

---

# 平成24年度 プロジェクトベース型 実践演習説明会 ( Project Based Learning編)

---

平成24年10月27日

主催：福岡市

実施：NPO法人 九州組込みソフトウェアコンソーシアム(QUEST)

# プロジェクトベース型実践演習

- プロジェクトベース型実践演習
  - Project Based Learning : 略してPBL
    - 演習となるプロジェクトを起こし要求仕様、設計、試験、プロジェクト管理を行うことにより、実践的なプロジェクトを学習する。

# なぜ、PBLなのか？

- 学習方法の違いが大きい
  - PBLでない学習方法（座学等）
    - 講師が学習内容を決定
    - 受講者がその内容を学習
    - 講師が問題（宿題）を提供
    - 受講者がレポート（宿題）を提出（学習の確認）
  - PBLによる学習方法
    - 講師が題材を提供
    - 受講者が題材をプロジェクトとして実行
    - 受講者が課題・問題を発見、自主的に学習、解決、をサイクル的に繰り返す

与えられた学習内容の  
理解が目的

プロジェクトで発生する問題を、チームもしくは個人で解決<sub>3</sub>

# プロジェクトの題材

- 本講座では、平成25年1月に開催される福岡市主催の Green ET Challenge 2012（以降より GETC2011記載）をプロジェクトの題材として実施します
- GETC2012は低消費電力をテーマとしたソフトウェア開発と、その成果を模型によるレースで競います
- 組み込みソフトウェアを設計開発する過程を演習として取り組み、その成果をGETC2012で確認します

# PBLの行い方

- プロジェクトチームの結成
- プロジェクトを開始する
  - PDCAサイクル
- 定期的なチームミーティング

# プロジェクトチームの結成

- チームリーダ（プロジェクト管理者）
- システムアーキテクチャ（システム全体の設計）
- 設計者/プログラマ（単体設計、コーディング）
- 試験担当者（システム全体の試験の設計）
- ドキュメント作成者（マニュアル等の設計）
- その他間接員
  - 上記は理想であり、実際には兼任で行うことが多い。
  - チームリーダがコーディングを行うと失敗することが多いので人員構成をよく考える。

# プロジェクトを開始する (PDCAサイクル)

- Plan→Do→Check→Action→Plan . . .
  - 計画を立て、実施し、問題点課題を洗い出し、それを対策・改善することを繰り返す。
- 簡単そうに見えるがすぐに挫折する。なぜか？
  - 計画を立てない
    - 面倒くさい。予測が難しい。経験不足。必要ないと考える。
  - 問題点を報告しない
    - 恥ずかしい。能力がないと思われたくない。責任を感じて言い出せない。黙って解決してみせる。etc
  - 管理を含めた見積もりをしていない

# プロジェクトを開始する (PDCAサイクル)

- PDCAサイクルを行う強い意識が必要
  - PDCAサイクルが当たり前になると、やらないと不安になる。きちんとしたやり方が身につく。
  - 「問題点の分析」「改善」「スケジュール・納期」がワンセットで議論できるようになってくる。
  - 必要な作業とそれ以外の明確化。



# 定期的なチームミーティング

- コミュニケーションの重要性
  - 議論は大事。しかし、上の人が押し付けるだけでは無理が生じる。論理的に諭す必要あり
  - 結論から先に言う
  - 批判は、誰でもできる。時間の無駄
    - **提案あるいは代替案を考えての発言**
  - 建設的な議論は、皆の頭の回転をよい方向へ向ける
  - 感情的ではなく、論理的思考を訓練する
  - 議論を発散させない
    - 得意分野の議論に誘導し横道にそれる

# 進捗遅れ・原因の洗い出し・対策

- チームリーダの腕の見せどころ。
- 担当を問い詰めない。雰囲気悪くなるとプロジェクト遅れる、または失敗する
- 本音を聞き出して問題の分析を行い対策を検討するのが仕事
- 残業して対策、土日出勤して対策するというのは、時間の取り方であって対策ではない。
- みんなで原因を考える。あるいは、原因を抽出するための手段を提案する。

# 原因の洗い出しの注意点（例）

- 場当たりの対処になっていないか？
- 過去に同様の問題はないか？
- 再現性は（条件は明確になっているか）？
- 真の原因を押さえて問題対策を検討したか？
  - たとえば、健康上の問題が真の原因の場合もある
  - ハードウェアが原因の場合もある
- 潜在的な問題なのか、新規作りこみなのか？
- 信頼できる開発ツールを用いているか？
- ソースの構成管理（版数の不整合など）

# プロジェクトの実践

- 題材：
  - GETC2012で優勝することを目的とする
- 管理：
  - PDCAサイクルを導入する

# PLAN (計画)

- 計画前の準備
  - コンテストの目的の理解
    - 「地域発信」
    - 「低炭素社会への技術」
    - 「組込み教育」
  - 基本モデルの理解
    - 11月17日(土)福岡県Rubyコンテンツ産業振興センターにて基本モデルの説明会があります。
    - 基本モデルを理解し、発展的設計方法あるいはまったく違う設計方法を考え実施してください。

# PLAN (計画)

- 計画前の準備
  - 要素技術の洗い出し
    - どのようなドメインが必要か？
      - 知識領域、専門領域
      - メンバ選定、確保
  - 提出書類(納品物)の原案をイメージする
    - 工夫した点、基本モデルとの違い
    - ここまでは、必ずやろうと決意した点
    - スケジュール
    - 今後の課題

# PLAN (計画)

- 実際に計画を立てる
  - 作業項目の洗い出し
    - 事前調査
    - 各設計書作成 (Outputの定義)
    - 製造 (コーディング)
    - 試験仕様書作成試験
    - ドキュメント作成
  - 担当者の割り振り
  - 作業項目に対する期日設定

# 計画表 (例)

Green ET Challenge 2011 計画線表の例

工程内容	担当	9月				10月				11月				12月				1月				2月				3月			
		5	12	19	26	3	10	17	24	7	14	21	28	5	12	19	26	2	9	16	23	30	6	13	20	27	5	12	19
1. 調査																													
1. 1 コンテストの調査	Aさん	S	—	E																									
1. 2 PBLの調査	Aさん	S	—	E																									
1. 3 NXTの調査	Aさん	S	—	E																									
2. 設計・試験																													
2. 1 コンテストの要求分析	全員			S/E																									
2. 2 基本仕様書の作成	Aさん				S/E																								
2. 3 詳細仕様書の作成	Bさん				S	E																							
2. 4 試験仕様書の作成	Bさん					S/E				S/E																			
2. 5 コーディング	Cさん						S/E			S/E			S/E																
2. 6 単体試験	Cさん						S/E			S/E			S	E															
2. 7 結合試験	Bさん						S/E			S/E			S	E															
2. 8 総合試験	全員						S/E			S/E			S	E															
3. ドキュメントの作成																													
3. 1 コンzeptシートの作成	Aさん												S/E																
3. 2 学校・会社内発表	Bさん													S/E															
3. 3 学会発表	Bさん													S	—	—	E												
3. 4 外部発表(Web等)	Bさん																												
4. イベント																													
4. 1 PBL概要及びコンテスト説明					△9/24																								
4. 2 ハードウェア、ソフトウェア説明会						△10/8																							
4. 3 試走会1									△10/29																				
4. 4 試走会2										△11/26																			
4. 5 大会													△12/17																
4. 6 情報処理学会全国大会																											△1/13締め切り		△3/6-8

ウォーターフォールモデル

スパイラルモデル

機能単位のイテレーションを作成していき、分析、設計、製造、テストを回す。機能拡張して、再度回す

ウォーターフォールモデルでは、上流工程が終わらないと次の工程には進めない



# DO（実施、実行）

## • 事前調査

- 情報収集その結果を必ず記録しておく
- 読んだだけでは時間をつぶしただけ
- キーワード、ブックマーク、概要等の記録
  - 様々な情報があとあと意味を持つことがある
  - 知識が増えると、ゴミも宝にかわる
- 低消費電力の実現方法を明らかにする。
  - そのためには、低消費電力走行を実現するためのドメインの知識が必要となる
  - 今回のGETC2012では、直接消費電力を測らない
  - いかにして電圧の変化がないような走行をするかがポイントとなる
  - 電圧の変化はどのようにして起きるかを調査する必要がある
  - 解らなければ実験してみるなど。

# DO（実施、実行）

- 設計書の作成
  - どのような設計書が必要かを吟味する。
    - 客先要求をまとめたものは、要求仕様書
    - 製造を行う時に必要となるものが設計書
    - 基本設計書
      - 客先要求をどのように実現するかをラフスケッチ
      - マシンスペックを列挙
      - 制限を列挙
      - 必要リソース（ROM、RAM、その他）

# DO（実施、実行）

- 設計書の作成

- 詳細設計書

- 基本設計に基づいてその詳細を書く
    - どのようなアルゴリズムで実現しているのか
    - 状態遷移図、クラス図等を使ったUML表示

- 機能設計書

- どのような機能があるのか、インタフェースはどのようなものか

- 関数設計書

- 関数の機能とインタフェースをわかりやすく形式的に書く

# DO（実施、実行）

- 設計書の作成

- UML

- 誰でも共通に理解できる。
    - 後から見ても分かりやすい
    - 面倒かもしれない
    - 時間がかかるかもしれない。
    - ドキュメントのメンテナンスが大変。小規模でも本当に必要なのかが疑問
    - が、将来必ず役に立つのでこの機会に慣れておいた方がGOOD

# DO（実施、実行）

- 設計書の作成

- 基本モデルでは取扱っていないマルチタスクを行なってみるのも新しい発見ができるかもしれない。
- Green ET Challengeではドキュメントの審査は行いません。設計手法の学習の行い方はETロボコンをお勧めします。
  - <http://www.etrobo.jp>

# DO（実施、実行）

- 製造（コーディング）
  - ソフトウェアの実装
    - 基本モデルのソフトウェアを実装してみる
    - またはオリジナルソフトウェアを組んでみる
- 試験仕様書の作成
  - 試験仕様書は、後で見なおしたときに非常に役に立つ。
  - 試験仕様書で行った試験は、実際に経験したことであり、憶測ではなく自信を持って「それはできます」と断言できる。

# CHECK（点検、評価）

- プロジェクトを実行すると必ず問題や課題が発生する
  - まずは動くのか？
  - 本当に低消費電力となっているか？
  - ユースケースどおりに動作しているか？
  - 計画したドキュメントの作成ができているか？
  - 期日を守れているか？
- 課題あるいは、問題の分析作業が必要
  - 課題、問題の原因は何か？
  - QC7つ道具（必要なものだけ使ってみる）
    - 特性要因図、パレート図、ヒストグラム、管理図、散布図、グラフ、チェックシート
    - 特性要因図は簡単で分析しやすい
    - あてずっぽうだと何の根拠もないので時間の無駄

# CHECK（点検、評価）

- 課題

- とりあえずそのままでも動作するが、改善必要な事柄
- しかし改善は大事。
- どのようにして改善したかを学習する事がとても重要



# CHECK（点検、評価）

## ● 問題

- 仕様あるいは期待する性能と合致しない現象。
- 問題をリストアップする時に、問題の現象を書かずに原因を書く人がいるので注意。
  - 現象欄に
    - 「XXのポインタが参照先を間違っていたために落ちた」

# CHECK（点検、評価）

## • 原因

- 問題となる現象の元。状況証拠で予測されることを書く人がいるがそれは、ただの妄想。
- 本当の原因が解らない場合は「原因不明」とすべき

# CHECK（点検、評価）

## • 対策

- 根本的な対策（永久対策）
- 対処療法的な対策（暫定対策）
  - 原因が特定できないが、現象を出さないような対策
  - いつかは、改善しなければならない
  - しかし、ほっとかれる例も多い
- 仕様とわりきる（未対策）
  - 状況によってリーダーが判断

# CHECK（点検、評価）

- 確認

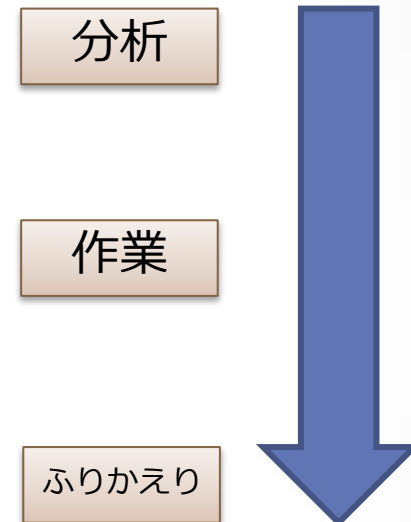
- 問題が解決されたかを確認する。
- 問題の発見者と確認者は別の人が望ましい。

# ACTION（処置、改善）

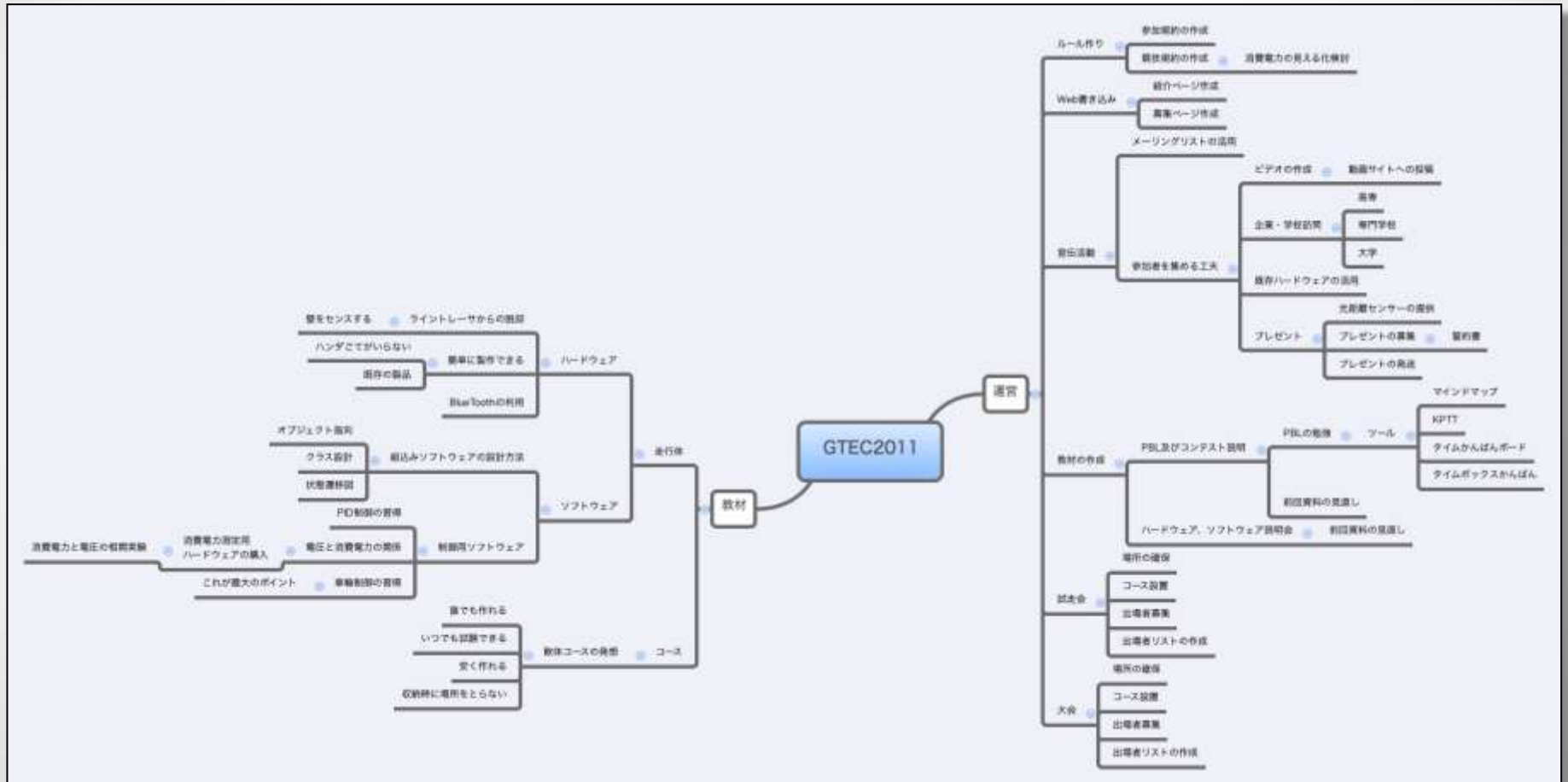
- 処置、改善を実施
  - お金をかけず
  - 時間をかけず
  - 安全に
  - 複雑でない方法
    - 「単純な方法が一番」の一言につきる

# PBLで使われるツール類

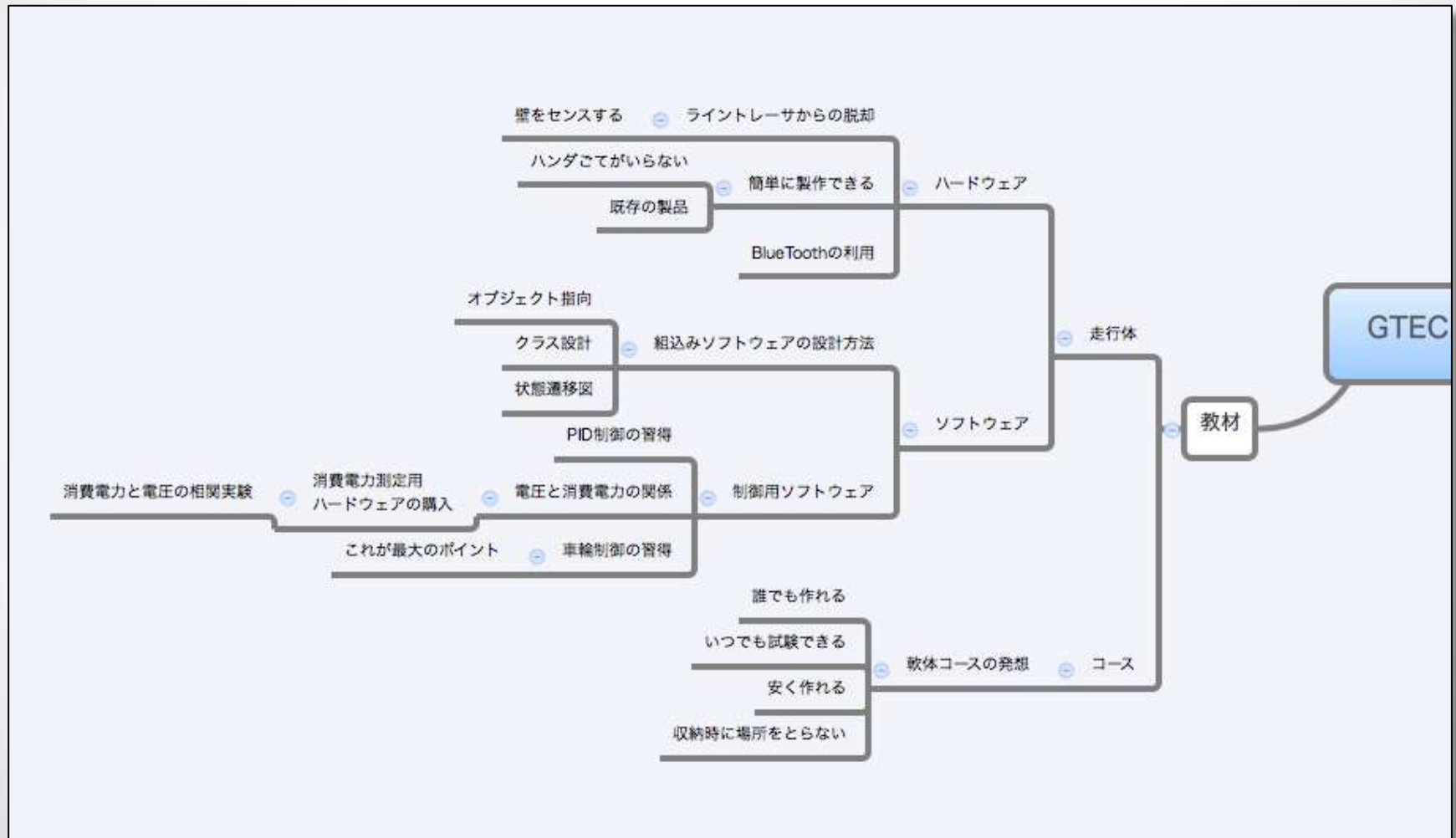
- これらのツールもPDCAに割り当てることが
- 分析、タスク割り当て (P)
  - **マインドマップ**
- 作業 (D)
  - 朝会
  - **タスクかんばんボード**
  - 夕会
- ふりかえり (C/A)
  - **KPTT**
- 実例
  - 学校法人赤山学園 九州技術教育専門学校 (KTEC) でのSQEP「MDDを用いた初学者に対するソフトウェアモデリング教育」でのPBL例



# 分析：マインドマップ (例：GETC2011)

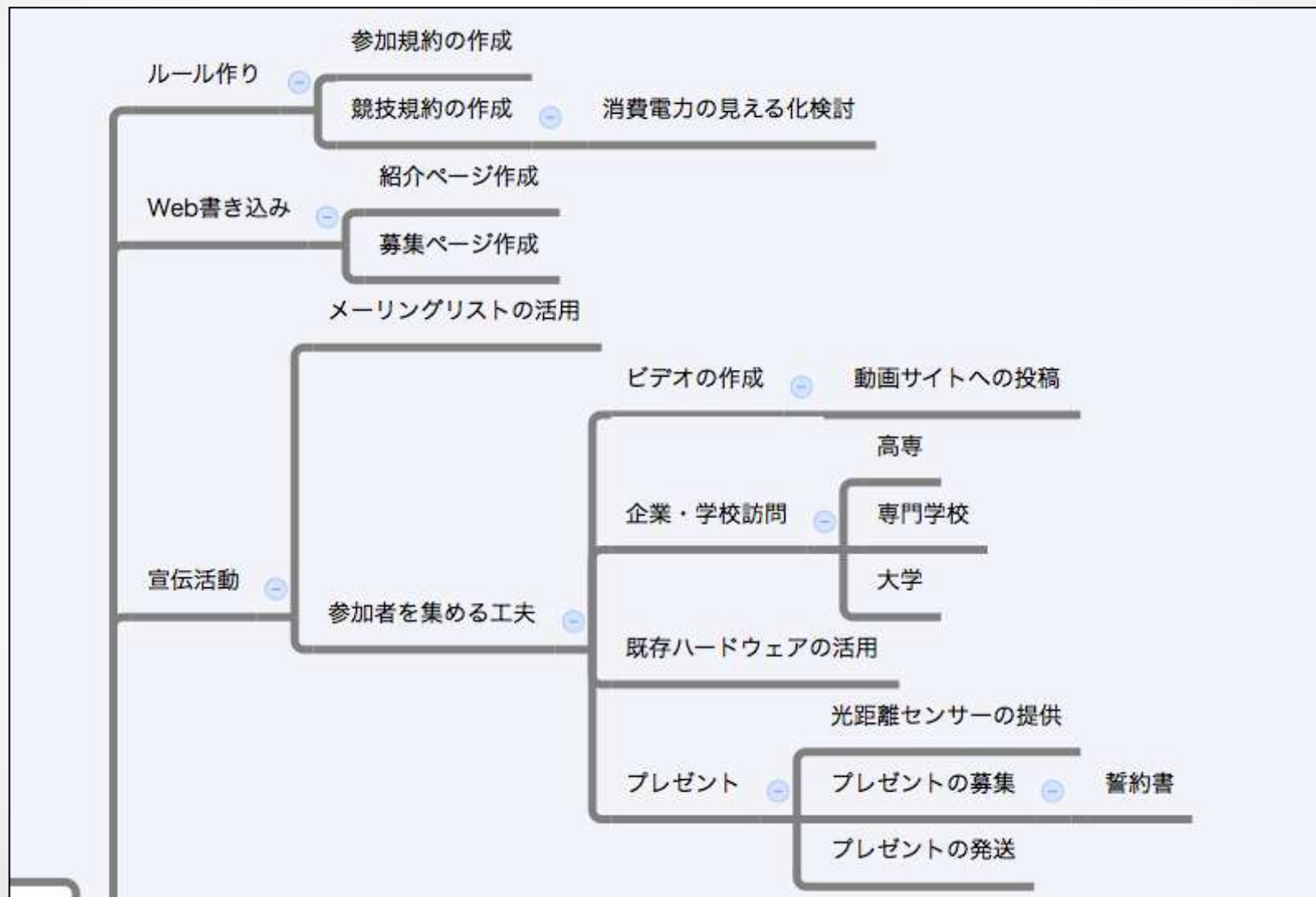


# 分析：マインドマップ（教材）

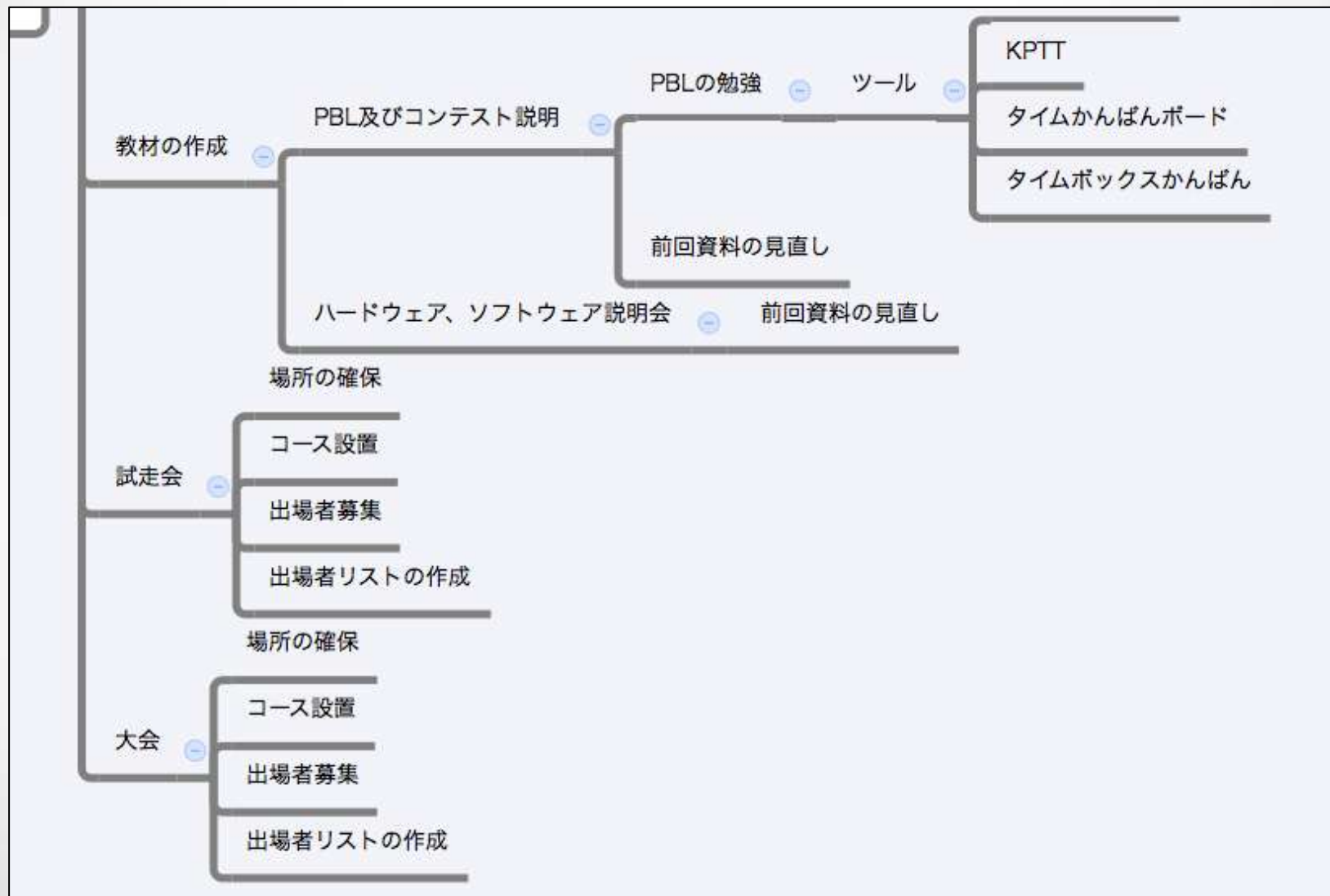




# 分析：マインドマップ：運営(上)

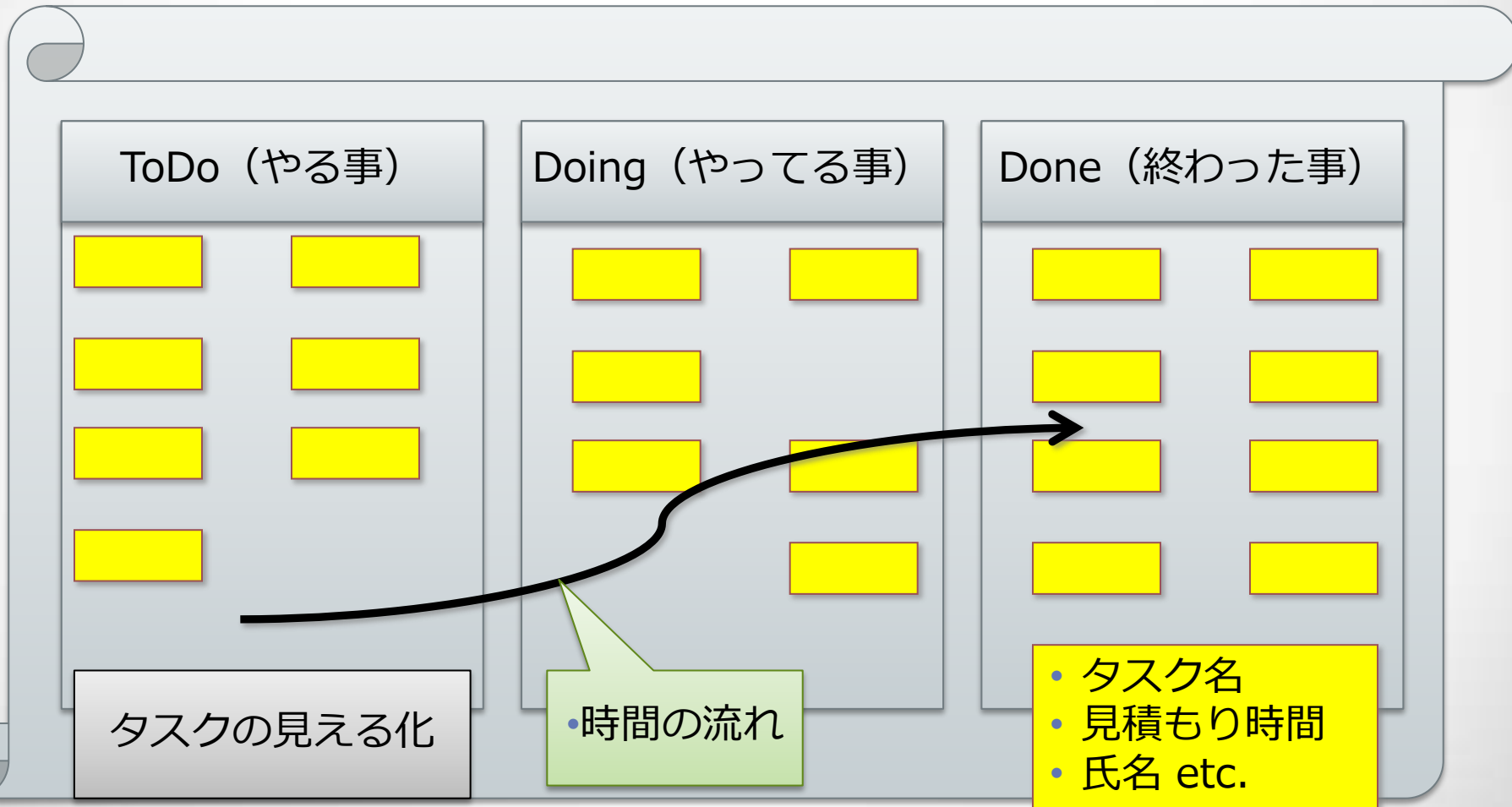


# 分析：マインドマップ：運営(下)



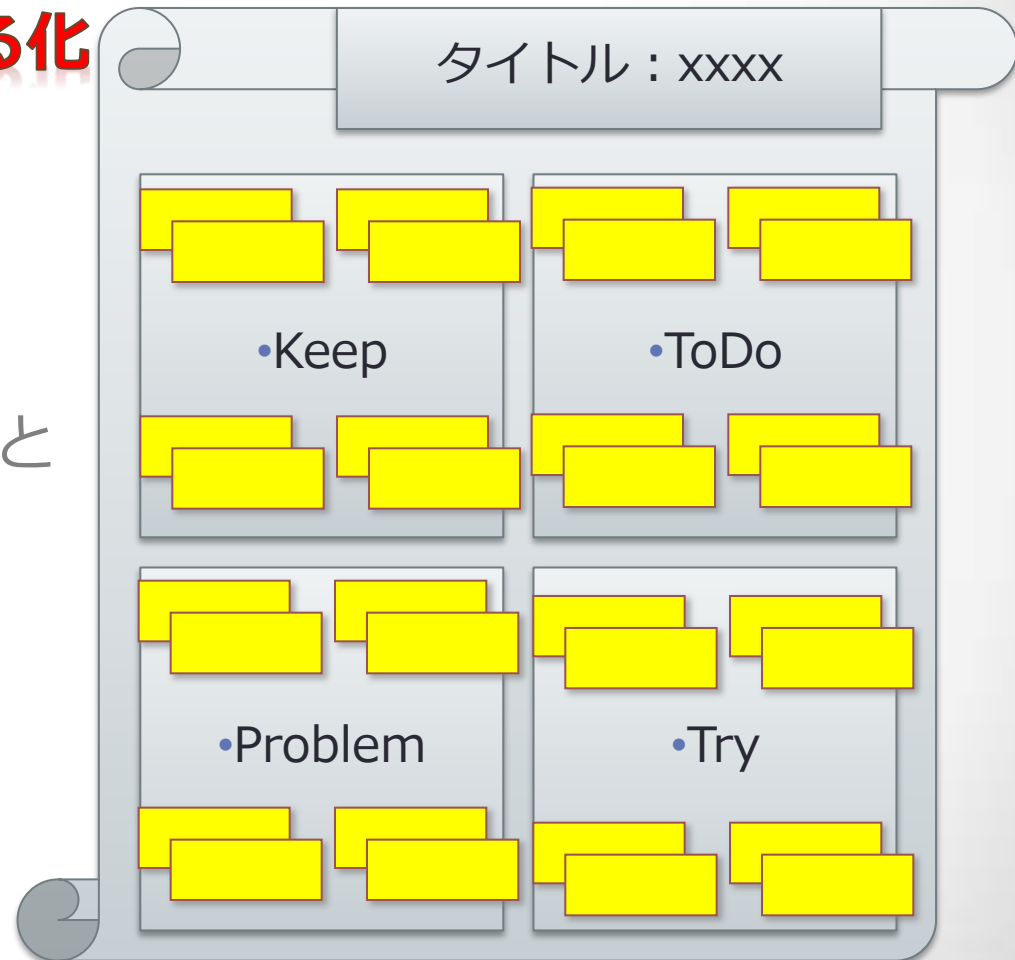
# 作業：タスクかんばんボード

- ToDo→Doing→Done



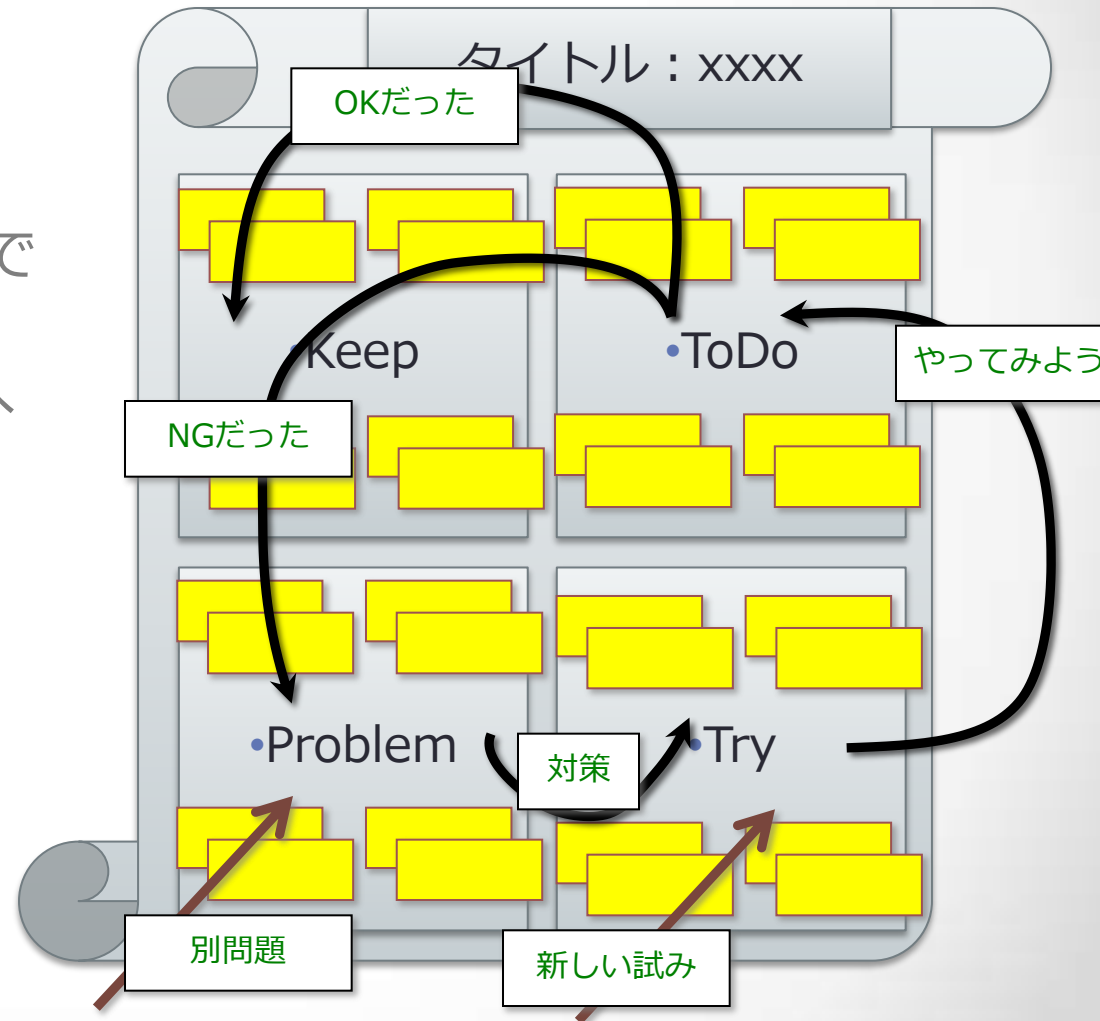
# ふりかえり : KPTT (けぷとつー)

- 目的 : **振り返りの見える化**
  - 改善に使われるツール
  - 振り返ることを見える化
- Keep : 良かったこと
- Problem : 問題点
- ToDo : すぐに出来ること
- Try : やりたいこと
- イベント後に実施する
  - 例えば
    - 夕会ミーティング
    - 週末ミーティング



# KPTTは振り返りの状態遷移図

- イベント後に書きこむ
- Tryでできそうな事をToDoへ移動させる
- ToDoでやってみてOKであれば、Keepへ
- NGであればProblemへ移動する
- Problemで解決方法を見つけTryへ移動
- PDCAに置き換えると
  - P:書き込み
  - D:ToDoの実行、
  - C:OK,NGへ遷移、
  - A:対策を考える



# 事例紹介

## 九州技術教育専門学校様の例

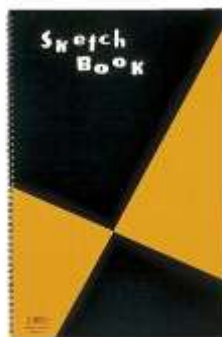
- 産業界と連携した高品質組込みソフトウェア技術者養成プロジェクト
  - (Super Quality Embedded software development education Project:SQEP)
  - 平成22年度にMDDを用いて初学者に対してPBLを使った実践教育を行った。
  - 独自に開発した、「タイムボックスかんばん」を使っている。

# PBL時に各チームに用意したグッズ

• ホワイトボード



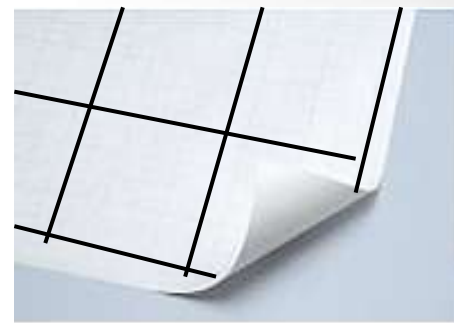
• スケッチブック



• 顔料系ペン



• 模造紙



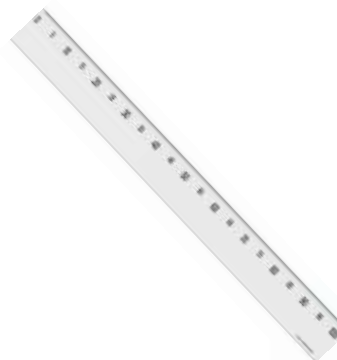
• 付箋紙



• タイマー



• 定規



• ノートPC



「産業界と連携した高品質組込みソフトウェア技術者養成プロジェクト」より引用

# PBLを円滑に進めるためのツール

- プロジェクトファシリテーションのツールを利用

## タイムボックスかんばん

班		年 月 日 ( 日目 )				
目標:		氏名1	氏名2	氏名3	氏名4	氏名5
1	目標	ToDo				
	OK	NG				
2						
3						
4						
5						
6						

## KPTT表



テーマ:	
<b>Keep</b> <ul style="list-style-type: none"> <li>・良かったこと</li> <li>・続けたいこと</li> </ul>	<b>ToDo</b> <ul style="list-style-type: none"> <li>・試してみることに</li> </ul>
<b>Problem</b> <ul style="list-style-type: none"> <li>・問題だと認識していること</li> </ul>	<b>Try</b> <ul style="list-style-type: none"> <li>・試してみたいこと</li> <li>・改善策</li> </ul>



# PBLの1日の流れ：朝会(15分以内)

## ・目的

- チーム全体が必要な情報を短い時間で共有すること。
- 昨日やったことをふりかえり、今日やることを各自認識して積極的に役割を担えるようにすること。
- 朝、気持ちよく仕事のスタートを切ること。

- ・必ず立って行います



「産業界と連携した高品質組込みソフトウェア技術者養成プロジェクト」より引用

## ・タイムボックスかんばん

班		年 月 日 ( 日目 )				
目標:		氏名1	氏名2	氏名3	氏名4	氏名5
1	目標 ToDo					
	OK NG					
2						

- ・1日の目標を記入

# PBLの1日の流れ：開発時間(50分)

- タイムスロット制

- 5分：計画
- 40分：作業
- 5分：ふりかえり
- 10分：休憩

- タイムボックスかんばん

班		年月日(日)	
目標:			
氏名1		氏名2	
1	目標 ToDo		
	OK	NG	
2			

1時間1スロット



「産業界と連携した高品質組込みソフトウェア技術者養成プロジェクト」より引用

# PBLの1日の流れ：夕会(15分以内)

## • 目的

- 一日の目標と作業内容のふりかえりを行い、開発の停滞を防ぐこと
- 効果的な活動や改善策を共有し実行に移せるようにすること
- 次の日の活動をスムーズに進めること

必ず立って行います



## • KPTT表の記入

テーマ：	
Keep ・良かったこと ・続けたいこと	ToDo ・試してみること
Problem ・問題だと認識していること	Try ・試してみたいこと ・改善策

# 修正履歴

日付	修正者	内容	備考
平成23年9月24日	山下	初版	1.00
平成23年9月29日	山下	Rubyコンテンツセンターを福岡県Ruby・コンテンツ産業振興センターへ変更	1.01
平成24年10月26日	馬場	24年版として改版	