ソフトウェア分析学研究室の研究テーマ

岡山大学大学院自然科学研究科
門田 暁人
monden@okayama-u.ac.jp

4つの研究分野

開発プロジェクト支援

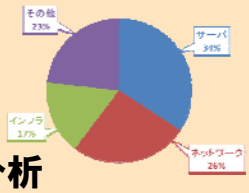
バグ予測

仕様書の分析

RFP	評価者			ページ数	無視範囲
	エキスパート	教員	エンジニア		
A.自治体	2.71	5.99	6.88	59	10
B.図書館	27.31	32.79	33.40	11	3
C.政府機関	18.91	15.16	16.91	21	5
D.大学	5.06	5.88	5.34	71	5
E.病院	42.75	44.08	64.90	160	12

開発者教育支援

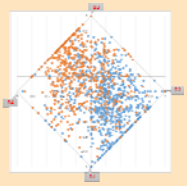
開発作業の分析



求人票の分析

データマイニング技術

原型分析



尖度，歪度の変換

$$H(x;\epsilon,\delta) = \sin h[\delta \sinh^{-1}(x) - \epsilon]$$

ソフトウェア保護

```
int func(int c)
{
    return c -'A'+'a';
}
```

→

```
int func(int c){int c1, c2,
x1=6, x2=13; c1=(c+x1)%
13; c2=(c+x2)%19; return
decode(c1, c2);}
int decode(int a, int b){ int
x=(a+13*(b-a)*3)%247;
if(x<0) x+= 247; return x;}
```

元のプログラム　　難読化後のプログラム

プログラムの難読化，バースマーク

開発プロジェクト支援：バグ予測

■ 機械学習により，プログラム中のバグ位置を推定し，デバッグに役立てます．

```
int check_bendchange(int p)
{
 static int last_bend = -1;
 if(p == -1){
  last_bend = -1;
  return -1;
 }
 if(p == last_bend) return -1;
 last_bend = p;
 return 0;
}
```

ソースコード

機械学習

バグ予測モデル

各行のバグ含有確率

```
15  getchar();                                                          39
16  for(i = 0; i < n; i++){                                             28
17  gets(room[i]);                                                      11
18  }                                                                   10
19  count = 0;                                                          41
20  for(i = 0; i < n; i++){                                             32
21  for(j = 0; j < m; j++){                                             42
22  if(room[i][j] != C) continue;                                       25
23  if(i >= 1 && room[i-1][j] != '.' && room[i-1][j] != C && !table[room[i-1][j] - 'A']){   1
24  table[room[i-1][j] - 'A'] = 1;                                      29
25  count++;                                                            13
26  }                                                                   20
```

*福谷 圭吾, 門田 暁人, Zeynep Yucel, 畑 秀明, "ソフトウェアバグの行レベル予測の試み," FOSE2017, pp. 105-110, Nov. 2017.

開発プロジェクト支援：仕様書の分析

■ 発注仕様書に含まれる**無理難題**の分析

- 東芝のケースでは，損失を電気会社ではなく東芝が負う契約になっていた．

〇〇病院システム

・・・・

新システムのテストに際し，現行システムの利用が必要となる場合はその費用を負担すること

・・・

将来の機能拡張等におけるデータ移行時に特別な費用が発生しないこと

■ 地方自治体，病院，図書館，政府機関，大学のシステム仕様書を分析した．

- すべての仕様書に「将来発生することに対応を求める要求」が含まれていた．

*門田暁人, 住吉倫明, 神谷芳樹, "提案依頼書に含まれる無理難題の分類," SEC journal, Vol.13, No.3, pp.18-25, Dec. 2017.

開発プロジェクト支援：テスト最適化

■ バグ予測に基づくソフトウェアテスト戦略の提案と評価

$$t_i = t_{total} \cdot (\hat{F}_i/S_i)/(\hat{F}_{total}/S_{total})$$
$$t_i = t_{total} \cdot \hat{F}_i/\hat{F}_{total} \cdot S_i/S_{total}$$
$$t_i = T_{total} \cdot \hat{F}_i/\hat{F}_{total} \cdot \log(S_i)/\sum_{i=1}^{n}\log(S_i)$$

テスト最適化戦略

バグ票

発生日	不具合内容	担当者
41116	・・・	A
41117	・・・	A
41118	・・・	A

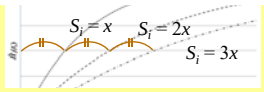
設計書　プログラム

過去の開発のデータ

機械学習

バグ位置推定モデル

$\hat{H}_i(t_i) = a_i[1-\exp(-b_i t_i)], \quad b_i = b_0 / S_i$



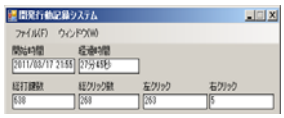
テストのシミュレーション

テストのコストは
25%削減可能である.

Akito Monden et al., “Assessing the cost effectiveness of fault prediction in acceptance testing,” IEEE Trans. Software Eng. 39(1), 1345-1357, 2013

開発者教育支援：開発作業の分析

開発者



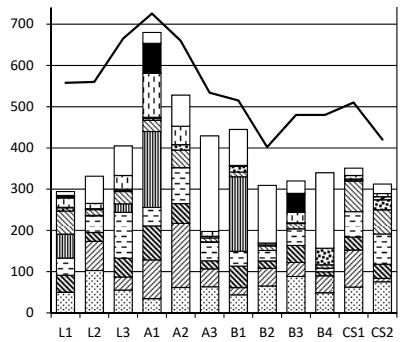
計測ツールTaskPit

アプリケーション実行履歴

実行ファイル名	開始時刻	左クリック	右クリック	打鍵回数
taskpit.exe:TaskPit	20XX年〇月〇日 15時21分54秒	1	0	0
firefox.exe:〇〇〇 - Mozilla Firefox	20XX年〇月〇日 15時22分08秒	1	0	0
firefox.exe:□□□ - Mozilla Firefox	20XX年〇月〇日 15時22分11秒	1	0	0
explorer.exe:Program Manager	20XX年〇月〇日 15時22分20秒	1	0	0
explorer.exe:解析系	20XX年〇月〇日 15時22分49秒	2	0	0
devenv.exe:△△△ - Microsoft Visual Studio	20XX年〇月〇日 15時22分51秒	1	0	0

開発作業の推定

行番号	候補1	候補2	候補3	自動推定
1	テスト	文書作成・確認		テスト
2				テスト
3				テスト
4	テスト	文書作成・確認		テスト
5	プログラミング	テスト		テスト
6	プログラミング	テスト		テスト
7	プログラミング	テスト		テスト
8	テスト	文書作成・確認		テスト
9	テスト			テスト
10	テスト			テスト
11	テスト			テスト



作業の改善点の分析

*清水良介, 門田暁人, Zeynep Yucel, 上野 秀剛, "計算機上のソフトウェア開発作業の自動推定の検討," WWS2018, Jan. 2018.

開発者教育支援：プログラミング作法の分析

プログラミング学習サイト

Webスクレイピング

```
int check_bendchange(int p)
{
 static int last_bend = -1;
 if(p == -1){
  last_bend = -1;
  return -1;
 }
 if(p      last_bend) return -1;
 last_bend = p;
 return 0;
}
```

ソースコード

pmd DON'T SHOOT THE MESSENGER

静的解析

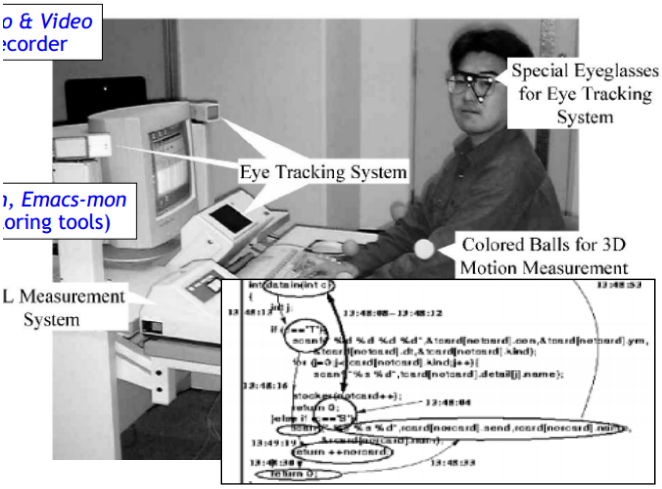
プログラミング作法	評判	優先度	小規模	人数
UUC	×		×	
UV		×		
FDA				
CT	×			
PLC				
SBR			×	
ARP				
SBE			×	×

初心者が守るべきプログラミング作法

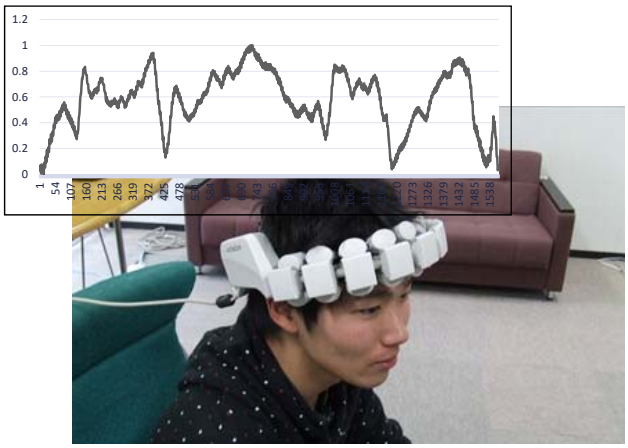
順位	プログラミング作法名	検出件数（件）	検出回数（回）	非順守率（%）	人数（人）
1	UseUtilityClass（UUC）	4843	4849	84.09	88
2	UseVarargs（UV）	487	879	8.46	39
3	FieldDeclarationsShouldBeAtStart OfClass（FDA）	479	1190	8.32	24
4	ConfusingTernary（CT）	388	580	6.74	39
5	PositionLiteralsFirstInComparisons（PLC）	335	750	5.82	40
6	SimplifyBooleanReturns（SBR）	280	311	4.86	20
7	AvoidReassigningParameters（ARP）	247	721	4.26	36
8	SimplifyBooleanExpressions（SBE）	236	486	4.10	17

*大西 臣弥, 門田 暁人, "ランキング上位者のプログラミング作法の評価," FOSE2017, pp. 231-232, Nov. 2017.

開発者教育支援：視線・脳活動の分析



視線追跡装置を用いたデバッグ行動の分析



脳血流計測（NIRS)装置を用いたプログラム理解行動の分析

■ 熟練者は，キーとなる部分を読んでいる．

■ 短期記憶が必要な箇所で脳が活性化している．

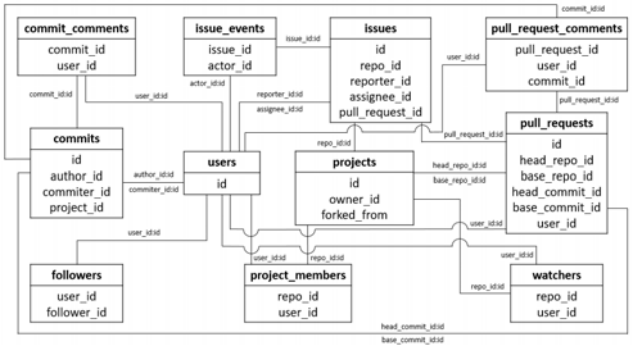
*中川, 亀井, 上野, 門田, 鵜林, 松本, "脳活動に基づくプログラム理解の困難さ測定," コンピュータソフトウェア, Vol.33, No.2, pp.78-89, June 2016.

開発者教育支援：GitHub上のプログラマ名鑑の開発

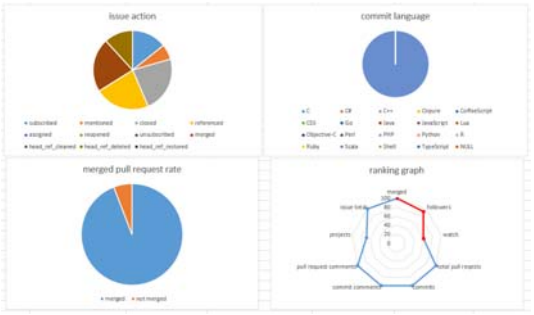
GitHub
（約2000万Gitリポジトリ，100万人）

GitHub APIの
実行履歴

データベース化



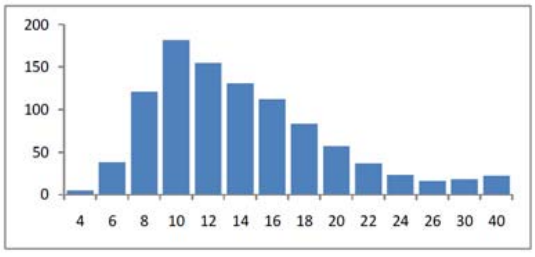
分析



開発者プロファイル・ランキング

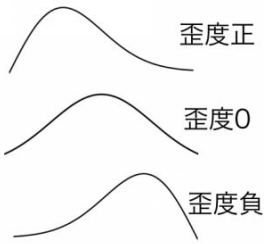
*池本 和靖, 門田 暁人, "GitHub上のプログラマ名鑑の作成に向けて," FOSE2017, pp. 233-234, Nov. 2017.

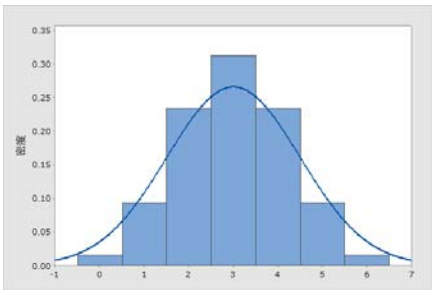
データマイニング技術：歪度・尖度変換



データ分布

歪度，尖度の算出

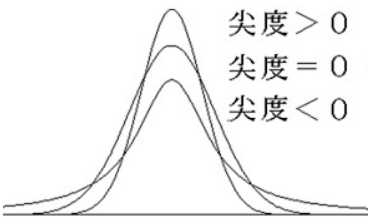




正規分布に近づける

歪度，尖度の変換

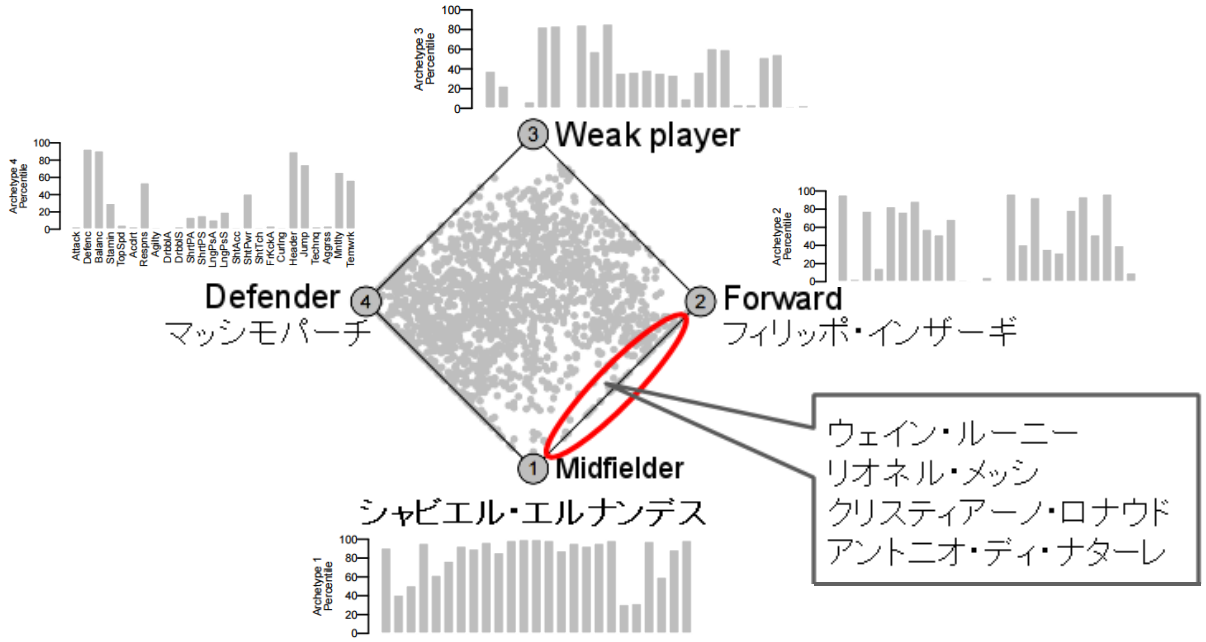
$$H(x;\epsilon,\delta) = \sin h[\delta \sinh^{-1}(x) - \epsilon]$$



*福居誠二, 門田暁人, "尖度と歪度を考慮した予測モデルの検討," WWS2018, Jan. 2018.

データマイニング：原型分析

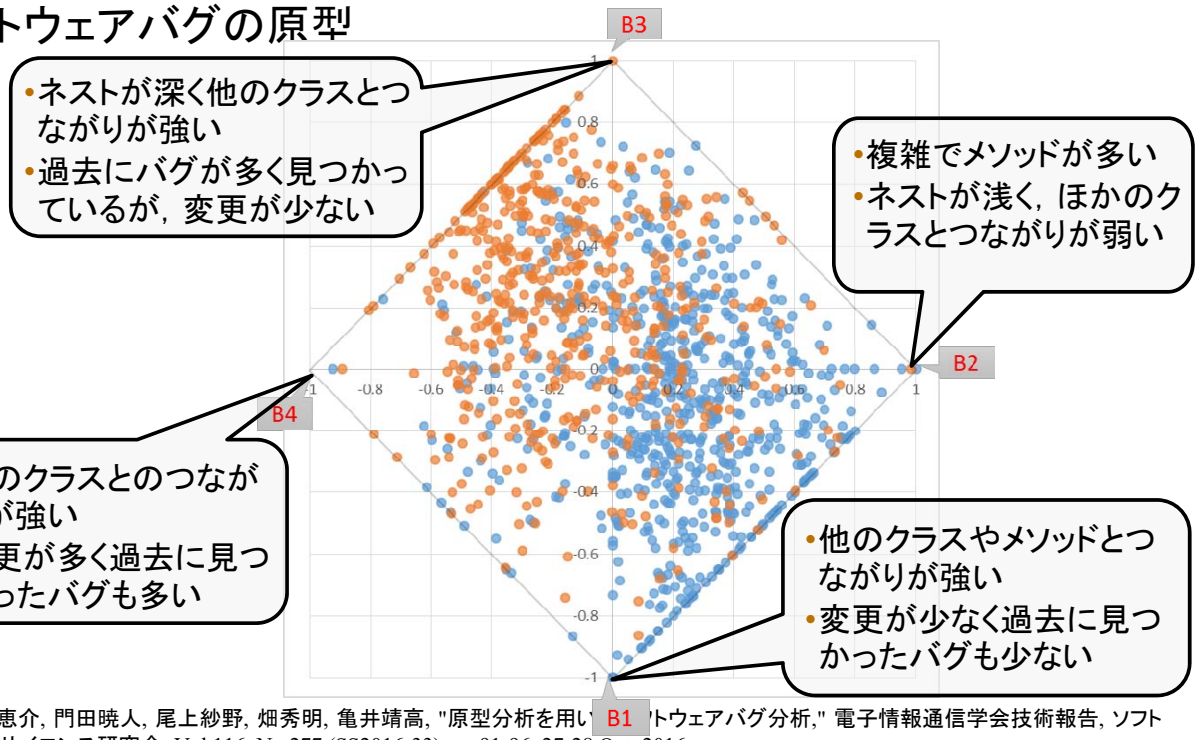
サッカーゲームの選手データの原型（データを代表する極端な値）



*瀧本恵介, 門田暁人, 尾上紗野, 畑秀明, 亀井靖高, "原型分析を用いたソフトウェアバグ分析," 電子情報通信学会技術報告, ソフトウェアサイエンス研究会, Vol.116, No.277 (SS2016-33), pp.91-96, 27-28 Oct. 2016.

データマイニング：原型分析

ソフトウェアバグの原型



*瀧本恵介, 門田暁人, 尾上紗野, 畑秀明, 亀井靖高, "原型分析を用いたソフトウェアバグ分析," 電子情報通信学会技術報告, ソフトウェアサイエンス研究会, Vol.116, No.277 (SS2016-33), pp.91-96, 27-28 Oct. 2016.

ソフトウェア保護：プログラム難読化

■ セキュリティに関わるソフトウェアには，様々な保護が必要となる．

- アクセス制御
- 改ざんの検出
- 暗号鍵の隠ぺい
- アルゴリズムの隠ぺい
- ・・・・

ソフトウェア保護技術の開発とその
セキュリティ分析を行う．

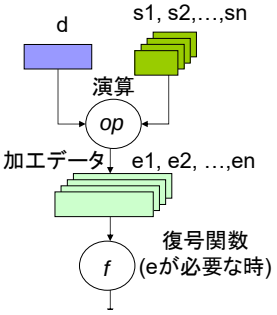
```
int func(int c)
{
    return c -'A'+'a';
}
```

元のプログラム

```
int func(int c){int c1, c2,
x1=6, x2=13; c1=(c+x1)%
13; c2=(c+x2)%19; return
decode(c1, c2);}
int decode(int a, int b){ int
x=(a+13*(b-a)*3)%247;
if(x<0) x+= 247; return x;}
```

難読化後のプログラム

プログラムの難読化



秘密分散による暗号鍵の隠ぺい

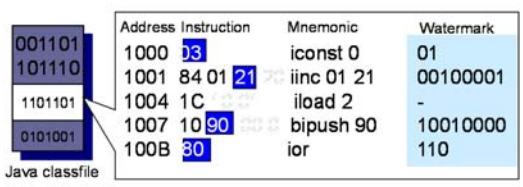
*Yuichiro Kanzaki, Clark Thomborson, Akito Monden and Christian Collberg, "Pinpointing and Hiding Surprising Fragments in an Obfuscated Program," Proc. 5th Program Protection and Reverse Engineering Workshop (PPREW-5), Article No. 8, Dec. 2015.

ソフトウェア保護：盗用の抑止

■ オープンソースソフトウェアの普及に伴い，著作権やライセンスに関する係争が多発している．

- Mac OS Xエミュレータ Cherry OSが，Pear PCのソースコードを盗用
- プロジー社のDivxコンバータが，DeCSS，DVD2avi等のソースコードを盗用
- Sigma Design社のReal Magic MPEG-4が，XVIDのソースコードを盗用
- ・・・

ライセンス保護・違反検出を行う．



プログラムに対する電子透かし法によるライセンス表記の埋め込み

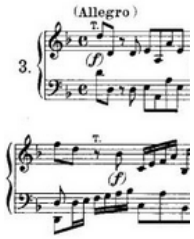
ソフトウェアの指紋（バースマーク）による盗用の検出

*Takehiro Tsuzaki, Teruaki Yamamoto, Haruaki Tamada, Akito Monden, "Scaling up software birthmarks using fuzzy hashing," International Journal of Software Innovation, Vol.5, Issue 3, pp.89-102, July 2017.

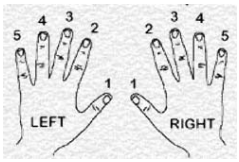
その他の研究テーマ

■ データ計測・分析ができるものであればなんでも.

- 自動デバッグ
  - AIによるソフトウェアバグの自動修正
- 音楽データの分析
  - ピアノ演奏の難易度推定, 類似する楽曲の推薦



楽譜



運指の推定

運指の難易度の推定

- ソフトウェア開発者育成ゲーム

2017年度の卒業研究

■ 計算機上のソフトウェア開発作業の自動推定
■ GitHub上のプログラマのプロファイリング
■ オンラインジャッジ参加者のプログラミング作法の分析
■ N-gram IDFによるバグレポートの自動分類
■ 二変数関数を扱うアソシエーションルール
■ ソフトウェアバグの行レベル予測

2016年の卒業研究

- ソフトウェアエンジニアに求められる技術の求人票に基づく分析
- ソフトウェア開発の提案依頼書における無理難題の分析
- 原型分析を用いたソフトウェアバグ分析
- プロジェクト間バグ予測方法の実験的評価
- バイナリコード中の文字列に着目したソフトウェアの流用検出
- ソフトウェア開発工数の二段階予測方法

2015年の卒業研究

- データの匿名化
- プロジェクト管理者のスキルの分析
- アソシエーションルールマイニングによるバグ分析
- ソフトウェア開発工数予測