

1	Einführung.....	3
1.1	Umfang und Lebenszeit von Software.....	4
1.2	Relevanz der Thematik.....	5
1.3	Image.....	6
1.4	Was ist Software-Wartung?.....	7
1.5	Warum ist Wartung schwierig (teuer)?.....	10
1.6	Szenarien.....	14
2	Organisation von Wartungseinheiten (WE).....	14
2.1	Einbettung der WE in das Unternehmen.....	16
2.2	Wer führt W durch?.....	17
2.3	Organisation von Wartungsprojekten.....	19
2.4	Richtlinien (RL).....	20
2.5	Wissensmanagement.....	21
3	Prozesse.....	22
3.1	Entwicklungsprozesse.....	22
3.2	Wartungsprozesse.....	23
3.3	Änderungsmanagement.....	25
3.4	Konfigurationsmanagement.....	30
3.5	System-Bewertung.....	40
4	Techniken.....	43
4.1	Software-Metriken.....	43
4.2	Redundananalyse (engl.: Clone Detection).....	55
4.3	Software-Visualisierung.....	59
4.4	Software Dokumentation.....	60
4.5	Software Generieren.....	67
4.6	Programm-Transformation.....	67
4.7	Impact Analyse (Slicing).....	68
4.8	Flexibilisierung.....	68
4.9	Release-Planning.....	69

## Kap 1.2. Relevanz

```
/* ugly.c */

#include <stdio.h>
main(t,_,a)
char *a;
{
return!0<t?t<3?main(-79,-13,a+main(-87,1-_,main(-
86,0,a+1)+a)):
1,t<_?main(t+1,_,a):3,main(-94,-27+t,a)&&t==2?_<13?
main(2,_,+1,"%s %d %d\n"):9:16:t<0?t<-72?main(_,t,
"@n'+,#'/*{ }w+/w#cdnr/+,{ }r/*de}+,/*{ *+,/w{ %+,/w#q#
n+,#{ l+,/n{n+,/+#n+,/#\
;#q#n+,/+k#;*+,/'r : 'd*'3,}{w+K w'K:'+'e#' ;dq#'l \
q#' +d'K#!/+k#;q#'r}eKK#}w'r}eKK{nl]'/#;#q#n' )}{)#}w'
)}{nl]'/+#n' ;d}rw' i;# \
){nl]!/n{n#' ; r{#w'r nc{nl]'/#{l,+'K {rw'
iK{;[{nl]'/w#q#n'wk nw' \
iwk{KK{nl]!/w{%'l##w#' i;
:{nl]'/*{q#'ld;r'}{nlwb!/*de}'c \
;;{nl}'-{}rw]'/+,}##' * }#nc, ',#nw]'/+kd'+e}+;#'rdq#w!
nr'/ ' ) }+}{rl#'{n' ' )#\
}'+'}##(!!/" )
:t<-50?_==*a?putchar(31[a]):main(-
65,_,a+1):main((*a=='/')+t,_,a+1)
:0<t?main(2,2,"%s"): *a=='/' || main(0,main(-61,*a,
"!ek;dc i@bK'(q)-[w]*%n+r3#l,{ }:\nuwloca-0im
.vpbks,fxntdCeghiry"),a+1);
}
```

On the first day of Christmas my true love gave to me  
a partridge in a pear tree.

On the second day of Christmas my true love gave to me  
two turtle doves  
and a partridge in a pear tree.

On the third day of Christmas my true love gave to me  
three french hens, two turtle doves  
and a partridge in a pear tree.

On the fourth day of Christmas my true love gave to me  
four calling birds, three french hens, two turtle doves  
and a partridge in a pear tree.

On the fifth day of Christmas my true love gave to me  
five gold rings;  
four calling birds, three french hens, two turtle doves  
and a partridge in a pear tree.

On the sixth day of Christmas my true love gave to me  
six geese a-laying, five gold rings;  
four calling birds, three french hens, two turtle doves  
and a partridge in a pear tree.

On the seventh day of Christmas my true love gave to me  
seven swans a-swimming,  
six geese a-laying, five gold rings;  
four calling birds, three french hens, two turtle doves  
and a partridge in a pear tree.

On the eighth day of Christmas my true love gave to me  
eight maids a-milking, seven swans a-swimming,  
six geese a-laying, five gold rings;  
four calling birds, three french hens, two turtle doves  
and a partridge in a pear tree.

On the ninth day of Christmas my true love gave to me  
nine ladies dancing, eight maids a-milking, seven swans a-swimming,  
six geese a-laying, five gold rings;  
four calling birds, three french hens, two turtle doves  
and a partridge in a pear tree.

On the tenth day of Christmas my true love gave to me  
ten lords a-leaping,  
nine ladies dancing, eight maids a-milking, seven swans a-swimming,  
six geese a-laying, five gold rings;  
four calling birds, three french hens, two turtle doves  
and a partridge in a pear tree.

On the eleventh day of Christmas my true love gave to me  
eleven pipers piping, ten lords a-leaping,  
nine ladies dancing, eight maids a-milking, seven swans a-swimming,  
six geese a-laying, five gold rings;  
four calling birds, three french hens, two turtle doves  
and a partridge in a pear tree.

On the twelfth day of Christmas my true love gave to me  
twelve drummers drumming, eleven pipers piping, ten lords a-leaping,  
nine ladies dancing, eight maids a-milking, seven swans a-swimming,  
six geese a-laying, five gold rings;  
four calling birds, three french hens, two turtle doves  
and a partridge in a pear tree.

```

/* primenum.c */

#define BeginProgram void main(int argc, char *argv[])
#define CloseBrace }
#define CommandLineArgument -1
#define Declare int i,j,n,Flag=1;
#define EndOfProgram return;
#define False 0;
#define ForLoop ;for
#define GetCommandLineArgument n=atoi(argv[1]);
#define i Flag
#define If if
#define Increment ++
#define Is ==
#define LessThan *(c&64)*
#define LessThanOrEqualTo !=
#define Modulo %
#define OpenBrace {
#define PossibleFactor j
#define PossiblePrime i
#define PossiblePrime (c=getchar())
#define PrimeNumber (c^(!i*n%64));
#define Print putchar
#define SetTo =
#define SmallestPrime 2
#define True 1
#define Variables char c;
#define Zero i%j

BeginProgram
OpenBrace
    Declare Variables
    GetCommandLineArgument

    ForLoop (PossiblePrime SetTo SmallestPrime ;
            PossiblePrime LessThanOrEqualTo CommandLineArgument;
            Increment PossiblePrime)
    OpenBrace
        Flag SetTo True
        ForLoop (PossibleFactor SetTo SmallestPrime ;
                PossibleFactor LessThan PossiblePrime ;
                Increment PossibleFactor)
            If (PossiblePrime Modulo PossibleFactor
                Is Zero)
                Flag SetTo False

        If (Flag Is True)
            Print PrimeNumber
    CloseBrace

    EndOfProgram
CloseBrace

```

```
> echo "Hallo" |./primenum.exe 10  
Bkffe
```

```
> echo "Hallo" |./primenum.exe 10|./primenum.exe 10  
Hallo
```

```
> echo "THIS IS ALL IN CAPS" |./primenum.exe 32  
this is all in caps
```

```

/* gcc expand_expr */

rtx
expand_expr_real (tree exp, rtx target, enum machine_mode tmode,
                 enum expand_modifier modifier, rtx *alt_rtx)
{
  rtx op0, op1, temp;
  tree type = TREE_TYPE (exp);
  int unsignedp = TREE_UNSIGNED (type);
  enum machine_mode mode;
  enum tree_code code = TREE_CODE (exp);
  optab this_optab;
  rtx subtarget, original_target;
  int ignore;
  tree context;

  /* Handle ERROR_MARK before anybody tries to access its type. */
  if (TREE_CODE (exp) == ERROR_MARK || TREE_CODE (type) ==
      ERROR_MARK)
    {
      op0 = CONST0_RTX (tmode);
      if (op0 != 0)
        return op0;
      return const0_rtx;
    }

  mode = TYPE_MODE (type);
  /* Use subtarget as the target for operand 0 of a binary
  operation. */
  subtarget = get_subtarget (target);
  original_target = target;
  ignore = (target == const0_rtx
            || ((code == NON_LVALUE_EXPR || code == NOP_EXPR
                || code == CONVERT_EXPR || code == REFERENCE_EXPR
                || code == COND_EXPR || code == VIEW_CONVERT_EXPR)
                && TREE_CODE (type) == VOID_TYPE));

  /* If we are going to ignore this result, we need only do something
  if there is a side-effect somewhere in the expression.  If there
  is, short-circuit the most common cases here.  Note that we must
  not call expand_expr with anything but const0_rtx in case this
  is an initial expansion of a size that contains a PLACEHOLDER_EXPR. */

  if (ignore)
    {
      if (! TREE_SIDE_EFFECTS (exp))
        return const0_rtx;

      /* Ensure we reference a volatile object even if value is ignored, but
      don't do this if all we are doing is taking its address. */
      if (TREE_THIS_VOLATILE (exp)
          && TREE_CODE (exp) != FUNCTION_DECL
          && mode != VOIDmode && mode != BLKmode
          && modifier != EXPAND_CONST_ADDRESS)
        {
          temp = expand_expr (exp, NULL_RTX, VOIDmode, modifier);
          if (GET_CODE (temp) == MEM)
            temp = copy_to_reg (temp);
          return const0_rtx;
        }
    }
}

```

<p>2800 loc ~ 50 Seiten!</p>
----------------------------------

The fellow who designed it,  
Is working far away;  
The spec's not been updated,  
For many a livelong day.

The guy who implemented it is  
Promoted up the line;  
And some of the  
enhancements  
Didn't match to the design.

They haven't kept the flowcharts,  
The manual's a mess,  
And most of what you need to  
know  
You'll simply have to guess.

We do not know the reason,  
Why the bugs pour in like rain,  
But don't just stand here gaping,  
Get out there and **MAINTAIN**.

# Kognitive Dissonanz: Beispiel

- Sachverhalt: Die Software funktioniert nicht so, wie sie sollte
- Wahrnehmung Programmierer: „Ich bin mit meiner Arbeit zufrieden. Ich finde mich gut.“ (ignoriert)
- Wahrnehmung QA: „Das funktioniert nicht.“
- Feedback QA: „In ihrem Programm ist ein Bug“
- -> kognitive Dissonanz: „Ich bin gut – Du hast Mist gebaut“
- Auflösung:
  - „Der Bug liegt sicher nicht bei mir! Fragen sie mal Kollegen X.“
  - „Es steht aber genau so in den Anforderungen! Lesen sie die mal!“
  - „Das ist kein Bug, die User haben es mir so erklärt.“
  - „Sie wissen ja nicht einmal, was ein Bug ist!“
- Kognitive Dissonanz führt also zu einer **Wirklichkeitskonstruktion**, die versucht, die Dissonanz aufzulösen

© J. Weidl-Rektenwald 02-03

## Beispiel für den psycho-sozialen Aspekt

- aus Petr Kroha „Softwaretechnologie“ [Kroha97]
- Windows of Opportunity [Yourdon92] - beschreiben günstigsten Zeitpunkt zur Neuentwicklung eines Softwaremoduls:
  - „der zuständige Programmierer geht in Rente oder kündigt
  - wenn die Komponente extrem gravierende Fehler aufweist und der Programmierer die Suche aufgegeben hat
  - *wenn es endlich möglich ist, den Widerstand leistenden Programmierer zu entlassen.*“



Spezifikation  
Verwaltung

Verantw. : Markus Pizka  
Version: 1.0  
Status: zum Review  
Stand: 18.08.2003

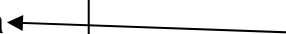
Ablage: xxx

TU-München  
Boltzmannstr. 3  
85748 Garching

Titel



genau ein Verantwortlicher



Versionsdaten

(z.B. mit den Zuständen  
- in Arbeit  
- zum Review  
- abgenommen)

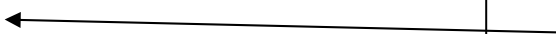


Zusammenfassung

Historie

Version	Status	Datum	Autor(en)	Erläuterung

Was soll das Dokument?  
Was steht ungefähr drin?



Wer hat das Dokument  
Wann und aus welchem Grund  
geändert?



# GNU Coding Standards

## [1. About the GNU Coding Standards](#)

## [2. Keeping Free Software Free](#)

### [2.1 Referring to Proprietary Programs](#)

### [2.2 Accepting Contributions](#)

### [2.3 Trademarks](#)

## [3. General Program Design](#)

### [3.1 Which Languages to Use](#)

### [3.2 Compatibility with Other Implementations](#)

### [3.3 Using Non-standard Features](#)

### [3.4 Standard C and Pre-Standard C](#)

### [3.5 Conditional Compilation](#)

## [4. Program Behavior for All Programs](#)

### [4.1 Writing Robust Programs](#)

### [4.2 Library Behavior](#)

### [4.3 Formatting Error Messages](#)

### [4.4 Standards for Interfaces Generally](#)

### [4.5 Standards for Graphical Interfaces](#)

### [4.6 Standards for Command Line Interfaces](#)

### [4.7 Table of Long Options](#)

### [4.8 Memory Usage](#)

### [4.9 File Usage](#)

## [5. Making The Best Use of C](#)

### [5.1 Formatting Your Source Code](#)

### [5.2 Commenting Your Work](#)

### [5.3 Clean Use of C Constructs](#)

### [5.4 Naming Variables, Functions, and Files](#)

### [5.5 Portability between System Types](#)

### [5.6 Portability between CPUs](#)

### [5.7 Calling System Functions](#)

### [5.8 Internationalization](#)

### [5.9 Mmap](#)

## [6. Documenting Programs](#)

### [6.1 GNU Manuals](#)

### [6.2 Doc Strings and Manuals](#)

### [6.3 Manual Structure Details](#)

### [6.4 License for Manuals](#)

### [6.5 Manual Credits](#)

### [6.6 Printed Manuals](#)

### [6.7 The NEWS File](#)

### [6.8 Change Logs](#)

#### [6.8.1 Change Log Concepts](#)

#### [6.8.2 Style of Change Logs](#)

#### [6.8.3 Simple Changes](#)

#### [6.8.4 Conditional Changes](#)

#### [6.8.5 Indicating the Part Changed](#)

### [6.9 Man Pages](#)

### [6.10 Reading other Manuals](#)

## [7. The Release Process](#)

### [7.1 How Configuration Should Work](#)

### [7.2 Makefile Conventions](#)

#### [7.2.1 General Conventions for Makefiles](#)

#### [7.2.2 Utilities in Makefiles](#)

#### [7.2.3 Variables for Specifying Commands](#)

#### [7.2.4 Variables for Installation Directories](#)

#### [7.2.5 Standard Targets for Users](#)

#### [7.2.6 Install Command Categories](#)

### [7.3 Making Releases](#)

## **4.1 Writing Robust Programs**

Avoid arbitrary limits on the length or number of *any* data structure, including file names, lines, files, and symbols, by allocating all data structures dynamically. In most Unix utilities, "long lines are silently truncated". This is not acceptable in a GNU utility. ...

...

## Kap 3.4. Change-Management

The screenshot shows the Rational ClearQuest application window titled "Rational ClearQuest - [RAMBU (Submitted by JJV (Defect))]". The main window displays a list of defects in a table. The selected defect is Rambu00021604 with the headline "multiil recoverpacket sets epoch estimate too low". Below the table, there is a detailed view of the defect record. The record includes fields for ID, State, Headline, Product, Project, Cust. Priority, Def. Severity, Cust. Impact, Bus. Priority, Owner, and Description. The description reads: "This is a spinoff / continuation of RAMBU21562. See that defect for history." The interface also shows navigation tabs for Platform, Engineering, and Planning, and a sidebar with workspace options like Personal Queries and Public Queries.

id	Headline	State	Pr
RAMBU00021604	multiil recoverpacket sets epoch estimate too low	Requires_Development	CQ/Mul
RAMBU00021603	syncreplica-import acknowledgement has extra slash on UNDI	Requires_Development	CQ/UNI
RAMBU00021602	"Cannot replay..." message does not show which database it's talki	Assigned_to_Developer	CQ/Mul
RAMBU00021601	"Cannot replay..." message shows time in GMT, not local time	Requires_Development	CQ/Mul
RAMBU00021595	MulSite on UNDI doesn't work with mixed-case family (database) na	Requires_Development	CQ/UNI

Result set | Query editor | Display editor | SQL editor /

Platform: Engineering | Planning

Escalation Records | Unified Change Management | Reminders

Main | Doc Changes | Notes | Resolution | Attachments | History | Origin

ID: Rambu00021604 | State: Requires\_Development

Headline: multiil recoverpacket sets epoch estimate too low

Product: ClearQuest | Mastership Belongs To: RAMBUREP1

Project: CQ/MulSite

Cust. Priority: | RAMBUCUP

Def. Severity: DS2 | RAMBULEX

Cust. Impact: Medium | RAMBUREP1

Bus. Priority: BP2 | RAMBUNIX

Owner: | Symptoms:

Description: This is a spinoff / continuation of RAMBU21562. See that defect for history.

The screenshot shows the Rational Project Console application window titled "Rational Project Console - Microsoft Internet Explorer". The main window displays a list of defects in a table. The table has columns for ID, Headline, State, and Priority. The defects listed include various issues such as "spelling error in login screen", "sales tax incorrect if item deleted from purchase", "cancel sale doesn't correctly reprint screen", "columns out of alignment", "delete item not working correctly", "override price does not work", "alt-C does not invoke cancel operation", "disk allowed to change to slash on credit card", "end-of-shift report fails if after midnight", "too many spaces in 'change due' field", "delete operation leaves blank line in form", "add item button not active after adding 3 other items", "sales tax incorrect for NH", "credit card refused message is unclear", "inventory report is not running correctly", "delete item button deletes two items", "overriding price operations allows negative number", "browsing of application looks too crowded", "part number column not wide enough", "add item button is out of line with the other buttons", "control sensitive help fails from reorder window", "formatting does not look right in inventory report", "add items fails for large quantities", "spelling error in cancel sale help", and "shortcut to logout does not work".

Query Results: Public Queries/All Defects

ID	Headline	State	Priority
CLSDC0000037	spelling error in login screen	Resolved	3-Normal Queue
CLSDC0000038	sales tax incorrect if item deleted from purchase	Resolved	2-Care High Attention
CLSDC0000039	cancel sale doesn't correctly reprint screen	Resolved	3-Normal Queue
CLSDC0000040	columns out of alignment	Resolved	2-Care High Attention
CLSDC0000041	delete item not working correctly	Opened	2-Care High Attention
CLSDC0000042	override price does not work	Resolved	2-Care High Attention
CLSDC0000043	alt-C does not invoke cancel operation	Resolved	3-Normal Queue
CLSDC0000044	disk allowed to change to slash on credit card	Resolved	3-Normal Queue
CLSDC0000045	end-of-shift report fails if after midnight	Resolved	3-Normal Queue
CLSDC0000046	too many spaces in 'change due' field	Assigned	3-Normal Queue
CLSDC0000047	delete operation leaves blank line in form	Resolved	3-Normal Queue
CLSDC0000048	add item button not active after adding 3 other items	Resolved	1-Resolve Immediately
CLSDC0000049	sales tax incorrect for NH	Assigned	1-Resolve Immediately
CLSDC0000050	credit card refused message is unclear	Resolved	3-Normal Queue
CLSDC0000051	inventory report is not running correctly	Assigned	1-Resolve Immediately
CLSDC0000052	delete item button deletes two items	Resolved	2-Care High Attention
CLSDC0000053	overriding price operations allows negative number	Assigned	2-Care High Attention
CLSDC0000054	browsing of application looks too crowded	Resolved	2-Care High Attention
CLSDC0000055	part number column not wide enough	Opened	3-Normal Queue
CLSDC0000056	add item button is out of line with the other buttons	Assigned	3-Normal Queue
CLSDC0000057	control sensitive help fails from reorder window	Assigned	2-Care High Attention
CLSDC0000058	formatting does not look right in inventory report	Opened	3-Normal Queue
CLSDC0000059	add items fails for large quantities	Opened	2-Care High Attention
CLSDC0000060	spelling error in cancel sale help	Resolved	2-Care High Attention
CLSDC0000061	shortcut to logout does not work	Assigned	3-Normal Queue

Microsoft Project Connect Dashboard - Microsoft Internet Explorer

Multi-Chart Display

**Dashboard**

- Private Metric Folders
- Public Metric Folders
- 1 Trend of Level 0 and Level 4 Risks
- 10 Trend of Total Defects
- 11 Trend of Open Defects
- 12 Trend of Excessively Failed Defs
- 2 Trend of Level 0 and Level 4 Risks
- 3 Trend of Level 0 and Level 4 Risks
- 4 Dist Analysis of Level 0 and Level 4
- 5 Dist Analysis of Level 0 and Level 4
- 6 Trend of Affects Architecture F
- 8 Distribution of Requirements I
- 9 Distribution of Level 0 Risks
- 7 Trend of Open Defects During C
- 6 Trend of Lines Added Modified
- 8 Trend of Non-Verified Requirer
- Distribution of Defects by Comp
- Distribution of Open Defects by C
- Distribution of Open Defects by S
- Distribution of Open Requirements
- Distribution of Requirements by I
- Distribution of Requirements by S
- Distribution of Total Defects by S
- MSP - Trend Milestone Tasks by I
- MSP - Trend of # of Late Tasks
- MSP - Trend of # of Tasks & Com
- MSP - Trend of 100% Complete Id
- MSP - Trend of Actual Duration of
- MSP - Trend of Actual vs. Budget
- MSP - Trend of Actual vs. Plannin
- MSP - Trend of ACWP vs. BCWP
- MSP - Trend of Baseline Current
- MSP - Trend of BCWP vs. SCPPS
- MSP - Trend of Earned Value vs. I
- MSP - Trend of Late Milestones
- Multi-Chart Display
- req of abc
- Trend of Requirements with Type
- Trend of Total Defects with State

**Open Requirements**

10.0

**Distribution of Open Requirements by Priority**

Priority	Count
High	2
Low	1
Medium	15

**Open Defects**

50

**Trend of Lines Added, Modified, and Deleted**

Count

Operation Date

Legend: Invested (blue), Unachieved (red), Line-Changed (yellow)

**Table of Defects by Owner with State**

	Submitted	Resolved	Opened	Assigned	Closed
U.S.L.	50				
Admin					20
Admin					20
Admin					34.0
Admin					30
Admin					3.0
Admin					3.0
Admin		10			4.0
Admin					3.0
Admin					3.0
Admin					30
Admin					3.0

Agent started

Local transit

# Kap 3.5.2. SRAH

## SOFTWARE REENGINEERING ASSESSMENT PROCESS OVERVIEW

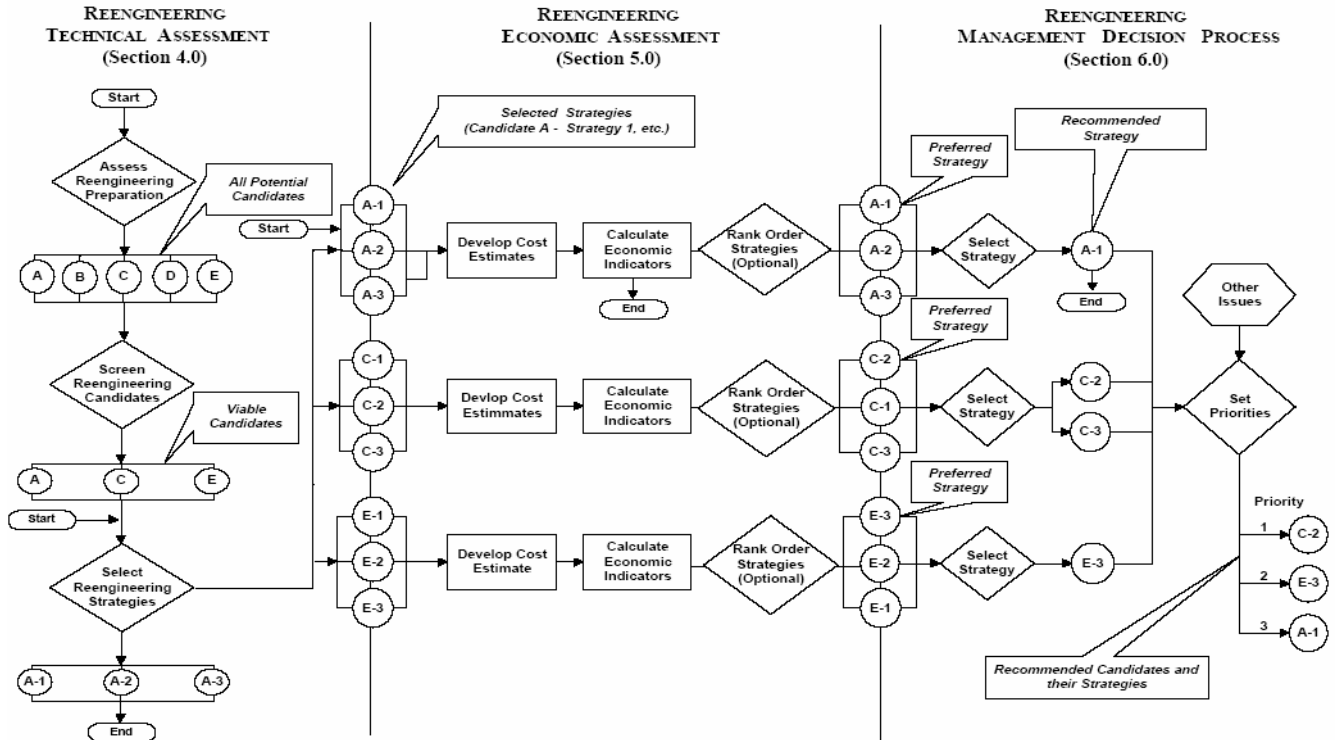
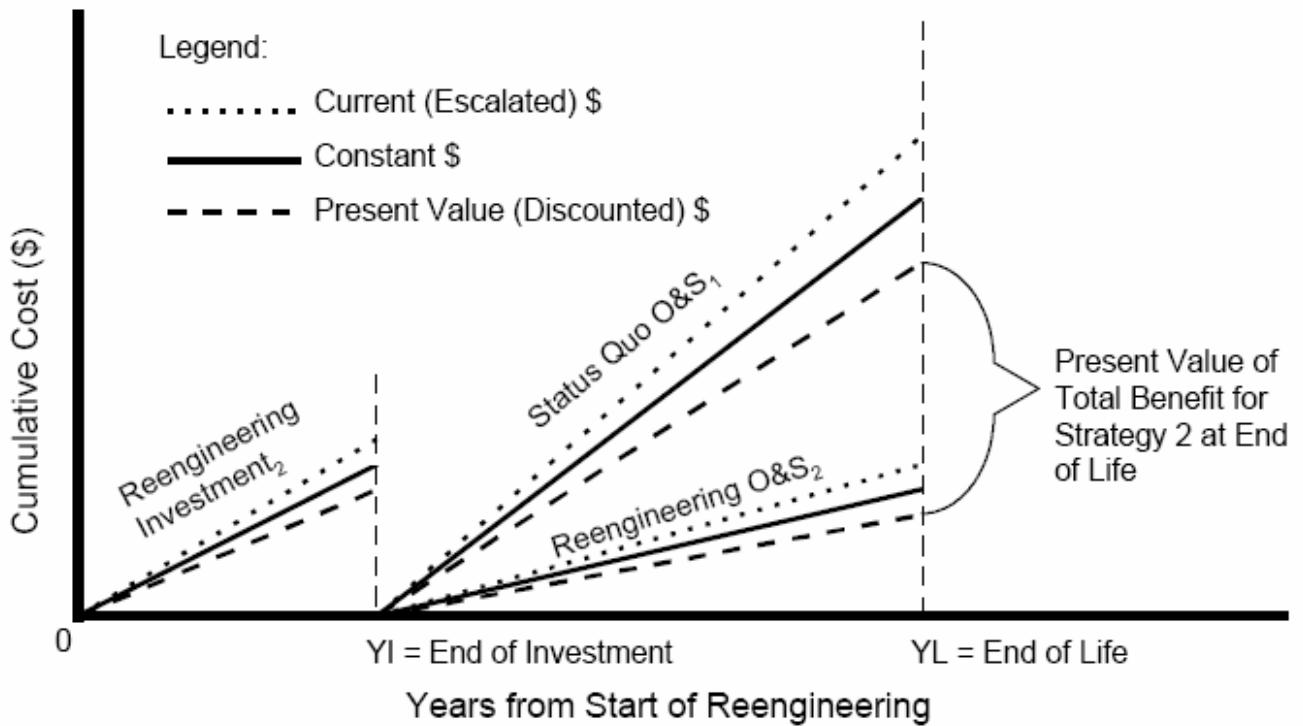


Figure 1-1. Software Reengineering Assessment (SRA) Process Overview

**Reengineering Technical Assessment  
Table 4-2 RTA Summary Worksheet**

Candidate Software Name & Description: _____ _____	Number of Questions Answered	Summation of Answers	Average: (Summation Divided by Number of Questions)	Indication of Strategy
Remaining Life (>3 yrs?)				yes no
Age (> 5 yrs?)				yes no
Preparation <sup>1</sup>				yes no
Importance <sup>1</sup>				yes no
Maintenance Environment <sup>1</sup>				yes no
Redocument <sup>2</sup>				yes maybe no
Restructure <sup>2</sup>				yes maybe no
Translate Source Code <sup>2</sup>				yes maybe no
Data Reengineer <sup>2</sup>				yes maybe no
Retarget <sup>2</sup>				yes maybe no
Reverse Engineer				yes no
Redevelop				yes no
Status Quo				yes no



$$NPV = PV(\text{Benefit}) - PV(\text{Investment})$$

$$= PV(O\&S_1) - PV(O\&S_2) - PV(\text{Investment}_2)$$

> 0: Positive Return (Discounted \$)

< 0: Negative Return (Discounted \$)

= 0: Break-even (Current \$)

$$BIR = PV(\text{Benefit}) / PV(\text{Investment})$$

$$= (PV(O\&S_1) - PV(O\&S_2)) / PV(\text{Investment}_2)$$

> 1: Positive Return (Discounted \$)

< 1: Negative Return (Discounted \$)

= 1: Break-even (Current \$)

**Figure 5-3. Net Present Value (NPV), Benefit Investment Ratio (BIR), and Break-even Point (BP)**

**Initial Ground Rules and Assumptions:**

- All estimates and comparisons will be made in discounted constant FY \_\_\_\_ dollars. (Fill in the blank to specify what fiscal year the dollars estimated represent.)
- Use the same cost element structure (CES) for all candidate strategies.
- The economic "clock" (or time = 0) for any candidate strategy begins when the strategy effort starts.
- Sunk costs, occurring prior to the start of the effort, will not be considered.
- A standard labor rate of \$\_\_\_\_\_/man-month (or \$\_\_\_\_\_/man-year) will be used for Government, and a standard labor rate of \$\_\_\_\_\_/man-month (or \$\_\_\_\_\_/man-year) will be used for contractors.
- Strategy 1 is reserved for the "status quo" candidate strategy.
- Operations & Support (O&S) cost for all candidate strategies includes O&S of the existing software during reengineering.
- The \_\_\_\_\_ software estimating model will be used for all estimates of reengineering and support efforts. Overall assessment (maturity, level of risk in using them, and differences between models), experience with, and track record using each model are/are not attached (circle are or are not).
- If code translation is indicated by the technical assessment, automated code translation will/will not be used to reduce the level of manual translation that must be performed (circle will or will not).
- For automated code translation, it will be assumed, that \_\_\_\_% of the code will be successfully translated by the automated tool based on the tool developer's specifications.
- If translation to Ada and restructuring are both indicated, the code will first be translated to Ada and then restructured using the Ada Programming Support Environment.
- The system documentation produced during the effort will be consistent with (full/tailored) (MIL-STD-xxx).
- To the maximum extent possible, automated CASE tools will be used to facilitate the effort. This includes the following tools: .....
- The staff that is currently maintaining the software will/will not be the same staff that performs the effort. If not, the \_\_\_\_\_ will perform the work.
- The effort will/will not include IV&V.



## Kap 4.2. Clone-Detection

### Beispiel 1: identischer Code

```
private int pathDistance(Permutation path) {  
    int result = 0;  
    for (int i = 0; i < numNodes; i++) {  
        int from = path.get(i);  
        int to = path.get((i+1) % numNodes);  
        result += distance[from][to];  
    }  
    return result;  
}  
  
private double pathAvgDistance(Permutation path) {  
    int result = 0;  
    for (int i = 0; i < numNodes; i++) {  
        int from = path.get(i);  
        int to = path.get((i+1) % numNodes);  
        result += distance[from][to];  
    }  
    return (double) result / numNodes;  
}
```

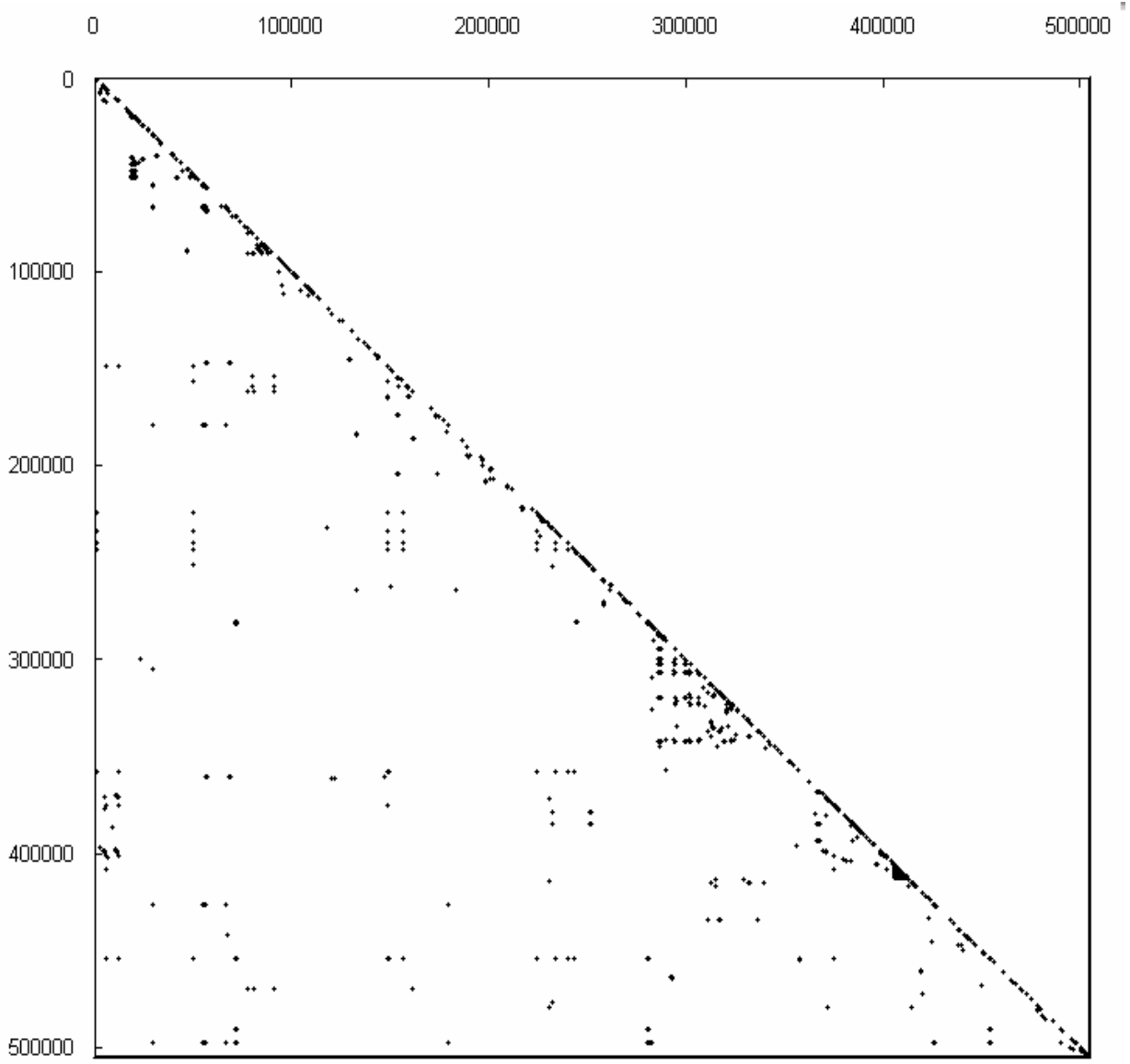


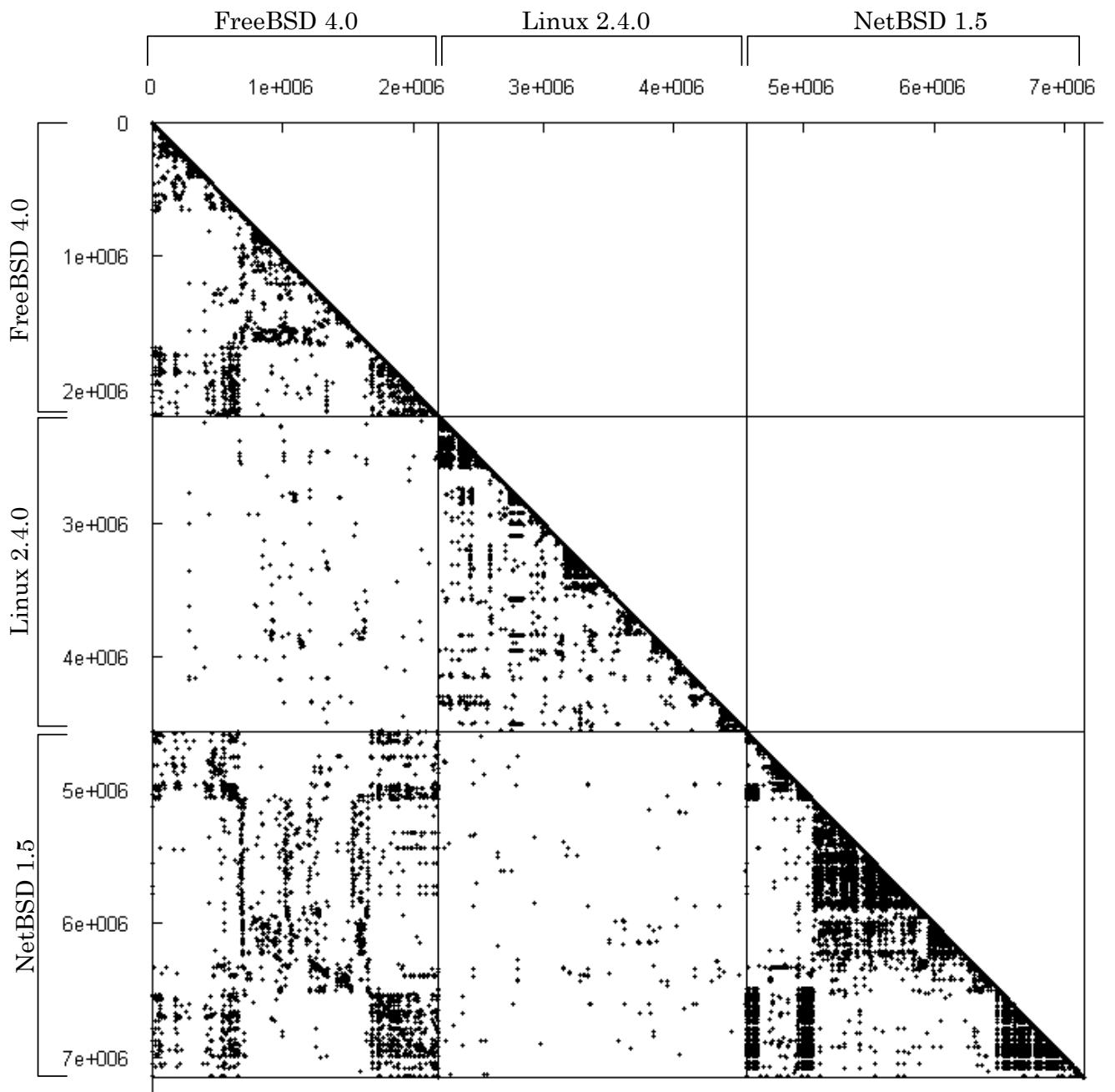
### Beispiel 2: identisch bis auf Identifikatoren

```
int r = 0;  
for (int j = 0; j < numNodes; j++) {  
    int f = path.get(j);  
    int t = path.get((j+1) % numNodes);  
    r += distance[from][to];  
}
```

### Beispiel 3: semantisch äquivalent (?)

```
private int pathDistance(Permutation path, int i, int r) {  
    if (i >= numNodes)  
        return r;  
    else  
        return pathDistance(path, i+1,  
            distance[path.get(i)][ path.get((i+1) % numNodes] + r)  
}
```

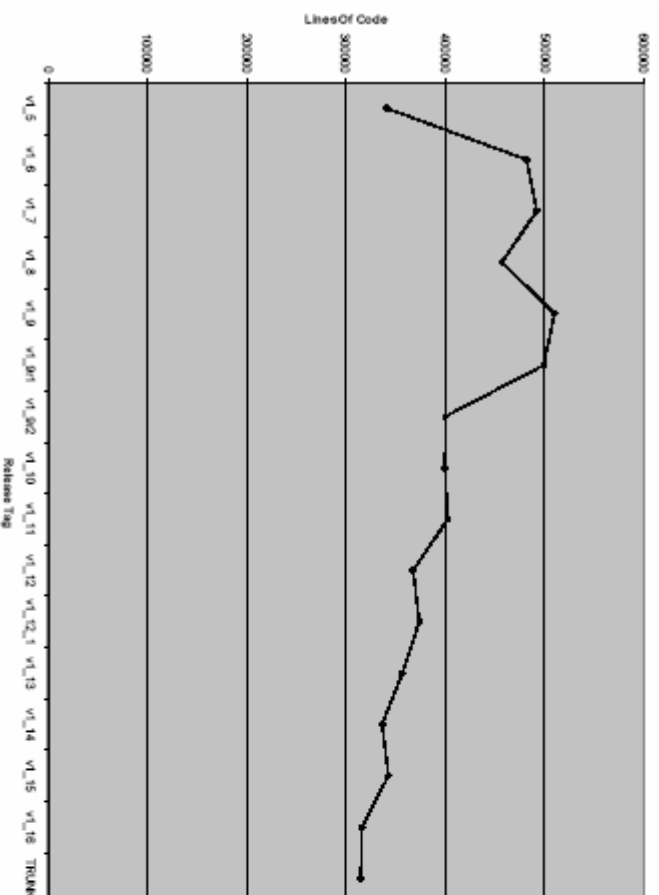




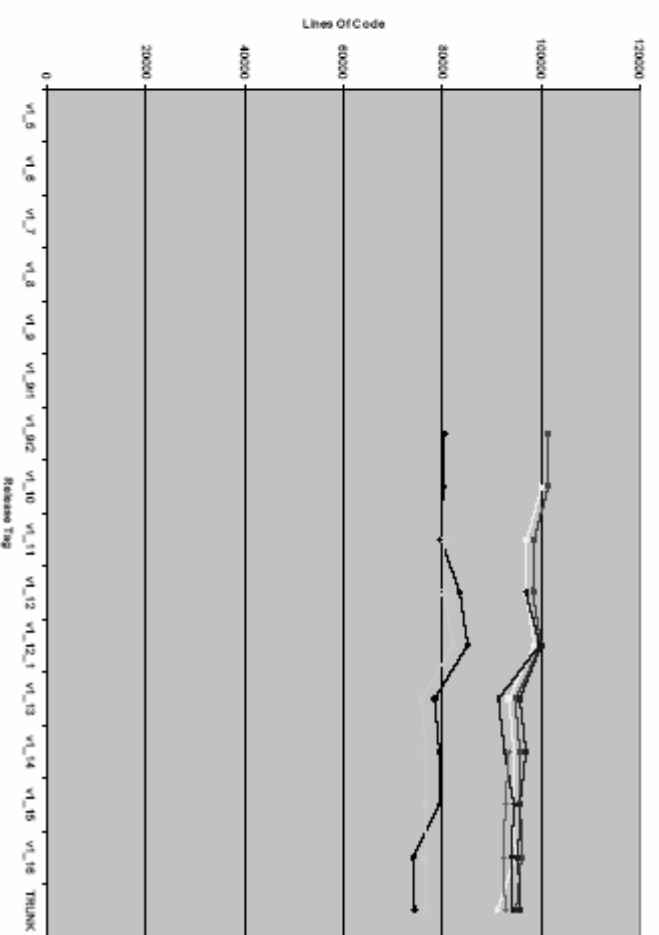
# Case Study: Salion (CRM)

- *Java Application: Core + 6 Customer variantsd*
  - *CloneDR runs every quarter; active clone removal by development team*
  - *Total SLOC reduction over time: ~40%*
- “Most of the drop in the source code graph (core) is due to clone detection removal.”**
- Dale Churchett, Architect**

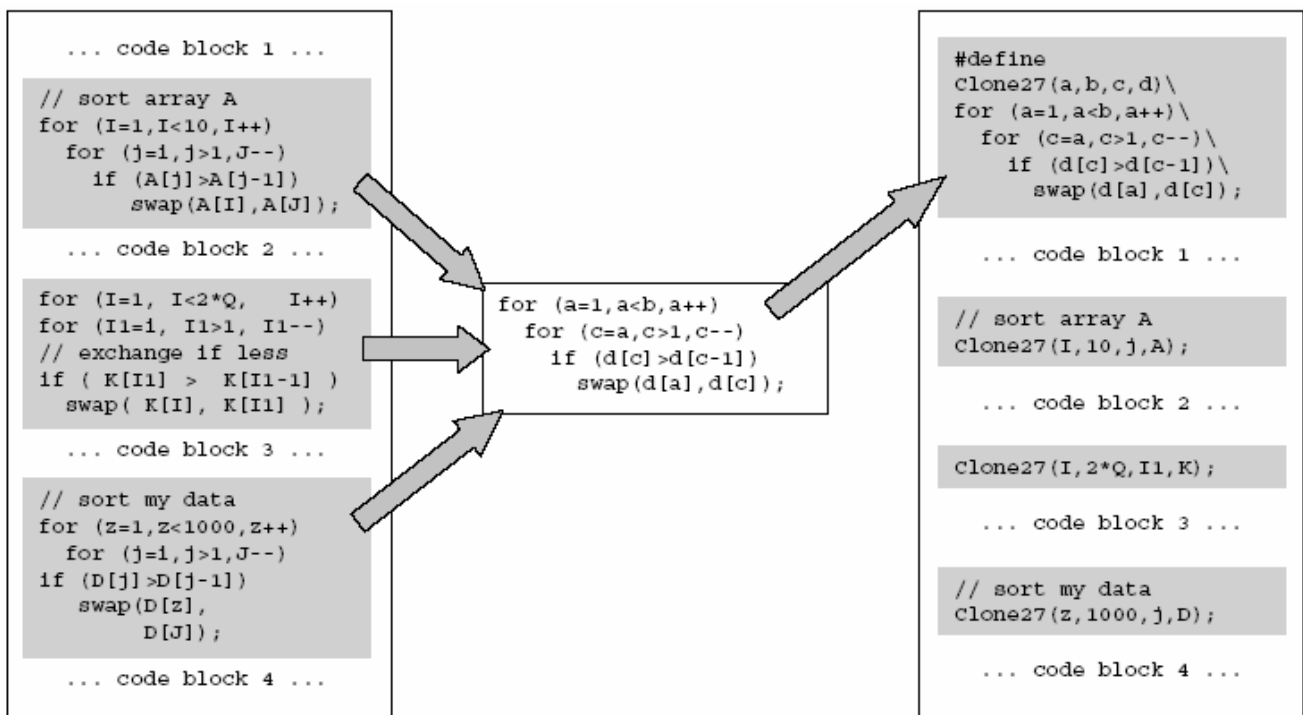
Core Code Trend Analysis



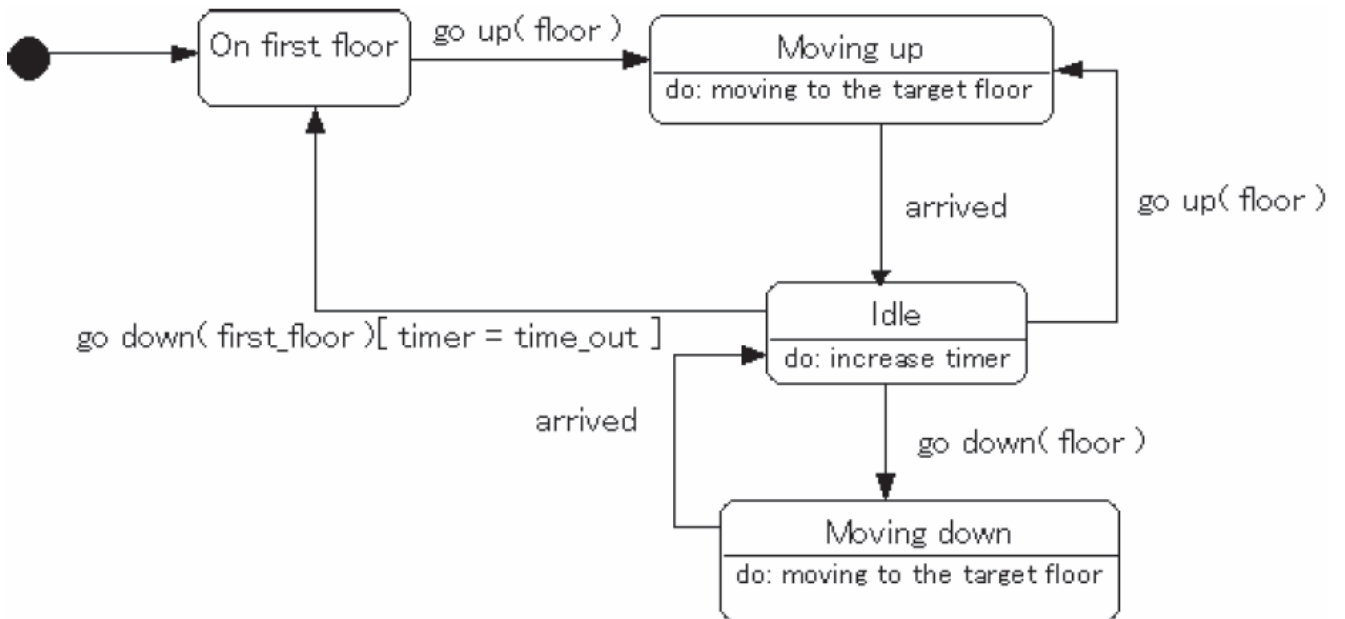
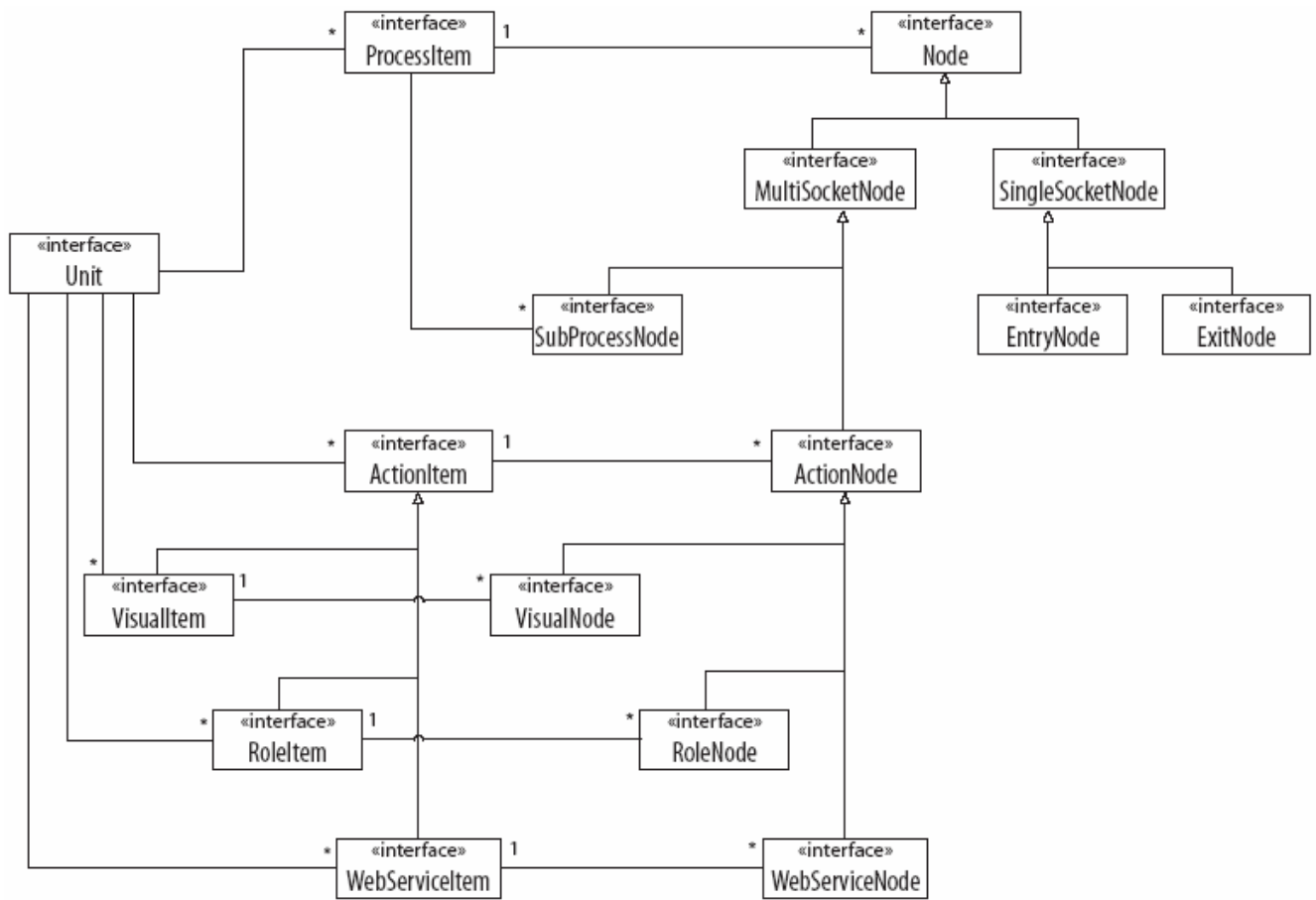
Custom Code Trend

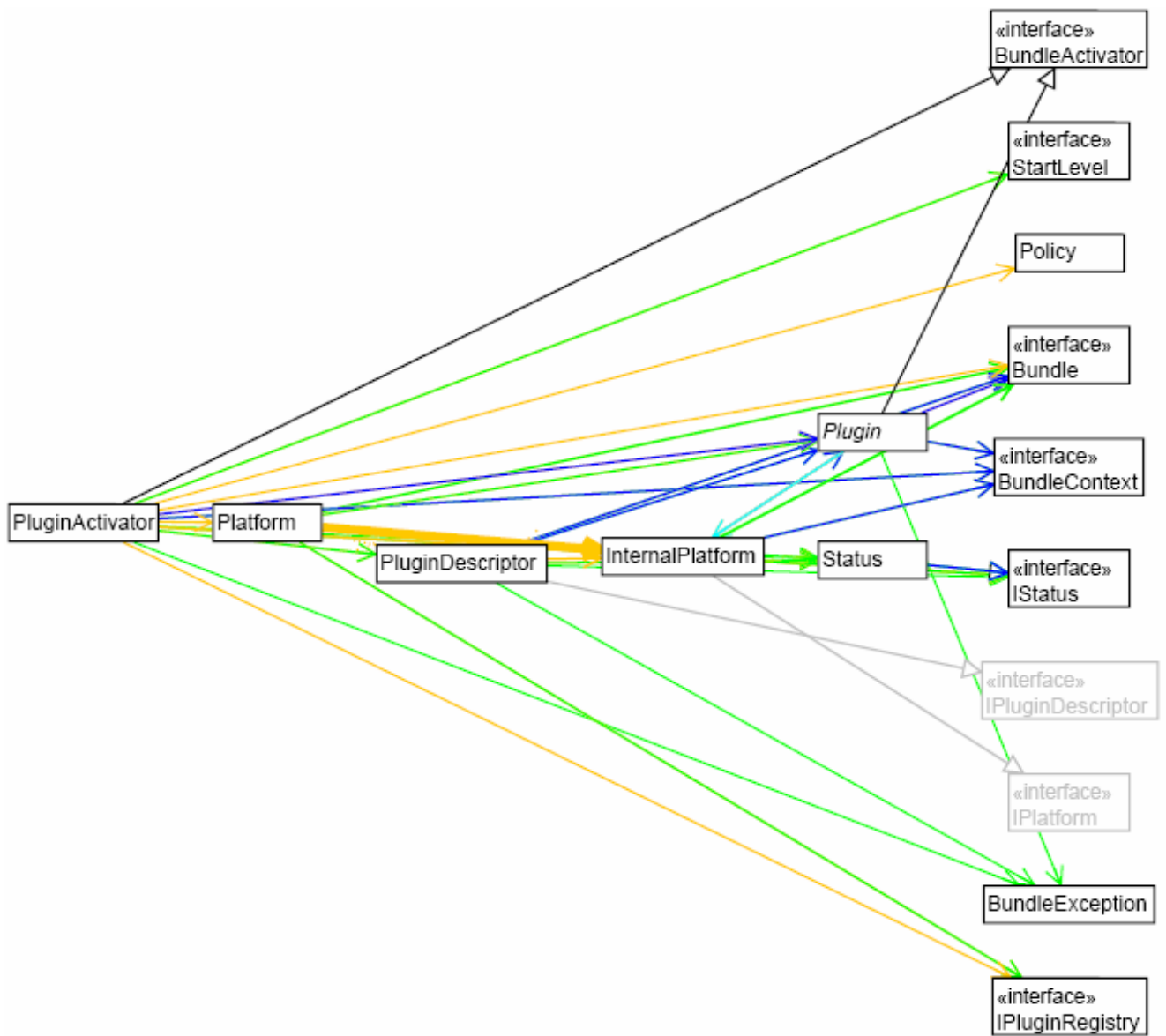


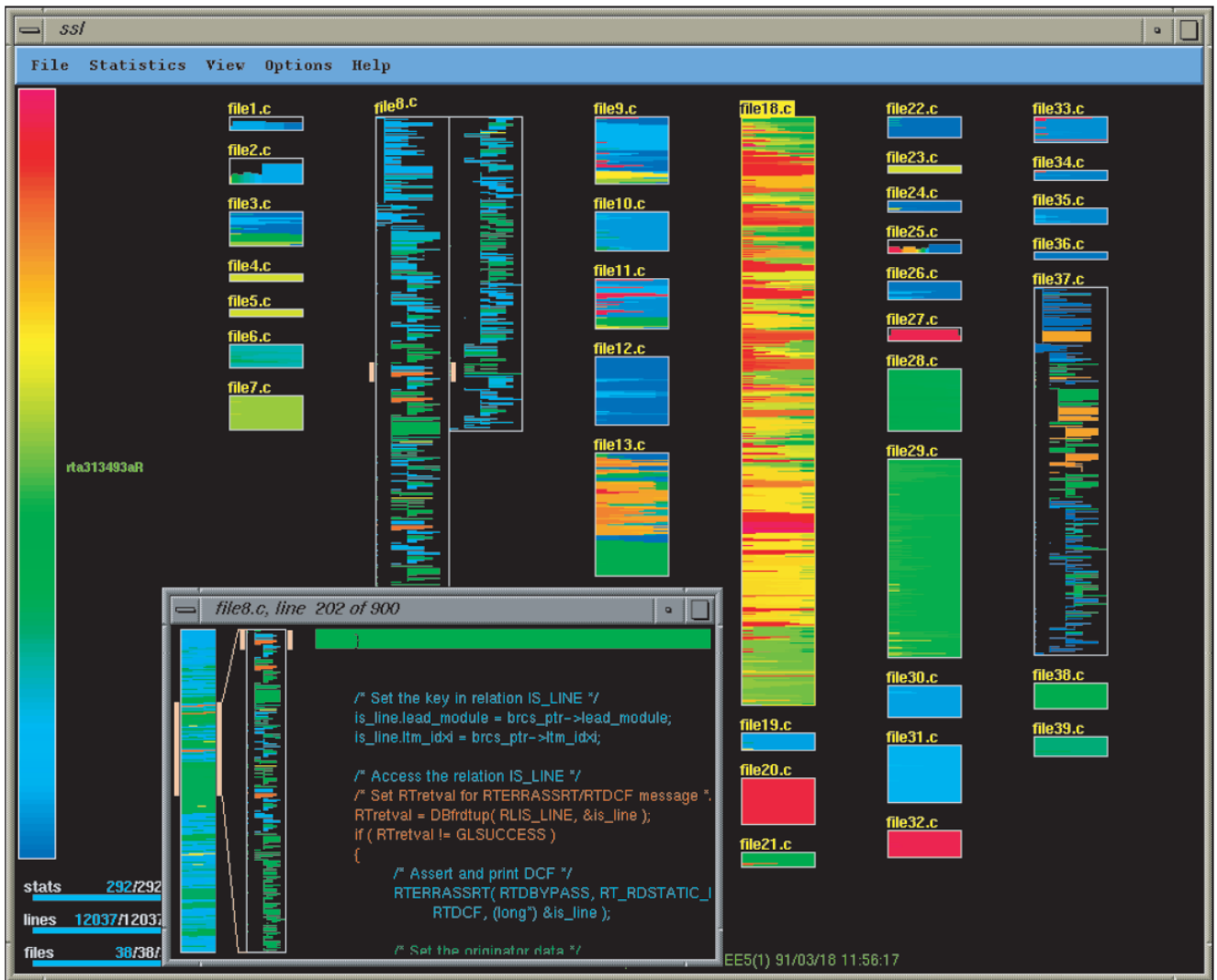
<i>Application</i>	<i>Language</i>	<i>SLOC</i>	<i>Removable SLOC (est.)</i>	<i>Removed SLOC</i>
Process Control	C	427921	54346 12.7%	N.A. (pre removal capability for C 6/98)
PARLANSE Compiler	C	42413	4133 9.7%	2568 6.0%
CCC Registration	COBOL85	77631	15500(39967) 20.0 (51.5)%	15034 19.9%
Roccade Accounting	IBM COBOL	567798	130494 23%	N.A. (not yet performed)
SLC Insurance	PROGRESS	356413	58783 16.5%	N.A. (not yet performed)
SWING Toolkit	Java	241460	26355 10.9%	N.A. (not yet performed)



### Kap 4.3. Visualisierung







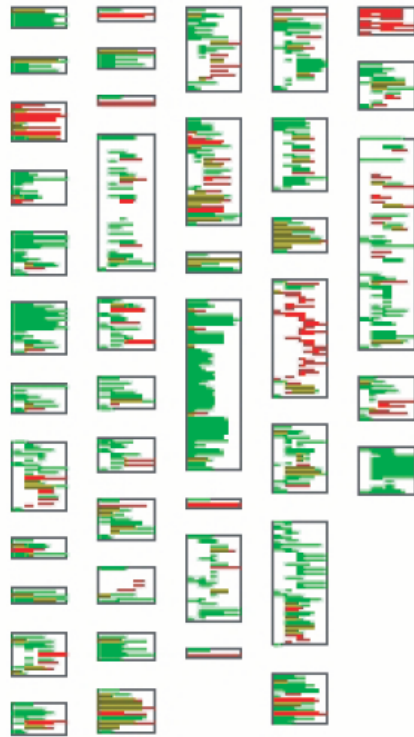




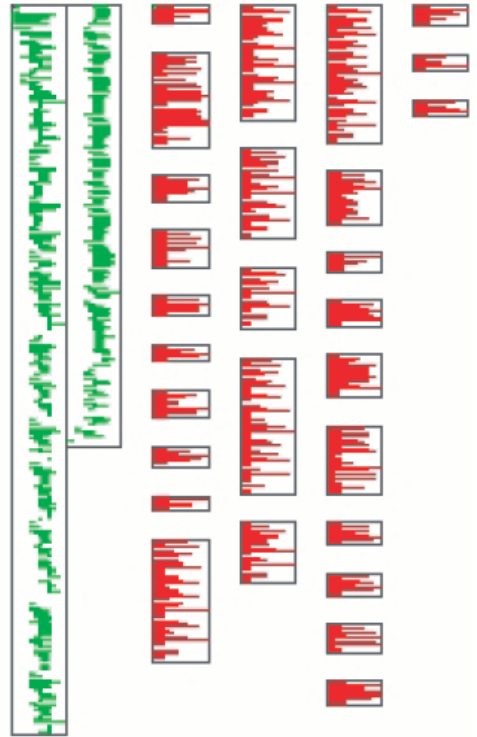
HTML Layout

Service Log

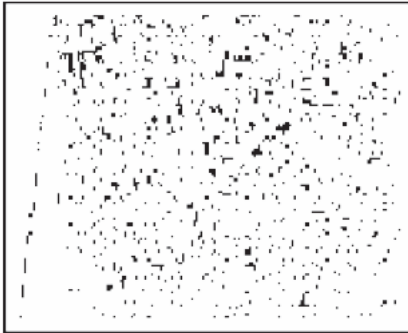
### Tools-Based Implementation



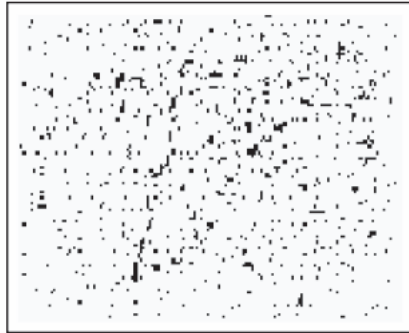
### MAWL Implementation



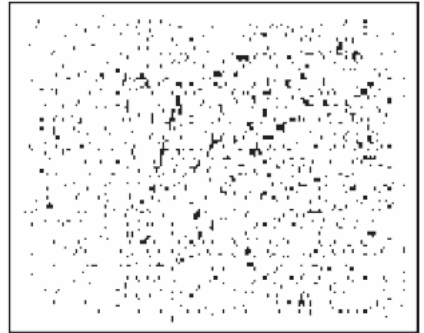
Linear insertion



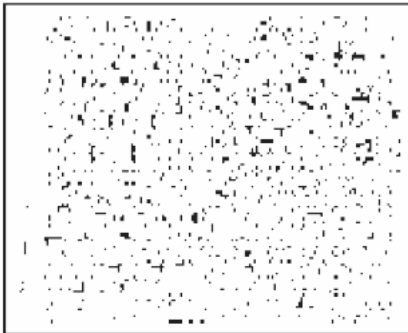
Bubblesort



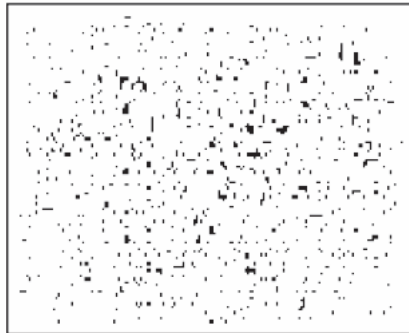
Straight Selection



Binary Insertion



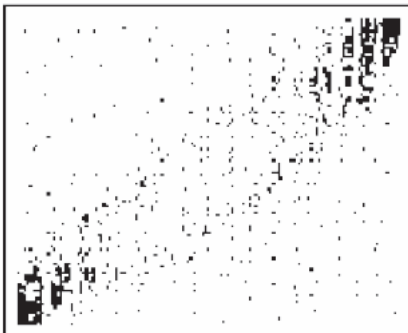
Shakersort



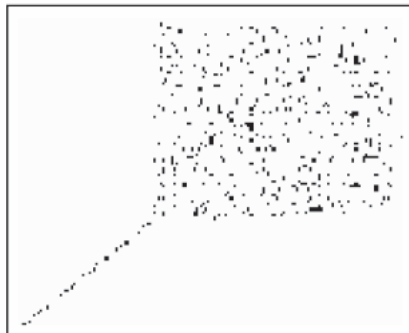
Tree Selection



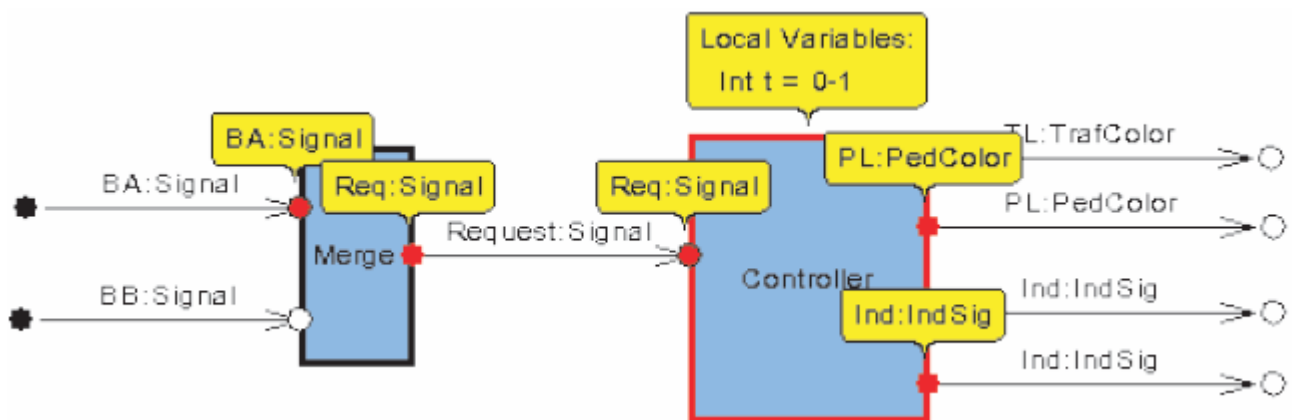
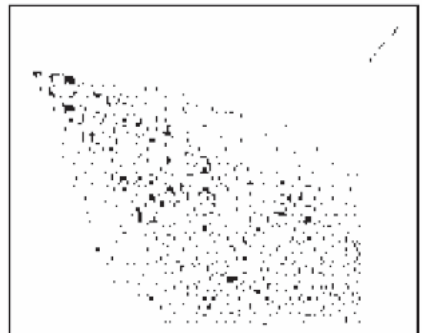
Shellsort



Quicksort



Heapsort



## Kap 4.4. Software-Dokumentation

(Visconti 2000) Umfrage zum Reifegrad von Dokumentationsprozessen;

Skala von „Very Low“ (VL) bis „Very High“ (VH).

	<b>Key Practices</b>	<b>VL</b>	<b>L</b>	<b>M</b>	<b>H</b>	<b>VH</b>
1.	Creation of basic software documents.	0%	8%	25%	17%	50%
2.	Management recognition of importance of documentation	0%	8%	25%	0%	67%
3.	<b>Existence of documentation policy or standard</b>	17%	8%	25%	0%	<b>67%</b>
4.	<b>Monitor implementation of policy or standard</b>	8%	8%	42%	25%	<b>17%</b>
5.	Existence of a defined process for creation of documents	25%	0%	33%	33%	8%
6.	Methods to assure quality of documentation	8%	25%	42%	25%	0%
7.	Assessments of usability of documentation	8%	67%	h	18%	0%
8.	Definition of software documentation quality and usability measures	67%	17%	8%	8%	0%
9.	Collection and analysis of documentation quality measures	42%	42%	17%	0%	0%
10.	Collection and analysis of documentation usability measures	100%	0%	0%	0%	0%
11.	<b>Process improvement feedback loop</b>	42%	33%	17%	<b>0%</b>	<b>8%</b>

I believe that the time is ripe for significantly better documentation of programs, and that we can best achieve this by considering programs to be works of literature.

Hence, my title: "Literate Programming."

– D. Knuth. "Literate Programming"

@ An `Xstring` constructed from a standard string needs space to hold the characters:

@<`Xstring`...@>=

```
Xstring::Xstring(const char* s)
```

```
{  
    p = new srep;  
    @<Allocate space for the string and put a copy of |s| there@>;  
}
```

@ There is always the possibility that a client will try something like ```Xstring x = NULL.`" We substitute the null string whenever we are given a null pointer.

@<Allocate...@>=

```
if (s==NULL) s="";  
p->s = new char[ strlen(s)+1 ];  
strcpy(p->s,s);
```

6. An `Xstring` constructed from a standard string needs space to hold the characters:

<`Xstring` members and friends 6>  $\equiv$

```
Xstring::Xstring(const char *s)
```

```
{  
    p = new srep;  
    <Allocate space for the string and put a copy of s there 7>;  
}
```

See also sections 9, 10, 13, and 17.

This code is used in section 3.

7. There is always the possibility that a client will try something like `"Xstring x =  $\Lambda$ ."` We substitute the null string whenever we are given a null pointer.

<Allocate space for the string and put a copy of s there 7>  $\equiv$

```
if (s  $\equiv$   $\Lambda$ ) s = "";  
p->s = new char [strlen(s) + 1];  
strcpy(p->s, s);
```

This code is used in sections 6 and 13.

```
/**
 * Returns true if this list contains the specified element.
 *
 * @param elem element whose presence in this List is to be tested.
 * @return true if the specified element is present;
 *         false otherwise.
 */
public boolean contains(Object elem) {
    return indexOf(elem) >= 0;
}
```

### **contains**

```
public boolean contains(Object elem)
```

Returns true if this list contains the specified element.

**Specified by:**

[contains](#) in interface [List](#)

**Overrides:**

[contains](#) in class [AbstractCollection](#)

**Parameters:**

elem - element whose presence in this List is to be tested.

**Returns:**

true if the specified element is present, false otherwise.

# Klausurtermine

ca. 90 Minuten

- Mittwoch 28.07.04 10:15
- Freitag 30.07.04
- Mittwoch 11.08.04 10:15
- Freitag 20.08.04 11:00
- Mittwoch 29.09.04
- Freitag 01.10.04