SOFTWARE REUSE IN AN INDUSTRIAL SETTING:
A CASE STUDY

Michael F. Dunn John C. Knight
What is Re-engineering?

“The examination and alteration of a system to reconstitute it in a new form”.

What is Re-engineering?

1. Manual Method
   • Converting identical functions/modules in the existing language to identical functions/modules in the destination language
   • Finding the recurring patterns to identify the potential functions/objects in the existing system and group the candidate methods/functions in the same class of new system
   • Starting from the first principles (e.g. problem statement, design) and develop the product from the ground up

2. Repeatable Method
   • Similar to option 2 in manual method but perform it automatically by using metric tools
On Going Case Study

• Sperry Marine Incorporated.
• Computerized marine electronic systems.

Do you think publishing the results of an ongoing case study present a threat to the study validity?
Goals

• To analyze the problems that limit reuse.
• To seek solutions suitable for industrial application.
1. Select a domain
Scoping exercise
Searching for reusable components in existing systems

Domain: Electronic Charts
2. Perform a domain analysis
List the features of sea- and land-based system
3. Prepare reuse library
Implement reuse library for the domain (from C to C++)
4. Construct & Measure
Implement sample sea- and land-based system using the reuse library
## Evaluation: Evaluation Framework

<table>
<thead>
<tr>
<th>Evaluation Steps</th>
<th>Key Issues</th>
<th>Actual Issues &amp; Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initial Reuse Library Development</strong></td>
<td>• Design Decisions</td>
<td>• Chart coordinate &lt;-&gt; window coordinate</td>
</tr>
<tr>
<td></td>
<td>• Implementation Language</td>
<td>• Ensuring the flexibility in the addition of new chart features</td>
</tr>
<tr>
<td></td>
<td>• Total resources required to create the initial reuse library</td>
<td>• Inherit from the base class</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Implementation language</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• C++</td>
</tr>
<tr>
<td><strong>Using the components in Application</strong></td>
<td>• Development Effort</td>
<td>• All the components from the reuse library were used in creating the sample sea- and land-based system.</td>
</tr>
<tr>
<td></td>
<td>• Effectiveness of domain analysis</td>
<td>• Sea-based</td>
</tr>
<tr>
<td></td>
<td>• Time &amp; space efficiency and levels of reliability and availability</td>
<td>• New classes created = 457 LOCs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Changes in existing classes = 15%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Reuse = 2600 LOCs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Land-based</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• New classes created = 420 LOCs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Changes in existing classes = 0%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Reuse = 2710 LOCs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Performance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Load: expected = 3-5 secs, actual = 6 secs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Zoom: expected = 1-3 secs, actual = 1.2 secs</td>
</tr>
<tr>
<td><strong>Characterizing Change</strong></td>
<td>• Amount of change necessary to use library components to build a new application</td>
<td>• Using version table</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Backtracking</td>
</tr>
<tr>
<td><strong>OO Design and Code Scavenging</strong></td>
<td>• The ease with which code scavenged from other non-OO systems into OO reuse library</td>
<td>• Manual method (option 2)</td>
</tr>
</tbody>
</table>
"Old software is not reusable" by R.Carle in *Reusable Software Components for Missile Applications* [3rd paragraph of page 331]
Software reuse was completed in real organization before!! (although it’s just a sample one)
Questions

• Do you think that the programming language a piece of software is written in affects the level of which the code can be reused? (i.e. If this program was written in C# would it be more reusable?)

• In this paper, for reuse they used c++. if we have used JAVA rather than c++ would it be more reusable? What do you think?
Questions

• Do you think if there were some kind of diagrams representing the functions of this system that it would be easier to analyze the code for reusability?
• In this paper they haven’t used any UML diagrams. If we used UML’s would it be more easy for reuse. What do you think?
• Do you agree with the paper that no design techniques exist that are specifically intended to take advantage of a set of reusable components?
Questions

• Is code scavenging a form of opportunistic reuse, or could it be part of the plan in future projects?
• Not a good argument for not including exception handling--yes handling exceptions is part of most "good" coding...why would this be ignored for reuse?
Questions

• Do you think the methods that the authors describe can be applied to other organizations or should the reuse strategies be unique to each organization?

• What could be the potential challenges in case of reengineering systems built using object oriented technology?
Questions

• What does “scavenging” mean? Are there any formal techniques to do it?
Questions

• How do you know if the reuse library contains the “right” parts and if they are presented in the “right” way? What is this “right” means?
References

• “An Empirical Comparison of Methods for Reengineering Procedural Software Systems to Object-Oriented Systems” by W.B.Frakes, G.Kulczycki & N.Moodliar
• “Software Reuse in an Industrial Setting: A Case Study” by M.F.Dunn & J.C.Knight
• www.cin.ufpe.br/~in1045/slides/app02.ppt)[Access Feb. 28, 2010].