SYNOPSYS®
Silicon to Software™

見た聞いた！
残念なソフトウェア開発現場と素敵な開発現場

2020/11/17

日本シノプシス合同会社
ソフトウェア インテグリティ グループ
シニアセールスエンジニア
吉井雅人

残念な
素敵な
開発現場

1 小手調べ

2 残念な開発現場環境と迷言

3 日本にもある！素敵な開発現場

4 まとめ

1 小手調べ

SYNOPSYS®
Silicon to Software™

残念な
素敵な 開発現場

バグはどこにあるでしょう?

```
#include <stdio.h>

int simple_example(int c) {
  void *p = malloc(10);
  if(c)
    return -1;
  /* ... */
  free(p);
  return 0;
}
```

```
1 #include <stdlib.h>
2 int simple_example(int c) {
```

1. **alloc_fn:** アロケーション関数 malloc が領域を返しています。
2. **var_assign:** 代入: p = malloc(10UL) から返された領域。

```
3         void *p = malloc(10);
```

3. 条件 c は true となりました。

```
4         if(c)
```

CID 29856 (#1/1): リソースリーク (RESOURCE_LEAK)

4. **leaked_storage:** 変数 p がスコープを外れるため、リソースがリークしています。

```
5                 return -1;
6         /* ... */
7         free(p);
8         return 0;
9 }
10
```

バグはどこにあるでしょう?

```
#include <stdio.h>

void print(int srcId, int dstId){
    printf("source Id is %d. ¥n", srcId);
    printf("destination Id is %d. ¥n", dstId);
}

void test() {
    int srcId = 1;
    int dstId = 2;
    print(dstId, srcId);
}
```

```c
#include <stdio.h>

void print(int srcId, int dstId){
    printf("source Id is %d. \n", srcId);
    printf("destination Id is %d. \n", dstId);
}

void test() {
    int srcId = 1;
    int dstId = 2;
    print(dstId, srcId);
}
```

CID 30260 (#1/1): 引数の順序が間違っています (SWAPPED_ARGUMENTS)

swapped_arguments: print の呼出しの引数の位置がパラメータ

- dstId が srcId に渡されています
- srcId が dstId に渡されています

の順序にマッチしていません。 [詳細の表示]

バグはどこにあるでしょう?

```
int square(int x) {
    return x*x;
}

int example(int a, int b, int x, int y) {
    int result = 0;
    if (a > 0) {
        result = square(a) + square(x);
    }
    if (b > 0) {
        result = square(a) + square(y);
    }
    return result;
}
```

```
1  int square(int x) {
2      return x*x;
3  }
4
5  int example(int a, int b, int x, int y) {
6      int result = 0;
7      if (a > 0) {
```

original: square(a) はコピー元のように見えます。

```
8          result = square(a) + square(x);
9      }
10     if (b > 0) {
```

CID 30261 (#1/1): コピーペースト エラー (COPY_PASTE_ERROR)

copy_paste_error: square(a) 内の a はコピーペーストのミスのように見えます。

b にすべきではありませんか？

```
11         result = square(a) + square(y);
12     }
13     return result;
14 }
```

現在のお仕事

- ソフトウェアの品質向上のためのソリューションを提供

良い点

- 自社の開発環境だけではなく様々な環境を知ることができる
- どの企業がどういった手法でソフトウェア開発をしているのかを把握できる

これまで遭遇した様々な開発環境、迷言を紹介します

1 小手調べ

2 残念な開発現場環境と迷言

残念な
素敵な
開発現場

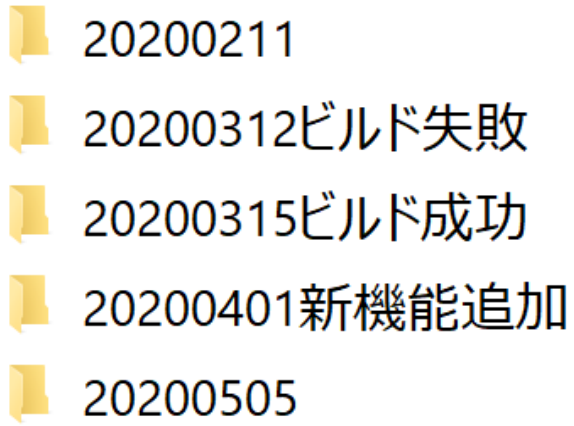
残念な現場 その1

ソースコードの管理は
yyyymmddフォルダで管理!!

残念な現場 その1 解説

- ソースコード管理システム(git、svnなど)の存在を知らなかった
- 以前からこのやり方でやってきて疑問を持たなかった

※イメージ図



残念な現場 その2

20年間動いているから
修正しなくてもOK!!

残念な現場 その2 解説

- 以下のような不具合を大量に検出
  - –リソースリーク
  - –バッファオーバラン
  - –など35種類のバグ

迷言

「20年間出荷し続けているので、不具合が顕在化しない限りはバグではない」

※不具合修正は都度実施している

残念な現場 その3

再起動すれば解決する問題は
バグじゃありません～

残念な現場 その3 解説

- 以下のような不具合が大量に検出
  - –リソースリーク(含むメモリリーク)
  - –バッファオーバラン
  - –二重解放
  - –Null参照

迷言

「再起動すれば解決する問題はバグではないため、メモリリークは問題にはならない」

「不具合は修正せずに、次の型番をリリース」

残念な現場 その4

どうせ工程遅延するなら
最初から工程表を2つ作る!!

残念な現場 その4 解説

- 上役・顧客に見せる用の表線表と実際の裏線表が存在する
- 表線表ではテスト工程に入っていることになっているが、実際には仕様も決定していない
- ウォータフォールだと問題が顕在化しない
- 当たり前だが現場は追い込まれることになる

1 小手調べ

2 残念な開発現場環境と迷言

残念な
素敵な 開発現場

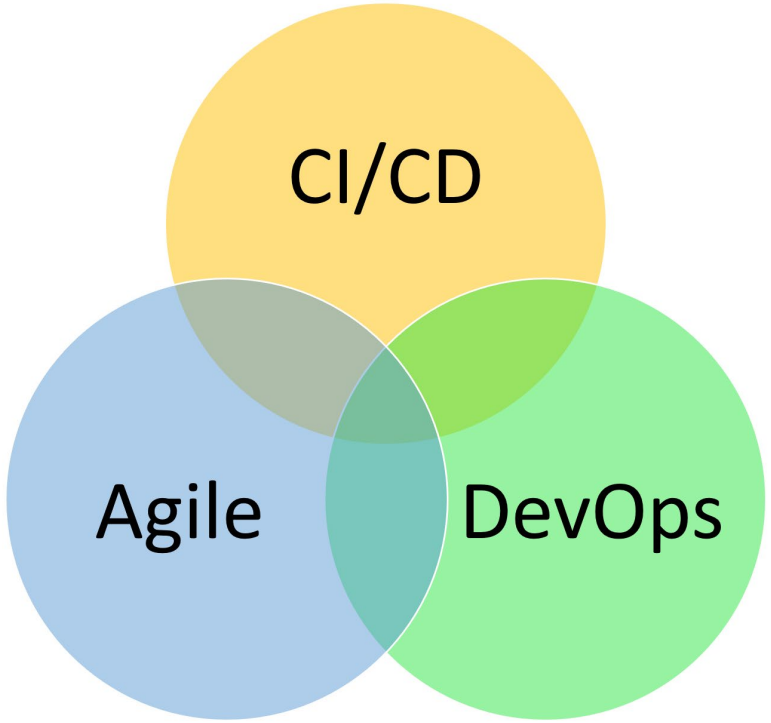
3 日本にもある！素敵な開発現場

素敵な現場 その1

アジャイル、CI/CD、DevSecOps

素敵な現場 その1 解説

- ウォーターフォールではなくアジャイル開発
- CI、自動化は大前提
- バズワードとして扱われがちなDevSec、DevSecOpsを当たり前に実践



素敵な現場 その2

ソリューションを見つけるのに
世界中回ってます

素敵な現場 その2 解説

- 「世界中を回って」開発現場に最適な解をもたらす
- ベンダーに負けない知識
- 常に最新情報を入手
- ベンダーロックインを避ける

素敵な現場 その3

Tシャツとジーパン
(※冬はネルシャツ)

素敵な現場 その3 解説

- 服装のカジュアルさと技術のレベルは正比例

※あくまで個人の見解です

- エンジニアの裁量が大きく、自律的
- バグかそうでないかの判断、修正するしないの判断が明確
- テクニカルな判断とビジネスの判断のバランスが絶妙
- 決断が早く問題の先送りがない

素敵な現場 その4

自社で開発してます

素敵な現場 その4 解説

- 自社に優秀なエンジニアを囲い込む
- エンジニアに対する投資も手厚い
- ベンダーに丸投げしない
- 当然責任も取る
- プロダクトオーナー

残念な
素敵な
開発現場

1 小手調べ

2 残念な開発現場環境と頂いた迷言

3 日本にもある！素敵な開発現場

4 まとめ

まとめ

- 時代も考え方も環境も変化が激しいので、それに対応しましょう
- 残念な環境で働いている人は、素敵な環境を求めて動きましょう
- エンジニアが幸せになれる環境もあります

SYNOPSYS®
Silicon to Software™

Thank You