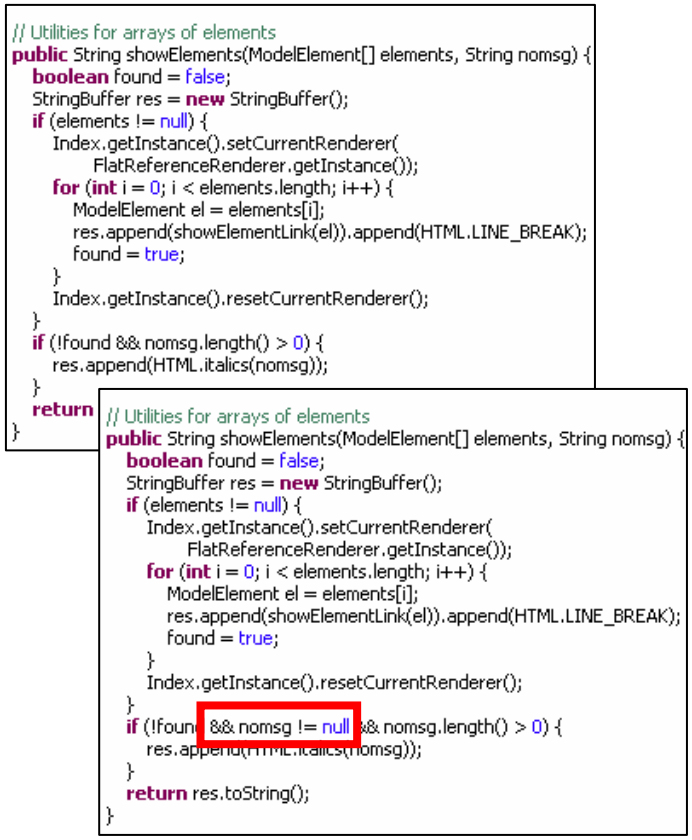
Elmar Juergens, Florian Deissenboeck

How Much is a Clone?

March 15th, 2010
4th SQM, Madrid

Problem

- Cloning abounds in real world SW
- Negative consequences established *qualitatively*
  - Size increase
  - Multiple modifications
  - Faults through inconsistencies
- But: *quantitative* impact unclear
- How harmful is cloning really?
- Unquantified issues likely neglected



```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
   boolean found = false;
   StringBuffer res = new StringBuffer();
   if (elements != null) {
      Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
      for (int i = 0; i < elements.length; i++) {
         ModelElement el = elements[i];
         res.append(showElementLink(el)).append(HTML.LINE_BREAK);
         found = true;
      }
      Index.getInstance().resetCurrentRenderer();
   }
   if (!found && nomsg.length() > 0) {
      res.append(HTML.italics(nomsg));
   }
   return res.toString();
}
```

```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
   boolean found = false;
   StringBuffer res = new StringBuffer();
   if (elements != null) {
      Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
      for (int i = 0; i < elements.length; i++) {
         ModelElement el = elements[i];
         res.append(showElementLink(el)).append(HTML.LINE_BREAK);
         found = true;
      }
      Index.getInstance().resetCurrentRenderer();
   }
   if (!found && nomsg != null && nomsg.length() > 0) {
      res.append(HTML.italics(nomsg));
   }
   return res.toString();
}
```

Better understanding of economic consequences of cloning required

Agenda

Terms

Maintenance Process

Clone Cost Model

Instantiation & Case Study

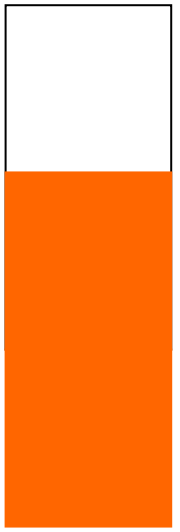
Conclusion

Terms

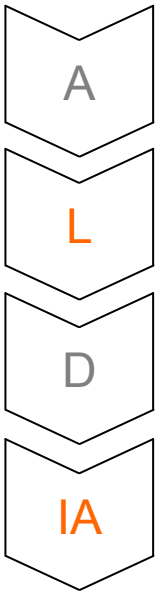
Clones Regions of similar code

- Change coupling due to implementation of common concept
- Syntactic similarity (required for detectability, not change coupling)

SS	Number of Source Statements (w/o comments, whitspace)	150
RFSS	Number of Redundancy Free Source Statements	100
Overhead	Relative size increase due to cloning	50%



Maintenance Process



Analysis Studies feasibility of CR and devises preliminary plan

- Takes place on problem domain -> not affected by cloning

Location Determines change start points.

- Involves inspection of code. *Effort ~ amount of inspected code*
- Cloning increases code size and thus effort.

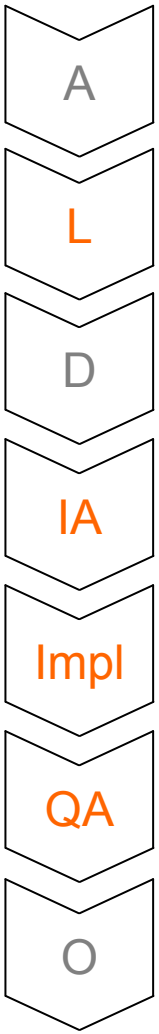
Design Design modification of system based on A&L results

- Not affected by cloning, since code not central artifact

Impact Analysis Determines all change points from change start points

- *Effort ~ number of change points*
- Cloning increases number of change points and thus IA effort.

Maintenance Process (2)



Implementation Realizes designed change in source code

- Effort ~ amount of added and modified code
- *Addition:* Unaffected by cloning
- *Modification:* Affected by cloning

Quality Assurance Validates change w.r.t. quality requirements

- Not limited to single QA type. Systematic QA assumed.
- Smart QA strategy assumed. *Effort ~ amount of affected code*
- Cloning affects QA since it increases size of code affected by change

Other Further activities (delivery, deployment, …)

- Not affected by cloning, since code not central artifact

Cost Model: Approach

- Total effort as sum of efforts of individual change requests
- Population of CR determines scope

$$E = \sum_{cr \in CR} e(cr)$$

- Effort for change request as sum of activity efforts

$$e = e_A + e_L + e_D + e_{IA} + e_{Impl} + e_{QA} + e_O$$

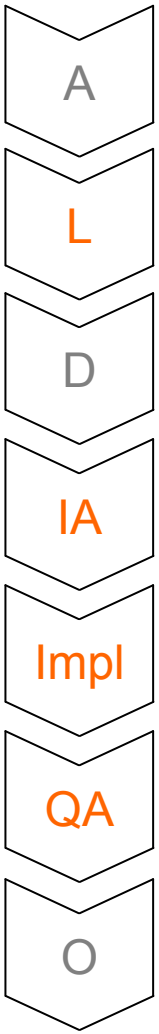
- Activity effort as sum of *inherent* and *cloning induced* effort

$$e = e^i + e^c$$

- Result: Effort increase relative to inherent effort

$$\Delta e = \frac{e^i + e^c}{e^i} - 1 = \frac{e^c}{e^i} = \Delta E$$

Activity Cost Models



Location

- Effort increases with code size

$$e^{c}_{L} = e^{i}_{L} \cdot overhead$$

Impact Analysis

- Effort increases with code size

$$e^{c}_{IA} = e^{i}_{IA} \cdot overhead$$

Implementation

- Effort for modifications and additions
- Mod effort increases with code size

$$e_{Impl} = e_{Impl_{Mod}} + e_{Impl_{Add}}$$

$$e_{Impl_{Mod}} = e_{Impl} \cdot mod$$

$$e^{c}_{Impl} = e^{i}_{Impl} \cdot mod \cdot overhead$$

Quality Assurance

- Additions & modifications affect QA
- Assumption: cloning in added and modified code similar

$$e^{c}_{QA} = e^{i}_{QA} \cdot overhead$$

$$e^{c} = overhead \cdot (e^{i}_{L} + e^{i}_{IA} + e^{i}_{Impl} \cdot mod + e^{i}_{QA})$$

Maintenance Effort Increase

Relative

$$\Delta e = \frac{overhead \cdot (e^{i}_{L} + e^{i}_{IA} + e^{i}_{Impl} \cdot mod + e^{i}_{QA})}{e^{i}_{A} + e^{i}_{L} + e^{i}_{D} + e^{i}_{IA} + e^{i}_{Impl} + e^{i}_{QA} + e^{i}_{O}}$$

Relative Cost Model

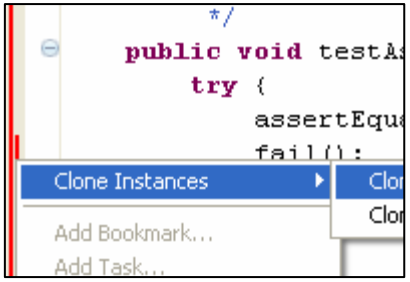
- Very many factors influence maintenance productivity
- Absolute cost models needs to include them (=> costly instantiation)
- *Relative* cost model compares cloned and non-cloned system
- Many factors remain constant and can be removed from the model

This model does not include potential additional efforts caused by increases in remaining field defects caused by cloning. (Future work)

Tool Support

Clone Indication

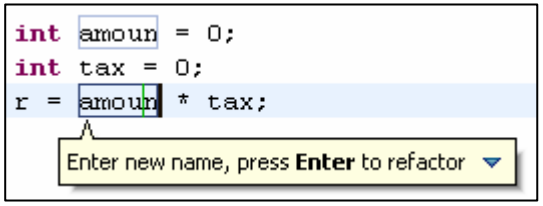
- Informs developers about cloning relationships
- Ideally reduces *IA* overhead to zero



Linked Editing

- Replicates edit operations to siblings.

  (Also: clone indication.)
- Ideally reduces *IA* and *Impl* overhead to zero



$$\Delta e = \frac{overhead \cdot (e^{i}_{L} + e^{i}_{QA})}{e^{i}_{A} + e^{i}_{L} + e^{i}_{D} + e^{i}_{IA} + e^{i}_{Impl} + e^{i}_{QA} + e^{i}_{O}}$$

Simplified Cost Model

- All parameters except overhead quantify effort distribution
- Gather into single parameter: Cloning affected effort (CAE)
- CAE possibly simpler to determine than individual parameter values

$$\Delta e = overhead \cdot \frac{e^{i}_{L} + e^{i}_{IA} + e^{i}_{Impl} \cdot mod + e^{i}_{QA}}{e}$$

$$CAE = \frac{e^{i}_{L} + e^{i}_{IA} + e^{i}_{Impl} \cdot mod + e^{i}_{QA}}{e}$$

Simplified Model:

$$\Delta e = overhead \cdot CAE$$

Instantiation

Overhead

- Computed automatically on detection results
- Tailoring required for accurate detection results and thus overhead

Effort parameters

- Project specific - determine for each project individually

Literature values

- Less accurate (but cheap)
- Effort distribution still insufficiently understood
- Modification ratio: 0.63 (based on CR type distros)

Activity	[31]	[6]	[36]	Estimate
Analysis			26%	5%
Location		13%		8%
Design	30%	16%	19%	16%
Impact Analysis				5%
Implementation	22%	29%	26%	26%
Quality Assurance	22%	24%	17%	22%
Other	26%	18%	12%	18%

Case Study

Goal

- Evaluation of cost model
- First quantification of impact of cloning on maintenance efforts

Objects (Selected based on existing contacts)

- 11 industrial software systems (7 companies, 5 languages)

Design & Procedure

- ConQAT used for clone detection and overhead computation
- Tailoring with developer feedback to achieve accurate detection results
- Effort values from literature used (=> limited result accuracy):

  L: 8%, IA: 5%, Impl: 26% * 0.63, QA: 22% => CAE: 50% (51.38%)

Validation of correctness of cost model results beyond scope of study

Results

System	Language	kLOC	kSS	kRFSS	overhead	ΔE	ΔE_{Tool}
A	XSLT	31	15	6	150.0%	75.0%	67.5%
B	ABAP	51	21	15	40.0%	20.0%	18.0%
C	C#	154	41	35	17.1%	8.6%	7.7%
D	C#	326	108	95	13.7%	6.8%	6.2%
E	C#	360	73	59	23.7%	11.9%	10.7%
F	C#	423	96	87	10.3%	5.2%	4.7%
G	ABAP	461	208	155	34.2%	17.1%	15.4%
H	C#	657	242	210	15.2%	7.6%	6.9%
I	Cobol	1,005	400	224	78.6%	39.3%	35.4%
J	Java	1,347	368	265	38.9%	19.4%	17.5%
K	Java	2,179	733	556	31.8%	15.9%	14.3%

- Overhead ranges between 5.2% (F) and 75% (A)
- ΔE: average 20%, median 15.9%; ΔE > 10% for A,B,E,G,I,J,K.

=> Determine actual effort parameters for these systems

Summary & Future Work

- Relative analytical cost model to quantify impact of cloning on maintenance efforts
- Basis to evaluate alternative clone management strategies
- Case study that instantiates model for 11 industrial projects

- Validation of assumptions
- Instantiation of cost model with actual (non-literature) parameters
- Validation of results of cost model
- Extension of cost model to quantify impact on remaining faults
- Sensitivity analysis

Thank You!