Happy Engineerになるために

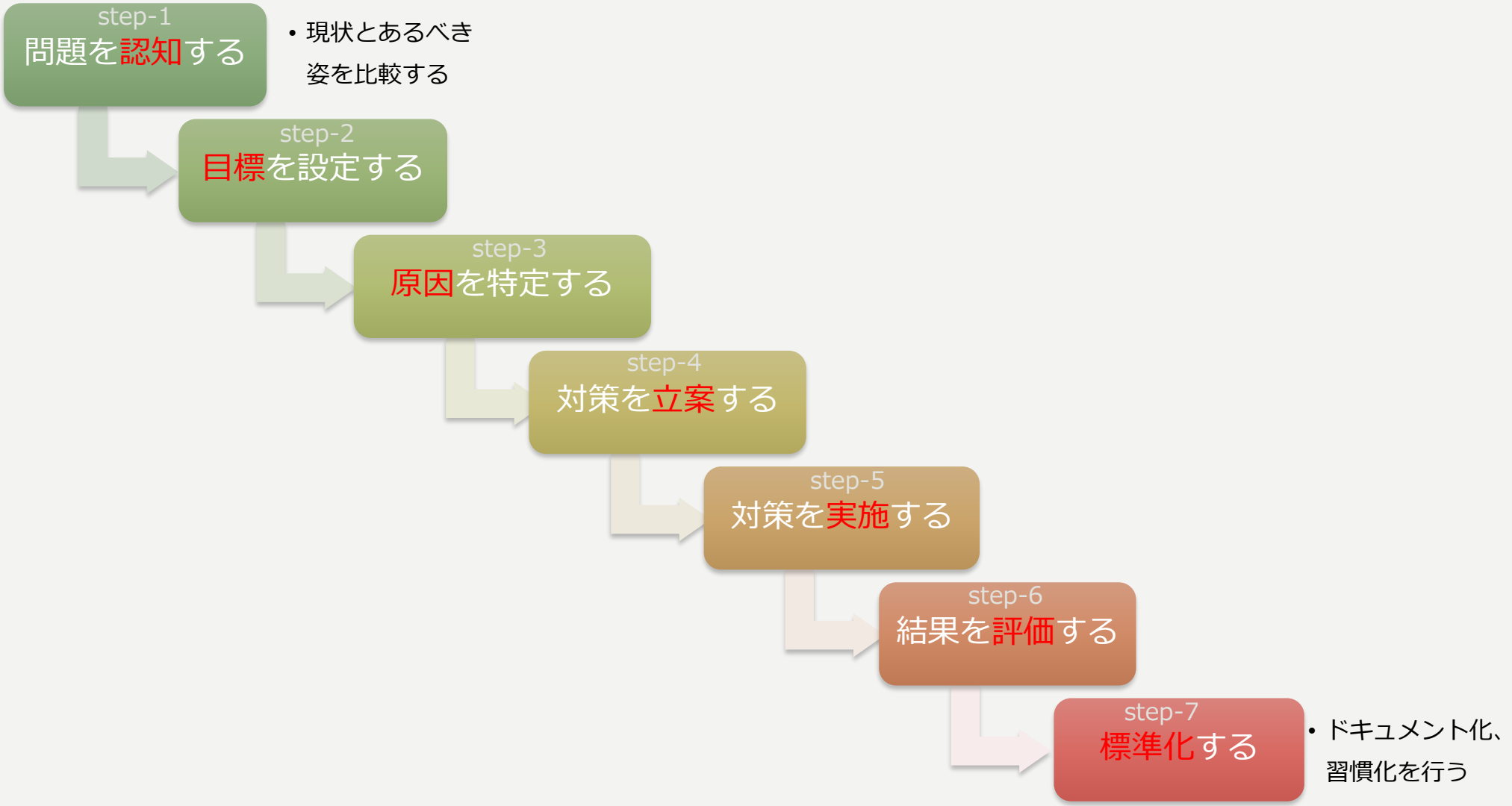
エンジニアのための問題解決手法

ソフトウエア研究会IN秋葉原
池田公平

問題解決しよう!

- 問題解決は「いますぐ」しよう！
- 「忙しいから」という理由で問題解決を先延ばしにしない！
- 効率の悪い仕事は、エンジニアの心と体を疲弊させます！
- エンジニアは「楽」をして仕事を「楽しもう」！
- トヨタ流の問題解決手法を参考に、エンジニア向けにアレンジしてみました。

問題解決のための7つのSTEP



STEP-１　問題を認知する

「問題」とは、あるべき姿と現状とのギャップです。

現状

ギャップ ＝ 問題

あるべき姿

「エンバグが多い」という問題の例

不具合修正を行うと、新たな不具合が発生する

不具合修正に、新たな不具合を発生させるリスクがある。

不具合修正を行うと問題が解決する

STEP-2 目標を設定する

問題解決にあたり、具体的な目標を設定します。

現状

↓ **ギャップ ＝ 改善値**

目標値

「エンバグが多い」という問題の例

平均すると、不具合修正1件につき、不具合件数は0.5件しか減少しない

↓ エンバグ発生率を50%から10%に低下させる。

不具合修正時の平均不具合減少件数を、0.9件以上にする。

STEP-3 原因を特定する

1.問題のカギとなってる要因を探ります

- ✓技術
- ✓フレームワーク・マネジメント
- ✓スキル・適正
- ✓文化、価値観、組織

2.問題発生のメカニズムを探ります

- ➢なぜ発生したか？
- ➢なぜ問題が成立するか？
- ➢関連する要素は何か？
- ➢問題の成立条件を阻害することはできるか？

「エンバグが多い」という問題の例

- ✓ 技術
  - ➢ **デバッグやテストが困難な開発環境**
- ✓ フレームワーク・マネジメント
  - ➢ **テストを実施する手順が確立されてない**
- ✓ スキル・適正
  - ➢ **エンジニアが経験不足**
- ✓ 文化、価値観、組織
  - ➢ **問題解決に消極的な組織にいる**
- ➢ なぜ発生したか？
  - ➢ **単体テストをせずコミットした**
- ➢ なぜ問題が成立するか？
  - ➢ **テストプログラムがない**
- ➢ 関連する要素は何か？
  - ➢ **テストを実施しないでコミットが容認されている**
- ➢ 問題の成立条件を阻害することはできるか？
  - ➢ **テストの義務化**

STEP-4 対策を立案する

対策を立案するにあたり、問題のどの時点に対策を入れるか考えます。
優先順位の高い順に列挙します。

1. **発生源を絶つ！**
   - ✓ 問題の根幹を解決
2. **見逃し要因を絶つ！**
   - ✓ 問題が解決されず放置されている原因を取り除く
3. **拡大源を絶つ！**
   - ✓ 問題が表面化してしまった原因を取り除く

「エンバグが多い」という問題の例

1. 発生源を絶つ！
   - ✓ テストを義務づける
2. 見逃し要因を絶つ！
   - ✓ テストを自動化する
3. 拡大源を絶つ！
   - ✓ テストを通過していないものはコミットさせない

STEP-5　対策を実施する

対策の実施は、タイミングや周囲の理解を得ることが大切です。特に、仕事の手順を変える事に抵抗感を持つ人はすくなくありません。以下の事の周知を徹底する必要があります。

1. Why（目的・狙い）・・・何のために
2. What（課題）・・・何をどの程度
3. Where（対象範囲）・・・どこを対象に
4. How（手段）・・・どのようにして
5. When（時期）・・・いつから
6. Who（体制）・・・誰が

「エンバグが多い」という問題の例

1. Why（目的・狙い）・・・何のために
   - ✓ エンバグを減らし、不具合件数を収束させるため
2. What（課題）・・・何をどの程度
   - ✓ エンバグ発生率を10%以下にする
3. Where（対象範囲）・・・どこを対象に
   - ✓ 自動テストが可能な個所すべて
4. How（手段）・・・どのようにして
   - ✓ 自動テストとコミットルールの徹底
5. When（時期）・・・いつから
   - ✓ 明日から
6. Who（体制）・・・誰が
   - ✓ プログラマ全員

STEP-6 結果を評価する

結果を確認します。

改善が不十分な場合や、予期せぬ弊害がある場合、解決策の見直しや改善を図ります。

1. 問題は改善したか?
2. 目標までの達成率は?
3. 弊害はないか?

「エンバグが多い」という問題の例

1. 問題は改善したか?
   - ✓ エンバグがほとんどなくなった
2. 目標までの達成率は?
   - ✓ エンバグの発生率は5%以下になった
3. 弊害はないか?
   - ✓ テストコードを書くコストが増えた

STEP-7　標準化する

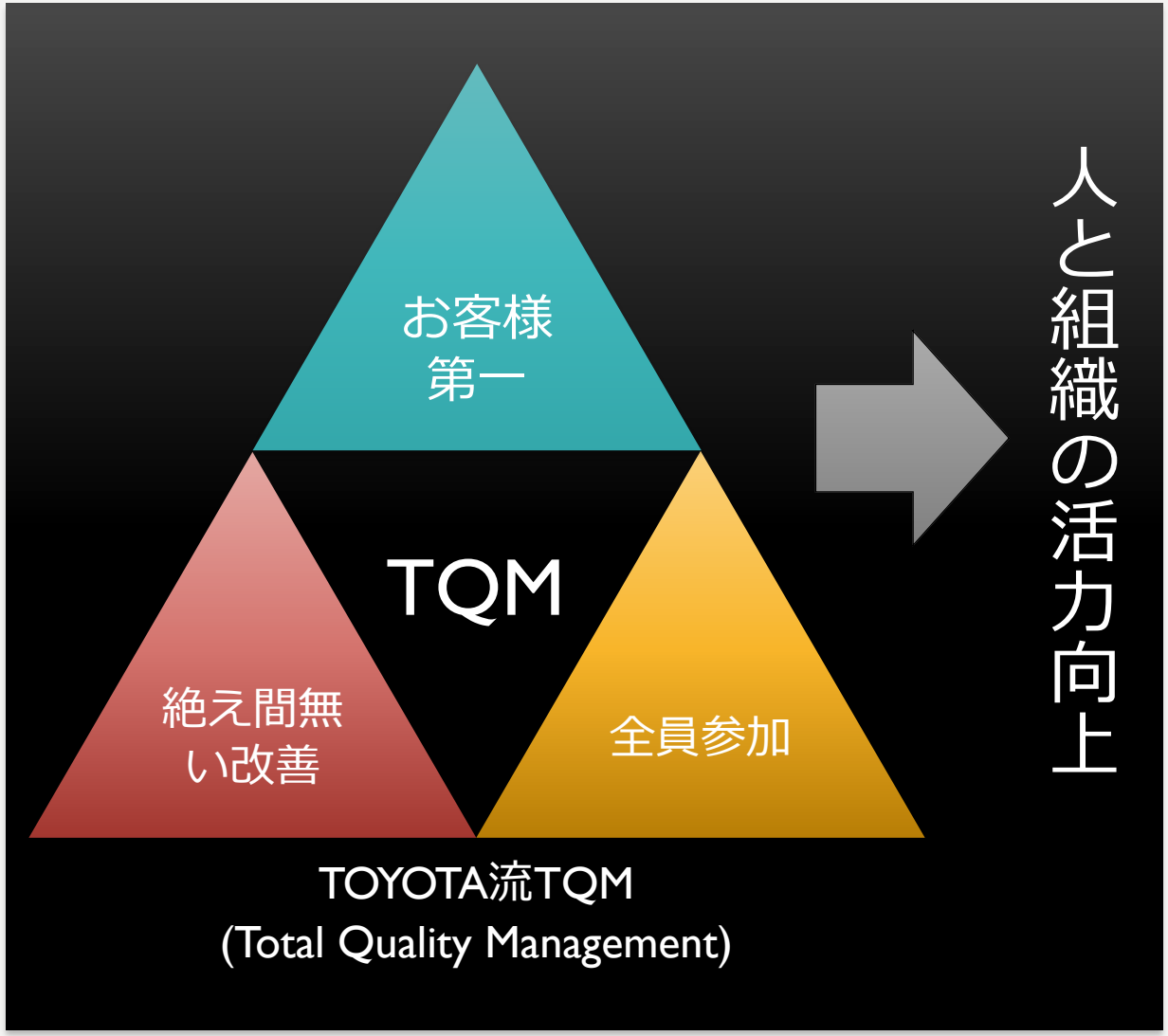
似たような問題の再発を防ぐために、今回行った対策を標準化・フレームワーク化を行います。

1. できるだけ汎用的なフレームワークに落とし込みます。
2. 問題解決の手順や結果をできるだけ多くの人に周知してもらいます。
3. 問題解決を行う習慣を定着させます。

「エンバグが多い」という問題の例

1. できるだけ汎用的なフレームワークに落とし込みます。
   ✓ドキュメント化して共有スペースに公開する
2. 問題解決の手順や結果をできるだけ多くの人に周知してもらいます。
   ✓勉強会で発表する
3. 問題解決を行う習慣を定着させます。
   ✓絶え間ない改善を社訓にする

おわりに



問題解決は、企業やチームの組織力および活力の向上に欠かせません。
トヨタグループの成功の裏には、このような取り組みを積極的に行ってきた経緯があります。
小さな会社や個人でも実践していきましょう。

このドキュメントは以下のgithubからダウンロードできます。

https://github.com/ssa-study/documents
ファイル名: ProblemSolvingForEngineer.pdf

参考文献
「トヨタ流問題解決「８ステップ」の誰も教えていない極意」
http://starhillplan.com/2016/09/26/problem-solving-3/
飯塚悦功、金子龍三(2012)：「原因分析～構造モデルベース分析術～」日科技連出版社

ソフトウエア研究会in秋葉原
http://ssa.techarts.co.jp/
池田公平 godai@techarts.co.jp