

一般財団法人 日本科学技術連盟
第36年度ソフトウェア品質管理研究会 成果発表会

リリース後の不具合発生予測モデルに基づく 効果的なプロセス改善への仕掛けの提案

2021年2月26日

演習コースⅡ ソフトウェアメトリクス 実践Ⅰ

研究員：星野 智彦（アイシン精機株式会社）

主査：小池 利和（ヤマハ株式会社）

副主査：小室 睦（株式会社プロセス分析ラボ）

副主査：柏原 一雄（株式会社デンソークリエイト）

はじめに

現状分析

課題

分析結果

考察

まとめ

ソフトウェア開発の理想と現実

でもテストで欠陥が出たら手戻りコストがかかってしまいますよ

レビューで欠陥を除去して品質作り込みしていきましょう

テストで動かした方が振る舞いが見えて分かりやすいんだよね

また言ってるよ...

開発現場

SEPG

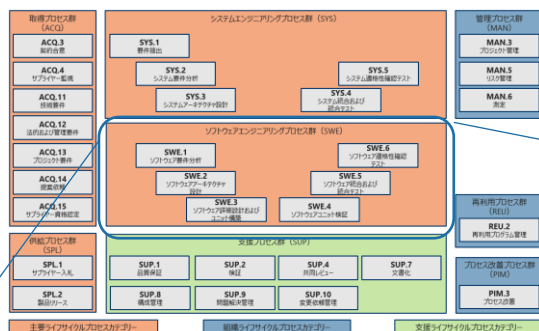
そんなこと分かってるけどさ、レビューじゃ品質状況とか見えないからモチベーション上がらないよ

それじゃあ見えるようにしちゃいましょう!

自組織のソフトウェア開発プロセス

はじめに 現状分析 課題 分析結果 考察 まとめ

Automotive SPICE® プロセス参照モデルをベースにしたV字開発プロセス



上流工程

SWE.1
要求分析

SWE.2
構造設計

SWE.3
詳細設計/実装

SWE.6
適格性確認テスト

SWE.5
統合テスト

SWE.4
単体テスト

自組織が抱える問題①

はじめに

現状分析

課題

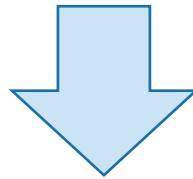
分析結果

考察

まとめ



IPA/SECのデータから
品質目標値を設定



目標値の
根拠は何だ！

ウチの開発
傾向と違う！



開発現場からは
受け入れられない！

自組織が抱える問題②

はじめに

現状分析

課題

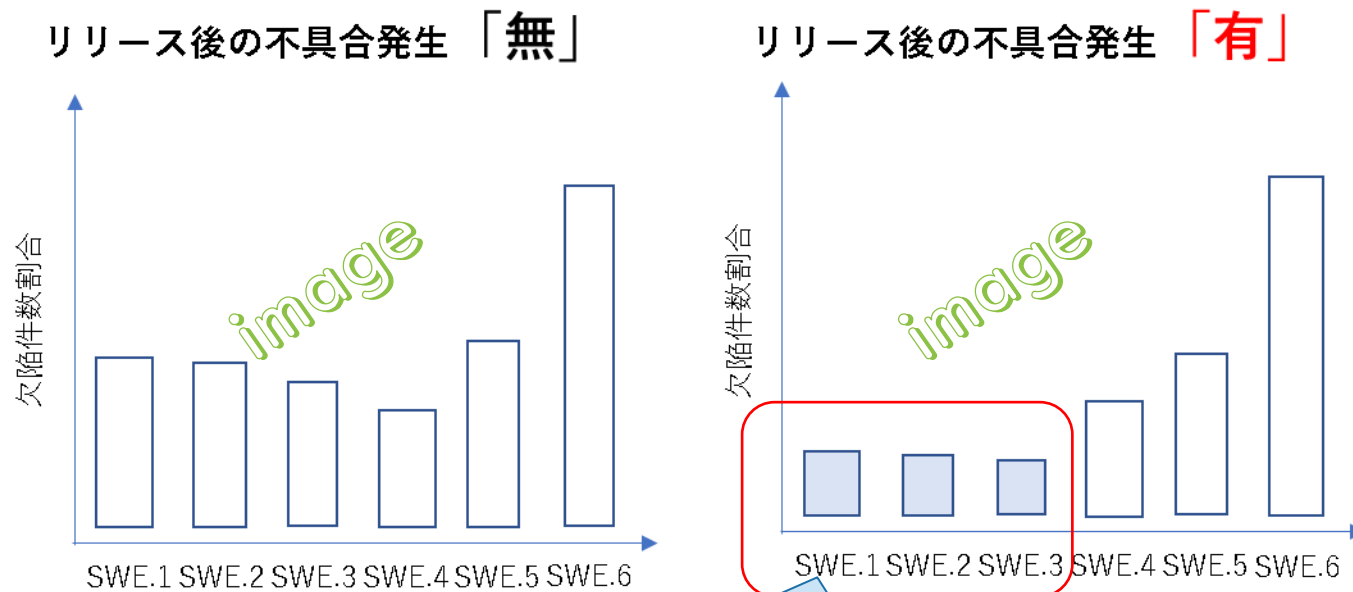
分析結果

考察

まとめ

■工程ごとの欠陥件数の割合を可視化

大人の事情により
イメージ図のみ
掲載します



上流工程での欠陥件数の割合が低い傾向

解決したい課題

はじめに

現状分析

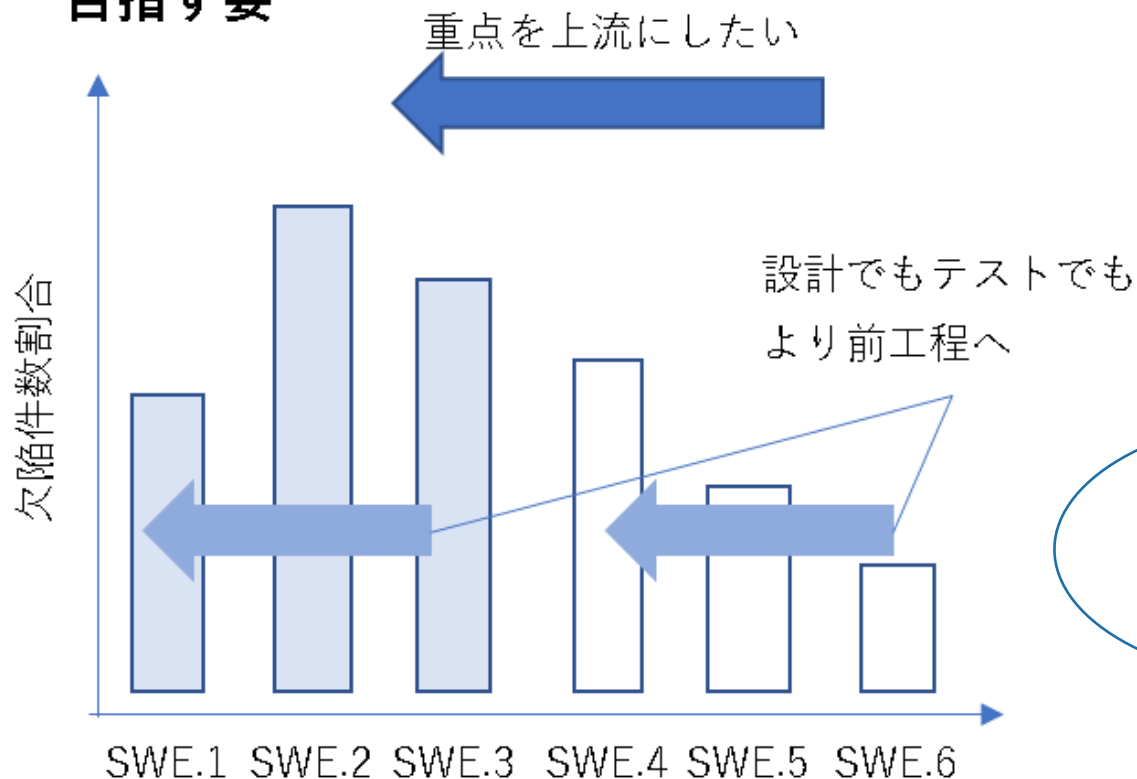
課題

分析結果

考察

まとめ

目指す姿



現場への
動機付け



課題

「自組織のデータによる分析結果」を使って
より上流での品質作り込みを開発現場に意識付ける

はじめに

現状分析

課題

分析結果

考察

まとめ

■分析手法

1. t検定
2. ロジスティック回帰分析

■結果を表すメトリクス(目的変数)

- ・ リリース後の不具合発生「有無」

■分析対象のメトリクス(説明変数)

4つのカテゴリ

- ・ 工数比率
- ・ 前倒し率
- ・ レビュー欠陥
- ・ テスト欠陥



各工程での品質作り込みが、リリース後の不具合発生にどのように影響するか

はじめに | 現状分析 | 課題 | 分析結果 | 考察 | まとめ

■t検定

リリース後の不具合発生「無」(A群)と「有」(B群)の平均の差



A:リリース後の不具合「無」の平均値

B:リリース後の不具合「有」の平均値

	p値	比較	5%有意差
テスト工数比率	0.667	A>B	無し
レビュー工数比率	0.088	A>B	無し
レビュー工数_テスト工数比率	0.449	A>B	無し
テスト及びレビュー工数比率	—	—	—
設計_コーディング工数比率	0.938	A>B	無し
設計レビュー工数_設計工数比率	0.007	A>B	有り
コードレビュー_コーディング工数比率	0.552	A>B	無し
前倒し率	0.985	A>B	無し
レビュー欠陥密度_SWE1	0.236	A>B	無し
レビュー欠陥密度_SWE2 (構造設計)	0.038	A>B	有り
レビュー欠陥密度_SWE3 (詳細設計/実装)	0.043	A>B	有り
テスト欠陥密度_SWE4	0.194	A>B	無し
テスト欠陥密度_SWE5 (統合テスト)	0.031	A>B	有り
テスト欠陥密度_SWE6	0.601	A>B	無し

期待通り、より
前工程での
欠陥除去が
有効

リリース後の不具合発生に影響するメトリクスを特定できた

分析～ロジスティック回帰分析とは

はじめに

現状分析

課題

分析結果

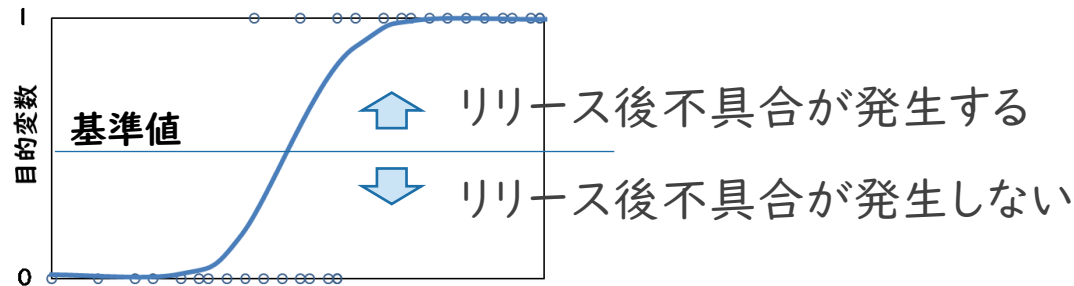
考察

まとめ

■特徴

- 目的変数が二値(無し(0) / 有り(1))の回帰分析

これにより
リリース後不具合の
発生確率を算出



- 次の回帰式の最適な β_i を求める

$$f(X) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n))} \quad (i = 0, 1, 2, \dots, n)$$

■最終結果(後述)

X: 説明変数(分析対象のメトリクス)

$\beta_i > 0$: Xが↑ → 発生確率↑

$\beta_i < 0$: Xが↑ → 発生確率↓

$$f(X) = \frac{1}{1 + \exp(-(-2.86) + 11.53 \times \text{テスト工数比率} + (-11.52) \times \text{レビュー欠陥比_SWE2} + (-5.33) \times \text{テスト欠陥比_SWE5})}$$

結果～予備分析

はじめに

現状分析

課題

分析結果

考察

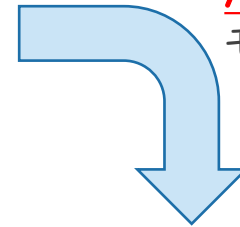
まとめ

以下の4つのメトリクスでロジスティック回帰分析すると・・・

回帰分析に用いたメトリクス

設計レビュー工数_設計工数比率
レビュー欠陥密度_SWE2（構造設計）
レビュー欠陥密度_SWE3（詳細設計/実装）
テスト欠陥密度_SWE5（統合テスト）

ステップワイズ法により
AICの値が最小となった
モデル



説明変数	係数 β	p値	説明変数の値が大きくなると
定数項	-12.6	0.03	—
テスト欠陥密度_SWE5 （統合テスト）	-1.80	0.14	不具合発生確率が低くなる
レビュー欠陥密度_SWE2 （構造設計）	-2.79	0.01	不具合発生確率が低くなる
レビュー欠陥密度_SWE3 （詳細設計/実装）	2.13	0.06	不具合発生確率が高くなる

開発現場への動機付けはしづらい

結果～メトリクス再考

はじめに

現状分析

課題

分析結果

考察

まとめ

■ 詳細設計/実装の欠陥密度が「係数 $\beta > 0$ 」となる要因

説明変数

欠陥密度 $\uparrow$

係数 $\beta > 0$

目的変数

リリース後の
不具合発生確率 $\uparrow$

例えば

難易度の高い開発

これって

説明変数と目的変数の
両方に影響
(すべての工程で効いてくる)

これらの影響を
排除する必要アリ!

はじめに

現状分析

課題

分析結果

考察

まとめ

■解決方法

異なる工程との欠陥比率をとる

例えば
難易度

比率を使えば、分母分子両方に働く要因が相殺される

レビュー
欠陥比率＝

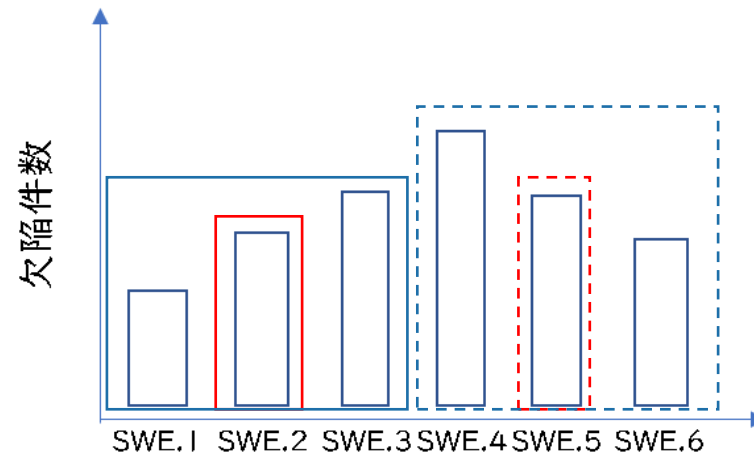
各工程の欠陥件数

レビュー欠陥総数

テスト
欠陥比率＝

各工程の欠陥件数

テスト欠陥総数



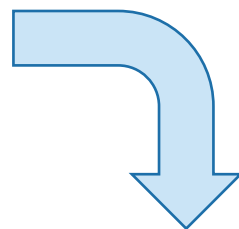
欠陥密度を欠陥比率に変更

結果～ロジスティック回帰分析

はじめに | 現状分析 | 課題 | **分析結果** | 考察 | まとめ

回帰分析に用いたメトリクス

テスト工数比率
レビュー工数比率
レビュー欠陥比率_SWE1(要求分析)
レビュー欠陥比率_SWE2(構造設計)
テスト欠陥比率_SWE4(単体テスト)
テスト欠陥比率_SWE5(統合テスト)



ステップワイズ法により
AICの値が最小となった
モデル

※多重共線性を回避

説明変数	係数 β	p値	説明変数の値が大きくなると
定数項	-2.86	0.33	—
テスト工数比率	11.53	0.18	不具合発生確率が高くなる
レビュー欠陥比率_SWE2 (構造設計)	-11.52	0.06	不具合発生確率が低くなる
テスト欠陥比率_SWE5 (統合テスト)	-5.33	0.19	不具合発生確率が低くなる

$$f(X) = \frac{1}{1 + \exp(-(-2.86) + 11.53 \times \text{テスト工数比率} + (-11.52) \times \text{レビュー欠陥比_SWE2} + (-5.33) \times \text{テスト欠陥比_SWE5})}$$

設計でもテストでもより上流で欠陥を摘出することが必要!!

考察～予測結果の利活用

はじめに

現状分析

課題

分析結果

考察

まとめ

■降水確率が何%なら傘を持っていますか？



考察～予測結果と基準値の設定

はじめに

現状分析

課題

分析結果

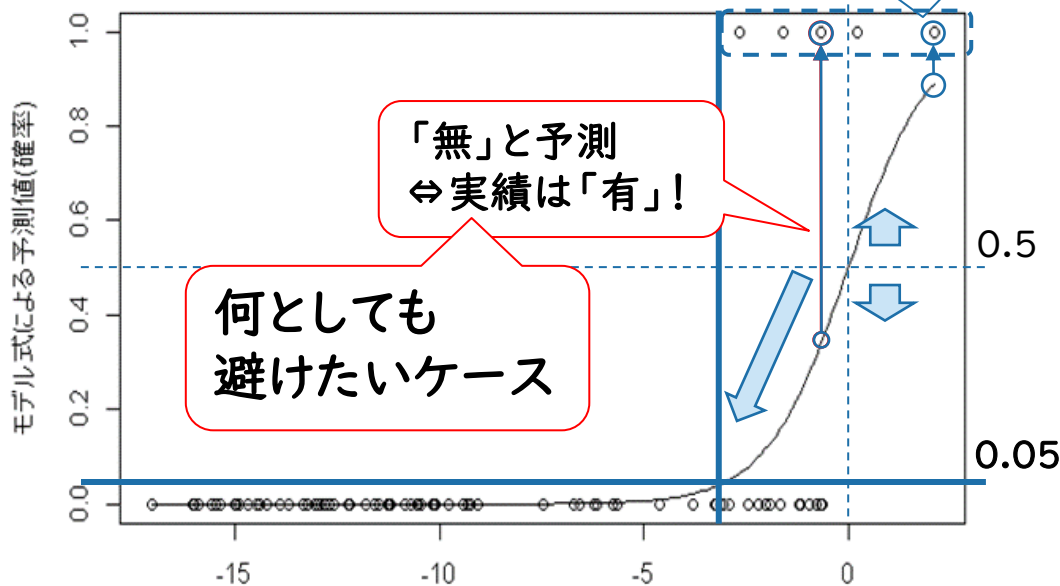
考察

まとめ

■ロジスティック曲線とクロス集計表

基準値:0.5

→基準値:0.05



実際にリリース後不具合発生
「有」であったプロジェクトをリリース後不具合「無」と予測すること、つまり取りこぼすことがなくなる

「有」と予測
→実績も「有」

「無」と予測
⇔実績は「有」!

何としても 避けたいケース

基準値：
予測値=0.5

	リリース後の 不具合「無」 (予測)	リリース後の 不具合「有」 (予測)
リリース後の 不具合「有」 (実績)	3	2
リリース後の 不具合「無」 (実績)	77	0

予測が当たる確率
 $79/82=96.3\%$
 (2+77=79件)

基準値：
予測値=0.05

	リリース後の 不具合「無」 (予測)	リリース後の 不具合「有」 (予測)
リリース後の 不具合「有」 (実績)	0	5
リリース後の 不具合「無」 (実績)	64	13

予測が当たる確率
 $69/82=84.1\%$
 (5+64=69件)
 ⇒精度は下がる

精度よりも
大事な事が
そこにある

考察～予測モデル構築の効果

はじめに

現状分析

課題

分析結果

考察

まとめ

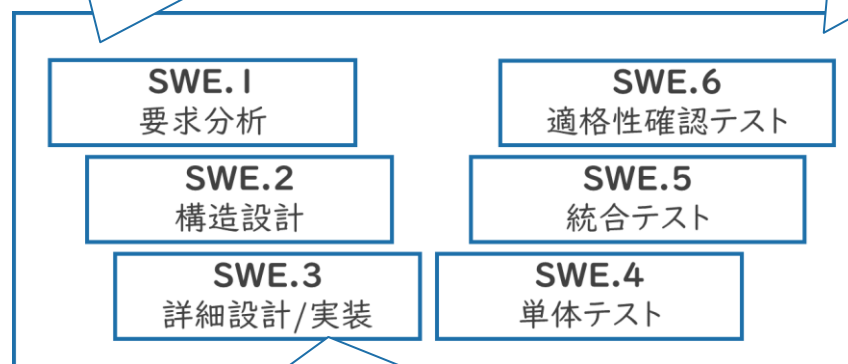
■想定される予測モデルの活用シーン

プロジェクト計画時

品質目標の妥当性や、改善対象のプロセスも見える化

プロジェクト完了後

達成状況を振り返り次へ生かす！



プロジェクト進行中

データを監視しながら、目標値の達成状況を確認

考察～組織的なプロセス改善への効果

はじめに

現状分析

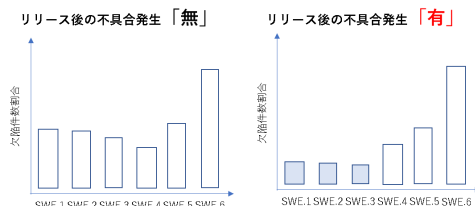
課題

分析結果

考察

まとめ

可視化



課題の明確化



プロジェクト毎に
状況を監視



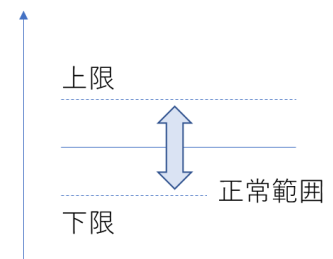
影響する
メトリクスの特特定

説明変数	係数β	p値
定数項	-2.862	0.3301
テスト欠陥比_SWE5 (統合テスト)	-5.328	0.1911
レビュー欠陥比率	11.528	0.1846
レビュー欠陥比_SWE2 (構造設計)	-11.52	0.0621

改善点に
気づきやすくなる



ベースラインの
確立→活用!



はじめに

現状分析

課題

分析結果

考察

まとめ

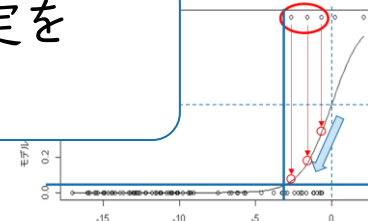
自組織の実績データを使って

リリース後不具合に影響する
メトリクスを特定した！

リリース後不具合発生率	0.088	A>B	無し
リリース後不具合発生率（テスト済み）	0.049	A>B	無し
レビューエラー率	0.038	A>B	無し
レビューエラー率（設計工程）	0.007	A>B	有り
レビューエラー率（コーディング工程）	0.052	A>B	無し
レビューエラー率（テスト工程）	0.085	A>B	無し
レビューエラー率（総計）	0.038	A>B	有り
レビューエラー率（SWE2（アーキ設計））	0.043	A>B	有り
レビューエラー率（SWE3（詳細設計及コーディング））	0.194	A>B	無し
レビューエラー率（SWE4（テスト））	0.031	A>B	有り
レビューエラー率（SWE5（統合テスト））	0.001	A>B	無し

上流での品質作り込み効果を
明確にするメトリクスの再整理
をして

リリース後の不具合「有無」の判定を
可能にするモデルを構築した！



上流での品質作り込みに対する重要性を
客観的、定量的に示すことができるようになった！

はじめに

現状分析

課題

分析結果

考察

まとめ



目標値の
根拠は何だ!

ウチの開発
傾向と違う!

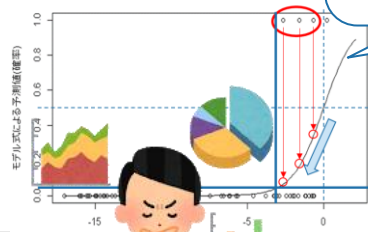


自分たちの
プロジェクトの
傾向も見えそう



実際のプロジェクトで
トライしてみよう!

分析結果を
開発現場に提示



テスト失敗率			
レビュー失敗率			
レビューとテスト失敗率			
テスト失敗率とレビュー失敗率			
設計レビューと設計失敗率	0.007	A>B	有り
レビューと設計失敗率	0.552	A>B	無し
レビューと設計失敗率	0.585	A>B	無し
レビューと設計失敗率	0.236	A>B	無し
レビューと設計失敗率	0.038	A>B	有り
レビューと設計失敗率	0.043	A>B	有り
レビューと設計失敗率	0.194	A>B	無し
レビューと設計失敗率	0.031	A>B	有り
レビューと設計失敗率	0.601	A>B	無し



開発現場と一緒に
改善活動を推進していく

END