17-708 SOFTWARE PRODUCT LINES: CONCEPTS AND IMPLEMENTATION

IMPLEMENTATION 1: VERSION CONTROL SYSTEMS PARAMETERS

CHRISTIAN KAESTNER
CARNEGIE MELLON UNIVERSITY
INSTITUTE OF SOFTWARE RESEARCH

READING ASSIGNMENT OCT 5

Apel, S., Batory, D., Kästner, C., & Saake, G. (2013). Feature-Oriented Software Product Lines. Berlin: Springer.

Chapter 4

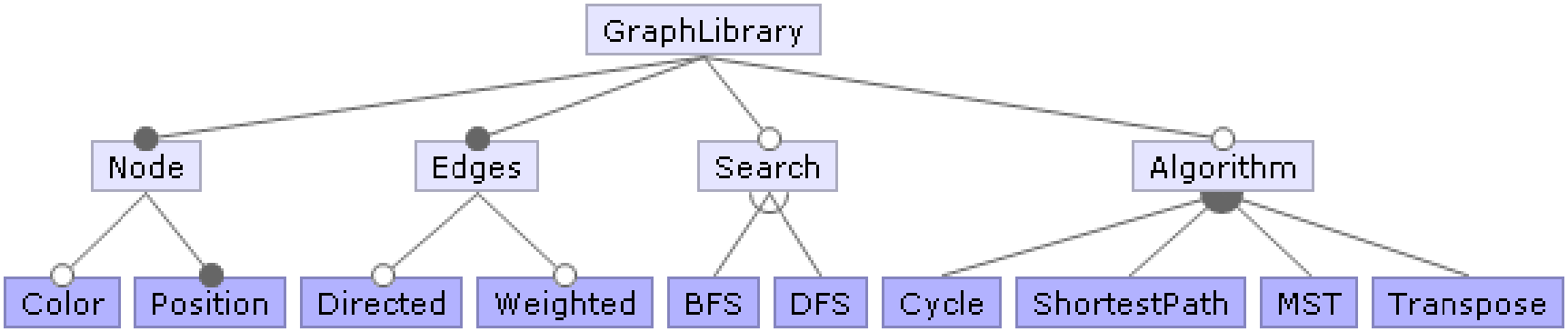
LEARNING GOALS

Explain the difference between versions, revisions and variants

Understanding the appeal and the problems of clone-based development

Use version control systems to implement and track minor variations

GRAPH PL


GraphLibrary
Node
Edges
Search
Algorithm
Color
Position
Directed
Weighted
BFS
DFS
Cycle
ShortestPath
MST
Transpose

GRAPH PL IMPL.

```
class Graph {
  Vector nv = new Vector(); Vector ev = new Vector();
  Edge add(Node n, Node m) {
    Edge e = new Edge(n, m);
    nv.add(n); nv.add(m); ev.add(e);
    e.weight = new Weight();
    return e;
  }
  Edge add(Node n, Node m, Weight w)
    Edge e = new Edge(n, m);
    nv.add(n); nv.add(m); ev.add(e);
    e.weight = w; return e;
  }
  void print() {
    for(int i = 0; i < ev.size(); i++) {
      ((Edge)ev.get(i)).print();
    }
  }
}
```

```
class Node {
  int id = 0;
  Color color = new Color();
  void print() {
    Color.setDisplayColor(color);
    System.out.print(id);
  }
}
```

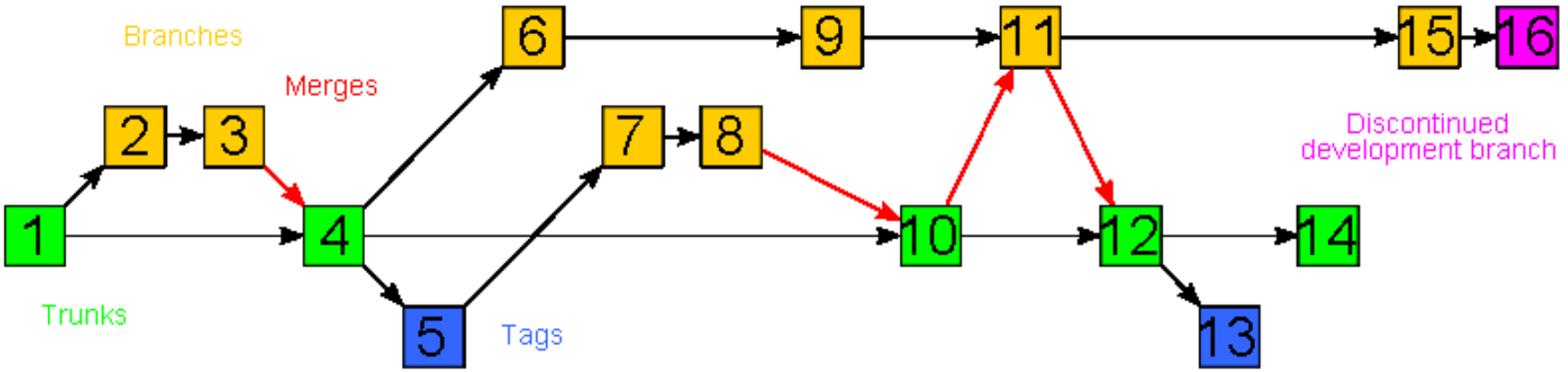
```
class Edge {
  Node a, b;
  Color color = new Color();
  Weight weight = new Weight();
  Edge(Node _a, Node _b) { a = _a; b = _b; }
  void print() {
    Color.setDisplayColor(color);
    a.print(); b.print();
    weight.print();
  }
}
```

```
class Color {
  static void setDisplayColor(Color c) { ... }
}
```

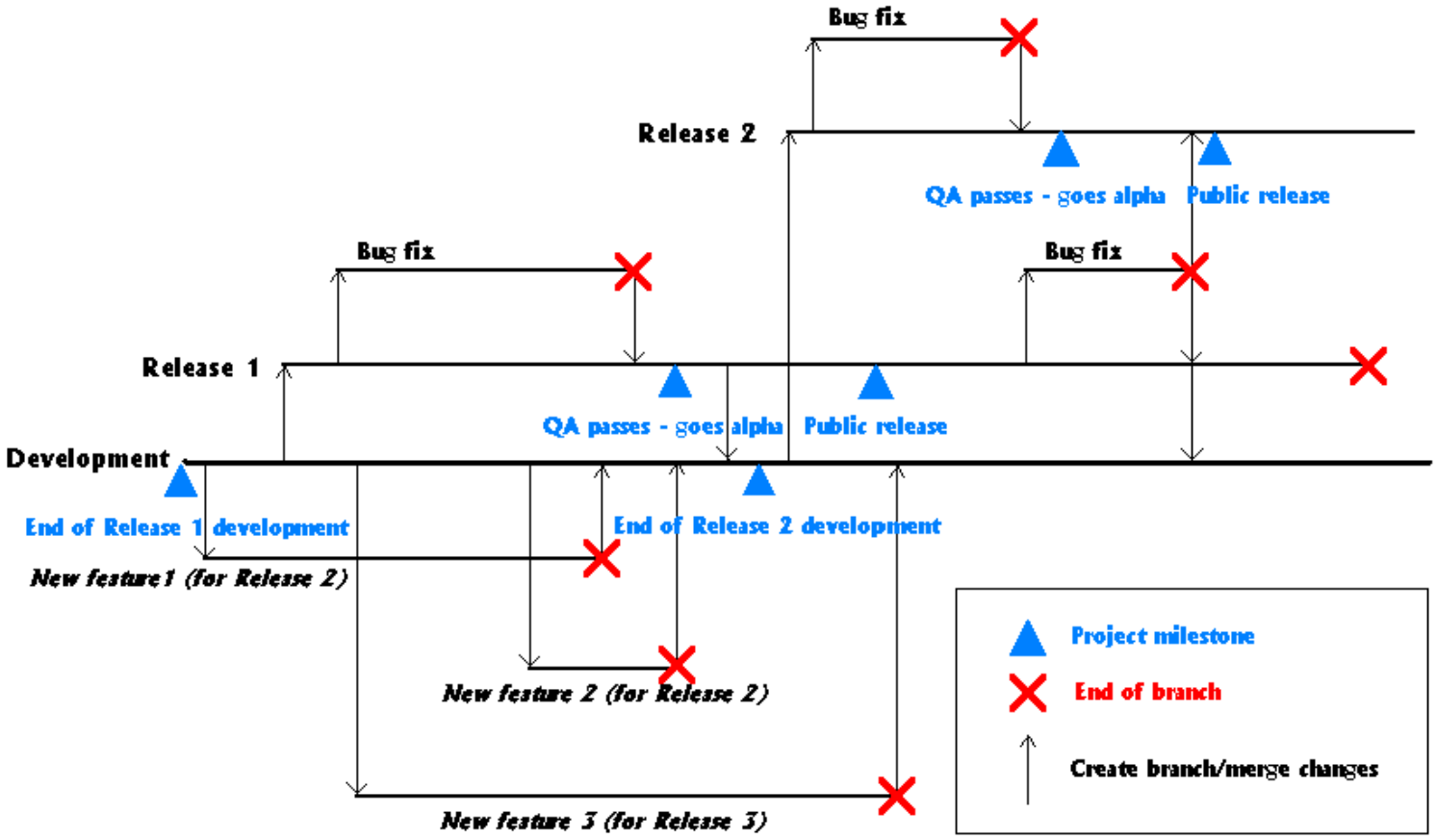
```
class Weight { void print() { ... } }
```

VERSION CONTROL SYSTEMS

BRANCHING & MERGING



RELEASE PLANNING



SOFTW. CONFIGURATION MANAGEMENT

"tracking and controlling change in software dev."

revision control

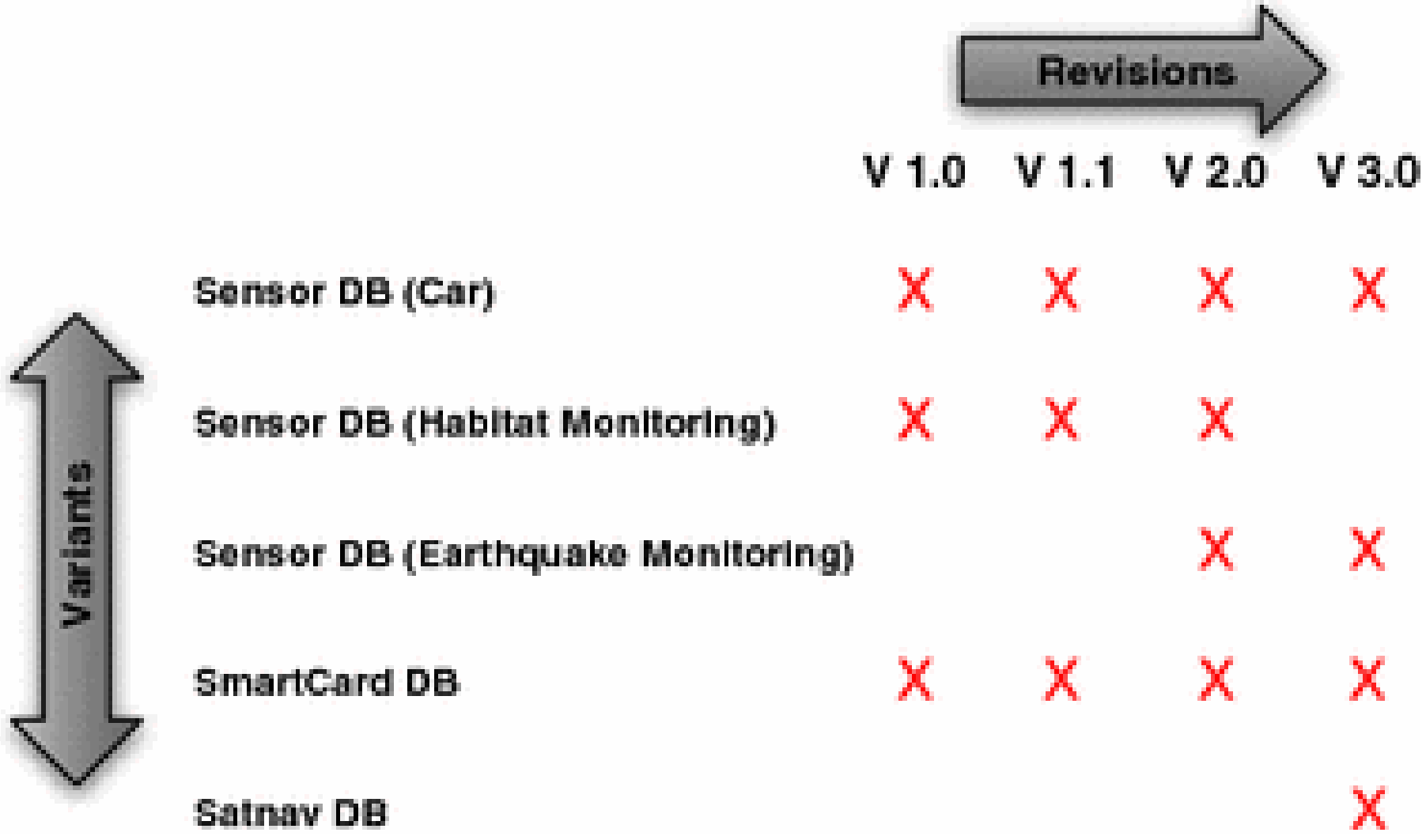
build consistency and reproducability

process enforcement

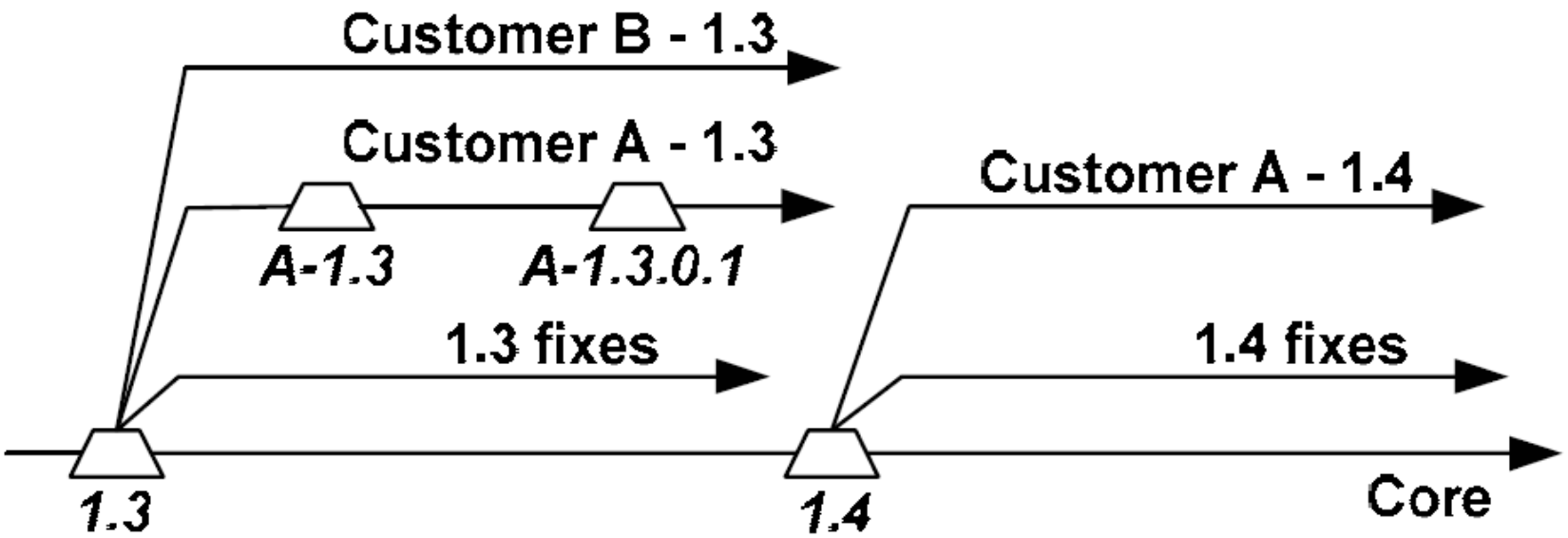
reporting and auditing

defect tracking

release planning

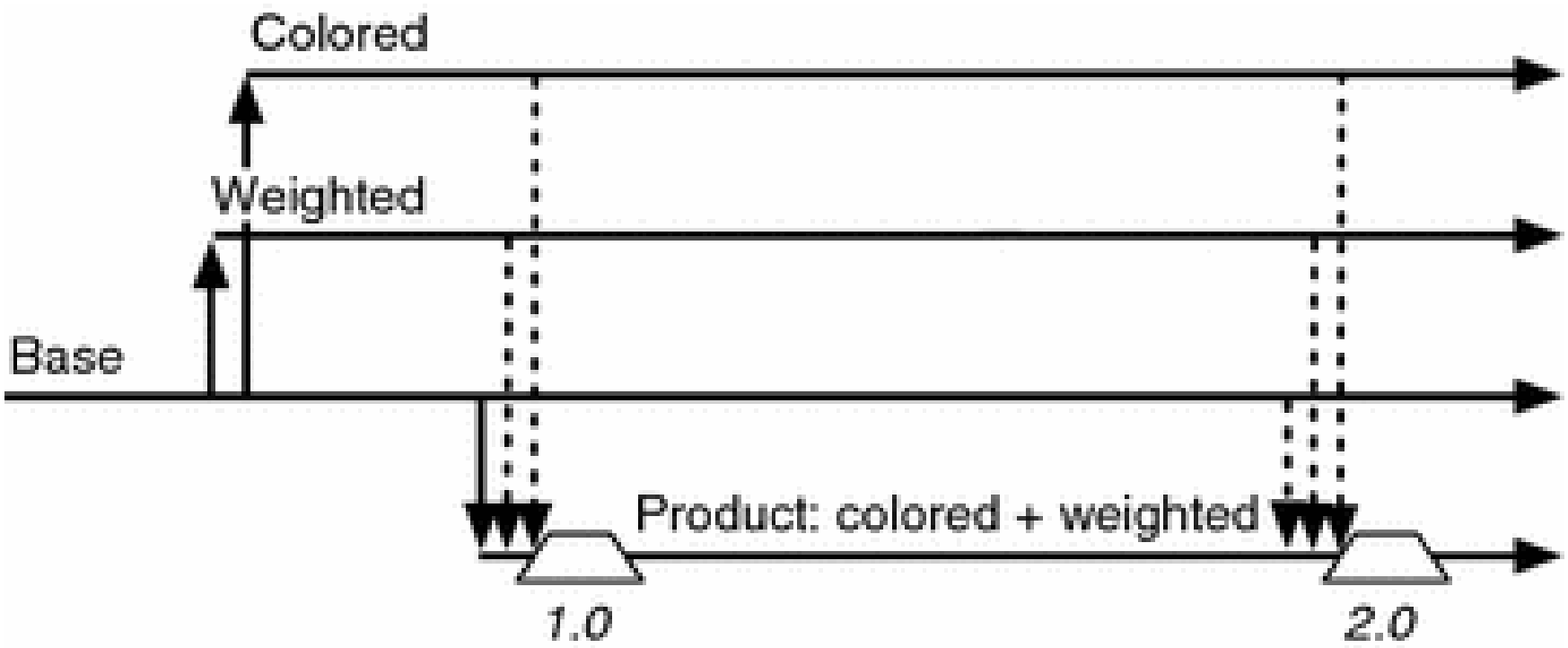

Revisions
V 1.0
V 1.1
V 2.0
V 3.0
Variants
Sensor DB (Car)
Sensor DB (Habitat Monitoring)
Sensor DB (Earthquake Monitoring)
SmartCard DB
Satnav DB

PRODUCT LINES WITH VERSION CONTROL



[Staples&Hill, APSEC'04]

FEATURE-ORIENTATION?



UNIFORMITY

Java Code

Documentation

Models

Build scripts

License

Grammars

Compiled binaries

HTML, JavaScript, CSS

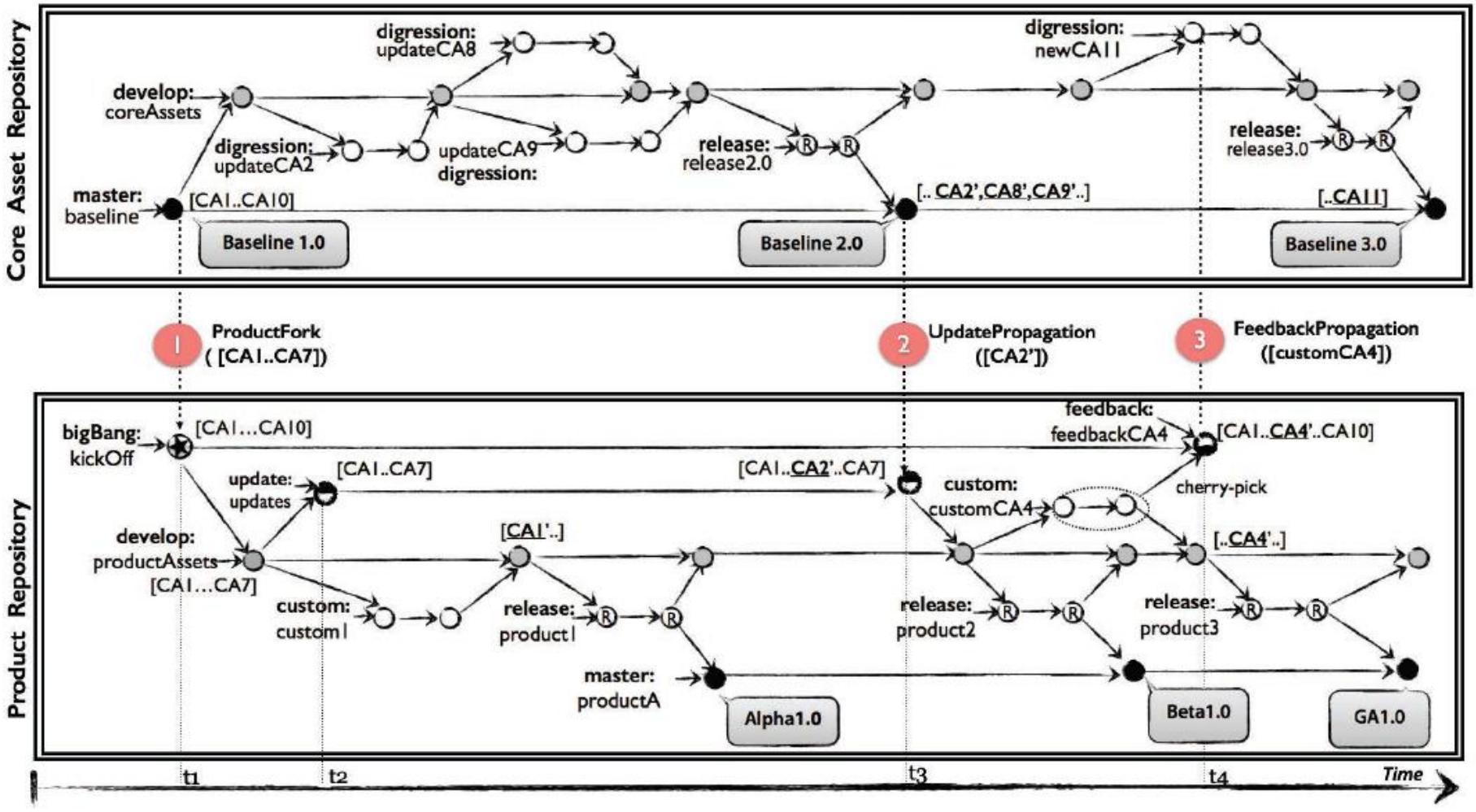
(binary files are difficult to merge)

APPEAL OF CLONE-AND-OWN

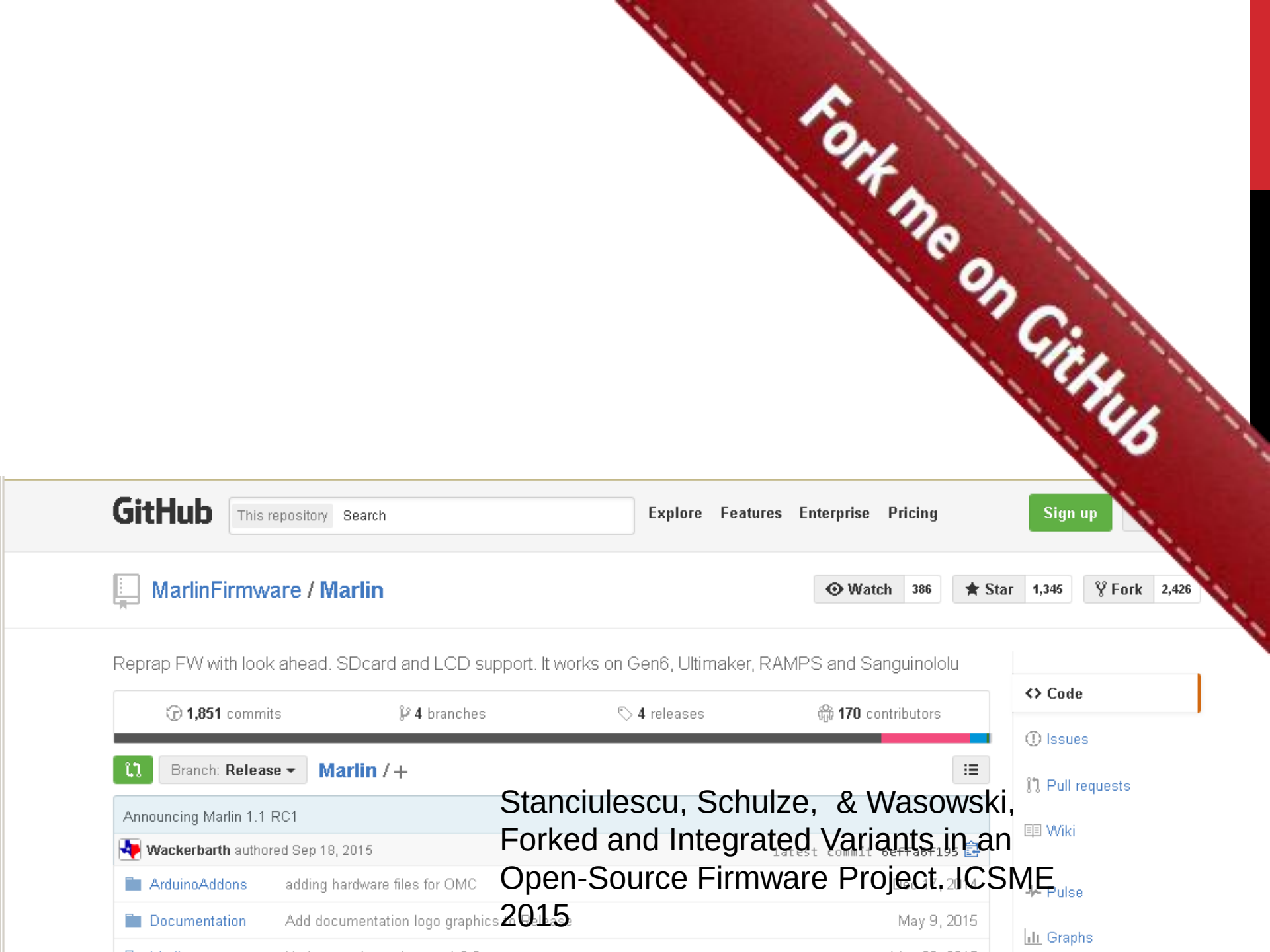
PROBLEMS OF CLONE-AND-OWN

	Statement	Not at all				To a great extent
1.	I'm aware of a product line strategy in my organization	1		2	2	6
2.	My team develops core assets that are later used by other teams	1	2	4	3	1
6.	We regularly clone pieces of code		3	5	2	1
10.	We measure how many times a certain core asset is used	5	4		2	
17.	People in my team know who should approve a change request			1	4	5
19.	We have work procedures that include cloning of artifacts	2	6	1	2	
20.	I feel our development process is well-defined		1	2	6	2
21.	We clone to reuse artifacts between products of a product line			5	5	1
22.	All project teams follow in practice the defined process	1		2	6	2
23.	We have relatively many clones of code		6	4	1	
27.	We do not change main APIs without receiving an approval			1	5	5
31.	We have relatively big clones of code		5	4	2	

[Dubinsky, CSMR'13]



Montalvillo, L., & Díaz, O. (2015, July). Tuning GitHub for SPL development: branching models & repository operations for product engineers. In *Proceedings of the 19th International Conference on Software Product Line* (pp. 111-120). ACM.



Stanciulescu, Schulze, & Wasowski, Forked and Integrated Variants in an Open-Source Firmware Project. ICSME 2015

Observation 6. *We have observed that Marlin developers preferred forking over integrated variability under the following reasons:*

S1. The fork extension has little relevance to other users.
S2. The maintenance time span for the developed code is expected to be short.
S3. The external developer has no control over the upstream project.
S4. A developer wants to create experimental code.
S5. An active project provides a good skeleton for adding new functionality.
S6. A defunct project contains code that can be reused.
S7. The developer wants to change the programming language.

Observation 4. *Decentralization of information in many forks is a challenge in fork-intensive development.*

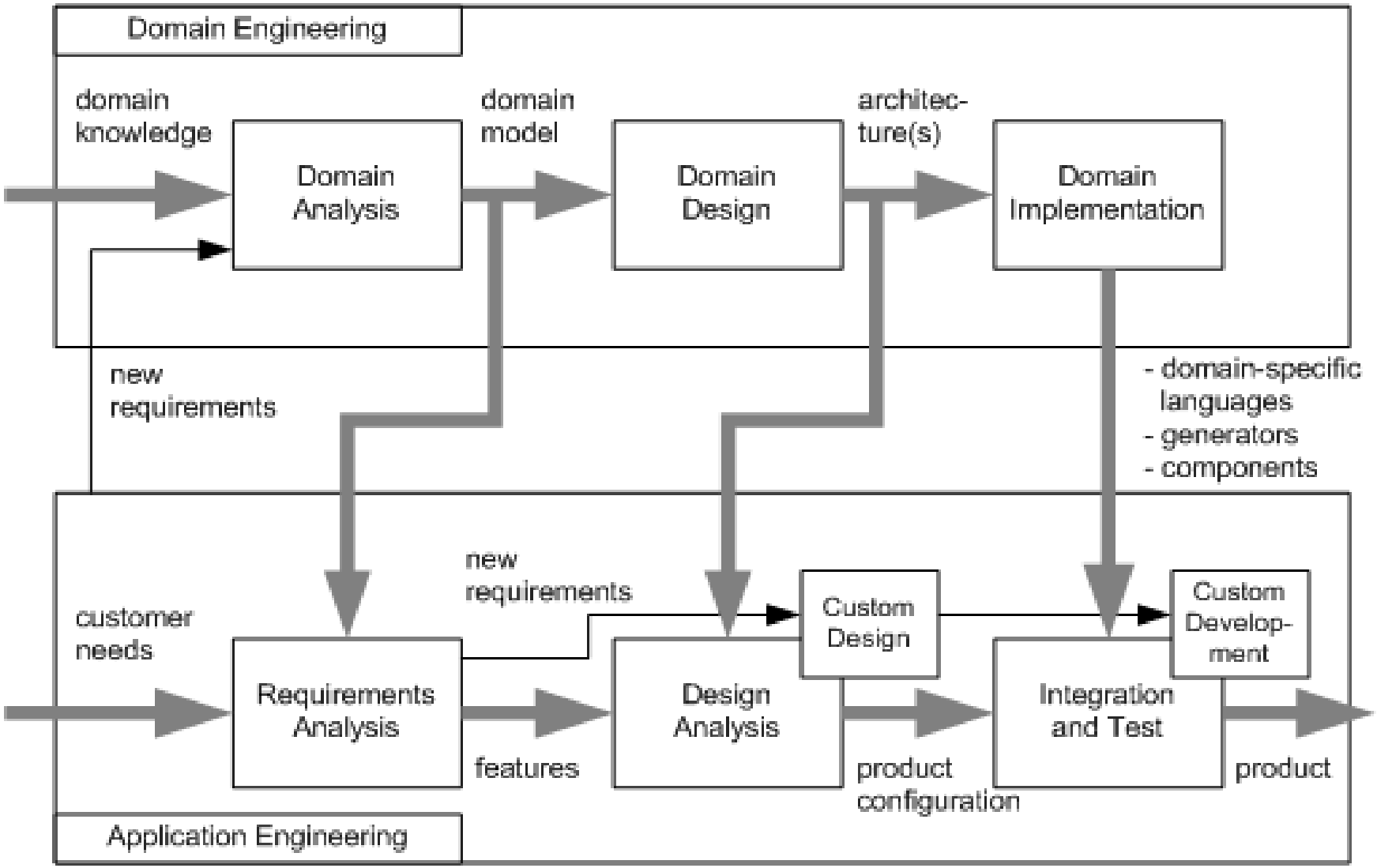
Observation 5. *Propagation of bug-fixes is a problem for forking, just like for cloning in-the-small, even though git offers facilities for selective download of patches from upstream.*

CLONE MANAGEMENT (VIRTUAL PLATFORM)

Products	BASE	LINE	RECT	COLOR	WIPE
Product P_1	✓	✓			✓
Product P_2	✓	✓		✓	
Product P_3	✓	✓	✓	✓	

Fischer, S., Linsbauer, L., Lopez-Herrejon, R. E., & Egyed, A. (2014, September). Enhancing clone-and-own with systematic reuse for developing software variants. In *Software Maintenance and Evolution (ICSME), 2014 IEEE International Conference on* (pp. 391-400). IEEE.

Antkiewicz, M., Ji, W., Berger, T., Czarnecki, K., Schmorleiz, T., Lämmel, R., ... & Schaefer, I. (2014, May). Flexible product line engineering with a virtual platform. In *Companion Proceedings of the 36th International Conference on Software Engineering* (pp. 532-535). ACM.


Domain Engineering
domain knowledge
Domain Analysis
domain model
Domain Design
architec-ture(s)
Domain Implementation
new requirements
- domain-specific languages
- generators
- components
customer needs
Requirements Analysis
new requirements
Custom Design
Design Analysis
Custom Develop-ment
Integration and Test
features
product configuration
product
Application Engineering

DISCUSSION

Modularity?

Traceability?

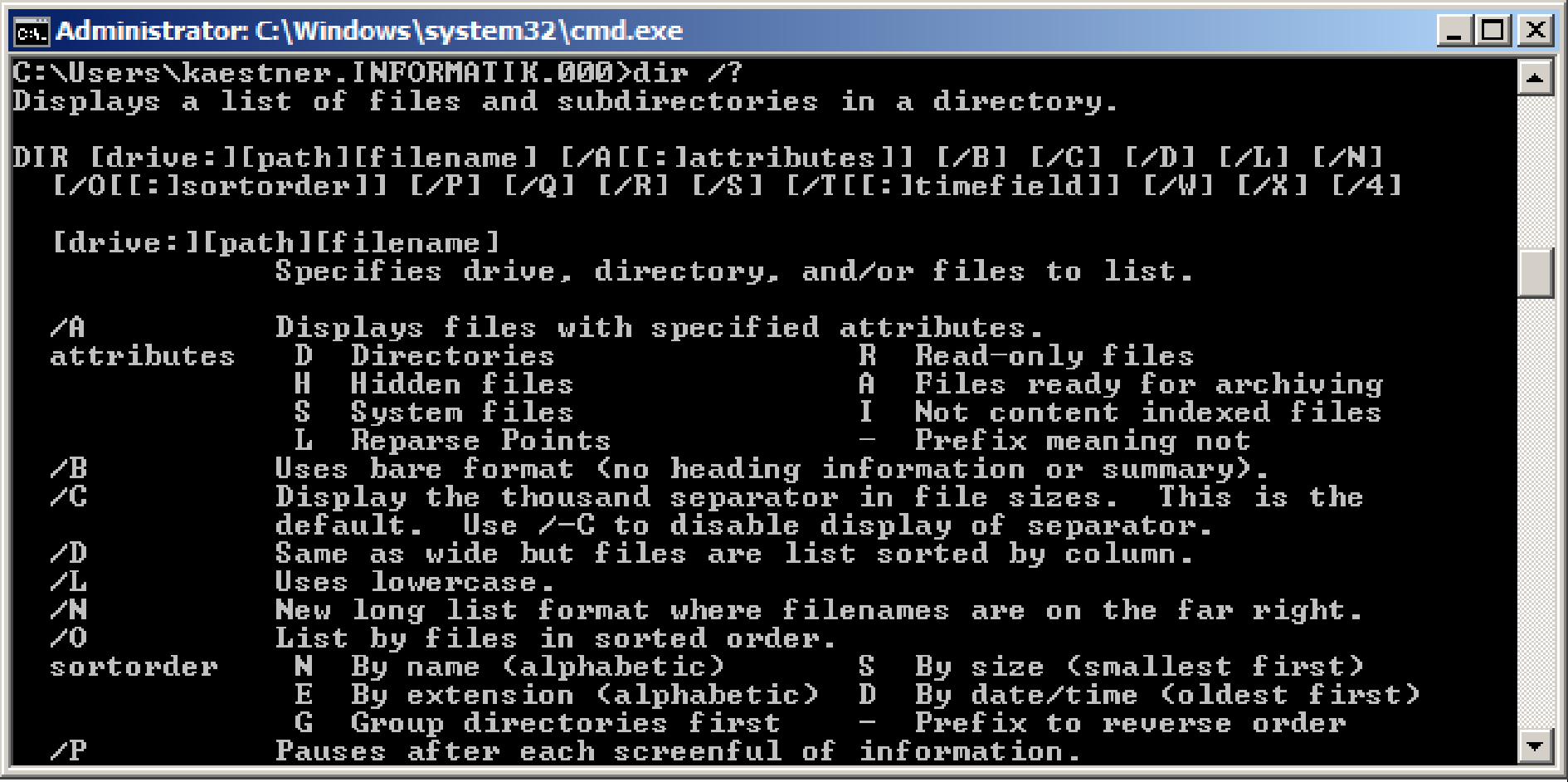
Effort?

Granularity?

Uniformity?

LOAD-TIME VARIABILITY

PARAMETER


Administrator: C:\Windows\system32\cmd.exe
C:\Users\kaestner.INFORMATIK.000>dir /?
Displays a list of files and subdirectories in a directory.
DIR [drive:][path][filename] [/A[[:]attributes]] [/B] [/C] [/D] [/L] [/N]
[/O[[:]sortorder]] [/P] [/Q] [/R] [/S] [/T[[:]timefield]] [/W] [/X] [/4]
[drive:][path][filename]
Specifies drive, directory, and/or files to list.
/A Displays files with specified attributes.
attributes D Directories R Read-only files
H Hidden files A Files ready for archiving
S System files I Not content indexed files
L Reparse Points - Prefix meaning not
/B Uses bare format (no heading information or summary).
/C Display the thousand separator in file sizes. This is the
default. Use /-C to disable display of separator.
/D Same as wide but files are list sorted by column.
/L Uses lowercase.
/N New long list format where filenames are on the far right.
/O List by files in sorted order.
sortorder N By name (alphabetic) S By size (smallest first)
E By extension (alphabetic) D By date/time (oldest first)
G Group directories first - Prefix to reverse order
/P Pauses after each screenful of information.

PARAMETER –I IN GREP

```
1  int match_icase;
2
3  int main (int argc, char **argv)
4  {
5    [...]
6    while ((opt = get_nondigit_option (argc, argv, &default_c
7      switch (opt)
8        {
9        [...]
10       case 'i':
11         match_icase = 1;
12         break;
13       }
14   }
15
16
17 static const char *
18 print_line_middle (const char *beg, const char *lim,
19                    const char *line_color, const char *match_color)
20 {
21   [...]
22   if (match_icase)
23     {
24       ibeg = buf = (char *) xmalloc(i);
25       while (--i >= 0)
26         buf[i] = tolower(beg[i]);
```

GLOBAL CONFIGURATION OPTIONS

```
class Conf {
   public static boolean Logging = false;
   public static boolean Windows = false;
   public static boolean Linux = true;
}
class Main {
  public void foo() {
     if (Conf.Logging)
       log(„running foo()“);
     if (Conf.Windows)
       callWindowsMethod();
     else if (Conf.Linux)
       callLinuxMethod();
     else
       throw RuntimeException();
  }
}
```

GRAPH IMPL.

```
class Conf {
  public static boolean COLORED = true;
  public static boolean WEIGHTED = false;
}
```

```
class Graph {
 Vector nv = new Vector(); Vector ev = new Vector();
 Edge add(Node n, Node m) {
  Edge e = new Edge(n, m);
  nv.add(n); nv.add(m); ev.add(e);
  if (Conf.WEIGHTED) e.weight = new Weight();
  return e;
 }
 Edge add(Node n, Node m, Weight w)
  if (!Conf.WEIGHTED) throw RuntimeException();
  Edge e = new Edge(n, m);
  nv.add(n); nv.add(m); ev.add(e);
  e.weight = w; return e;
 }
 void print() {
  for(int i = 0; i < ev.size(); i++) {
   ((Edge)ev.get(i)).print();
  }
 }
}
```

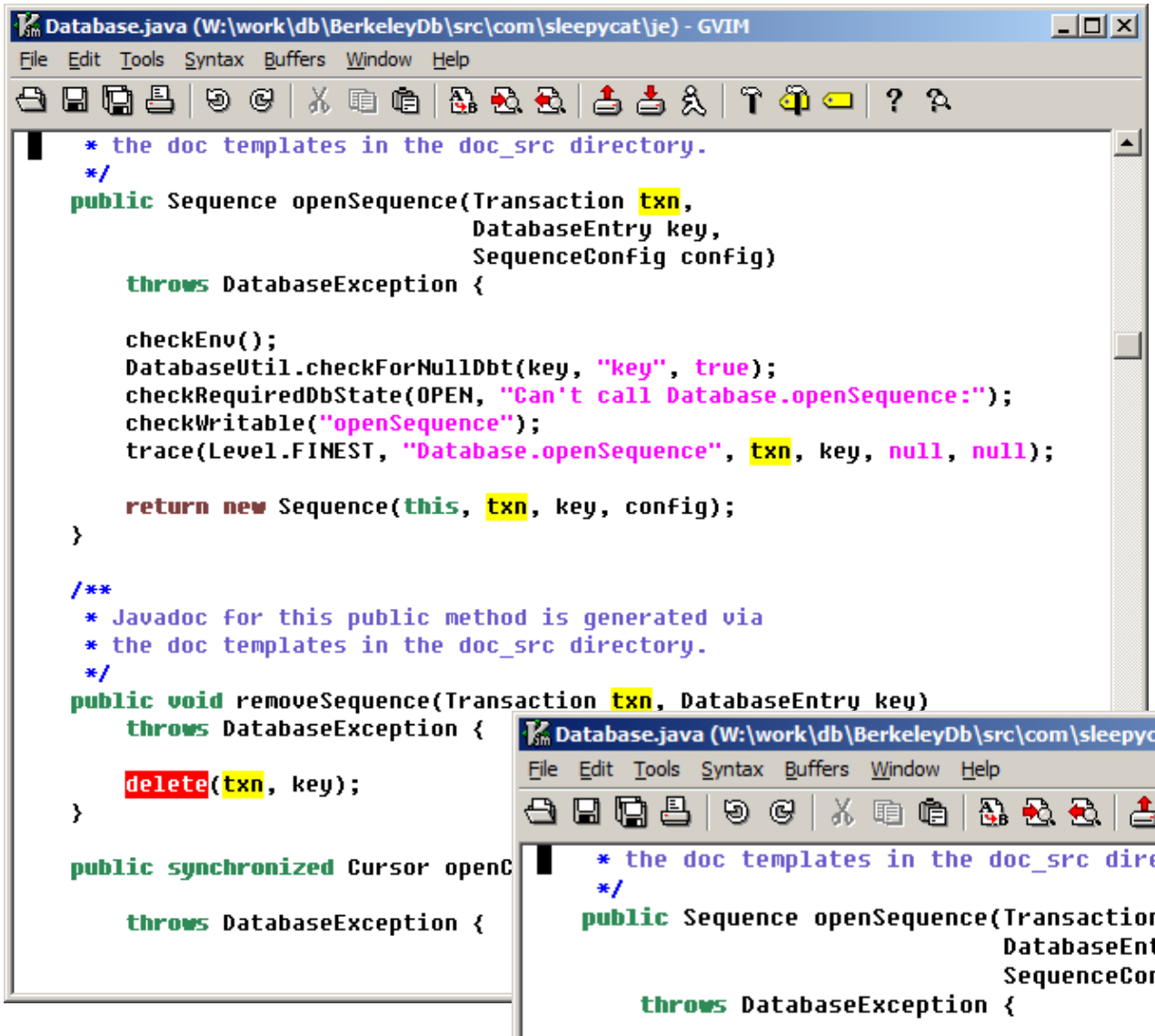
```
class Node {
 int id = 0;
 Color color = new Color();
 void print() {
  if (Conf.COLORED) Color.setDisplayColor(color);
   System.out.print(id);
 }
}
```

```
class Edge {
 Node a, b;
 Color color = new Color();
 Weight weight = new Weight();
 Edge(Node _a, Node _b) { a = _a; b = _b; }
 void print() {
  if (Conf. COLORED) Color.setDisplayColor(color);
  a.print(); b.print();
  if (Conf.WEIGHTED)  weight.print();
 }
}
```

```
class Color {
 static void setDisplayColor(Color c) { ... }
}
```

```
class Weight { void print() { ... } }
```

PROPAGATING PARAMETERS



replacing global variables by parameters

CONFIGURATION

httpd.conf -- win32 Apache
Building a Web Server, for Windows

```
Listen 80
ServerRoot "/www/Apache2"
DocumentRoot "/www/webroot"

ServerName localhost:80
ServerAdmin admin@localhost

ServerSignature On
ServerTokens Full

DefaultType text/plain
AddDefaultCharset ISO-8859-1

UseCanonicalName Off

HostnameLookups Off

ErrorLog logs/error.log
LogLevel error

PidFile logs/httpd.pid

Timeout 300

KeepAlive On
MaxKeepAliveRequests 100
```

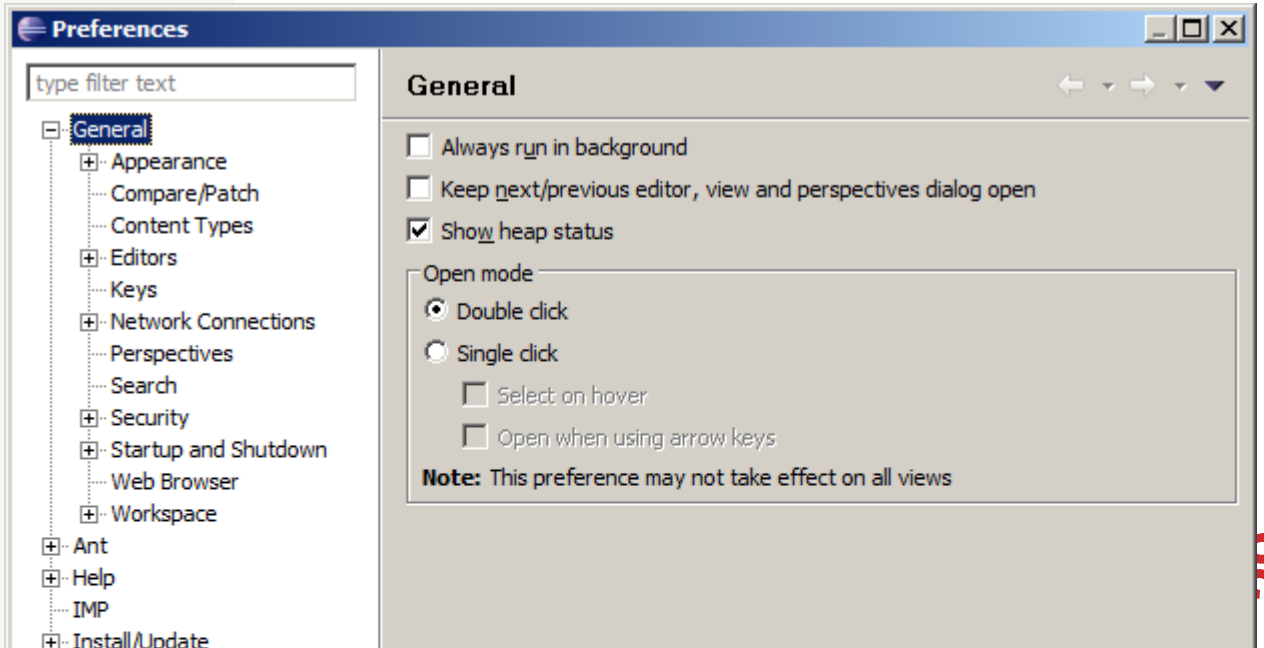
Command line parameter

Config files

Dialogs

Source code

…



DISCUSSION TRADEOFFS

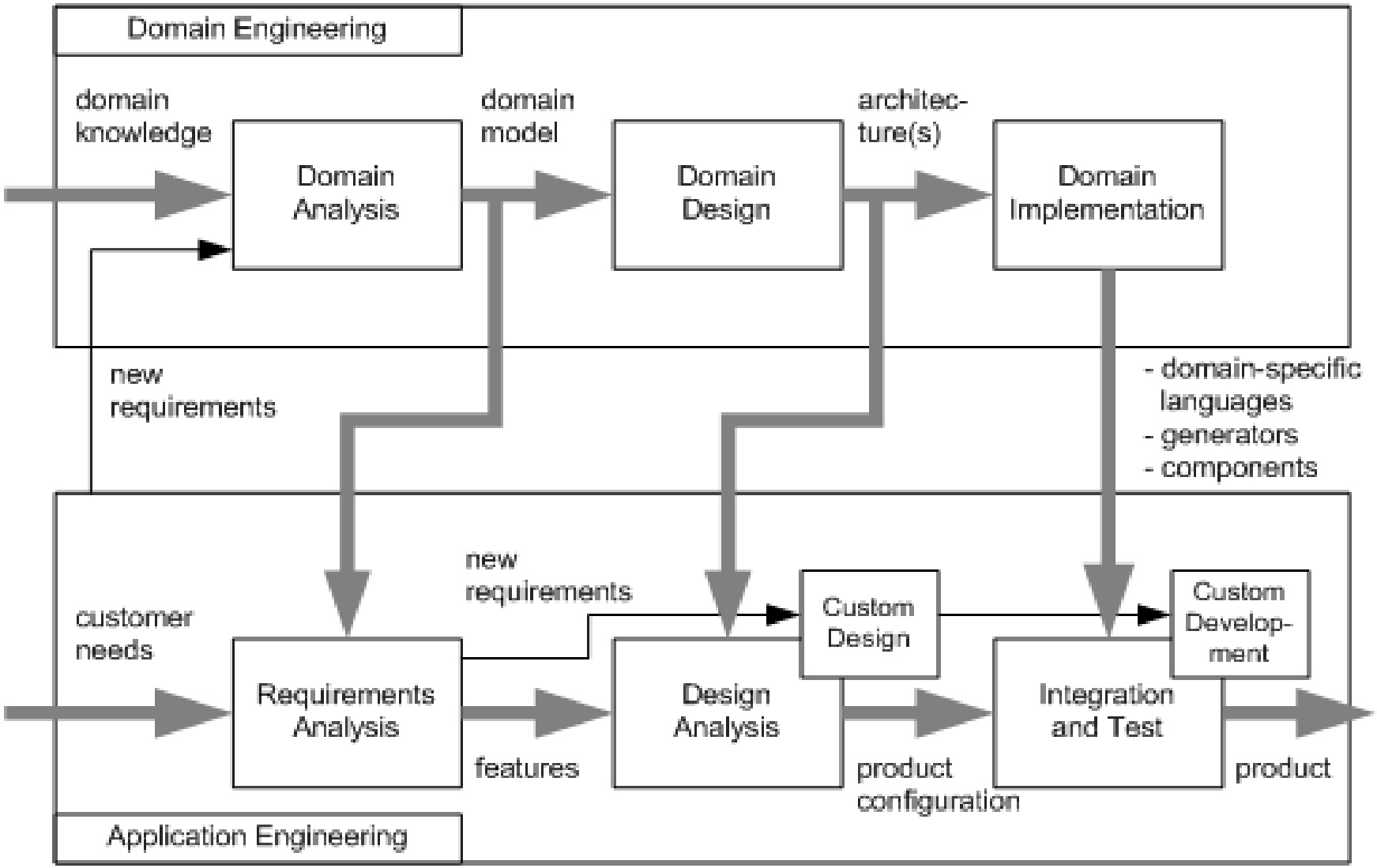
Modularity?

Traceability?

Effort?

Granularity?

Uniformity?


Domain Engineering
domain knowledge
Domain Analysis
domain model
Domain Design
architec-ture(s)
Domain Implementation
new requirements
- domain-specific languages
- generators
- components
customer needs
Requirements Analysis
new requirements
Custom Design
Design Analysis
Custom Develop-ment
Integration and Test
features
product configuration
product
Application Engineering

FURTHER READING

Staples, M., & Hill, D. (2004, November). Experiences adopting software product line development without a product line architecture. In *Software Engineering Conference, 2004. 11th Asia-Pacific* (pp. 176-183). IEEE.

Dubinsky, Y.; Rubin, J.; Berger, T.; Duszynski, S.; Becker, M.; Czarnecki, K., "An Exploratory Study of Cloning in Industrial Software Product Lines," in 17th European Conference on Software Maintenance and Reengineering (CSMR), pp.25-34, 2013

Antkiewicz, M., Ji, W., Berger, T., Czarnecki, K., Schmorleiz, T., Lämmel, R., ... & Schaefer, I. (2014, May). Flexible product line engineering with a virtual platform. In *Companion Proceedings of the 36th International Conference on Software Engineering* (pp. 532-535). ACM.