住友電工情報システム

SPI Japan 2019
「つなげる」　～共通性を見つけて取り組みを体系化しよう！～

データ分析技術を活用したプログラム開発の生産性向上

住友電工情報システム株式会社
QCD改善推進部　生産技術グループ　豊田　裕智

2019年10月10日

SUMITOMO ELECTRIC GROUP

住友電工情報システム株式会社　概要

- 設　立　　：　1998年10月1日
- 資　本　金　：　4.8億円
  - 住友電気工業株式会社　：　60%
  - 住友電装株式会社　　：　40%
- 従　業　員　：　500名
- 代　表　者　：　代表取締役社長　奈良橋　三郎

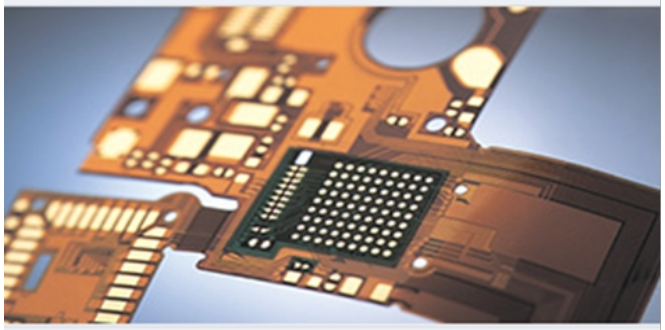
- 事業内容
  - パッケージソフトの開発、販売
  - 情報処理システムの開発受託
  - コンピュータ運用業務の受託
  - 情報機器の販売
- URL
  - https://www.sei-info.co.jp/

住友電気工業(親会社)の製品

自動車

情報通信

システム

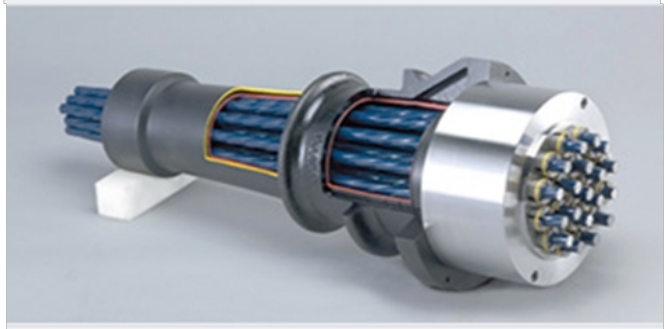


エレクトロニクス

半導体・デバイス

エネルギー

環境・エコロジー



インフラ設備

工業用製品・材料

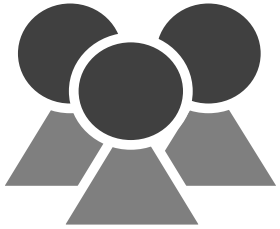
アジェンダ

1. 背景

1.1. 組織目標

システム開発の生産性向上

Working Group 活動は活発



向上つながる確実な施策は見つからず・・・

革新的な施策が必要

1.2. ブレインストーミング

生産性が悪くなる要因は？

- プロセス起因
  - テストを こまめにしない
  - テストが すべて自動
  - テストデータ作成に時間がかかる
- 成果物起因
  - プログラムが 複雑
  - 仕様書の 品質が悪い
  - コードが汚い 標準無視
- 人起因
  - ショートカットを 使わない
  - 入力ミスが 多い

1.3. 要因の決定分析

データ量が多く、分析しやすい項目を選定

	テストをこまめにしない	テスト全自動	データ作成	PG複雑	仕様書品質悪	標準違反	ショートカット未使用	タイプミス多
効果性	◎	◎	○	○	○	△	△	△
容易性	△	△	△	◎	○	△	×	×
新規性	◎	△	○	△	△	△	○	○
グループ方針	○	○	○	○	○	◎	×	×
測定しているデータ量	◎	○	○	◎	○	△	△	△
合計	19	13	13	17	13	9	5	5

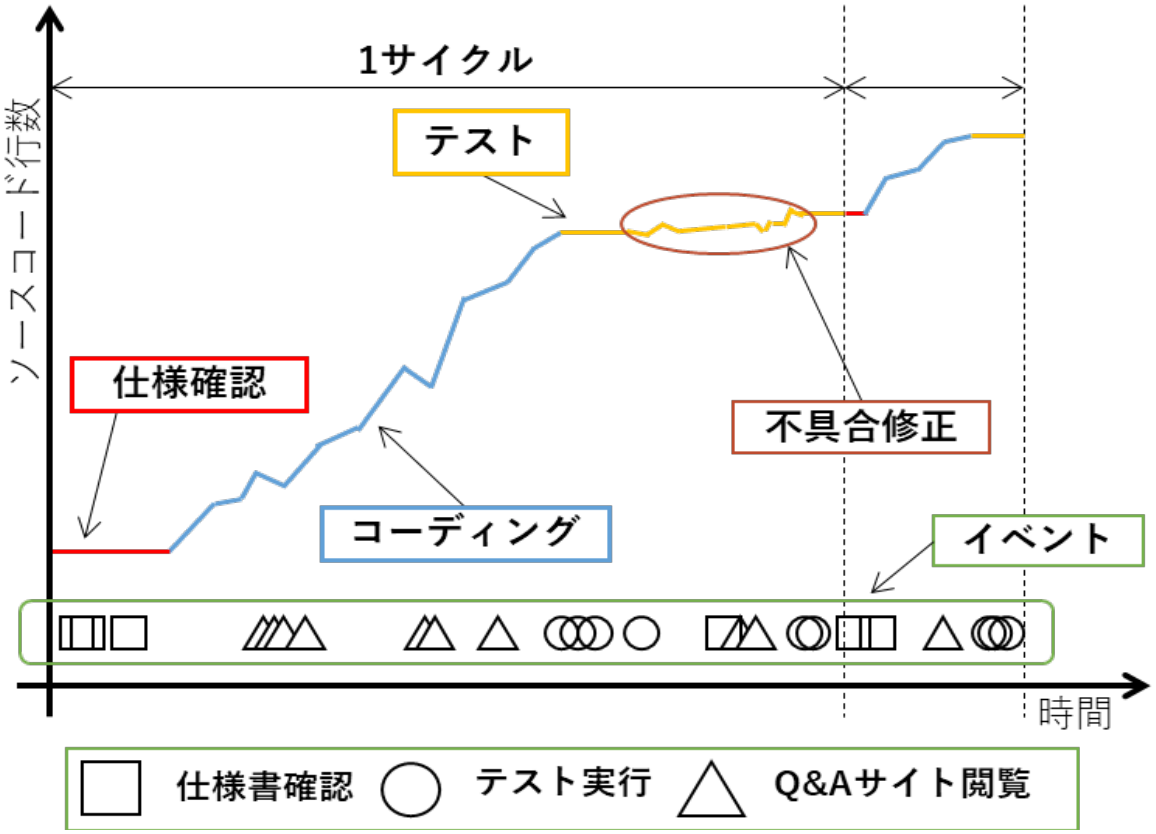
テストをこまめにしない とプログラムが複雑に注目

2. サイクルタイム低減による
生産性の向上

2.1. サイクルタイムとは

仕様理解 → コーディング → テスト(不具合修正)

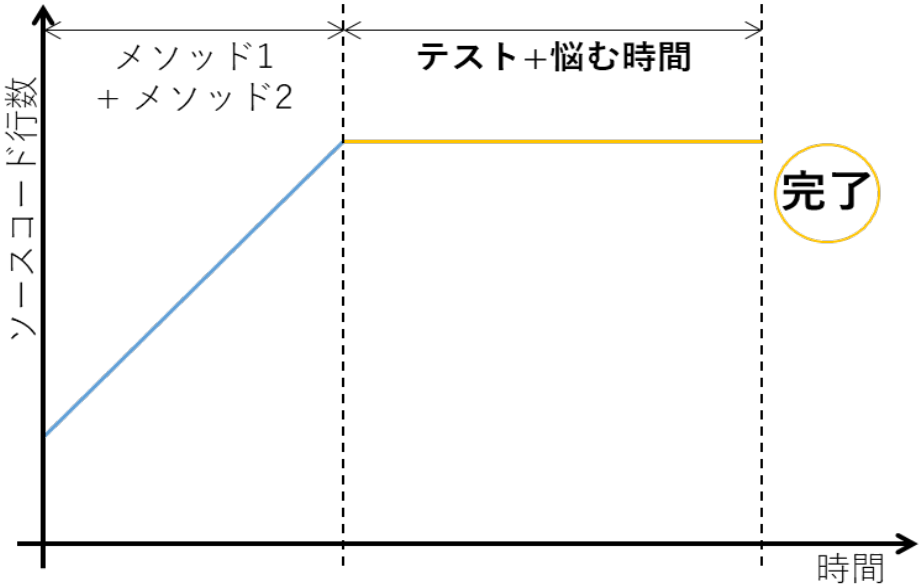
にかかる時間のこと



2.2. 仮説

サイクルタイムが長い人は生産性が低い

生産性が低い人

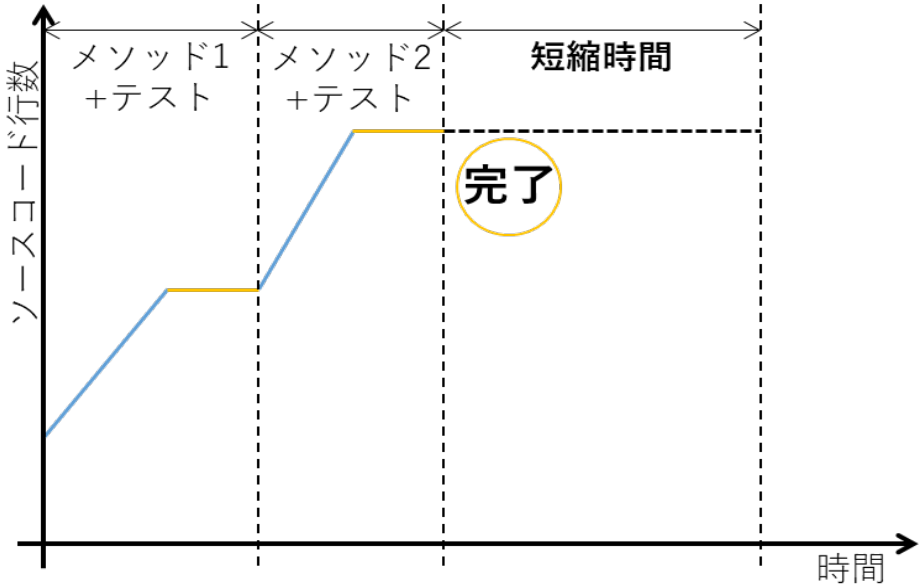


1回に書く量が多い

調査範囲が広い

デバッグに時間がかかる

生産性が高い人



1回に書く量が少ない

調査範囲が狭い

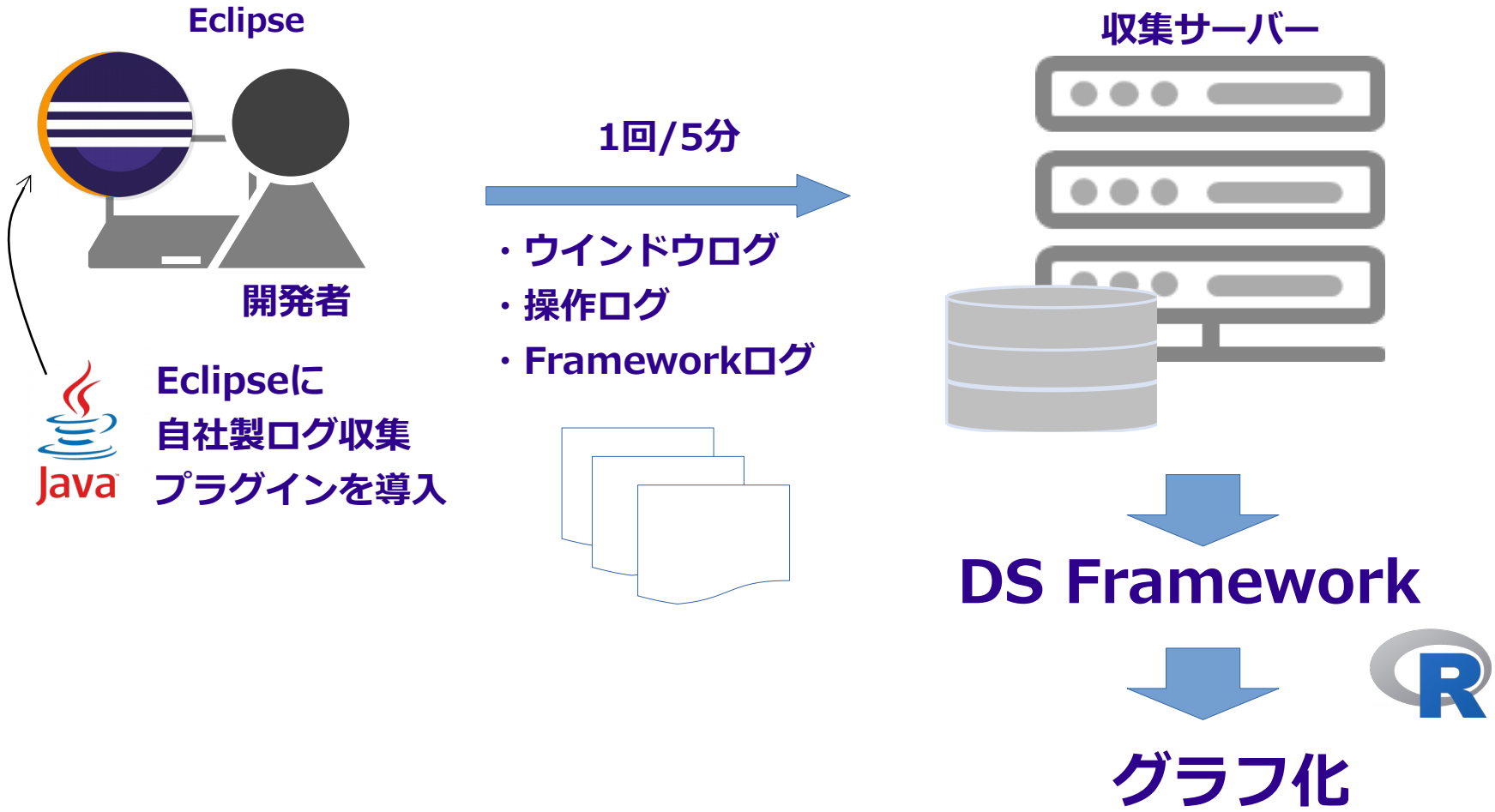
効率的にデバッグできる

サイクルタイムと生産性の関係を分析するために

開発者の PC操作ログ(行動) を活用する

2.3. 操作ログデータ収集方法

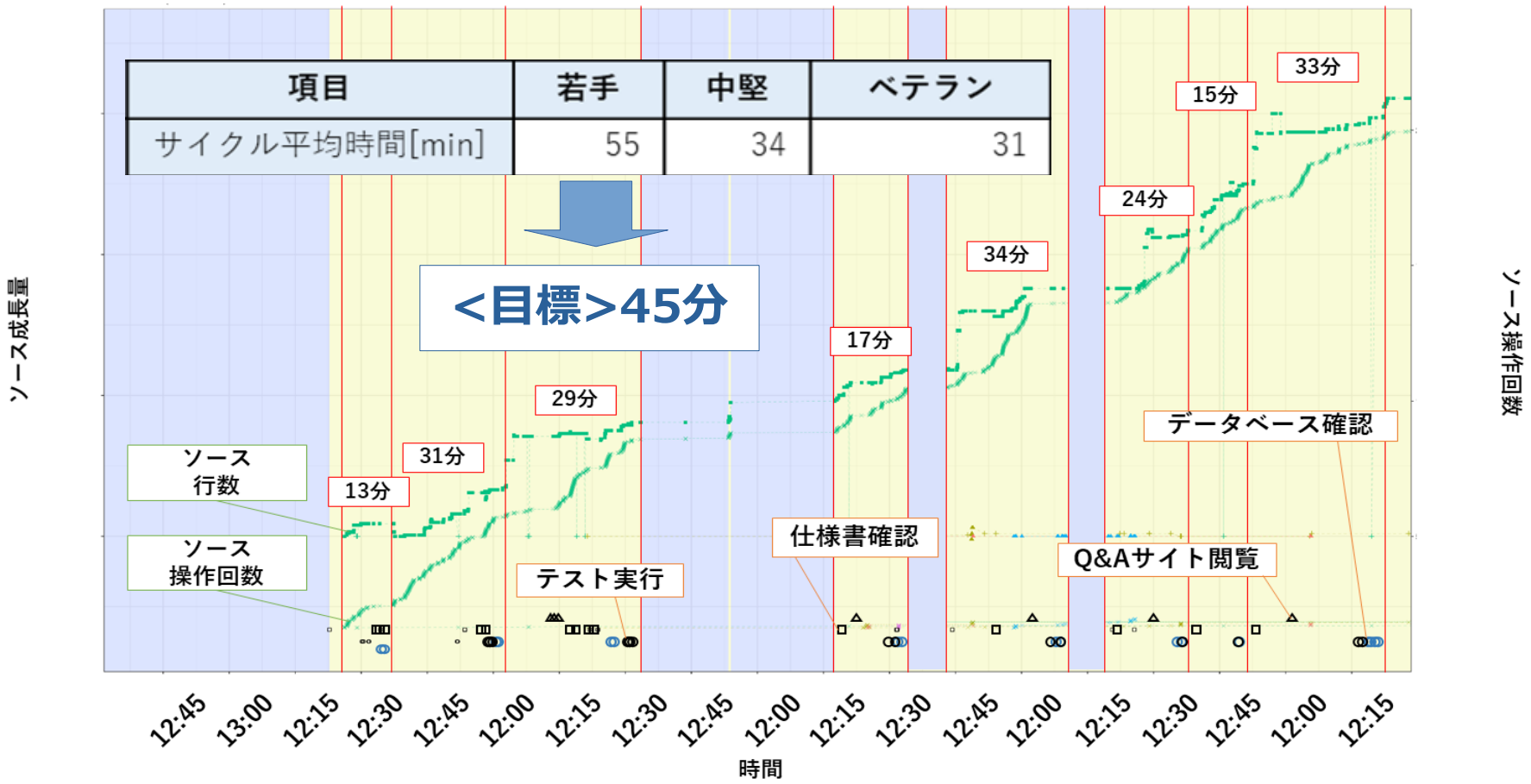
自社製のプラグインと DS Framework* を活用



*改善活動効率化を目的としたOSS利用によるデータ収集・分析の自動化 野尻優輝(SPI Japan 2018)

2.4. データ分析結果

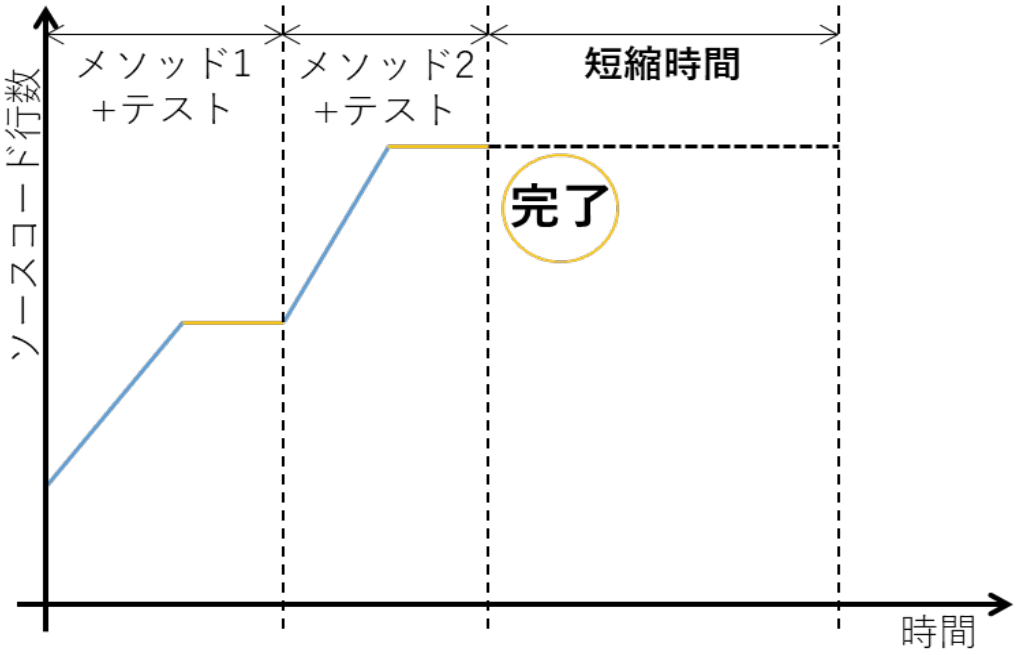
サイクルタイムが短い人は生産性が高い傾向が得られた



2.5. 施策1 メソッド単位でのコーディング

メソッド単位でコーディングとテストを実施する

生産性が高い人



1回に書く量が少ない

↓

調査範囲が狭い

↓

効率的にデバッグできる

課題

規模が大きいメソッドは1回に書く量が多くなる

(途中でテストすることができない)

2.5. 施策2 スタブを用いたテスト

スタブを用いて途中でテストできるようにする

スタブなし

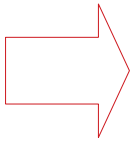
```
main(a){
   int i,j = 0;
   処理A
   処理B
   return 0;
}
```

処理Aを作り終わらないと
処理Bをテストすることができない

スタブを用いて処理を分割することで

1回に書く量を減らし、

調査範囲を狭くする



サイクルタイムの短縮

スタブなし

```
main(a){
   int i,j = 0;

   stab_a(i,j);
   stab_b(i,j);
   return 0;
}
stab_a(int i, int j){
   float n = 0;

   return n;
}
stab_b(int i, int j){
   float m = 0;

   return m;
}
```

2.6. 試行結果

2つの施策により悩んでいる時間は減少

ただし、関数やコマンドを調べている時間が我々の想定よりも多く、目標としては未達

当日のみ

3.McCabe低減による
生産性向上

3.1. McCabe とは

メソッド単位の複雑度を表す指標

循環複雑度(Cyclomatic complexity number：CCN)のこと

以下のプログラムのMcCabeはforが1つ、ifが2つなので「4」になる

```
for(i = 0; i <= 5; i++){
    if(i == 0){
        printf("i は 0 です");
    }
    if(i == 5){
        printf("最後のループです");
    }
}
```

今回はMcCabe測定ツールの

Lizard*を用いてソースを測定

*https://github.com/terryyin/lizard

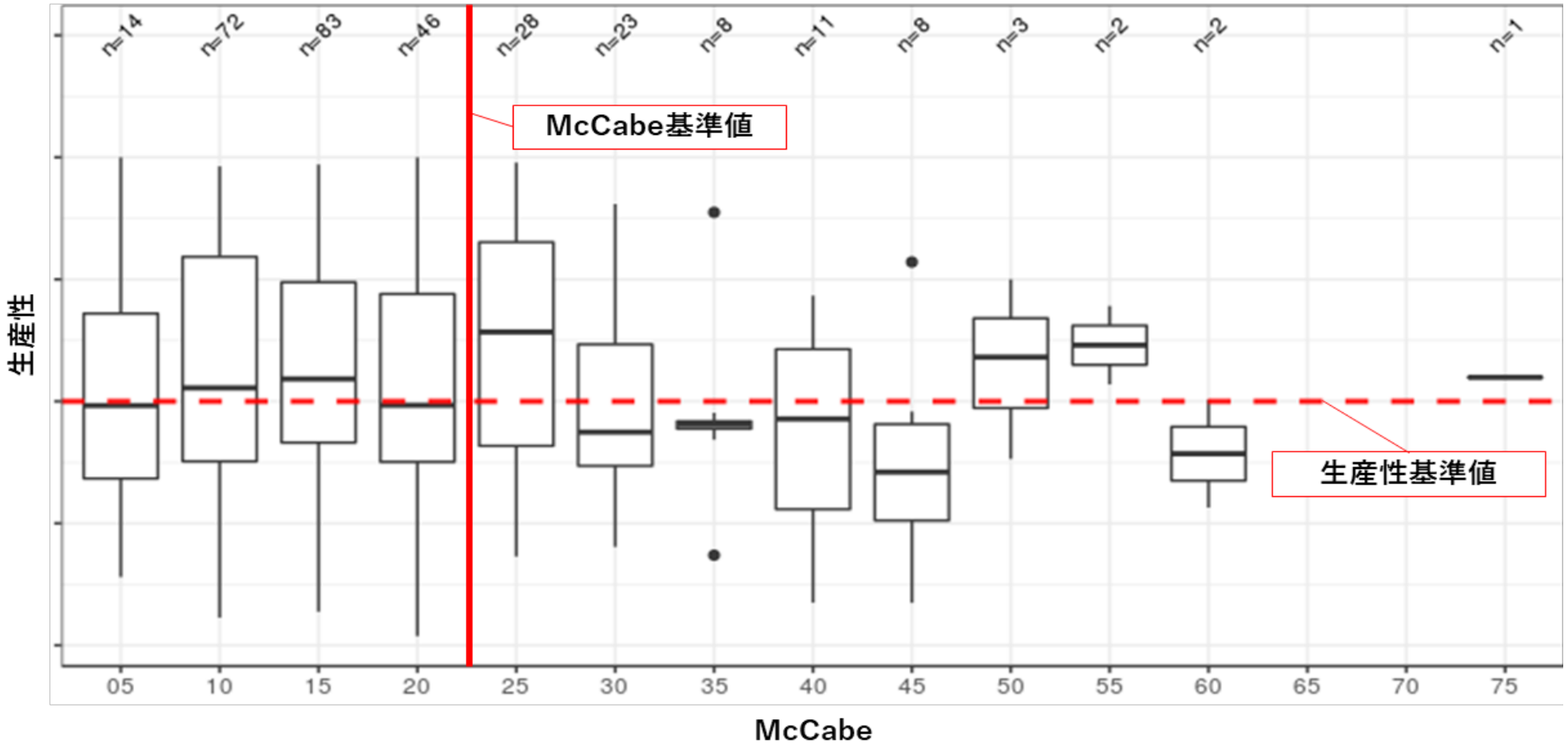
3.2. 仮説

プログラムが複雑になってしまっている

仮説：　McCabeが高いプログラムは生産性が低い

3.3. データ分析結果

McCabeが基準値を超えるプログラムは生産性が低かった



3.4. 施策1 プログラムの簡素化

プログラムの簡素化

今回は一例としてディシジョンテーブルを活用することで

複雑な条件分岐をシンプルに実装する方法を紹介

```
main(){
 if(A1 && !A2 ・・・){
  return O1;
 }
 if(A2 && !A3 ・・・){
  return O2;
 }
 if(A2 && !A4 ・・・){
   return O3;
 }
 if(A3 && !A3 ・・・){
  retrun O4;
 (以下略)
}
```

複雑な条件分岐

A1	A2	A3	A4	結果
T	T	T	T	O1
T	T	T	F	O2
T	T	F	T	O3
T	T	F	F	O2
T	F	T	T	O3
		・		
		・		
F	F	F	F	O1

ディシジョンテーブルで
条件を整理

```
main(){
 if(A1){
  return O1;
 }else if(A1 && A2){
  return O2;
 }else if(A1 && A3){
   return O3;
 }else{
   return O4;
 }
```

コードをシンプル化

3.4. 施策2 McCabeが高いプログラムの通知

課題：肌感覚で McCabe 20 超を認識

施策：超過時に通知をすることで感覚を養う

Digital Assistant @assistant Admin Owner 5:36

all

昨日に更新されたプログラムに McCabe や MLoC が高いものが見つかりました。
プログラム保守の際にデグレードが発生する可能性があるため、修正をお願いします。
(閾値：McCabe 20 以上、MLoC 100 以上)

filename	method	McCabe	MLoC	updateTime
		5	108	2019-09-03 13:43:16
		36	129	2019-09-03 19:48:10
		2	158	2019-09-03 19:48:10
		8	102	2019-09-03 13:44:31
		8	133	2019-09-03 13:44:31
		21	51	2019-09-03 18:52:27
		6	130	2019-09-03 21:00:03
		4	125	2019-09-03 19:48:10
		19	125	2019-09-03 18:52:27

3.5. シミュレーション結果

施策によって複雑なプログラムを簡素化することができ、
生産性を向上させる可能性を示すことができた

<適用前>

McCabe	96
メソッド行数 (論理行数)	176

<適用後>

McCabe	8
メソッド行数 (論理行数)	25

(※処理内容：4つのフラグから装置の稼働状況をチェックする)

3.6. 試行結果

対象プロジェクトはMcCabeの基準値超えが約2%
全社のプログラムは約11%

	対象プロジェクト	全社
全数	**約100**	約80,000
基準値超え	**2**	約8,500
割合	**2%**	約11%

4.まとめ

4.1. 成果

基準値に対して1.13倍の生産性

<施策>

- 1回に書く量を減らしてサイクルタイムを短縮
- コードの簡素化や通知によってMcCabeを低減

4.2. さいごに

データ収集・分析によって
従来とは異なる視点でQCD向上に貢献

<成果>

- **ベテランと初心者の違いの可視化**
- **定量的な評価**

<今後の展望>

- **リアルタイム測定、リアルタイムアドバイス**

住友電工情報システム

https://www.sei-info.co.jp

SUMITOMO
ELECTRIC
GROUP

参考文献

[1] 原田大輔(2016). ビデオ撮影及び画面キャプチャによる分析プロセスの改善. SPI Japan 2016.

[2] 野尻優輝(2018). OSSを利用したデータ収集・分析の自動化による改善活動の効率化. SPI Japan 2018.