinese Academy of Sciences

2009.05.17

Outlines

- Introduction
- Methodology
- Experiment
- Conclusion

Outlines

- *Introduction*
- Methodology
- Experiment
- Conclusion

Introduction

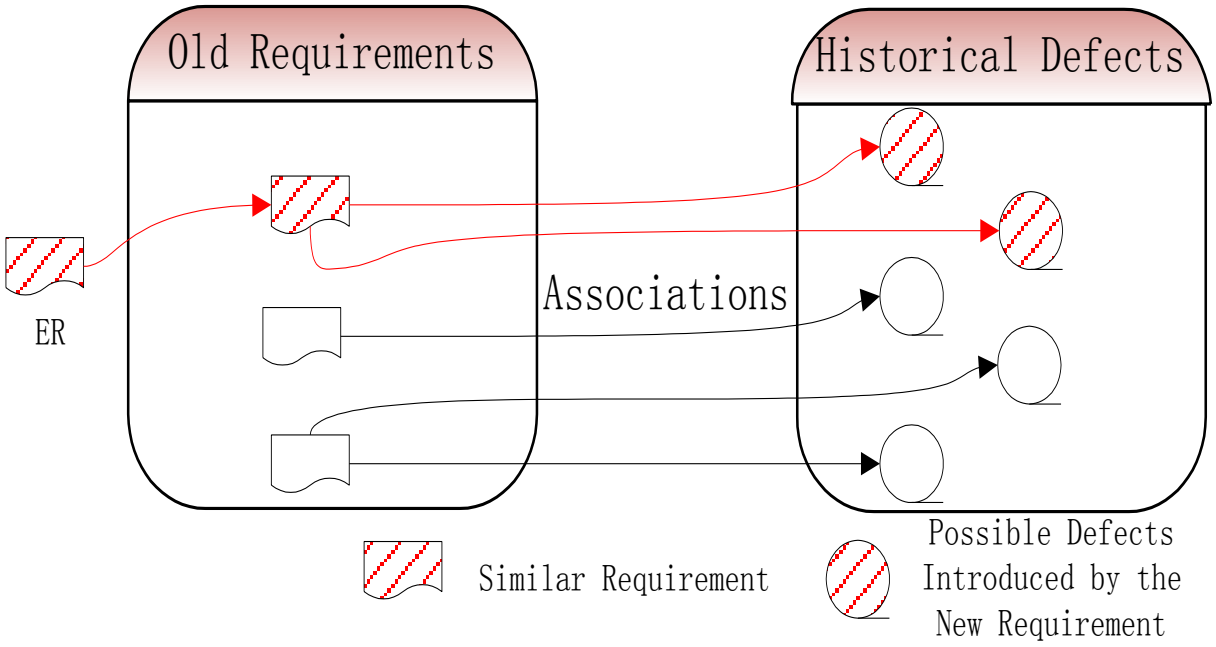
- Background
  - Upgrade Projects: delivering updated release or version of the software product
  - Enhancement Requirements (ER): requirement additions and modifications – Inject defects
- Motivation
  - ER impact analysis: risk management and cost reduction in early software lifecycle
  - Decision-making support: predict properties of defects may introduced (quantity, workload, type, priority…)

Introduction-cont. 1

- Prior work
  - Static defect prediction approaches applying Support Vector Machines(SVM)
  - Requirement measurement technologies (consistent, complete, correct…)
- Problem & Solving
  - Gap between requirement and defect: requirement→design→coding→testing(defects)
  - Solving: An empirical methodology applying Information Retrieval and Machine Learning technology (skip details)

Introduction-cont. 2

- Basic Idea

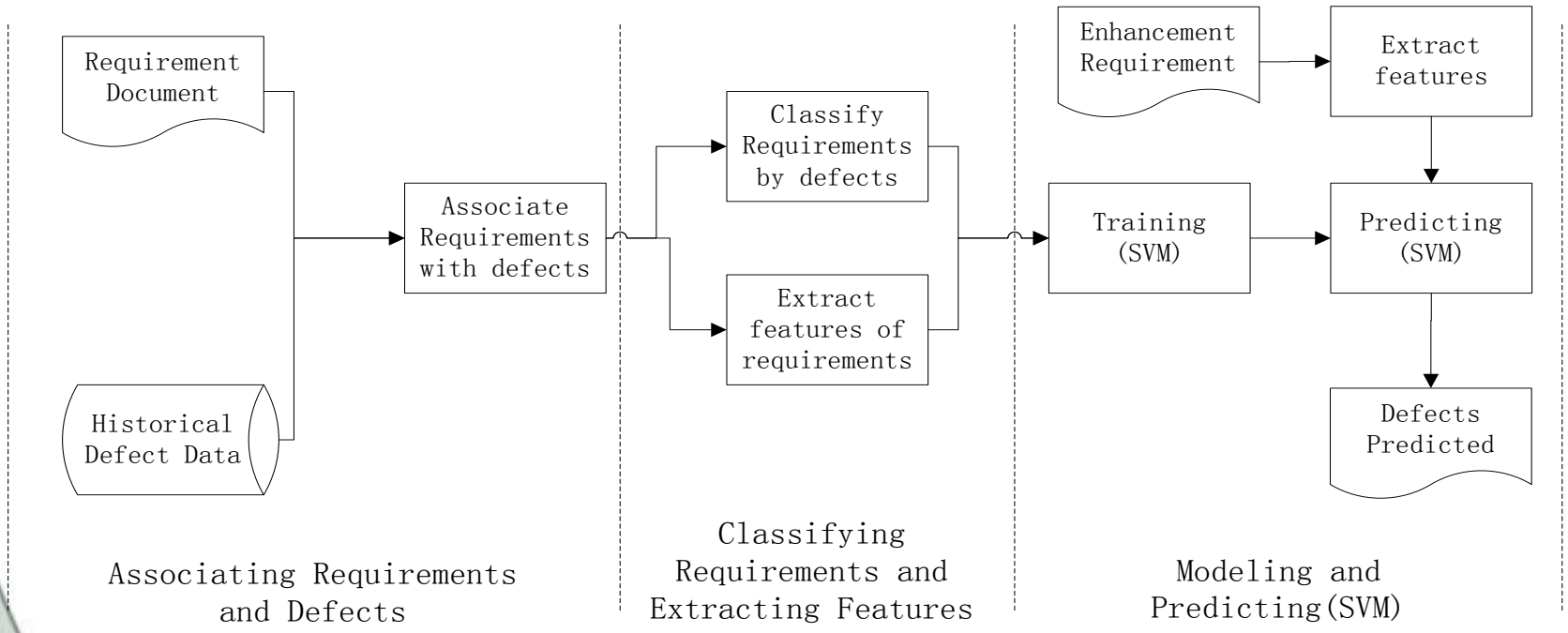


Outlines

- Introduction
- *Methodology*
- Experiment
- Conclusion

Methodology-overview

- Methodology overview: Coverting prediction problem to a classification problem



Methodology

- Associating historical requirements and defects
- Classifying historical Rqmt and extracting Rqmt features
- Modeling and predicting using SVM

Methodology

- *Associating historical requirements and defects*
- Classifying historical Rqmt and extracting Rqmt features
- Modeling and predicting using SVM

Step1: Associating

- Historical requirements(docs): use cases

·Name: Submit Work Plan;
·Description: To save and submit work plan in the work space;
·Pre-conditions: User logged in;
·Basic Event Flow: 1. Enter the work space, open the work plan table. 2. Input work plan.
3. Click 'submit' Button; Additional Work Flow: None;
·Post-conditions: A new plan added in project window. ……

- Historical defects(database records)

Name	Description
Title	Title of the defect.
Description	Brief description of the defect.
Workload	Workload to fix the defect (man-hour).
Priority	Priority of the defect: , Serious and Critical.
Type	Type of the defect: UI, Internal or Integrated.

Step1: Associating – cont.1

- Associating (IR technique)
  - Defect title +description as query
  - Requirement items (use case) as target indexing documents
  - Automated get matching scores using formula:

$$\sum_{t \text{ in } d} TF(t \text{ in } r) * IDF(t) * Boost(t.field \text{ in } r) * LengthNorm(t.field \text{ in } r)$$

  - Couples which [matching score > threshold(0.6 as default)] are considered as associated defects and requirements
  - Manual filtering

Methodology

- Associating historical requirements and defects
- *Classifying historical Rqmt and extracting Rqmt features*
- Modeling and predicting using SVM

Step 2: Classifying historical Rqmt and extracting Rqmt features

- Classifying historical requirements
  - According to the statistic properties of defects related, historical requirements can be divided to several classes.
  - Different classification standards are for predicting different defect properties. E.g.: classify requirements according to defect quantity related is to predict quantity of defects may introduced.
  - Discover more defects properties from DB records to predict more detailed information from different aspects.

Step 2: Classifying historical Rqmt and extracting Rqmt features –cont. 1

- Extracting features from requirements
  - Common Features: CHI metrics. Basic context metrics of requirement, counting classification weight of each term in the documents
  - Special Features: metrics perform differently in models predicting different properties
    - Module of the requirement (MoR)
    - Text length of the requirement item (LoR)
    - Number of items in event flow (NEF)
    - …

Step 2: Classifying historical Rqmt and extracting Rqmt features -cont.2

- Generating SVM input: each requirement item are converted to one vector

R: 3 1 3 2 5 0 0 2 4… 4 2 1 0 0 1 5…

class lable

Common Features

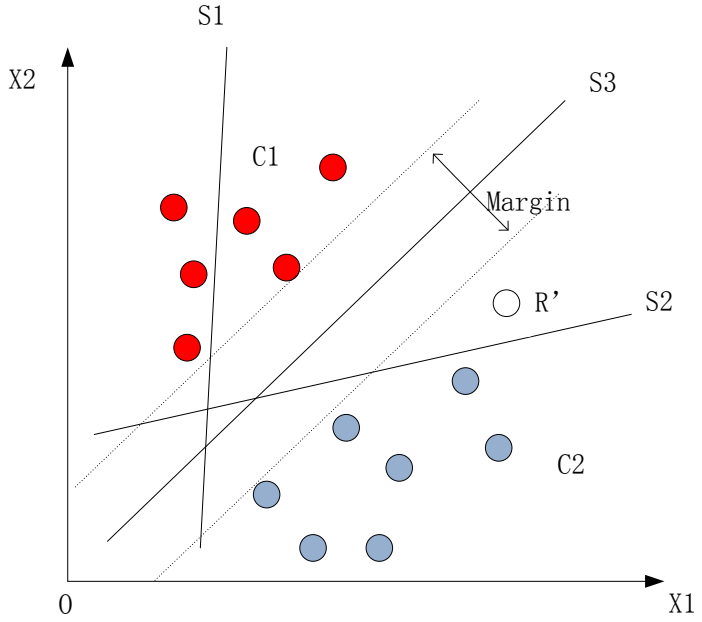
Special Features

Methodology

- Associating historical requirements and defects
- Classifying historical Rqmt and extracting Rqmt features
- *Modeling and predicting using SVM*

Step3: Modeling and Predicting

- SVM Models



- Advantages
  - High dimensional
  - Global optimum solution
  - Non linear
  - Prevent over-fitting

Outlines

- Introduction
- Methodology
- *Experiment*
- Conclusion

Experiment

- Data Set:
  - Defects and requirement data of 9 versions of SoftPM
    - SoftPM project is a software quality management platform, targeting to facilitate the process improvement initiatives in many small and medium software organizations(Since 2002).
  - 4893 defects and 581 requirement items (use case).
  - Establish four models of common defect properties
    - Quantity, workload, type, priority

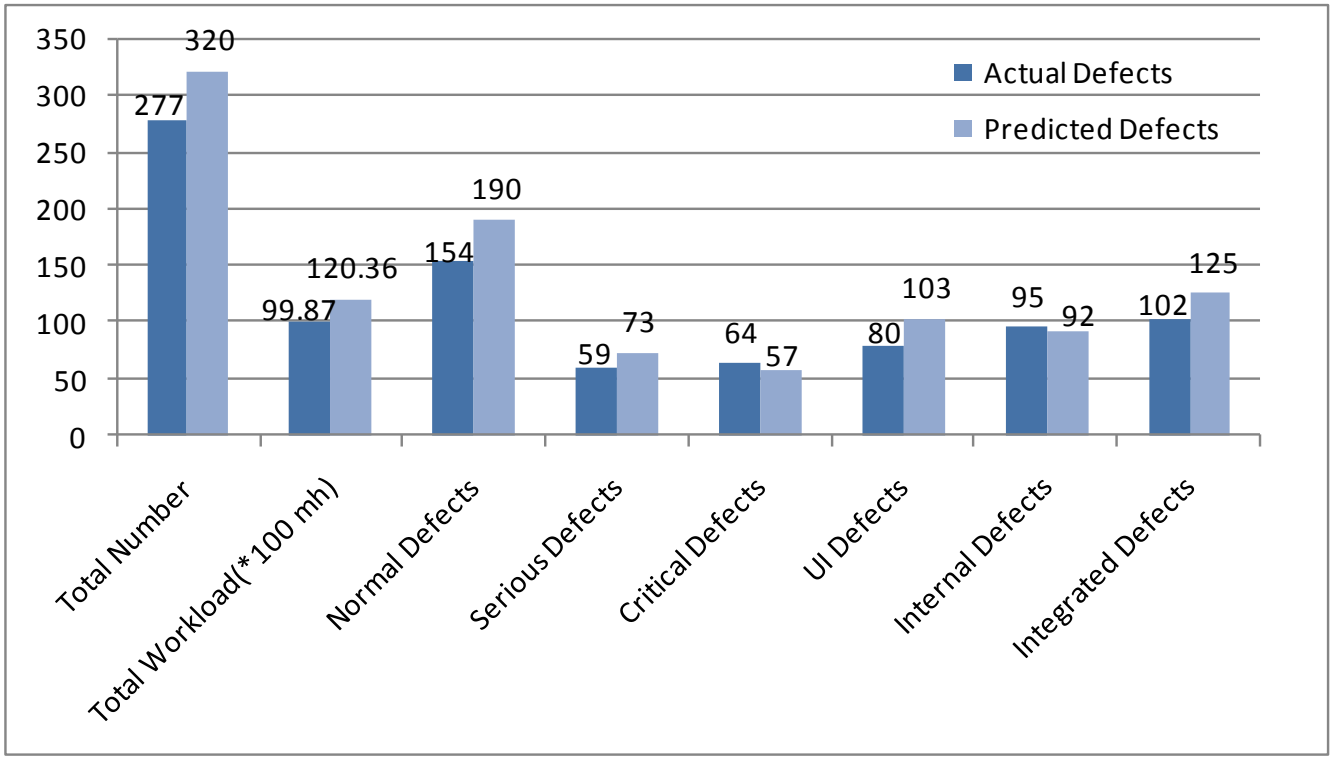
Experiment-cont.1

- Step 1 and 2 : historical data association and classification

Classification Standard	Classes	Boundary (Range)	Requirement Features
By **Quantity** of Related Defects	Extra Few	0~5	CHI (1000) MoR (50) LoR (100) NEF (100) PCR (50)
	Few	6~10	
	Medium	11~29	
	High	30~49	
	Extra High	More than 50	
By **Average Workload** of Related Defects	Extra Low	0~24 (man-hour)	CHI (1000) MoR (100) LoR (50) NEF (150)
	Low	25~48	
	Medium	49~144	
	High	145~288	
	Extra High	More than 289	
By **Priority** of related Defects	Normal	Regarding to defect properties recorded in the database	CHI (1000) MoR (200) NEF (100) PCR (50)
	Serious		
	Critical		
By **Type** of related Defects	UI	Regarding to defect properties recorded in the database	CHI (1000) MoR (100) NEF (100) DUI (50) RDP (50)
	Internal		
	Integrated		

Experiment- cont.2

- Step 3: modeling and predicting
  - 10-fold cross validation accuracy is around 65%
  - Quantified predicted results: a test set of 48 ERs



Outlines

- Introduction
- Methodology
- Experiment
- *Conclusion*

Conclusion

- Advantage
  - Novel and promising impact analysis for requirement evolution and software maintenance
  - Defect prediction in early software lifecycle
- Disadvantage
  - Empirical model, Differ from projects.
  - Requirement deletion?
- Future Work
  - Improve Accuracy: optimize metric selection
  - More data sets

Thank You!