ソフトウェア工学 第14回
― ソフトウェア開発管理 ―

大連理工大学・立命館大学 国際情報ソフトウェア学部

大森 隆行

講義内容

- ソフトウェア開発管理
  - プロジェクト管理
    - 開発計画
    - 開発工数の見積もり
    - 品質管理

プロジェクト管理

- プロジェクト管理
  - ソフトウェア開発全体の計画、
    進行状況の確認、その結果に応じた対策を行うこと
- プロジェクト管理の目的
  - 要求を満たすソフトウェアの開発
  - 決められた予算と期間で開発

- プロジェクト管理知識体系
  (PMBOK: project management body of knowledge)
  として、基本的な知識体系がまとめられている
  - 統合管理：プロジェクト計画の策定・実施・変更管理
  - 時間管理：スケジュールの作成、作業時間見積り、進捗管理
  - コスト管理：資源計画、コストの見積もりと予算化
  - リスク管理：リスクの洗い出し、対策　　等の視点がある

risk

PMBOKにおけるプロジェクトのプロセス分類

- PMBOKにおいてはプロジェクトのプロセスは以下の5グループに分類
  - 立ち上げ
    - プロジェクトどのようなフェーズ(段階)に分けて進めるかを定め、認可する
  - 計画
    - プロジェクトの目標を定める
    - 目標を達成するための活動を計画する
    - 必要な資源(予算、人、時間、機器等)の見積もり・調達
  - 遂行
    - 計画に沿ってプロジェクトを進める
    - 作業や成果物の検証、品質保証
  - 制御
    - 実績の報告、進捗の測定 → 計画と実際の差異を分析
  - 終結
    - 契約完了、プロジェクト終了

講義内容

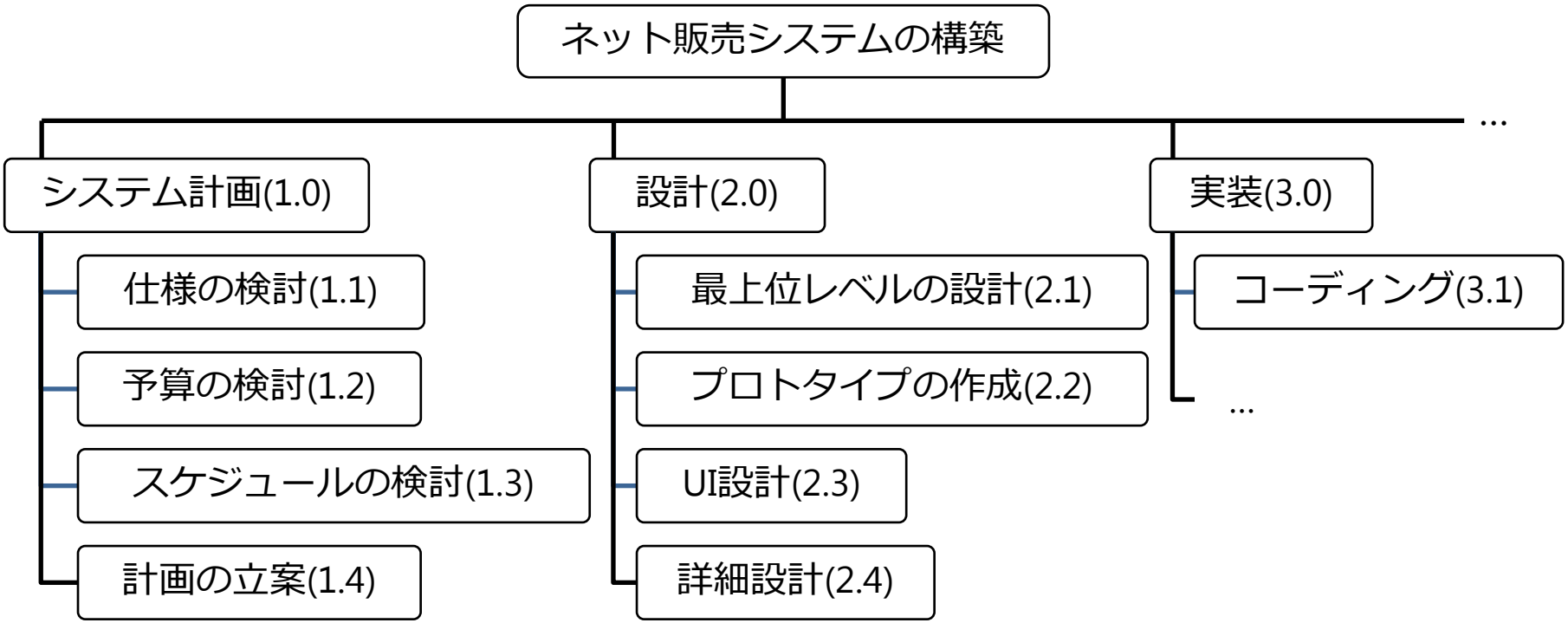
- ソフトウェア開発管理
  - プロジェクト管理
  - 開発計画
  - 開発工数の見積もり
  - 品質管理

開発計画の策定

- 開発計画では、以下の点を明確にする
  - 開発の目的・目標
  - 開発対象業務・運用方針
    - ソフトウェアがどのような業務でどのように使われるか
  - 開発システムの基本構成
    - 採用するアーキテクチャや要求する稼働条件
      (OS、ハードウェア、ネットワーク構成等)
  - 開発工数・開発コスト
    - 開発に要する工数(e.g., 人数×時間)とコスト、
      およびそれらの見積もり
  - 開発スケジュール
    - 開発工数に応じた開発期間の決定
  - 開発体制
    - 開発要員間のコミュニケーションの方法
  - 開発環境・方法論、構成管理手法
    - 利用するIDE、開発方法論、コーディング規約等
  - リスク管理(risk management)手法

開発計画の道具 - 作業明細構造

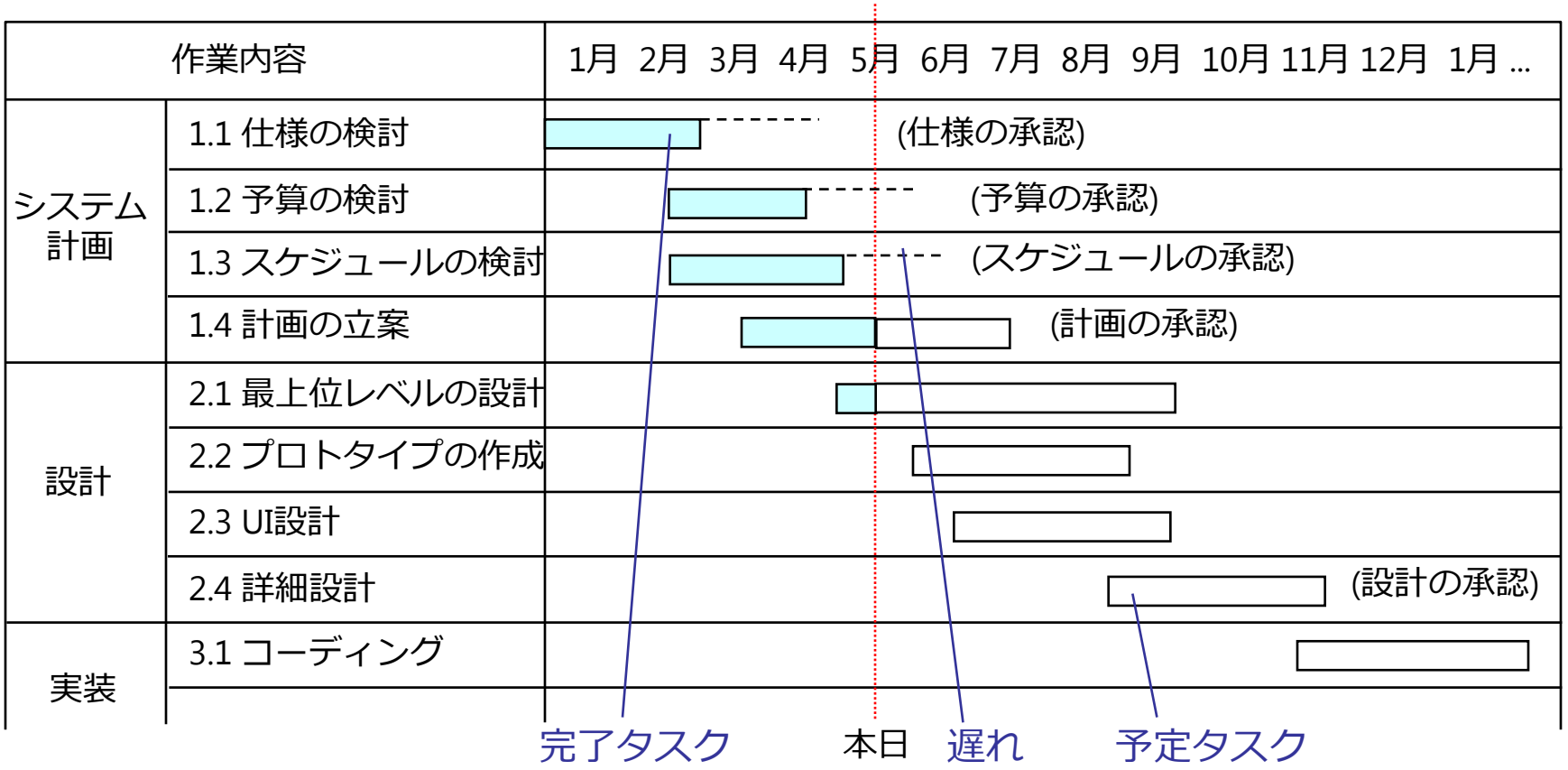
- 作業を分割、詳細化して表記



開発計画の道具 - ガントチャート

甘特图（Gantt chart）又称为横道图、条状图(Bar chart)

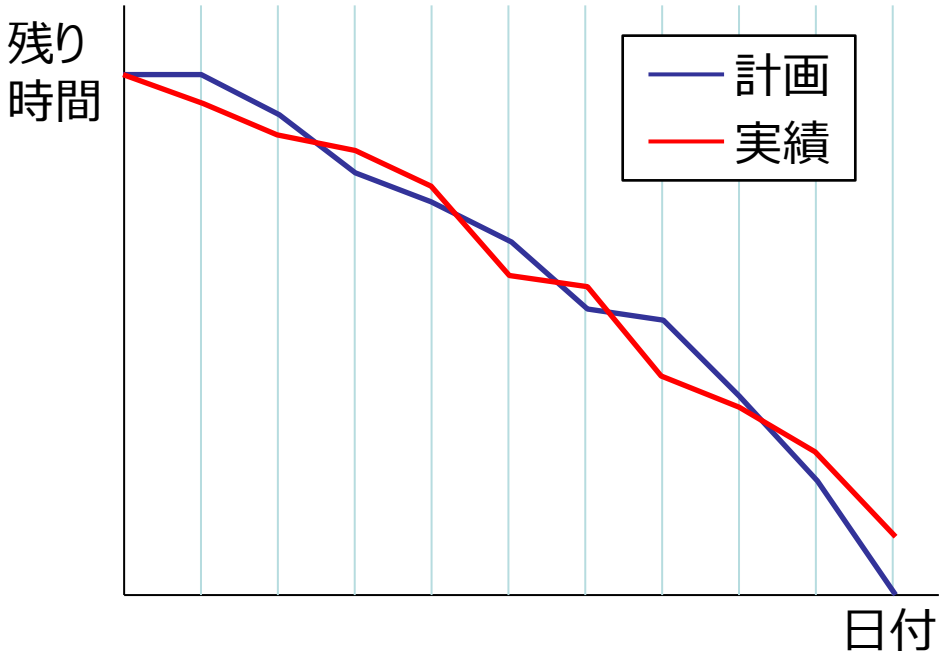
- 各作業の予定と進捗を棒グラフで記述
  - 並列作業や進捗を視覚化



開発計画の道具 - バーンダウンチャート

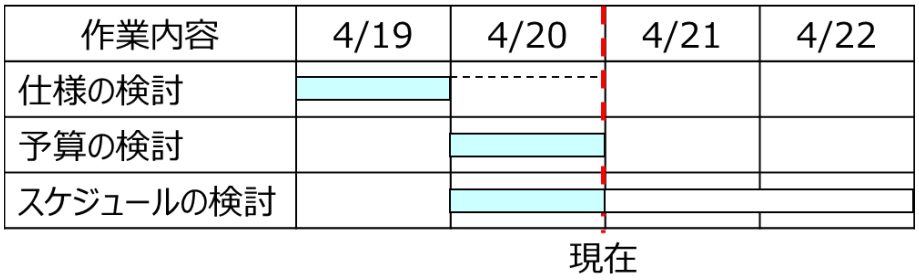
燃尽图（burn down chart）是在项目完成之前，对需要完成的工作的一种可视化表示。

- 開発計画と進捗(推定残り時間)を視覚化
  - チーム内で随時共有する
  - アジャイルプロセス手法スクラム等で採用
  - 見積もりが正確にできるかが重要

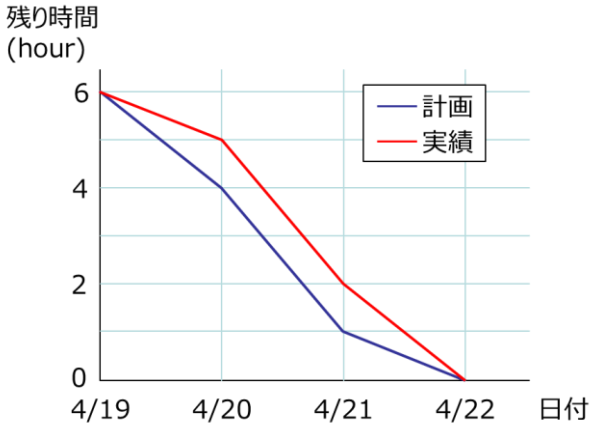


確認問題

- 左側に示したガントチャートおよびバーンダウンチャートからわかることを説明した文の空欄を埋めよ。



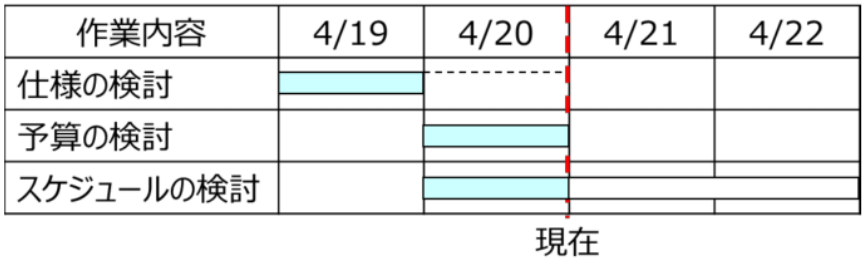
・仕様の検討は4/(1)に完了予定だったが、4/20に遅延した。
・予算の検討は4/(2)に行った。
・スケジュールの検討は4/(3)に開始し、4/(4)に終了する予定である。



・作業全体に要する時間は(5)時間であると見積もられた。
・4/20時点では、計画より実績が(6)時間分遅延している。
・4/21時点では、計画より実績が(7)時間分遅延している。
・4/(8)に作業は終了した。

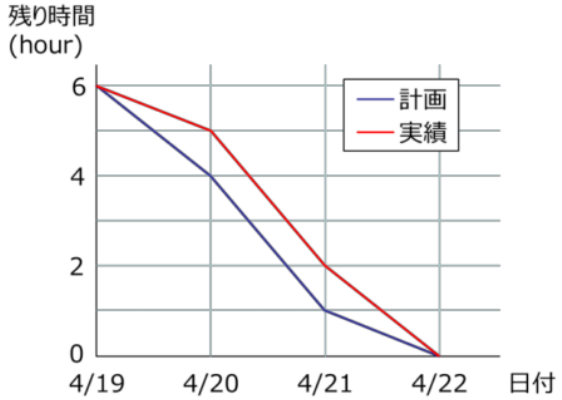
確認問題

- 左側に示したガントチャートおよびバーンダウンチャートからわかることを説明した文の空欄を埋めよ。



・仕様の検討は4/(1)に完了予定だったが、4/20に遅延した。
・予算の検討は4/(2)に行った。
・スケジュールの検討は4/(3)に開始し、4/(4)に終了する予定である。

(1).19 (2).20 (3).20 (3).22



・作業全体に要する時間は(5)時間であると見積もられた。
・4/20時点では、計画より実績が(6)時間分遅延している。
・4/21時点では、計画より実績が(7)時間分遅延している。
・4/(8)に作業は終了した。

(5).6 (6).1 (7).1 (8).22

講義内容

- ソフトウェア開発管理
  - プロジェクト管理
  - 開発計画
  - 開発工数の見積もり
  - 品質管理

開発工数の見積もり

- プロジェクトの計画においては、
  まず開発工数を見積もり、
  それに基づき開発コストを見積もる

- 開発工数の単位
  - 人：何人の開発者が必要か
  - 時間(e.g., 月、日、時間)：どれだけの時間が必要か
  - 人月(にんげつ)：人×月
    - 1人月＝1人が1か月かかる作業量
    - すべての開発者が同じ能力？
    - 3人×1か月＝1人×3か月？
    - 開発者を増やすと、効率は低下する
      - コミュニケーションや管理のコストが大

開発工数見積もり技法

- 標準タスク法
- COCOMO
- ファンクションポイント法

標準タスク法

- タスク(task)の種類ごとに開発工数をあらかじめ設定する
- プロジェクト内で各タスクがどの程度必要になるか(作業件数)を推測する
- タスクの開発工数×作業件数により工数を見積もる

(例) GUI部品実装の作業工数(人日)

規模＼複雑度	単純	普通	複雑
小	1	2	3
中	1.5	3	5
大	2	4	7

GUI部品実装の必要件数(件)

規模＼複雑度	単純	普通	複雑
小	10	5	0
中	10	30	5
大	0	10	10

GUI部品実装の工数見積もり(人日)

規模＼複雑度	単純	普通	複雑
小	10	10	0
中	15	90	25
大	0	40	70

必要なタスクの
種類・数・規模が事前に
判明している必要がある
→ 仕様は事前に確定
している必要

GUI部品実装の工数
=260人日

COCOMO (constructive cost model)

- 開発規模と開発工数の関係を統計的モデルから推測する技法
  - ソースコードの行数から工数を推測

    $y = ax^b$

    x:コード行数　y:工数

    a:係数　b:係数(大規模な開発ほど急激に工数増加)

- より係数決定の方法を詳細化した COCOMO IIもある

ファンクションポイント法(FP法)

Function point

- ファンクションポイントに基づく工数見積もり
- ファンクションポイント：
  ソフトウェアの機能ごとの規模と複雑度に基づく点数

EIの複雑度

関連ファイル数	データ項目数		
	1～4	5～15	16～
0～1	単純	単純	普通
2	単純	普通	複雑
3～	普通	複雑	複雑

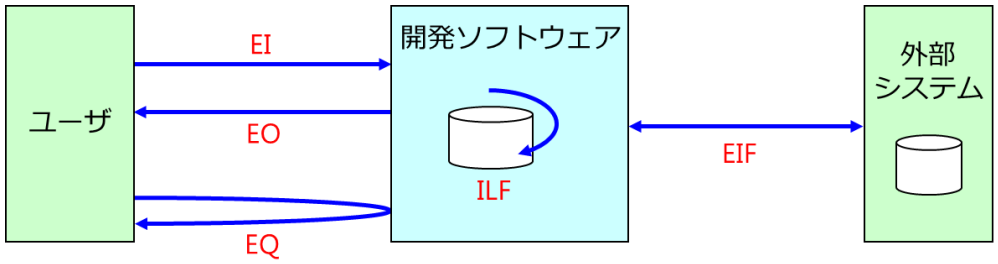
EO, EQの複雑度

関連ファイル数	データ項目数		
	1～5	6～19	20～
0～1	単純	単純	普通
2	単純	普通	複雑
3～	普通	複雑	複雑

ILF, EIFの複雑度

関連ファイル数	データ項目数		
	1～19	20～50	51～
0～1	単純	単純	普通
2	単純	普通	複雑
3～	普通	複雑	複雑

EI: external inputs
EO: external outputs
EQ: external query
ILF: internal logical files
EIF: external interface files



ファンクションポイント法(FP法)

複雑度別の機能数

	単純	普通	複雑
EI	10	12	14
EO	11	13	15
EQ	1	3	5
ILF	2	4	6
EIF	3	5	7

重み付け係数

	単純	普通	複雑
EI	3	4	6
EO	4	5	7
EQ	3	4	6
ILF	7	10	15
EIF	5	7	10

	単純	普通	複雑
EI	30	48	84
EO	44	65	105
EQ	3	12	30
ILF	14	40	90
EIF	15	35	70

FP=685

ファンクションポイント法(FP法)

- 算出されたFPに、システムの特性に応じた係数を割り当て

システム特性	ポイント
1 データ通信	0
2 分散処理	0
3 パフォーマンス	4
4 高負荷環境	4
5 トランザクション量	2
6 オンラインデータ入力	1
7 エンドユーザの作業効率	2
8 マスターデータベースのオンライン更新	2
9 内部処理の複雑さ	3
10 再利用を考慮した設計	4
11 導入の容易性	1
12 運用の容易性	3
13 複数サイトでの使用	0
14 変更の容易性	2
合計	28

0: まったく関係ない
1: ほとんど影響を受けない
2: 適度に影響を受ける
3: 平均的な影響を受ける
4: 大きな影響を受ける
5: 非常に大きな影響を受ける

1±0.35

調整用係数
= 0.65+(0.01×28)
= 0.93

FP
= 0.93×685
≒ 637

FPから実際の開発工数への変換は個々のプロジェクトで
過去の実績等に基づき行う

開発工数見積もり

- 工数に影響を及ぼす要因を的確に
  洗い出す(事前に明らかにする)ことが大切
- 企業やプロジェクトごとに、
  過去の経験を蓄積することで
  正確さを向上可能

講義内容

- ソフトウェア開発管理
  - プロジェクト管理
  - 開発計画
  - 開発工数の見積もり
  - 品質管理

ソフトウェアメトリクス

- プロジェクト管理においては、
  プロダクトやプロセスを評価する必要
  → 定量的な評価尺度(metrics)を導入
- ソフトウェアメトリクス：
  ソフトウェア開発に関する定量的な評価尺度
- ソフトウェアメトリクスの種類
  - プロダクトメトリクス
    - 成果物の評価尺度
  - プロセスメトリクス
    - 開発作業の評価尺度

プロダクトメトリクス

- ソースコードの行数
  - LOC (lines of code)
- プログラムの命令数(ステップ数)
  - 定義は会社等により異なる
- McCabeのサイクロマティック複雑度(循環的複雑度)

step

Cyclomatic

```
//sample program
int main(int argc, char* argv[]){
  int x;

  scanf("%d", &x);
  printf("%d", x);
  return 0;
}
```

LOC: 8
LOC(NBNC): 6

詳細な定義は状況により異なることに注意

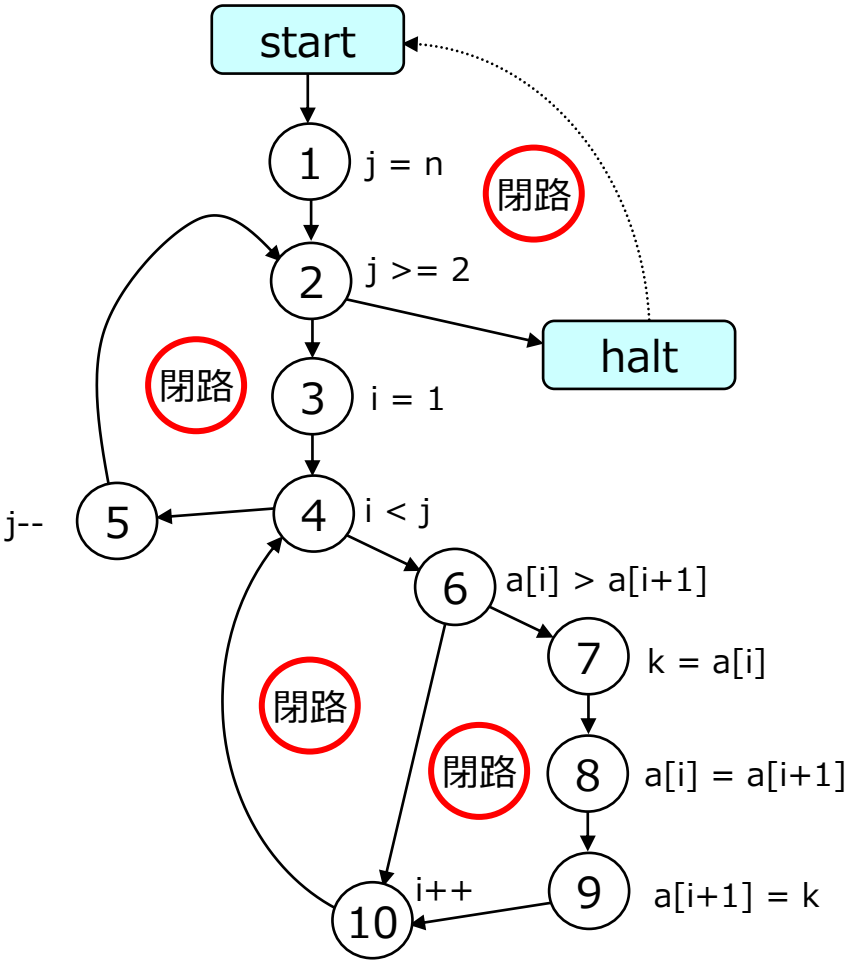
そもそも行ごとに内容も異なる
(同じLOCでも複雑度は異なる)

サイクロマティック数

- プログラムの流れを有向グラフで表現
  → 一時独立な閉路の数で複雑度を測定

```
for (int j = n; j >= 2; j--) {
  for (int i = 1; i < j; i++) {
    if (data[i] > data[i+1]) {
      int k = a[i];
      a[i] = a[i+1];
      a[i+1] = k;
    }
  }
}
```

サイクロマティック数
= 閉路の数
= 4



プロセスメトリクス

- 工数
- 工期(成果物の作成に要した時間)
- 生産性
- 顧客との対話時間
- 仕様変更回数・変更率
- ソースコード変更回数・変更率
- プロトタイピング実施の有無

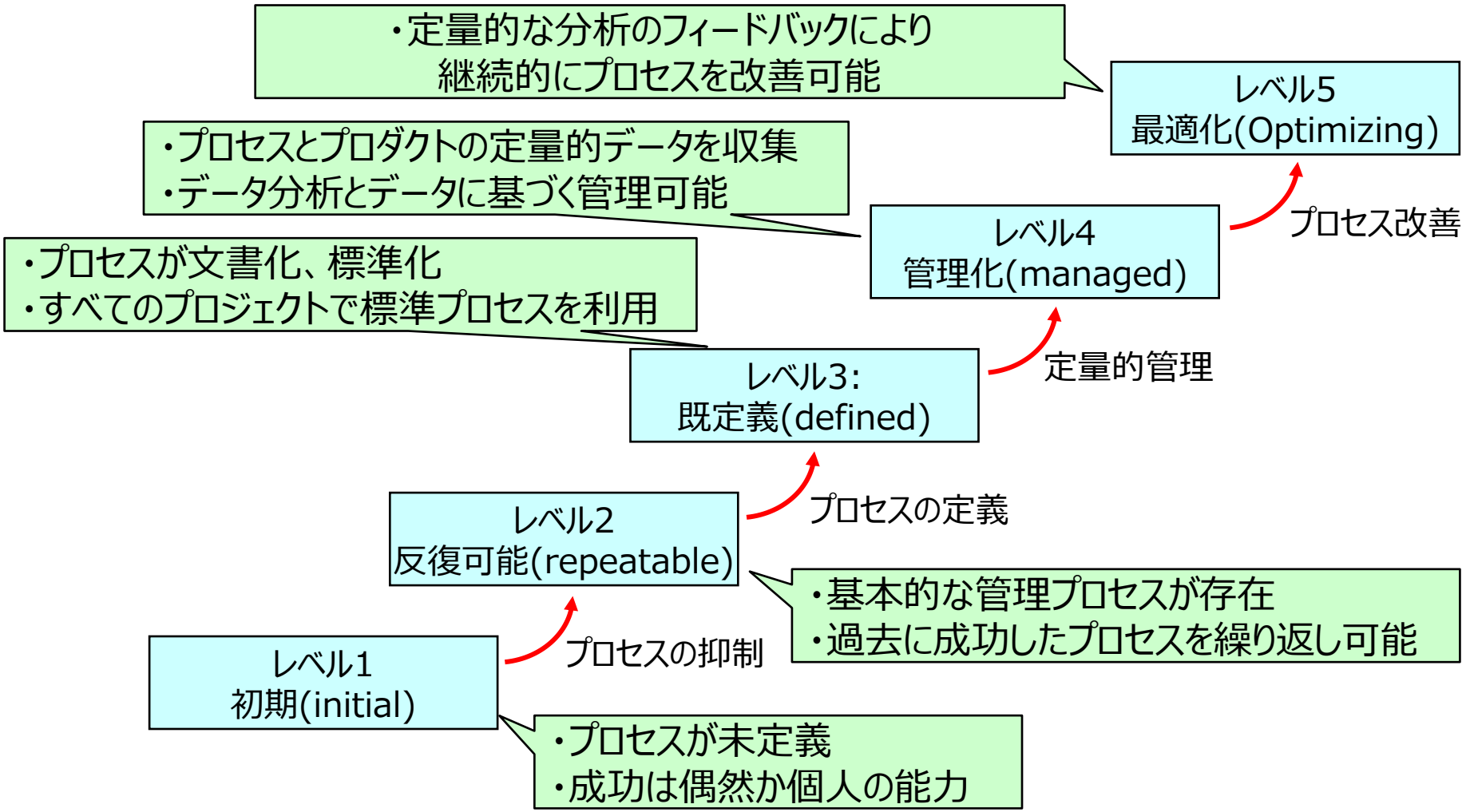
† 阿萬、野中、水野、「ソフトウェアメトリクスとデータ分析の基礎」、コンピュータソフトウェア, vol.28, no.3, 2010

プロセス成熟度

- プロダクトの品質を高めるために
  プロセスの品質を改善することが有効だと
  言われている
- CMM (capability maturity model) :
  組織におけるプロセス改善のために
  遵守すべき指針を体系化
  → 現在はCMMI (CMM integration)一体化
- SW-CMM: CMMIのうちソフトウェア開発に
  関するもの
  - 成熟度を5段階で規定

CMM是指“能力成熟度模型”，其英文全称为Capability Maturity Model for Software，英文缩写为SW-CMM，简称CMM。它是对于软件组织在定义、实施、度量、控制和改善其软件过程的实践中各个发展阶段的描述。
CMM的核心是把软件开发视为一个过程，并根据这一原则对软件开发和维护进行过程监控和研究，以使其更加科学化、标准化、使企业能够更好地实现商业目标。

SW-CMMにおける成熟度



確認問題

- 以下の説明に合う用語を答えよ
  - プロジェクト管理に関する基本的な知識体系をまとめたもの
  - ソフトウェア開発に関する定量的な評価尺度
  - 組織におけるプロセス改善のために遵守すべき指針を体系化したもの
  - 開発工数の単位。1人が1か月でできる作業の量を意味している
- 以下の各文は正しいか。○か×で答えよ。
  - 通常、プロジェクトの開発者を増やせば増やすほど、開発効率は向上する
  - 見積もりとは、過去のプロジェクトで要したコストを明らかにする作業である
  - 開発工数見積もりにおいては、工数に影響する要因をあらかじめ洗い出すことが大切である
  - 開発工数見積もりにおいては、過去の経験の蓄積・活用により正確さを向上させることができる
  - LOCが100のプログラムよりも1000のプログラムの方が必ず複雑である
  - コード中の条件分岐が増えると、サイクロマティック数も増える

確認問題

- 以下の説明に合う用語を答えよ
  - プロジェクト管理に関する基本的な知識体系をまとめたもの PMBOK
  - ソフトウェア開発に関する定量的な評価尺度 ソフトウェアメトリクス
  - 組織におけるプロセス改善のために遵守すべき指針を体系化したもの CMMI(CMM)
  - 開発工数の単位。1人が1か月でできる作業の量を意味している 1人が1か月かかる作業量
- 以下の各文は正しいか。○か×で答えよ。
  - 通常、プロジェクトの開発者を増やせば増やすほど、開発効率は向上する X
  - 見積もりとは、過去のプロジェクトで要したコストを明らかにする作業である X
  - 開発工数見積もりにおいては、工数に影響する要因をあらかじめ洗い出すことが大切である O
  - 開発工数見積もりにおいては、過去の経験の蓄積・活用により正確さを向上させることができる O
  - LOCが100のプログラムよりも1000のプログラムの方が必ず複雑である X
  - コード中の条件分岐が増えると、サイクロマティック数も増える O

ソフトウェアの品質特性 (ISO9126)

1. 機能性 (functionality) :
   必要な機能が実装されているか
2. 信頼性 (reliability) :
   機能が正常に動作し続けるか
3. 使用性 (usability) :
   利用者にとって使いやすいか
4. 効率性 (efficiency) :
   目的達成のために使用する資源は適切か
5. 保守性 (maintainability) :
   改訂作業に必要な労力は少ないか
6. 移植性 (portability) :
   他の環境へ移しやすいか

参考文献

- 「ソフトウェア工学」
  高橋直久、丸山勝久 著、森北出版、2010
- 「スクラム実践入門」
  貝瀬岳志、原田勝信、和島史典、栗林健太郎、柴田博志、家永英治 著、
  技術評論社、2015
- 「ずっと受けたかったソフトウェア開発管理の集中研修」
  宇治則孝 監修、大森久美子、岡崎義勝、西原啄夫 著、
  翔泳社、2010