

# トレーサビリティとインパクト分析

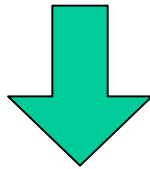
2011年7月13日

海谷 治彦

# 背景: ソフトウェア成果物

プロの開発では多様な成果物が作成される.

- 要求仕様書
- 設計仕様書
- ソースコード
- テストケース

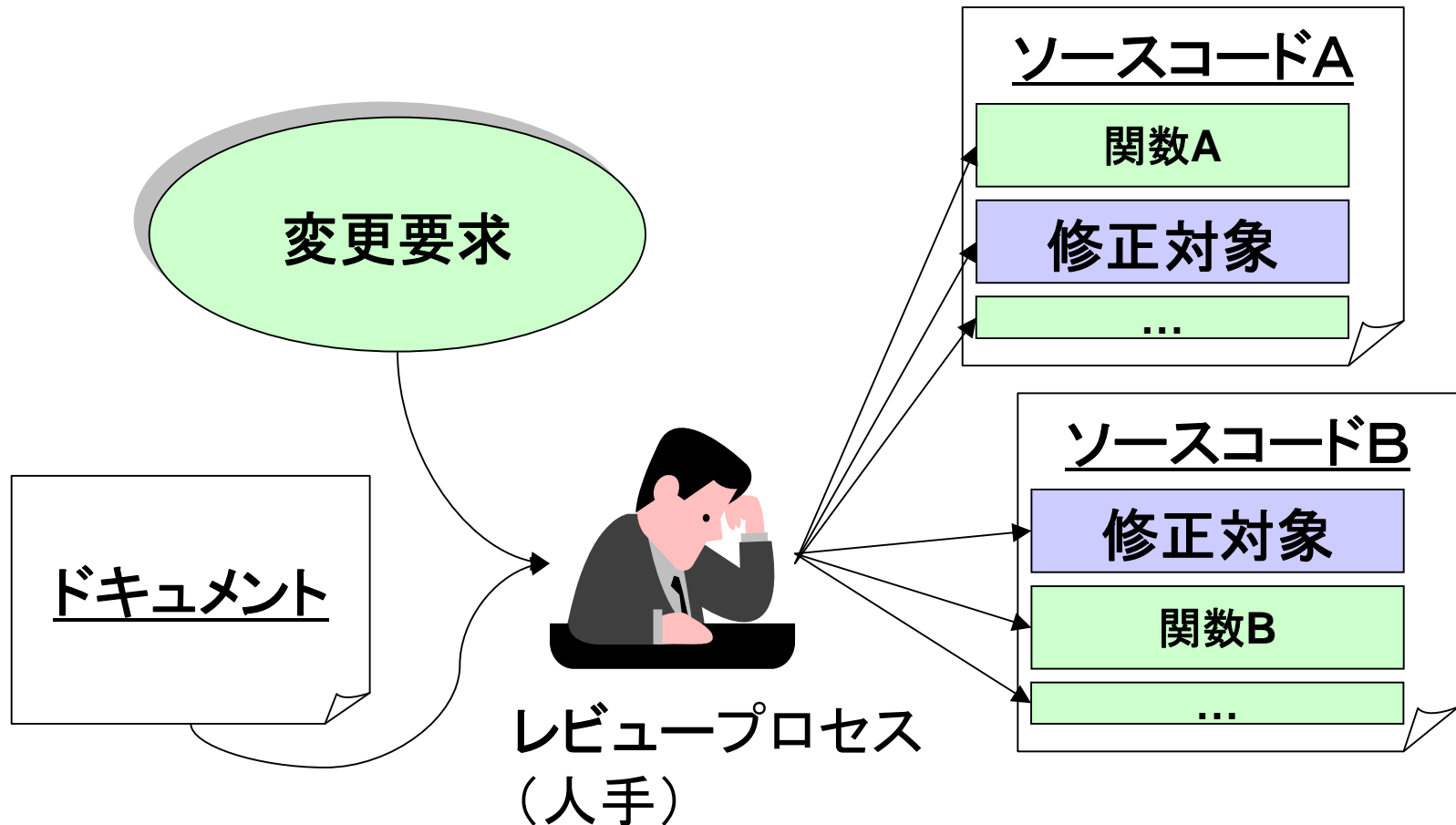


成果物内, 成果物間の相互の関係(トレース)を知ることが重要.

# トレースする理由

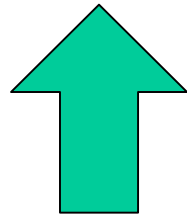
- 開発中に, どの機能が実現済か確認する.
  - 要求仕様書 ⇒ コード
- バグの原因を探す.
  - 仕様書 ⇒ コード
- 既存ソフトに新機能追加や性能アップのため, 改造・改良する際に, どこを書き直せば良いか探す.  
(インパクト分析)
  - 仕様書 ⇒ コード
- 仕様が見当たらない実システムの仕様を明確にする. (リバースエンジニアリング)
  - コード ⇒ 仕様

# 人手が主の変更要求への対応



# 例: 演習2,3の解答例

- 要求項目は14個
- ユースケースは6個
- (概念)クラスは5 or 6個
- ソースは6 or 7個
- 予想される要求変更が7個



- それぞれ単純に対応付いていないため、トレースをするための技術が必要.

# トレーサビリティの分類

- 水平方向のトレーサビリティ
  - 関数から関数, クラスからクラス等.
  - 変更や依存性.
  - 呼び出し関係, 共有関数の存在
- 垂直方向のトレーサビリティ
  - 要求項目から設計要素
  - 設計要素からコード片(関数やクラス)
  - 情報検索的なアプローチ

# 技術の分類

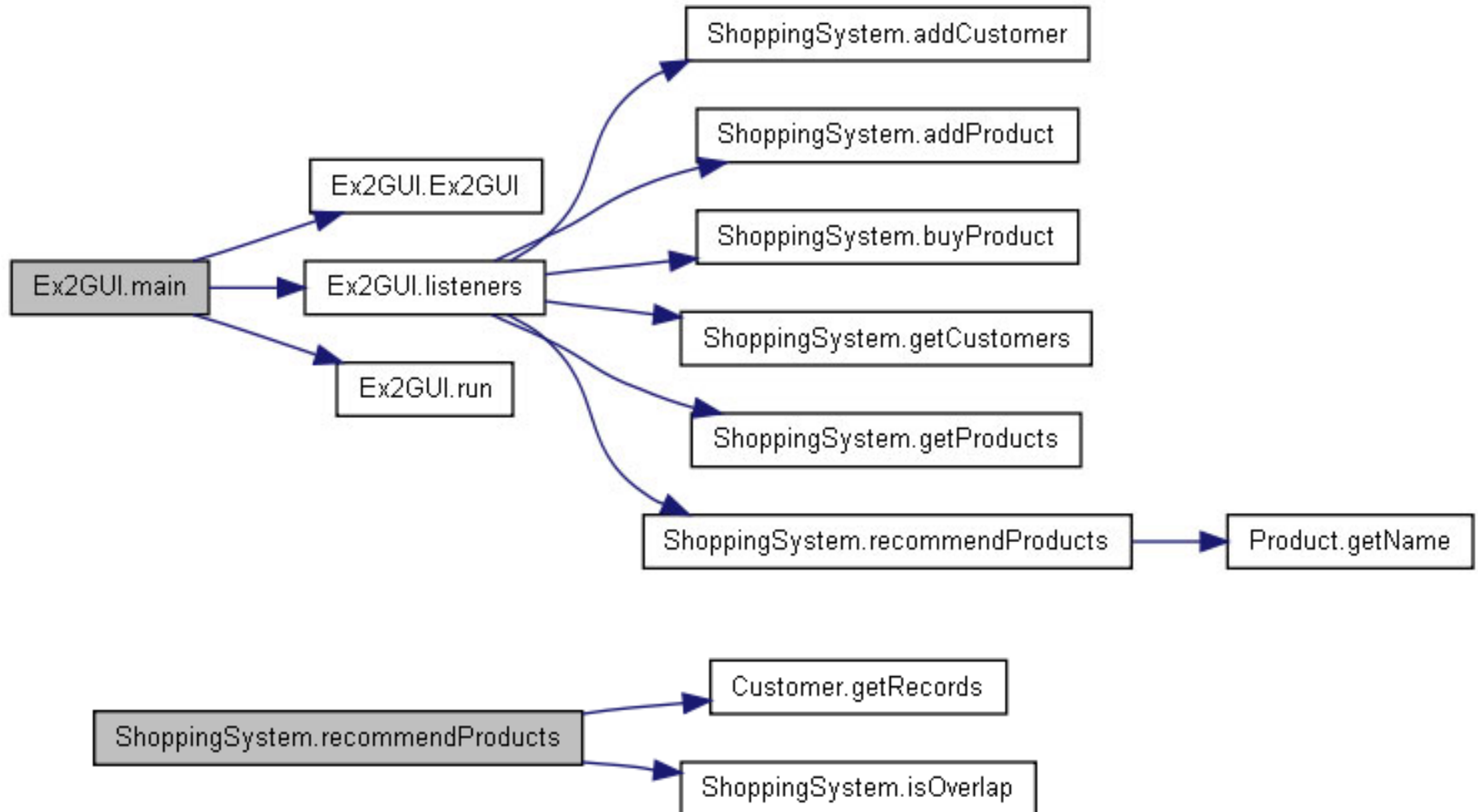
- 静的アプローチ
  - プログラムや文書の記述内容をもとにトレースをとる.
  - コールグラフ, データフロー解析.
  - 文書の場合は, 章構成や段落構成, 文内の係り受け等.
- 動的アプローチ
  - プログラムを動かしてみてトレースをとる.
  - プロファイリング.
  - デバッガ等を利用した解析.
  - 文書は当然, 動かないので原則, 適用できないが, ユースケース記述やシーケンス図を描いてみるのは, ある意味, 動的なアプローチ.

# コールグラフ

- 水平型トレースのための静的技術.
- 要は関数やメソッドの呼び出し関係を明確にする.
- 呼び出し関係がある関数間には, 変更の波及がある場合が多いため.
- コールグラフを作成するツールもある.
  - doxygen
  - 描画には graphviz を用いる



# 例



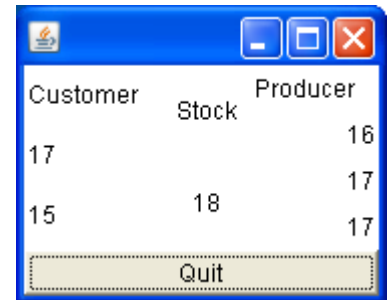
# プロファイラ

- プログラムの実行時の情報を収集するツール.
- 本来はパフォーマンスの向上のための情報等を集めるために使われる.
- ちょっとしたプログラムでも長大なデータになる.
- コレも水平方向のトレースに役立つ.

# 例 hprofの出力例

C:¥> java -agentlib:hprof -jar Run.jar

スレッドのところでやった生産者・消費者問題  
数秒動かしただけで、7万行の情報を出力.



Customer	Stock	Producer
17		16
		17
15	18	17

Quit

TRACE 302139:

java.awt.Component.<init>(<Unknown Source>:Unknown line)

java.awt.Label.<init>(<Unknown Source>:Unknown line)

TabacoSale.init(TabacoSale.java:29)

TabacoSale.main(TabacoSale.java:66)

TRACE 302140:

IntLabel.<init>(IntLabel.java:5)

TabacoSale.init(TabacoSale.java:32)

TabacoSale.main(TabacoSale.java:66)

TRACE 302141:

java.lang.Thread.<init>(<Unknown Source>:Unknown line)

LabelUpdater.<init>(LabelUpdater.java:8)

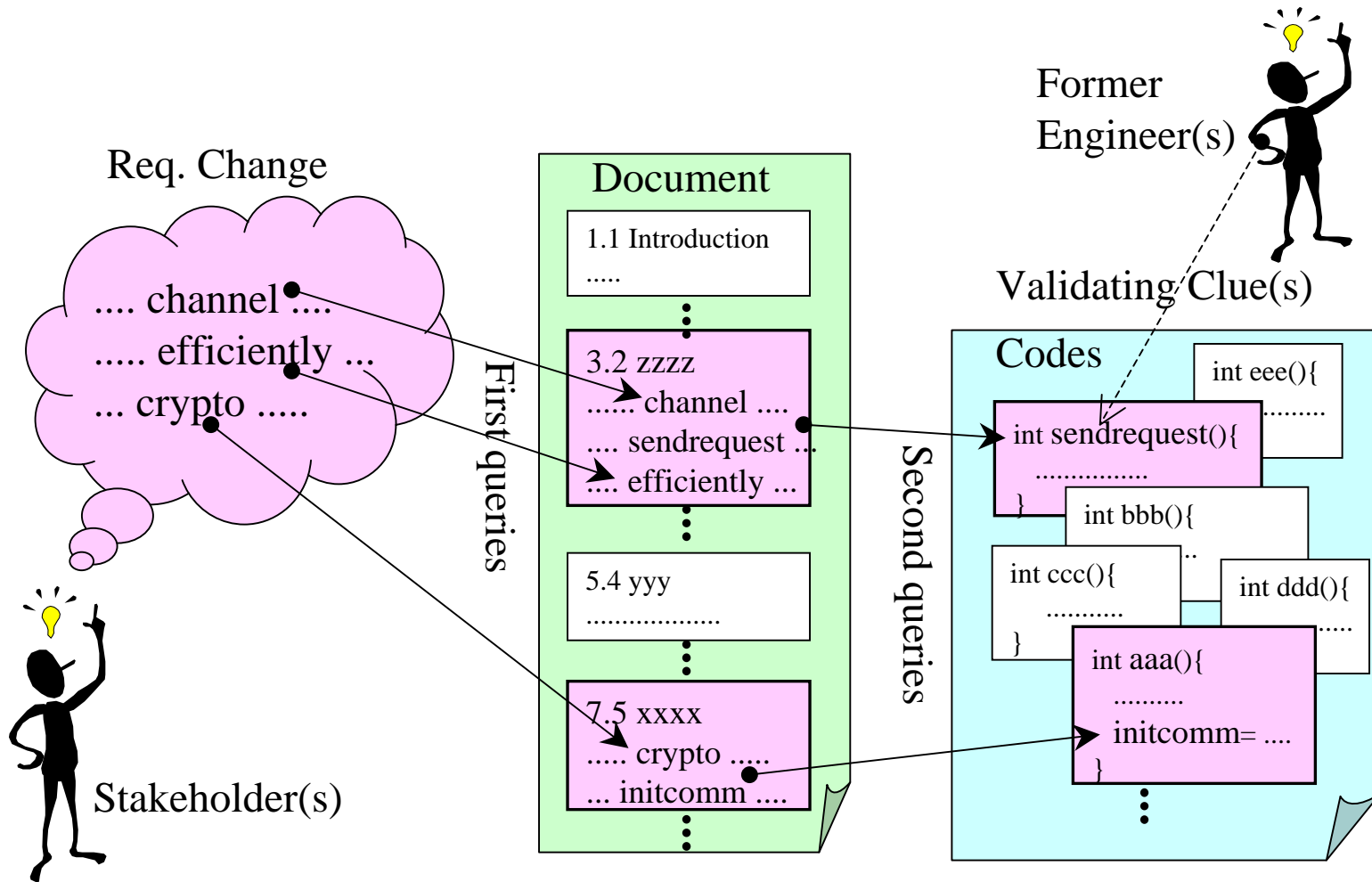
Customer.<init>(Customer.java:3)

TabacoSale.init(TabacoSale.java:32)

# 垂直の例と技術

- まだ気軽に使えるツールはあまり無い.
  - ソースだけでなく, 文書解析をしないといけな  
いたため.
- 情報検索
  - 文書の出現語句に基づき, トレースをとる.
- 同義語の解決
  - 「言い回し」の違いの吸収.
  - 辞書(オントロジー)の利用

# 我々の取り組み



**First Query**

send receive connect

Query	Count	Rate	Min	1st	Median	3rd	Max
send	41.0	10.448	0.0	0.0	1.0	2.0	21.0
receive	36.0	7.463	0.0	0.0	1.0	2.0	18.0
connect	0.0	0.0	0.0	0.0	0.0	0.0	0.0

**FirstQuery Selector**

Noun	Count	Verb	Count
file	107	send	41
command	74	issue	38
char.	63	display	37
directory	33	receive	36
argument	28	connect	33
socket	26	setup	31
function	26	use	24
mode	23	retrieve	24
login	21	specify	22
option	20	quit	22

Query: \_\_\_\_\_ Count: \_\_\_\_\_

OK Cancel

**Section**

ID	Caption	Query Type	Query Cou..	Use	Never
22	2.0. sendrequest	2	24	<input checked="" type="checkbox"/>	<input type="checkbox"/>
23	2.9. abortrecv	1	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
24	2.10. recvrequest	2	19	<input checked="" type="checkbox"/>	<input type="checkbox"/>
25	2.11. initconn	1	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
30	2.16. pswitch	2	4	<input checked="" type="checkbox"/>	<input type="checkbox"/>
32	2.18. proxtrans	3	7	<input checked="" type="checkbox"/>	<input type="checkbox"/>
35	2.21. abort_remote	1	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>

19/134

**Second Query**

0 < Count < 1,000 in Function in Section

ID	Sec..	Cou..	Rate	Min	1st	Med	3rd	Max	Cou..	Rate	Min	1st	Med	3rd	Max	Use	Neve..
849	rsstat	2.0	1.449	1.0	1.0	1.0	1.0	1.0	0.0	2.980	1.0	1.0	1.0	1.0	2.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
707	initco..	2.0	1.449	1.0	1.0	1.0	1.0	1.0	3.0	2.239	1.0	1.0	1.0	1.0	1.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
476	stor	2.0	1.449	1.0	1.0	1.0	1.0	1.0	7.0	2.985	1.0	1.0	1.5	2.75	3.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
668	putch..	4.0	1.449	2.0	2.0	2.0	2.0	2.0	19.0	4.478	1.0	1.75	2.0	6.0	6.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
325	type	6.0	1.449	3.0	3.0	3.0	3.0	3.0	51.0	9.701	1.0	1.0	2.0	7.0	11.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
700	mode	1.0	0.725	1.0	1.0	1.0	1.0	1.0	0.0	2.230	2.0	2.0	2.0	4.0	4.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>

89/1261

**Impact-prone Functions**

Function n..	Path	DirID	LOC	MCV	2Q Count	Predict	Training	Classified
remglob	/ftp/cmds.c	2	82	0	12	HIT_POSI		0.0
checkglob	/ftp/cmds.c	2	53	0	1	HIT_NEGA		0.6
dotrans	/ftp/cmds.c	2	25	0	1	HIT_NEGA		0.6
domap	/ftp/cmds.c	2	168	0	1	HIT_NEGA		0.6
globulize	/ftp/cmds.c	2	26	0	2	HIT_NEGA	NO	
confirm	/ftp/cmds.c	2	28	0	4	HIT_POSI	YES	
getit	/ftp/cmds.c	2	125	0	53	HIT_NEGA	NO	
quote1	/ftp/cmds.c	2	18	0	12	HIT_NEGA	NO	
pipeprotect	/ftp/cmds.c	2	19	8	3	HIT_POSI	YES	
do_settype	/ftp/cmds.c	2	22		7	HIT_NEGA		0.6

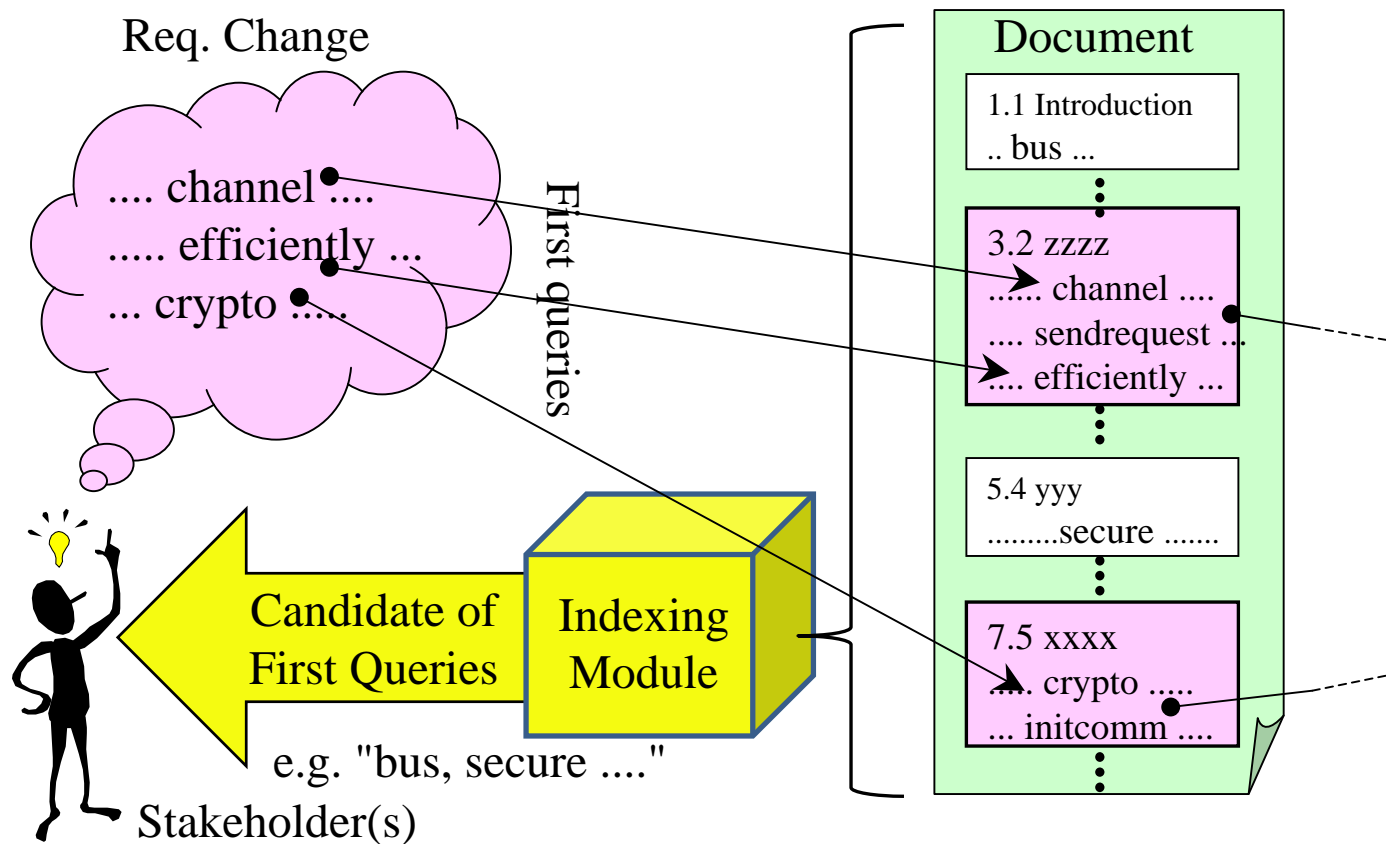
ML Tree

positive 94  
negative 38

LOC order  
100 %

# 要求変更を特徴付けるには？

## 索引付けの利用



# 画面例

- 変更要求を特徴付けるキーワードを見つけるための支援.
- 候補となる索引単語を提示する方法.

The screenshot displays a software interface with a search results table and a 'FirstQuery Selector' dialog box.

**Search Results Table:**

第1クエリ	総出現数	被含有率	Min	1st	Median	3rd	Max
送信	41.0	9.589	0.0	0.75	1.0	2.0	21.0
受信	36.0	6.849	0.0	0.0	1.0	2.25	18.0
コネクション	0.0	0	0.0	0.0	0.0	0.0	0.0

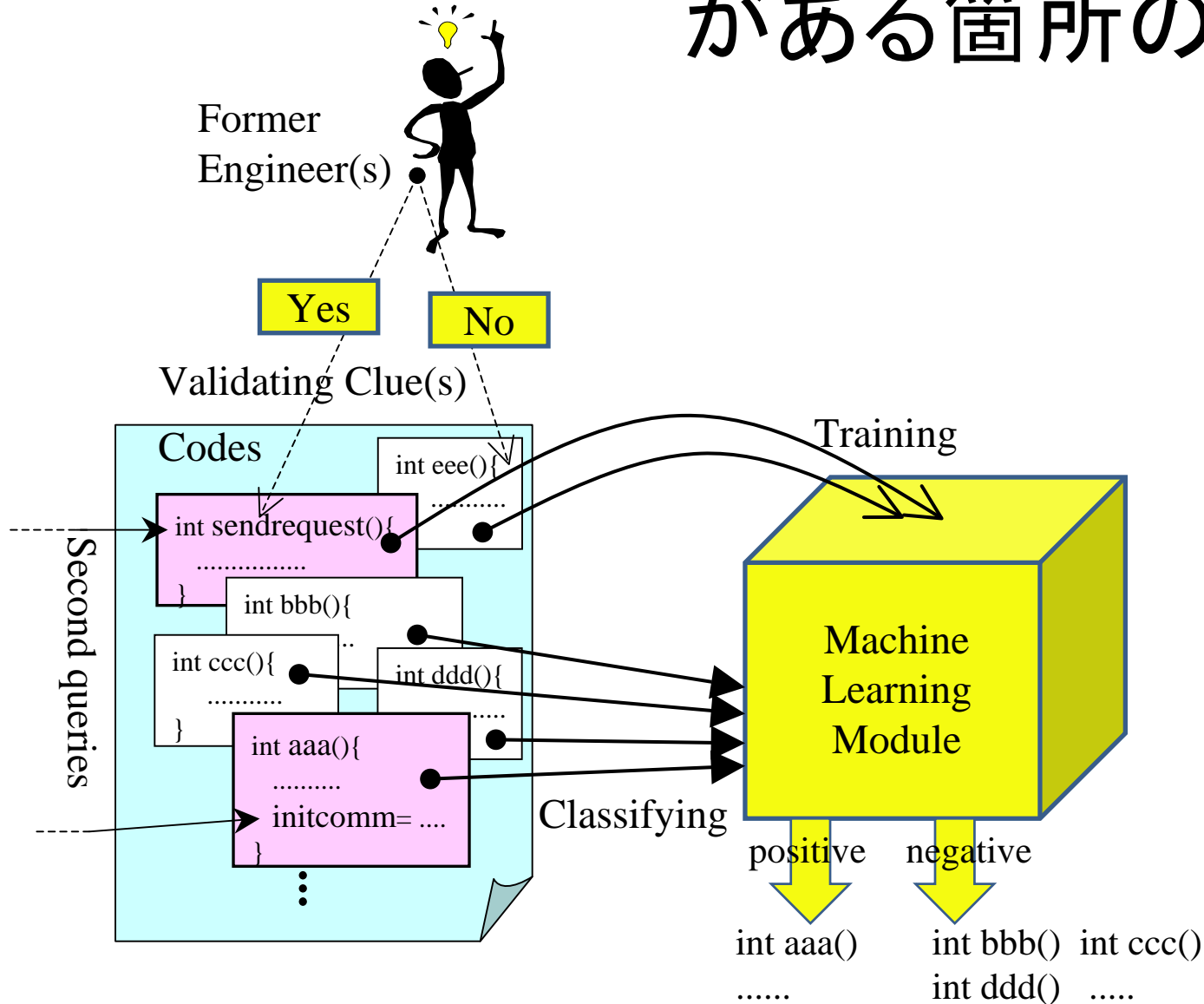
**FirstQuery Selector Dialog Box:**

名詞	Count	サ変名詞	Count
ファイル	107	送信	41
コマンド	74	発行	38
文字	63	表示	37
ディレクトリ	33	受信	36
引数	28	接続	33
ソケット	26	設定	31
関数	26	使用	24
モード	23	取得	24
ログイン	21	指定	22
オプション	20	終了	22

The dialog box also includes a 'Query' field, a 'Count' field, and an 'OK' button.

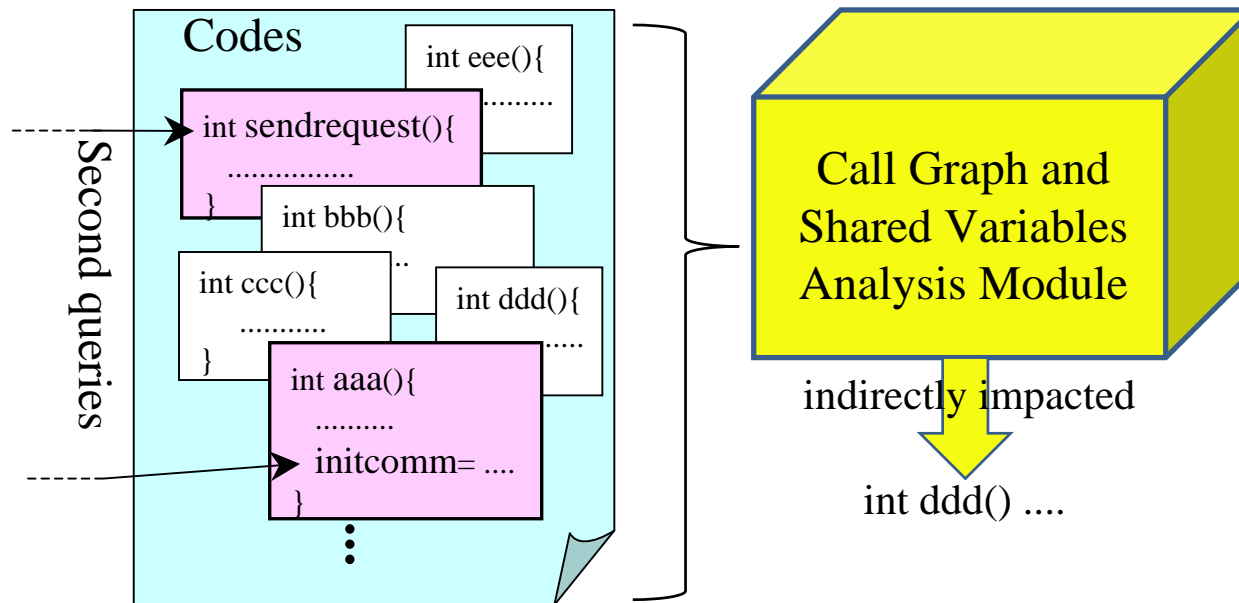


# 機械学習の利用による変更の可能性 がある箇所の予測



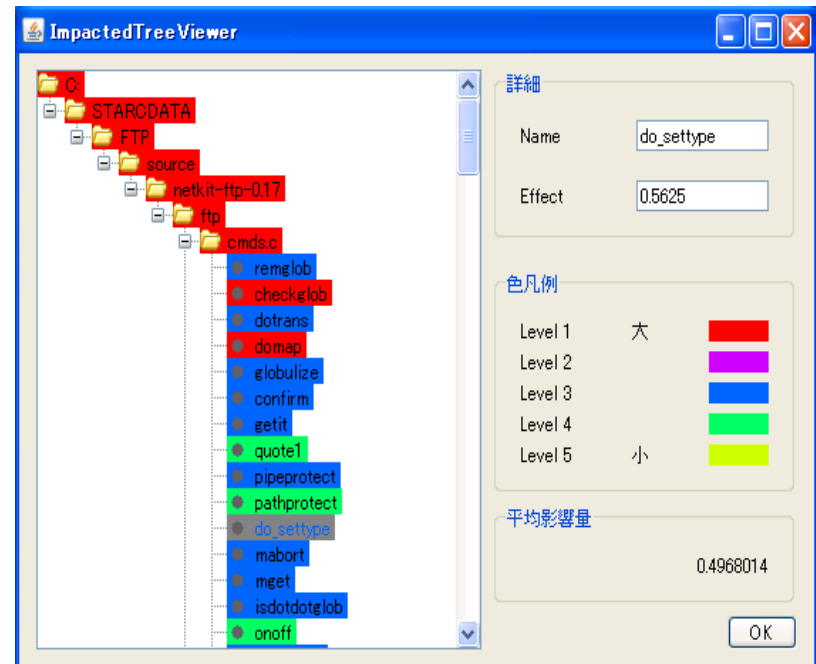
# 変更波及の可視化

## コールグラフと共有変数の利用



# 画面例

- シード関数を起点
  - 「関数の呼び出し関係」  
+0.5
  - 「共有変数による関係」  
+0.75
- 影響値が大きい関数ほど赤に近い色で表現され、小さい関数ほど黄に近い色を示す.



# インパクト分析ツール Jripples

- Eclipse上で変更波及の分析(インパクト分析)を行うツール.
- Wayne State Univ. (ミシガンの大学らしい)で開発された.
- 演習で行う小規模なプログラム開発では、あまりピンと来ないが、クラスの数が数百、数千のプログラムでの分析に有効.
- 基本的に対話的にインパクトを探すツール.

# インストールと使い方

- 基本的に公式HP参照

<http://jripples.sourceforge.net/>

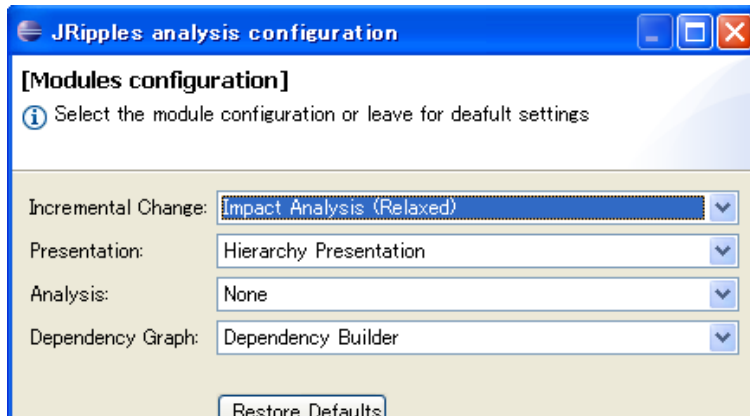
# 三つの主要概念

- Concept Location
  - ソース中で変更最初にインパクトがある部分を見つける作業.
- Impact Analysis
  - 最初のインパクトから波及してインパクトを受ける部分を探す作業.
- Change Propagation
  - 変更波及の実現によって、整合性がとれなくなった部分を探す作業.

# TIPS

- Configuration において, Relaxed を指定すること.
  - そうしないと, mainメソッドからしか辿れなくなる.
- Impactedをクラスやメソッドにマーキングして, 変更の種を植え付け, その波及箇所を予測するのに使う.

# 画面例



Class	Mark	Change Probability (CCIR)	Full Name
[-] Customer	Next		Customer
● setSales			Customer::setSal...
□ sales			Customer::sales
● <sup>c</sup> Customer			Customer::Custo...
□ name			Customer::name
● <sup>2</sup> addSale	Next		Customer::addSa...
● setName			Customer::setNa...
● getRecords			Customer::getRe...
● toString			Customer::toStri...
● <sup>c</sup> Customer			Customer::Custo...
● getSales			Customer::getSal...
● getName			Customer::getNa...
[+] Ex2			Ex2
[+] Ex2GUI			Ex2GUI
[+] LoadSave			LoadSave
[+] Product			Product
[-] Sale	Next		Sale
□ customer			Sale::customer
□ product			Sale::product
● getCustomer			Sale::getCustomer
● <sup>c</sup> Sale			Sale::Sale
● getAmount			Sale::getAmount
□ amount			Sale::amount
● setCustomer			Sale::setCustomer
● setProduct			Sale::setProduct
● <sup>2</sup> <sup>c</sup> Sale	Next		Sale::Sale
● getProduct			Sale::getProduct
● setAmount			Sale::setAmount
[-] ShoppingSystem	Impacted		ShoppingSystem
□ <sup>2</sup> buyProduct	Impacted		ShoppingSystem::...
● getRproducts			ShoppingSystem::...
● recommendProducts			ShoppingSystem::...
● setRcustomers			ShoppingSystem::...
● buyProduct			ShoppingSystem::...
● getCustomers			ShoppingSystem::...
● <sup>2</sup> rproducts	Next		ShoppingSystem::...
● <sup>c</sup> ShoppingSystem			ShoppingSystem::...
● getProducts			ShoppingSystem::...
● <sup>2</sup> rcustomers	Next		ShoppingSystem::...
● setRproducts			ShoppingSystem::...
● recommendProducts			ShoppingSystem::...
● addProduct			ShoppingSystem::...