
リスクの高いプロジェクトの目標を達成する方法

～TOC/CCPMによる不確実性のマネジメント適用事例報告～

スコープ調整を考慮したCCPM適用事例 (株)ビーイング 後藤智博

「*スコープ調整を考慮したCCPM*」導入事例

株式会社ビーイング 後藤



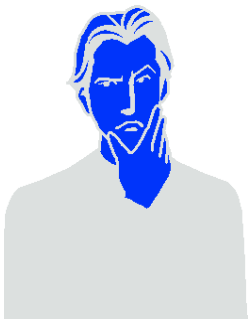
WHAT to Change?

何を変えたのか？

～抱えていた課題～

弊社では、2004年よりソフトウェア開発部隊を中心にCCPMによるプロジェクト管理を実践し、現在ではあらゆる場面で活用しています。しかし、未だに下記の課題を完全には解決できず更なる改善を必要としていました。

1. 営業やお客様からの要望は沢山あるが、どの要望に対応すれば売れるのかよくわからない
2. 営業はある程度完成してからしか製品をみることができず、イメージが違ふと何度も開発手戻りが発生する
3. ビジネス上の理由で製品リリース日は厳しい日程になっている





To What To Change?

何に変えたのか？

～スコープ調整を考慮したCCPM～

解決の方向性

抱えていた課題を何に変えればより改善が進むのか、変えていくべき解決の方向性を営業・開発で検討しました。

何を変えたのか？ (抱えていた課題)	何に変えたのか？ (解決の方向性)
営業やお客様からの要望は沢山あるが、どの要望に対応すれば売れるのかよくわからない	スコープの優先順位を明確にし、製品リリース直前まで営業・開発一体となって 優先順位の検討を継続しながら開発を進める
営業はある程度完成してからしか製品をみることができず、イメージが違っていると何度も開発手戻りが発生する	短いサイクル で開発したモノを営業に 何度も社内リリース し、営業からフィードバックしていくことで、営業の製品開発参画意識を高めると同時に、開発の方向性に合意を得ていく
ビジネス上の理由で製品リリース日は厳しい日程になっている	スコープの増大や開発速度の低下によって、製品リリース日が危険になった場合は、 優先順位の低いスコープをぎりぎりまで調整 しリリース日を守る

今までの方法とは大きく変えようとしたのは上記の赤字部分であり、改善のポイントは「スコープの調整方法」にあることがわかってきました。

スコープ調整を考慮したCCPM

より良いスコープ調整方法を実現するためにCCPMによるプロジェクト管理に加え、下記の方法を取り入れ、「スコープ調整を考慮したCCPM」としてスタート。

改善のポイント	必要条件	適用ツール
優先順位の検討を継続しながら開発を進める	営業と開発が調整できるスコープは何か明確に分かること	・狩野モデル(「望ましさ」による優先順位付け)
短いサイクルで開発したモノを営業に何度も社内リリースする	開発は全力で開発し、営業は全力で調査・フィードバックできること	・単位期間内スコープ固定 ・短サイクル開発(開発ー社内リリースーフィードバック)
	営業と開発が密にコミュニケーションをおこなえること	・単位期間開発会議
優先順位の低いスコープをぎりぎりまで調整する	開発状況が製品リリース日に対して危険であるかどうか判断できること	・バッファ傾向グラフ ・スコープ優先順位



How To Cause The Change?

どのように変えたのか？ ～導入事例紹介～

スコープを明確にして分析し、全体計画(リリース計画)内で開発したいスコープの優先順位付けを行う

1. スコープの洗い出し(ユースケース分析)
2. スコープの優先順位付けと規模の見積り
 - 狩野モデル(「望ましさ」による優先順位付け)
 - スコープアンケート(フィーチャアンケート)
 - プランニングポーカーとスコープポイント(ストーリーポイント)

システムでPERT図とガントチャートを同時に表示し、一方の操作が他方でも反映されながら作業できたら、どう思いますか？	気に入る	<input type="radio"/>
	当然である	<input type="checkbox"/>
	何とも感じない	<input type="checkbox"/>
	しかたない	<input type="checkbox"/>
	気に入らない	<input type="checkbox"/>

システムでPERT図とガントチャートを同時に表示し、一方の操作が他方でも反映されながら作業できないとしたら、どう思いますか？	気に入る	<input type="checkbox"/>
	当然である	<input type="checkbox"/>
	何とも感じない	<input type="checkbox"/>
	しかたない	<input type="radio"/>
	気に入らない	<input type="checkbox"/>

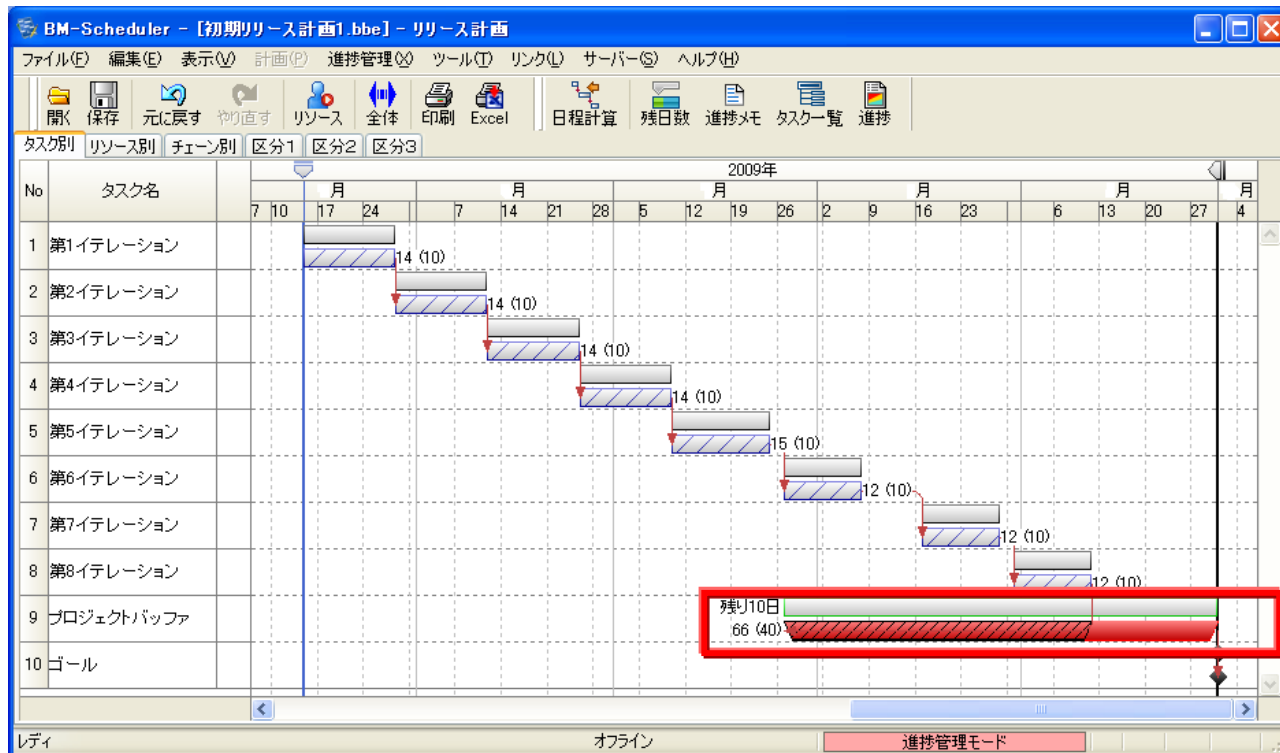


1. 必須スコープ(必須フィーチャ)
2. 線形スコープ(線形フィーチャ)
3. 魅力的スコープ(魅力的フィーチャ)
4. 無関心スコープ(無関心フィーチャ)
5. 懐疑的スコープ(懐疑的フィーチャ)
6. 逆効果スコープ(逆効果フィーチャ)

スコープの優先順位付け結果をもとにどのように開発を進めていくのかを計画する

1. 全体計画(リリース計画)の策定

- ・ スコープポイント合計と単位期間内の開発速度(イテレーション内のベロシティ)から全体計画を算出
- ・ 弊社では単位期間=2週間と設定し、単位期間内の開発速度を算出



製品リリース予定日がきまっていたため、赤バッファスタート

2. 全体計画とスコープ調整

- 全てのスコープを実現することが困難なことが判明したため、スコープ優先順位付け結果をもとに検討した結果、下記のようにスコープを分け、営業・開発両方の合意を得て、開発をスタートさせました。

	A	B	C
1	状況	スコープ	SP
2	必達 初版実装	プロジェクトマネージャーとして、システムでPERT図とガントチャートを同時に見たい。なぜなら、PERT図とガントチャートそれぞれに得意な部分があり、一方の操作を他方でも確認しながら作業できると、さらに効率よく工程表を作ることができるからだ。	8
3	必達 初版実装	プロジェクトマネージャー	3
4	必達 初版実装	プロジェクトマネージャー	8
5	順調に進めば初版実装可能	プロジェクトマネージャーとして、システムで	8
6	順調に進めば初版実装可能	プロジェクトマネージャー	8
7	順調に進めば初版実装可能	プロジェクトマネージャーとして、システムで	8
8	初版実装可能性低	プロジェクトマネージャーとして、システムで	40
9	初版実装可能性低	プロジェクトマネージャー	20
10	初版実装可能性低	プロジェクトマネージャー	13
11	初版以降実装	プロジェクトマネージャーとして、システムで	20
12	初版以降実装	プロジェクトマネージャー	20
13	初版以降実装	プロジェクトマネージャー	13

必達初版実装スコープ

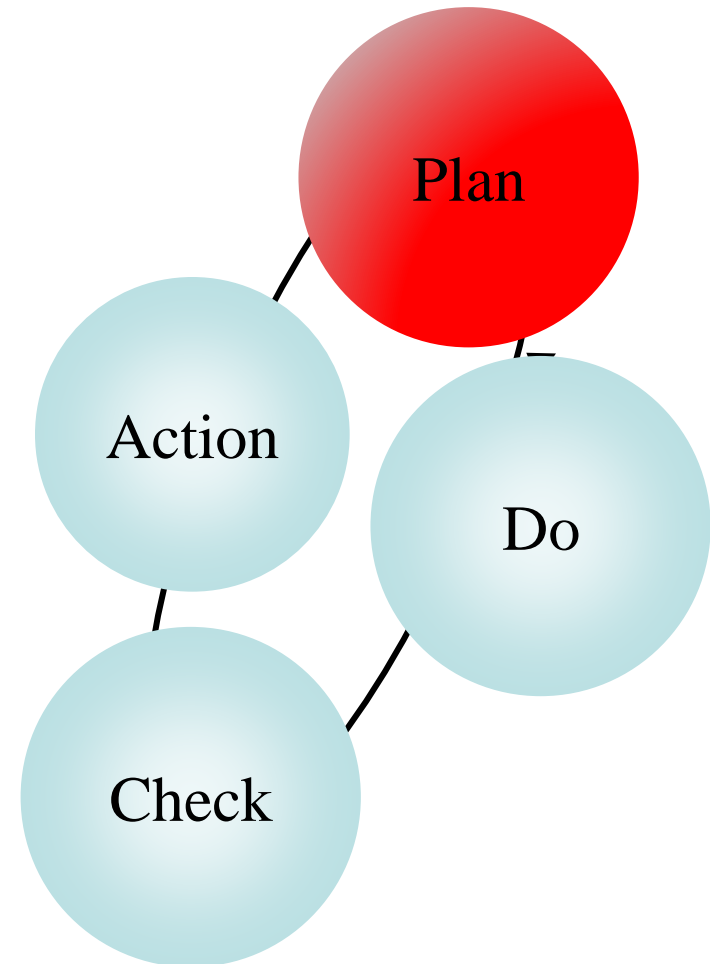
順調に進めば初版実装スコープ

初版実装可能性低スコープ

初版以降実装スコープ

Plan:

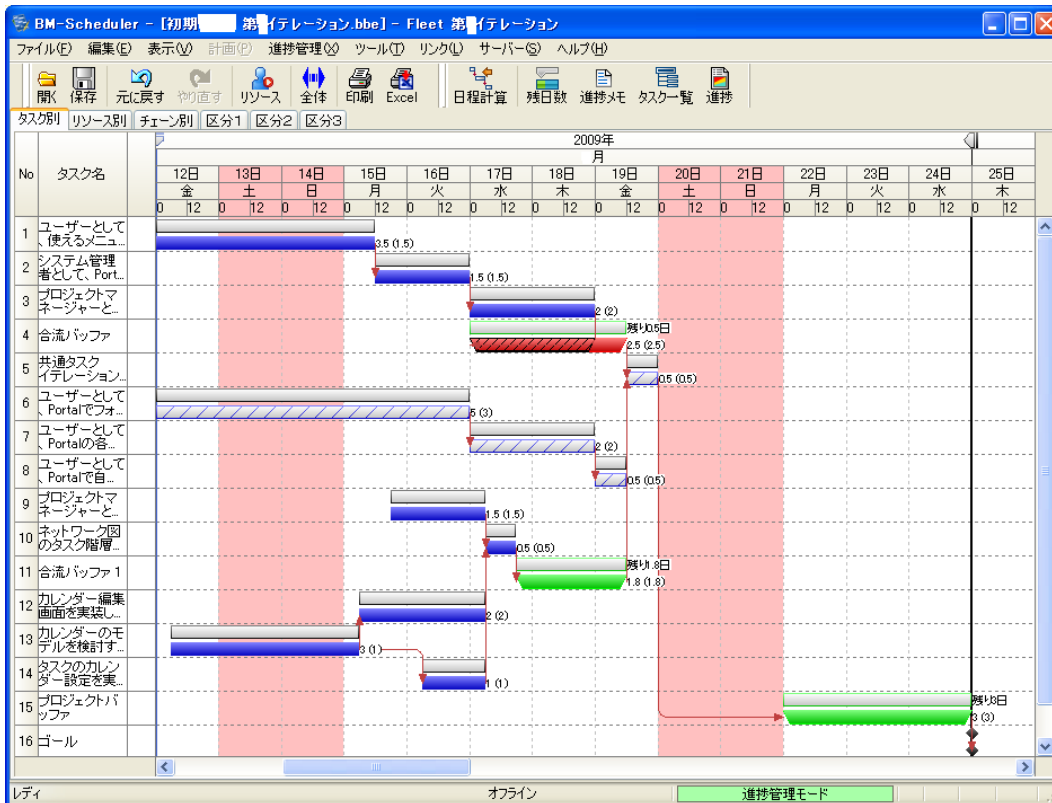
単位期間(イテレーション)
を計画する



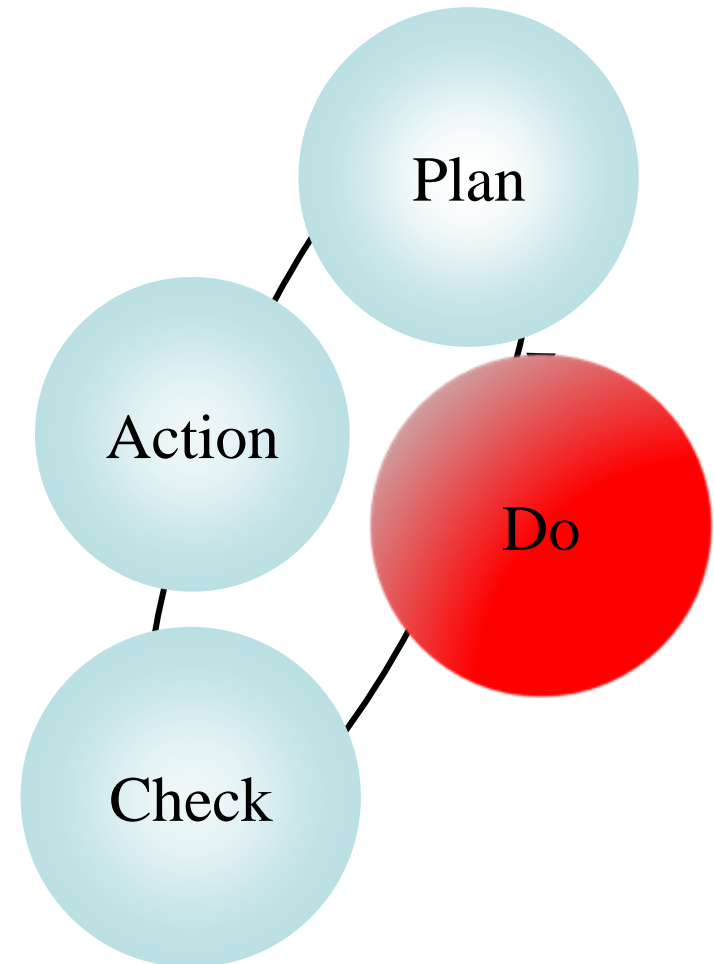
Plan 単位期間を計画する

営業・開発参加のもとで単位期間開発会議を実施し、
単位期間内に実装するスコープの決定と計画を策定する

1. 単位期間内に実装するスコープ決定と固定
2. 単位期間の計画策定(課題バラシ)



Do: 進捗状況を把握する



Do 進捗状況を把握する

CCPMによる単位期間工程表を営業・開発で共有し、 残日数管理により工程表を更新する

1. 残日数管理によるバッファの更新(朝会)

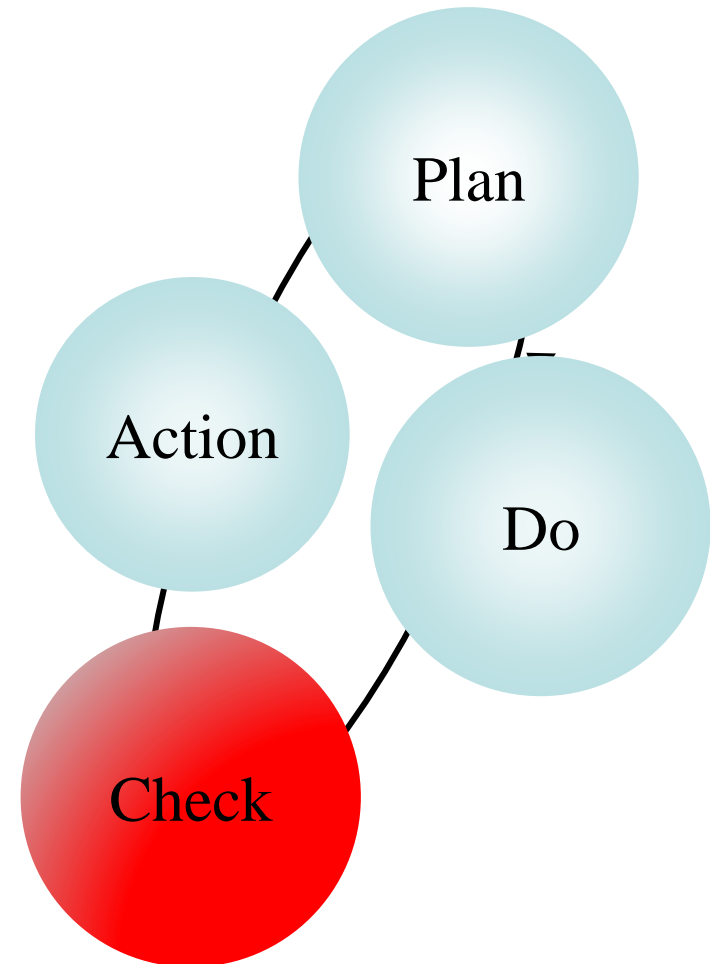
タスク名	開始	終了	見込み	残り(日)	バッファ
ユーザーとして、使えるメニュー	2009/06/12	2009/06/15	0.5	0.5日	合流バッファ
ユーザーとして、Portalでフォルダのタスク階層を整理する。	2009/06/12	2009/06/16	2	3日	プロジェクトバッファ
カレンダーのモデルを検討する。	2009/06/12	2009/06/15	0.5	1日	合流バッファ

残日数を入力



バッファの色が変化

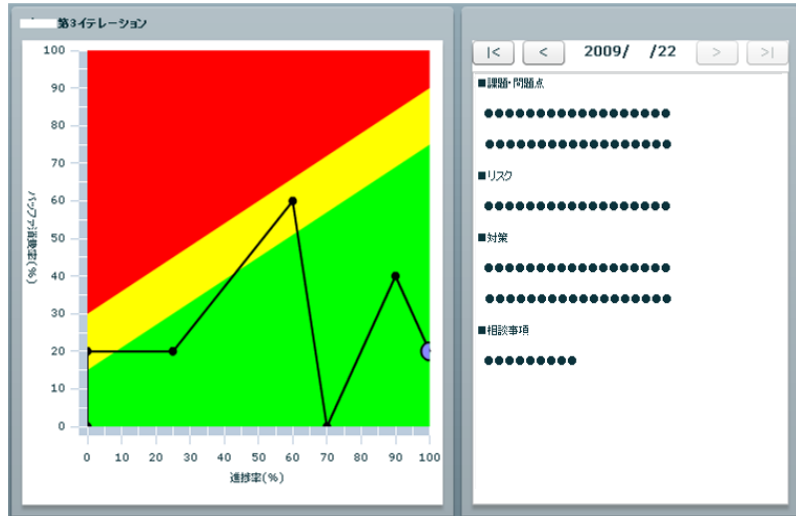
Check: 状況を確認
する



Check 状況を確認する

CCPMによる単位期間工程表を営業・開発で共有し、 単位期間内のバッファマネジメントを行う

1. 単位期間バッファ(イテレーションバッファ)の確認
2. 単位期間バッファの状況と組織全体の開発状況の確認



BeingManagement CCPM
Critical Chain Project Management

マイページ 進捗管理 プロジェクト一覧 個人設定

お気に入り
プロジェクト

検索条件一覧
全プロジェクト
プロジェクト

最近アクセスしたプロジェクト
プロジェクト
プロジェクト
プロジェクト
プロジェクト
プロジェクト

ブロックの追加
進捗表示ブロックの追加
ドキュメントブロックの追加

クライアントダウンロード

プロジェクト名	リーダー	現在日	期限	バッファ消費タスク
プロジェクト	TL	2009/11/20 (1日前)	--	結合処理(動作確認)
プロジェクト	GL	2009/11/21 (0日前)	■■■■■■■■■■	結合処理(動作確認)
プロジェクト		2009/11/17 (4日前)	--	仕様を顧客とレビューして承認を得る
プロジェクト		2009/11/07 (14日前)	--	βシステム結合テスト
プロジェクト		2009/11/17 (4日前)	--	βシステム結合テスト
プロジェクト		2009/11/14 (7日前)	--	■■■■を結合する
プロジェクト		2009/11/15 (6日前)	--	ODSOを基に仕様書を作成する
開発プロジェクト		2009/11/14 (7日前)	--	βシステム結合テスト
プロジェクト		2009/11/14 (7日前)	--	αシステムを開発する
プロジェクト		2009/11/17 (4日前)	--	ODSOを基に仕様書を作成する

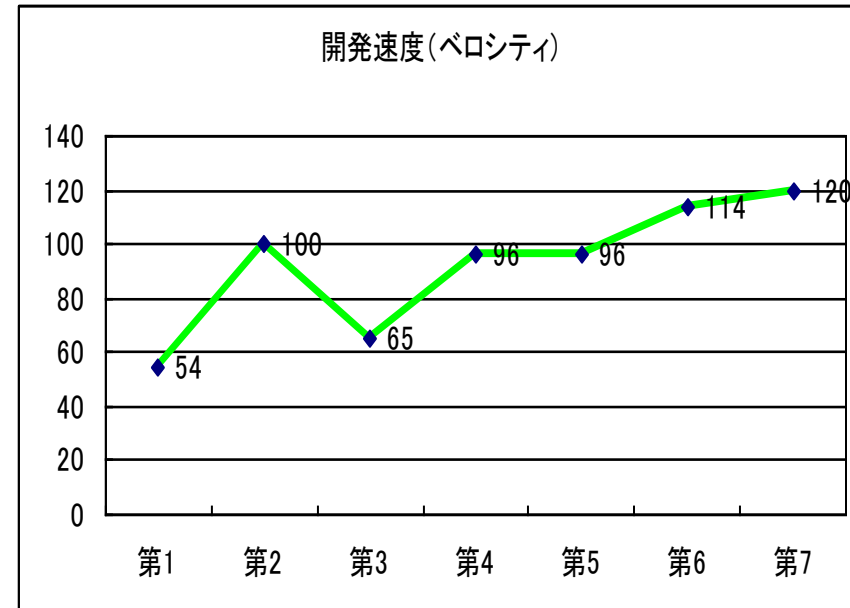
3. 社内リリースとフィードバック

- まとめて社内リリースするのではなく、短いサイクルで「開発ー社内リリースーフィードバック」を行う。
- 営業としては開発状況がわかる上に、開発されたモノに触れることによって製品への愛着が沸いたり、製品教育も行える

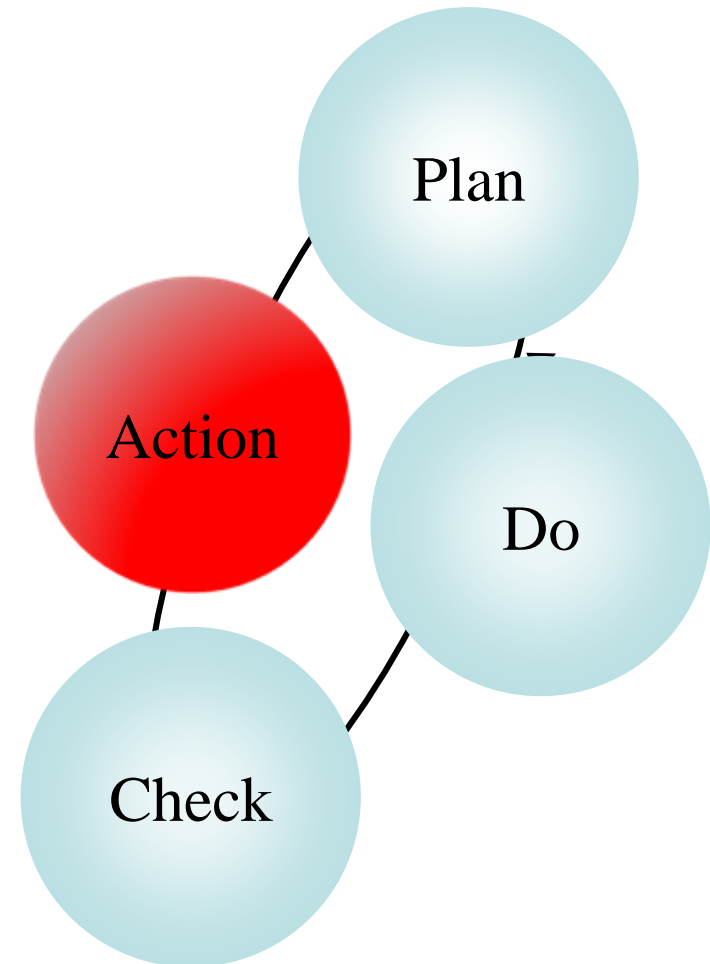


4. 開発速度の確認

- 単位期間の開発速度を計測・確認し、開発速度が上がった要因や下がった要因等を確認する



Action: バッファ回復
策を実施する

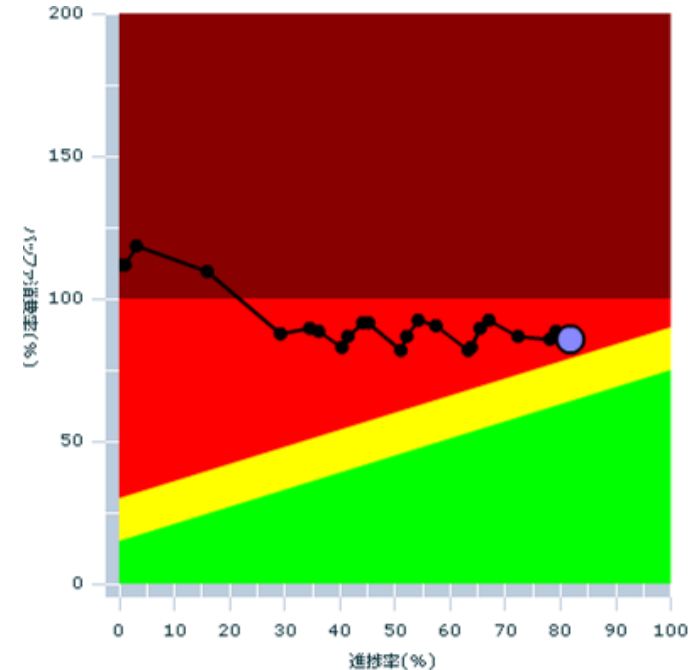
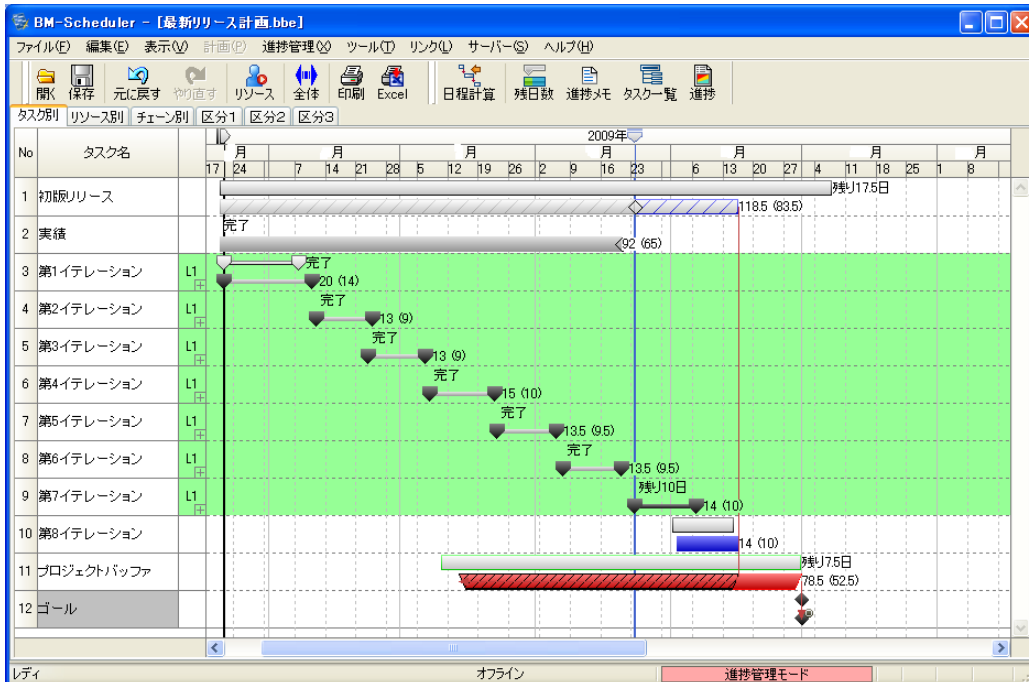


Action 全体計画のバッファ回復策を検討する

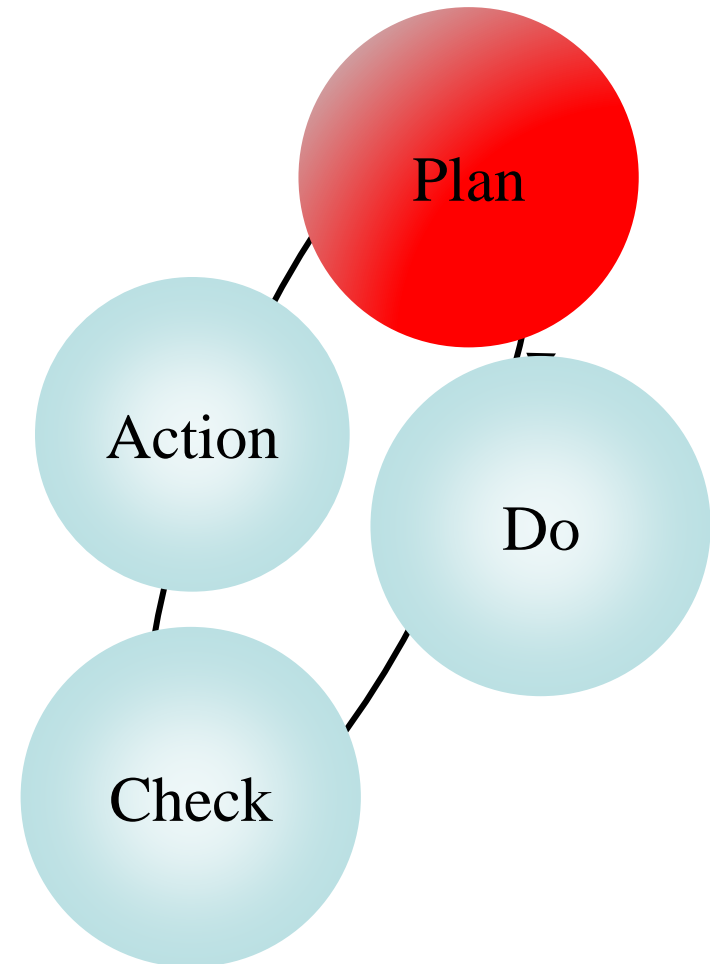
単位期間の結果を全体計画に反映させ、全体計画のバッファマネジメントを行う

1. 全体計画(リリース計画)の更新

- 次の単位期間に持ち越しになったスコープを踏まえ、全体計画工程の更新とスコープの優先順位付表を更新する。
- ここでのスコープ調整では、スコープのランク繰上げ繰り下げ等を行い、製品リリースに含まれるスコープを決定する



Plan: 次の単位期間
を計画する



スコープ調整を考慮したCCPMを実践して

■営業の声

- ・いつ、どのスコープが製品に搭載されるかがわかるようになり、お客様への提案などの活動が格段にやりやすくなった
- ・社内リリース毎に最新のモノに触れるので、開発部隊へのフィードバックがすぐにできる上に、次の社内リリースには改善されているとチームと一緒に商品を作り上げている感じがした
- ・製品の操作や機能も習得でき、正式リリースと同時に問題なく扱える

■開発の声

- ・スコープの優先順位が明確になっただけでなく、スコープ調整により今は必要でないスコープを後回しにできるなどの対策が実行できた
- ・営業VS開発ではなく、一緒になってよい商品を作る体制になった
- ・営業からのフィードバックや率直なリアクションがすぐにもらえるので、改善すべきポイントがすぐにわかり、すぐに製品に反映できた
- ・短いサイクルでPDCAが回るので、開発ノウハウをすぐに活用できた

「スコープ調整を考慮したCCPM」によって完成した商品がこちらです！