Requirement Evolution
Traceability and Measurement

Juan Li
Requirement Engineering Research Group
iTechs Lab
Institute of Software, Chinese Academy of Sciences
June 4, 2009
Agenda

• Overview of the Research Group
• Research Background
• Research Topics and Framework
• Research Progress
  – Dynamic Requirement Evolution Traceability
  – Requirement Evolution Measurement
Overview of the Research Group

• Faculty
  – Prof. Mingshu Li, Prof. Qing Wang, Prof. Yongji Wang, Associate Prof. Juan Li

• Students
  – Ph. D.(2): Fei Dong, Yin Li
  – Ms. (7): Zhongsen Qin, Lingjun Kong, Jian Gao, Lei He, Hailong Wang, Lin Shi, Peng Wan

• Supports
  – National Basic Research Program (973 Program), Requirement evolution modeling, Requirement engineering – Basic Research of software engineering for complex system, 2007-2011
  – National Natural Science Foundation (NSFC)
    • Research on multidimensional software requirements description and control method, 2006-2008
    • Research on requirements evolution measurement method, 2009-2011

• Research Focus: Requirement Evolution Traceability and Measurement
Research Background

• Evolution
  – a process of continuous change from a lower, or worse to higher, more complex, or better state (Merriam-Webster Dictionary)

• Software Evolution
  – After software system is released, it needs to be maintained continuously because of changing user’s expect and application environment

• Requirement Evolution
  – Is the main cause of software evolution
  – express the changing nature of requirements over the system development life cycle
  – has not been well-defined in requirement engineering research literature.
    • encompasses requirements change, volatility, creeps
Impact of requirement evolution

• Requirement evolution is inevitable in the whole software lifecycle
  – Stark G. analyzed 44 releases of 7 products in one software company and found 64% releases were changed. The requirements changed were 48% [Stark 1998]
  – Ebert, C. analyzed 15 projects in 2002. In these projects, 73% requirements changed [Ebert 2006]

• Requirement uncertainty is thought a major factor affecting project success
  – Leads to the overrun of cost and schedule, decrease of quality
Requirement Evolution Process

- Propose change request
  - What to be changed
  - When happen

- Analyze evolution risk
  - Impact scope
  - Impact on project performances
    - cost, schedule, quality

- Assess evolution effectiveness
  - Management effort
  - Review efficiency

- Execute Evolution
  - Modify impacted artifact
  - Test if new needs are met
Research Framework

Dynamic Requirement Evolution Traceability

- Req Dependency
- Req and artifact traceability
  - Req and code
  - Req and defect

Requirement Evolution Measurement

- Evolution Measures
- Data Collection
- Evolution Analysis

Requirement Evolution Data Mining

- Comparing requirement document
- Mining process repository
1 Dynamic Requirement Evolution Traceability
Requirement Traceability

- Requirement traceability
  - the ability to describe and follow the life of an artifact (requirements, design model, code, tests, etc.) developed during the software lifecycle in both forward and backward directions
- Providing important insights into system development and evolution impact analysis
- Giving essential support in understanding the relationships existing within and across software requirements, design and implementation
Dynamic Requirement Traceability

• Lack of automatic or semi-automatic traceability link generation in traditional software development

• Dynamic requirement traceability adopts Information Retrieval (IR) technology to establish traces automatically
Related work

- Directly applied IR method
  - recover traces between code and requirement
    [Antoniol 2002] [Andrian 2003]
  - Boolean model, Vector Space IR, Probability IR, Latent semantic indexing (LSI)

- Revised IR method according to project or artifact characteristic
  - introduce key-phrase and revise the similarity formula
    [Hayes 2003]
  - use hierarchical information of requirement and artifact to improve the precision [J. Cleland-Huang 2005]
The Process of Dynamic Requirement Evolution Traceability

**Requirement Dependency**
- Dependency Identification
  - Generate candidate reference and similarity
  - dependencies manually
- Maintain the dependencies
- Update Requirements

**Impact Analysis**
- Traceability Evaluation
  - Analyze the traces
- Impact Analysis
  - recursive

**Req and code Traceability**
- Trace Identification
  - Generate candidate
  - Filter out the candidate Traces manually
  - Requirement specification
  - Code Traces
- Update Requirements
- Maintain the traces

Change Proposal

Decision Making
- User Decision
  - Make decision according to the report
  - Agree the change proposal

Agree the change proposal
Requirement dependency

• Requirement dependency is useful to handle requirement evolution
  – Show how requirements affect each other
• If requirement A depend on requirement B, then when B is changed, A must be modified to maintain consistency with B.

  A  Depend on  B

• We define two kinds of requirement dependencies and corresponding rules to identify them
Two kinds of dependency

- **Similarity Dependency**
  - Requirement 1 and requirement 2 use same concept

- **Reference Dependency**
  - Requirement 2 refers to requirement 1
Identification Rule

• Clustering Rule
  – Identify the similarity dependency
  – If one Req contains sentences similar to sentences in another Req, then these two Reqs may have a dependent relation.

• Referring Rule
  – Identify the reference dependency
  – If the title of Req A is used in sentences of other Reqs then these Reqs depend on A

Compute similarity in requirements
Dependency identification for Requirement Evolution (DiRE)

- Requirement document
- XML
- Transform
- Input
- Clustering Rule
- Referring Rule
- Information Retrieval Engine
- Similarity Dependency
- Reference Dependency
- Output
- Vector Space IR Model
Experiment

- We collected more than 300 pages of requirement documents from 3 projects in a Chinese software company

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Req number</th>
<th>Doc Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Point Estimation System (FPE)</td>
<td>41</td>
<td>Use Case</td>
</tr>
<tr>
<td>Requirement Management System (RM)</td>
<td>54</td>
<td>non structured req doc</td>
</tr>
<tr>
<td>Test Management System (TM)</td>
<td>76</td>
<td>Use Case</td>
</tr>
</tbody>
</table>
Experiment Design

• Three analysts were involved in experiment
  – Analyst A
    • 8-year requirement analyst experience in 15 projects
  – Analyst B
    • 3-year requirement analyst experience in 4 projects
  – Analyst C
    • 4-year requirement analyst experience in 3 projects
<table>
<thead>
<tr>
<th>Project</th>
<th>Req No</th>
<th>CRN</th>
<th>Phase</th>
<th>FRN</th>
<th>FCRN</th>
<th>Precision (FCRN/FRN)</th>
<th>Recall (FCRN/CRN)</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>RM</td>
<td>54</td>
<td>36</td>
<td>DiRE</td>
<td>initial</td>
<td>38</td>
<td>20</td>
<td>52.6%</td>
<td>55.6%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>filtering</td>
<td>26</td>
<td>26</td>
<td>100%</td>
<td>72.2%</td>
</tr>
<tr>
<td></td>
<td>Manual</td>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td>13</td>
<td>65%</td>
<td>36.1%</td>
</tr>
<tr>
<td>TM</td>
<td>76</td>
<td>104</td>
<td>DiRE</td>
<td>initial</td>
<td>143</td>
<td>85</td>
<td>59.4%</td>
<td>81.7%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>filtering</td>
<td>88</td>
<td>88</td>
<td>100%</td>
<td>84.6%</td>
</tr>
<tr>
<td></td>
<td>Manual</td>
<td></td>
<td></td>
<td></td>
<td>36</td>
<td>20</td>
<td>55.6%</td>
<td>19.2%</td>
</tr>
<tr>
<td>FPE</td>
<td>41</td>
<td>34</td>
<td>DiRE</td>
<td>initial</td>
<td>62</td>
<td>25</td>
<td>40.3%</td>
<td>73.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>filtering</td>
<td>30</td>
<td>30</td>
<td>100%</td>
<td>88.2%</td>
</tr>
<tr>
<td></td>
<td>Manual</td>
<td></td>
<td></td>
<td></td>
<td>24</td>
<td>17</td>
<td>71%</td>
<td>50%</td>
</tr>
</tbody>
</table>

CRN: The number of the correct relations  
FRN: The number of the relations found  
FCRN: The number of the correct relations found  
Precision : The percentage of the correct retrieved relations on the retrieved relations  
Recall: The percentage of the correct retrieved relations on all the correct relations  
Time: Time spent on identifying the dependency
Req and code traces identification

Dependency Identification
- Generate candidate reference and similarity dependencies
- Filter out the candidate dependencies manually
- Maintain the dependencies

Impact Analysis
- Traceability Evaluation
  - Analyze the traces
- Recursive

Dependency Evaluation
- Analyze the dependencies

Decision Making
- User Decision
  - Make decision according to the report
  - Agree the change proposal

Change Proposal
- Update Requirements
- Maintain the traces

Requirement dependencies

Requirement traceability

Trace Identification
- Generate candidate traces between
- Traces manually

Code, Tasks and Requirements
- Update Requirements

Req and code Traceability
Requirement traces establishing and refining

- The requirement is query, the artifact is document, and the process of establishing the RT is to execute the search. Computing the similarity between requirement and code to identify the traces.
Relevance Feedback

• We use RF strategy to improve the accuracy of dynamic requirement traceability
  – A classical way in information retrieval to improve the retrieval performance

• Process of feedback
  – First do an initial retrieval and next update the query based on user’s evaluation on the retrieved document, and then do retrieval again with the new query which will hopefully have better retrieval performance

• Evaluation mainly consists of user’s effort to verify the retrieved document relevant (positive feedback information) or irrelevant (negative feedback information) to user’s query
Requirement traces refining

• Four feedback methods

<table>
<thead>
<tr>
<th>IR Models</th>
<th>Feedback Methods</th>
<th>Simple Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSM</td>
<td>SR</td>
<td>Standard Rochio method in VSM</td>
</tr>
<tr>
<td></td>
<td>nSR</td>
<td>modification of SR (omitting negative feedback in SR)</td>
</tr>
<tr>
<td>LM</td>
<td>MM</td>
<td>Mixture Model in LM</td>
</tr>
<tr>
<td></td>
<td>eMM</td>
<td>extension of MM (adding negative feedback to MM)</td>
</tr>
</tbody>
</table>

• Conduct experiment to assess the effectiveness of these methods
  – Question 1: Is relevant feedback method effective to refine requirement traceability?
  – Question 2: If feedback method’s performance varies with the increase of number of iterations of feedback?
  – Question 3: Is there any performance difference between eMM and SR?
Experiment and Evaluation

• Subject of experiments
  – One industrial project, 43 requirements, 468 classes
  – 1090 correct traces

<table>
<thead>
<tr>
<th>Project Characteristics</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project type</td>
<td>Web based application</td>
</tr>
<tr>
<td>Development process</td>
<td>Iterative</td>
</tr>
<tr>
<td>Development tool</td>
<td>Java/applet.struts/jsp/ajax, mysql, tomcat4</td>
</tr>
<tr>
<td>Team size</td>
<td>5 members</td>
</tr>
<tr>
<td>Developer skill</td>
<td>Experience</td>
</tr>
<tr>
<td>Project duration</td>
<td>30 weeks</td>
</tr>
<tr>
<td>Project scale</td>
<td>43 use cases, 70 KLOC, 468 classes</td>
</tr>
<tr>
<td>Logical module number</td>
<td>7</td>
</tr>
<tr>
<td>Deployment package</td>
<td>pmreq.jar, pmapplet.jar, pmwss.jar</td>
</tr>
</tbody>
</table>
Experiment design

• Establish correct traces manually
• Preprocess requirement and code
• Establish the initial traces automatically
• Incorporate analyst’s feedback
  – Feedback was performed by a feedback simulator
  – Four subsets were elicited from the relationship set
  – Simulate 8 iterations of feedback
Results analysis and evaluation

- Question 1: Is relevant feedback method effective to refine requirement traceability?

<table>
<thead>
<tr>
<th>Subset Proportion</th>
<th>IR methods</th>
<th>Initial Evaluation (avg. Prec./Recall)</th>
<th>Evaluation with 8 Iterations of Feedback (avg. Prec./Recall)</th>
<th>% delta in Prec./Recall (all are increase)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>LM (eMM)</td>
<td>46.7/33.2</td>
<td>59/42</td>
<td>21/14.8</td>
</tr>
<tr>
<td></td>
<td>VSM (SR)</td>
<td>50/55.6</td>
<td>57.6/41</td>
<td>70/49.8</td>
</tr>
<tr>
<td>0.1</td>
<td>LM (eMM)</td>
<td>36.2/51.5</td>
<td>41.7/59.3</td>
<td>49.7/70.8</td>
</tr>
<tr>
<td></td>
<td>VSM (SR)</td>
<td>39.8/56.6</td>
<td>45.7/65</td>
<td>52.3/74.6</td>
</tr>
<tr>
<td>0.15</td>
<td>LM (eMM)</td>
<td>27.6/58.9</td>
<td>31.1/66.4</td>
<td>37.6/80.3</td>
</tr>
<tr>
<td></td>
<td>VSM (SR)</td>
<td>32.1/68.5</td>
<td>35.1/74.9</td>
<td>38.2/81.7</td>
</tr>
<tr>
<td>0.2</td>
<td>LM (eMM)</td>
<td>23.2/66.1</td>
<td>26.4/75.2</td>
<td>30/85.4</td>
</tr>
<tr>
<td></td>
<td>VSM (SR)</td>
<td>25.8/73.6</td>
<td>27.7/78.9</td>
<td>29.5/84</td>
</tr>
</tbody>
</table>

Subset proportion: the subset we choose from the initial relationship set
Experiment and Evaluation

• Question 2: If feedback method’s performance varies with the increase of number of iterations of feedback?
  
  – The first two iterations of feedback result in significant increases both in precision and recall
Experiment and Evaluation

- Question 3: Is there any performance difference between eMM and SR?
  - eMM is generally more stable than SR, but SR can get better performances
Requirement Traceability Tool

Req and Code Traceability

Dependency
2 Requirement Evolution Measurement
Questions

• What are the causes of requirement evolution?
• What is the impact of requirement evolution?
• What are the characteristics of requirement evolution?
• How to assess the organization effectiveness at handling evolution?
Measures

• **Cause**
  – Requirement
    • size, dependency density, traceability density
  – Requirement quality
    • defect, req activity effort
  – Project characteristic
    • human capability, development process, technology maturity

• **Impact**
  – req Impacted, artifact impacted, effort caused, defect caused, schedule impacted

• **Evolution characteristics**
  – change cause, change phase, change type, change effort, requirement stability, change frequency

• **Management Capability**
  – Schedule variance, cost variance, product quality variance, change request acceptance rate, change management effort, change review effort and efficiency
Requirement Measurement Tool

- Prototype Tool was developed and integrated with Qone
  - Collect data from Requirement Management Tool, Project Management Tool
Data Collection and Analysis (1)

• One enhanced project in a Chinese medium-sized organization
  – Domain: Software development and project management
• 8 releases (2004 – 2008)
• Document collected
  – requirement specification, project plan, project summary, bug report, user feedback report

<table>
<thead>
<tr>
<th>Release</th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
<th>V5</th>
<th>V6</th>
<th>V7</th>
<th>V8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Cases</td>
<td>186</td>
<td>189</td>
<td>195</td>
<td>221</td>
<td>276</td>
<td>310</td>
<td>303</td>
<td>339</td>
</tr>
</tbody>
</table>
Use case changes in different releases

- From V1 ~ V4, changes include modification and addition. From V5 ~ V8, function deletion was conducted in every version
Deleted use case type distribution

- V5 ~ V8, about 40 functions were deleted
- Half of these functions were refined
  - combined, split, replaced
Use case evolution type distribution

- The ratio of modification is biggest
• Users’ business rules are various. To satisfy their needs, the product needs to be modified.
• The company is trying to adopt the workflow engine to let user reconfigure the business process.
• Employ business analyst in the project team.
Next work

• Analyze requirement change impact
  – Size, effort, duration

• Establish prediction model
  – Size, effort, duration
Data Collection and Analysis (2)

- One ongoing project

<table>
<thead>
<tr>
<th>Category</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>Browser / Server</td>
</tr>
<tr>
<td>Development process</td>
<td>Analyzing requirements, designing, coding and testing</td>
</tr>
<tr>
<td>Development cycle</td>
<td>2009/01 – 2009/07</td>
</tr>
<tr>
<td>Total stuff</td>
<td>25</td>
</tr>
<tr>
<td>Requirements</td>
<td>User requirements: 169, software requirements: 140, Average length of requirement is 81.149 words.</td>
</tr>
<tr>
<td>User</td>
<td>&gt;=1</td>
</tr>
</tbody>
</table>
• Model requirements with EA (Enterprise Architect) tool. Then export the requirement model into RM tool developed by themselves.
• Requirement baseline is used as basis of development.
  – 7 baselines
• Requirement change is regarded as baseline change
  – 6 requirement changes
  – Including requirement addition, deletion, and modification
Change phase distribution

- Requirement changes happened in late phases, e.g. coding and testing

<table>
<thead>
<tr>
<th>Change ID</th>
<th>Cause</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>user proposed new req</td>
<td>Coding</td>
</tr>
<tr>
<td>2</td>
<td>Unclear req</td>
<td>Coding</td>
</tr>
<tr>
<td>3</td>
<td>user proposed new req</td>
<td>Coding</td>
</tr>
<tr>
<td>4</td>
<td>incorrect req</td>
<td>Coding</td>
</tr>
<tr>
<td>5</td>
<td>missing req</td>
<td>Testing</td>
</tr>
<tr>
<td>6</td>
<td>users proposed new req</td>
<td>Testing</td>
</tr>
</tbody>
</table>
Requirement baseline stability

• Requirement baseline stability is above 69%

Requirements baseline stability: unchanged requirements / total requirements in one baseline
Requirement changes

• Requirement changes mainly includes requirement addition and modification.
Preliminary Analysis

• Change happened in later phases
• Requirement changes were not controlled effectively
  • 100% acceptance rate
  • Lack of impact scope analysis
    – Did not use the RM tool effectively. No tracking info is recorded.
      » Requirement and artifacts
      » Requirement and tasks
Thanks for your attention!