



PLATEAU  
by MLIT

Handbook of 3D City Models  
3D都市モデル導入のためのガイドブック



地下埋設物等の  
3D都市モデル作成・活用マニュアル

Manual for Underground Infrastructure Modeling and Integration

series  
No. 13

# CONTENTS

0.

はじめに

1.

**地下埋設物等の現状課題と関連取組**

- 1.1 地下埋設物等を取り巻く現状と課題
- 1.2 国の最新取組

2.

**使用するデータの準備**

- 2.1 既設管路の整備に必要なデータ
- 2.2 新設・移設管路の整備に必要なデータ

3.

**地下埋設物等の3D都市モデルの新規整備・更新**

- 3.1 管路データを作る際の基本的な考え方
- 3.2 マンホール等の道路付属物の基本的な考え方
- 3.3 データ更新時の作業フロー

4.

**システム運用の考え方・体制**

- 4.1 既往システム等の運用方法・体制
- 4.2 市町村における先行事例

5.

**今後の取組展開に向けて**

付録 データ作成ツールマニュアル

# CONTENTS

0.

はじめに

1.

地下埋設物等の現状課題と関連取組

- 1.1 地下埋設物等を取り巻く現状と課題
- 1.2 国の最新取組

2.

使用するデータの準備

- 2.1 既設管路の整備に必要なデータ
- 2.2 新設・移設管路の整備に必要なデータ

3.

地下埋設物等の3D都市モデルデータ作成

- 3.1 管路データを作る際の基本的な考え方
- 3.2 マンホール等の道路付属物の基本的な考え方
- 3.3 データ更新時の作業フロー

4.

システム運用の考え方・体制

- 4.1 既往システム等の運用方法・体制
- 4.2 市町村における先行事例

5.

今後の取組展開に向けて

付録 データ作成ツールマニュアル

## 本書の目的と位置づけ

我が国において、高度経済成長期以降に集中的に整備されたインフラは、建設後50年以上を経過する施設の割合が、今後加速度的に高くなる見込みである。今後、我が国の労働人口の減少が予測される中、インフラの老朽化に伴う維持管理の負担が増加するため、インフラの維持管理に対する更なる一層の効率化が求められている。また、都市の価値・持続性の向上に資する都市再開発では、調査・設計段階において、関係するさまざまなステークホルダーが協議を重ね、合意形成を図ることが重要である。特に地下埋設物は、ステークホルダーが多岐にわたり、その調査や調整に多大な時間を要しているため、プロセスの効率化やインフラ情報の精度向上が大きな課題となっている。

また、これまでのインフラ情報は、図面が紙で保管されていたり、必要な情報が不足している等のケースも散見され、インフラ情報のデジタル化が十分に進んでこなかった。そこで、Project PLATEAUでは、地下埋設物等の3D都市モデル作成のための標準仕様化、モデル新規整備・更新方法の確立、実証モデルエリアにおける3D都市モデルの作成、インフラの維持管理や都市再開発での関係者合意形成等を支援するツール開発を通じて、当該分野のDX推進を図ってきたところである。

本書では、インフラを取り巻く社会課題への対応やProject PLATEAUの取組等を踏まえつつ、自治体等が保有する地下埋設物等(上下水道関連)の情報を活用し、地下埋設物等の3D都市モデルの新規整備や更新方法、モデルを活用するためのシステム運用の考え方、想定される利用シーン等を解説することを目的としている。デジタル技術の進歩は日進月歩であり、本書が、各自治体等において、地下埋設物等の3D都市モデルへの理解を深め、インフラのデジタルデータ管理・活用の一助となれば幸いである。

## 本書の対象者と主な想定利用シーン

- 本書では、自治体等において地下埋設物等(上下水道関連)の維持管理や、関連情報の管理運用・活用を行っている行政担当者を主な対象とする。
- また、熊本県玉名市の保有データや実証結果にもとづき、主に以下の利用シーンを想定して、3D都市モデルの考え方や具体的な作成方法を解説している。

利用シーンとしては以下の3点を想定。

1. 地下埋設物の3Dデータを新規に整備する場合
2. 地下埋設物の3Dデータを更新する場合
3. 地下埋設物の3Dデータを庁内や関連事業者等と共有し、運用・活用する場合

# CONTENTS

0.

はじめに

1.

## 地下埋設物等の現状課題と関連取組

1.1 地下埋設物等を取り巻く現状と課題

1.2 国の最新取組

2.

## 使用するデータの準備

2.1 既設管路の整備に必要なデータ

2.2 新設・移設管路の整備に必要なデータ

3.

## 地下埋設物等の3D都市モデルデータ作成

3.1 管路データを作る際の基本的な考え方

3.2 マンホール等の道路付属物の基本的な考え方

3.3 データ更新時の作業フロー

4.

## システム運用の考え方・体制

4.1 既往システム等の運用方法・体制

4.2 市町村における先行事例

5.

今後の取組展開に向けて

付録 データ作成ツールマニュアル

# 1.1 地下埋設物等を取り巻く現状と課題

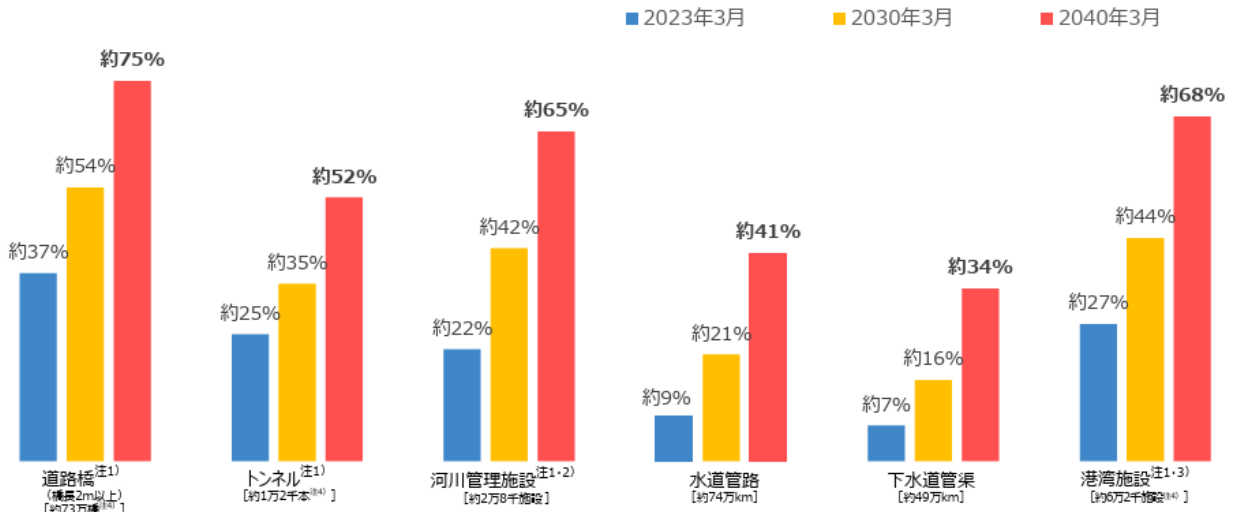
高度経済成長期以降に集中的に整備されたインフラの老朽化が、加速度的に進行すると見込まれている。例えば2025年の1月には、埼玉県八潮市の流域下水道管に起因すると考えられる道路陥没事故も発生しており、インフラの老朽化は喫緊の課題であり、計画的なインフラの維持管理・更新を進めていく必要がある。

計画的な維持管理・更新方法として、インフラに不具合が生じてから対策を行う「事後保全」から、不具合が生じる前に対策を行う「予防保全」に転換されつつある。適切なインフラの維持管理が求められる中で、多種多様なインフラを管理する自治体においては、財政面・体制面への対応が課題となっている。

## インフラの老朽化の現状・予測

【建設後50年以上経過する社会資本の割合】（2023年3月時点）

[ ] : 各施設の総数（総延長）



出典：国土交通省インフラメンテナンス情報「建設後50年以上経過する社会資本の割合」

水道・下水道の領域では、水道事業者のうち、全ての管路情報を電子媒体で管理している割合は67%（2025年度当初）、下水道事業を実施している自治体のうち、全ての管路情報を電子媒体で管理している割合は71%（2025年度当初）にとどまっており、水道・下水道情報のデジタル化は十分に進んでいない状況である。

このような現状を踏まえ、上下水道DX推進検討会では、上下水道DXの推進に向けて、各種取組の進捗状況や成果について、定期的にフォローアップするためのKPIを掲げている。次ページに示すとおり、水道・下水道の管路情報を電子媒体で管理する割合を2027年度までに100%とする目標を設定しており、これらのデジタルデータを活用することで、効率的な3D都市モデルの作成が期待される。また、水道・下水道の共通プラットフォームの導入事業者数を2027年度末に2024年度末に比ベ倍増することも目指しており、当該情報に関する統一的な仕様にもとづくデータ整備や利用環境の向上等も期待できることから、広域での3D都市モデルの整備推進も視野に入ってくるといえる。

## 上下水道DX推進に向けたKPI

- ①水道事業者※1のうち、メンテナンスに関する上下水道DX技術の導入率※2  
・2025年度当初 34% → 2027年度末 100%
- ②下水道事業を実施している自治体のうち、メンテナンスに関する上下水道DX技術の導入率※2  
・2025年度当初 21% → 2027年度末 100%
- ③水道事業者のうち、全ての管路情報を電子媒体で管理している割合  
・2025年度当初 67% → 2027年度末 100%※3
- ④下水道事業を実施している自治体のうち、全ての管路情報を電子媒体で管理している割合  
・2025年度当初 71% → 2027年度末 100%※3

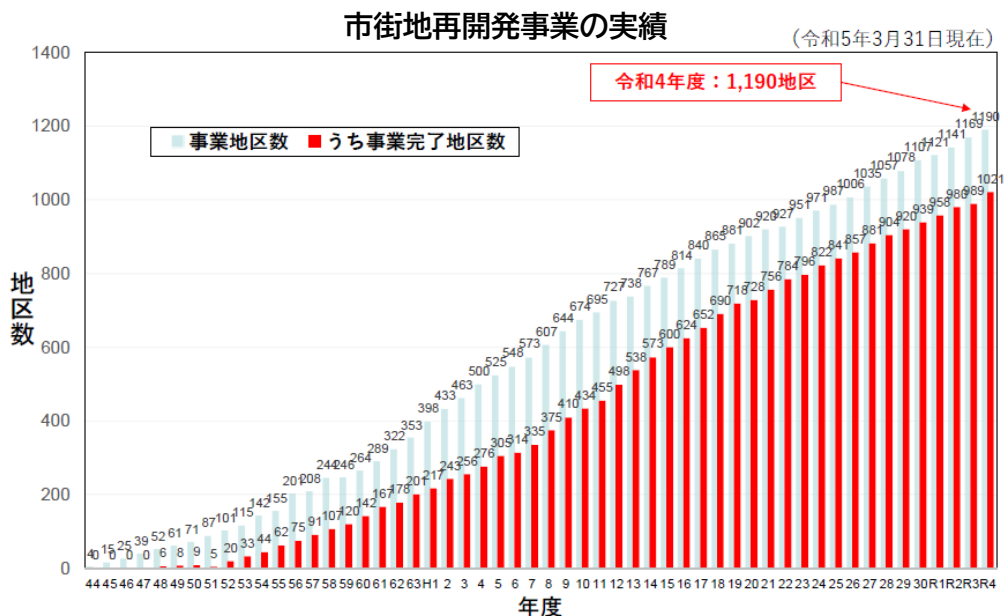
※1 水道事業者とは、上水道事業及び水道用水供給事業とし簡易水道事業は含まない。

※2 メンテナンス(上下水道施設の維持管理等)に関する上下水道DX技術の導入とは、DX技術が標準的なツールとして活用されている状態とする。ただし、DX技術により得た情報を活用し、長期的なメンテナンス計画を立案するなどの活用も含まれる。

※3 維持管理情報が含まれることや、GIS(地理情報システム)によるものが望ましいが、これらの対応が困難な場合は、文字判別可能な汎用的なファイル形式で電子化されたものを含む。

出典：最終とりまとめ報告書、上下水道DX推進検討会、2025年6月

また、都市再開発に関して、市街地再開発事業の事業地区数は、2022年度までに累計で1,190地区実施しており、そのうち、事業が完了している地区数は1,021地区である。市街地再開発事業の認可地区数は、1998年度頃をピークに減少傾向であるものの、2009年度の5地区以降、近年は20地区前後が認可されている。2022年度においては、27地区で事業が認可されており、本事業における地下埋設物等の3D都市モデルの活用が期待できる。



出典：国土交通省資料

## 1.2 国の最新取組

国における主な関連取組としては、国土交通省が推進しているProject PLATEAU、経済産業省が実施しているデジタルライフライン全国総合整備計画に関する取組が挙げられる。



### 1.2.1 国土交通省:Project PLATEAU

Project PLATEAUでは、地下埋設物等の3D都市モデル作成のための標準仕様化、モデル新規整備・更新方法の確立、実証モデルエリアにおける3D都市モデルの作成、インフラの維持管理や都市再開発での関係者合意形成等を支援するツール開発を行っている。これらの取組の中で、2023年度・2024年度にモデル地区での実証を行っており、以下に実証内容・結果の概要を紹介する。

#### (1) 2023年度の実証の目的

都市再開発を行う際などには、開発事業者と地下インフラ事業者がそれぞれ個別に調整しながら情報収集し、地下埋設物の敷設状況などを確認しながら建設設計を行い、進めていく必要がある。これまでは協議のやり直しや掘削調査の手戻り等が発生し、各ステークホルダーにとって大きな負担となっていた。そこで、本実証では、多くの地下インフラ事業者の協力のもと、地下埋設物に関する各種情報を収集し、これを3D都市モデル(地下埋設物モデル)の標準仕様に従ってデータ化する手法を開発することで、各地下埋設物情報を3次元GISデータとして整備した。さらに、標準化されたデータを活用し、開発による影響確認やBIMモデルとの統合設計が可能なシステムを開発することで、都市開発や地下インフラの維持管理業務の効率化を推進した。

#### (2) 2023年度の実証・モデル整備の対象地

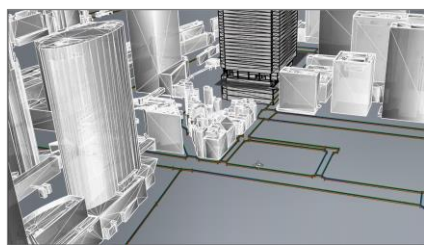
項目	内容	
実証地	大手町・丸の内・有楽町エリア (東京都千代田区・中央区)	品川駅港南口エリア (東京都港区・品川区周辺)
面積	1.21km <sup>2</sup>	1.09km <sup>2</sup>
マップ (対象エリア は赤枠内)		

### (3)2023年度に検証を行った仮説

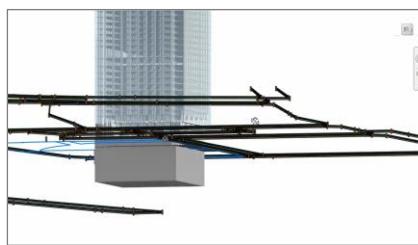
地下埋設物データを3D都市モデルで統合・可視化することで、建設設計における確認コスト削減・正確性向上・開発計画の精度向上が実現できる。また、3Dでの可視化による建設設計協議により、事業者間の共通認識醸成や課題理解・合意形成を促進することができる。下表は実際の業務フローをまとめたものであり、改善点をオレンジで示した。

業務フロー	既存フロー(2D図面での検討)	
	所要時間	備考
地下埋設物資料収集	20時間 (2.5日)	<ul style="list-style-type: none"> <li>● 埋設物情報の収集1社当たり0.5日と想定 (0.5日/社×5社=2.5日) ※埋設物情報:水道、下水道、電気、ガス、通信</li> </ul>
地下埋設物平面図・断面図作成	32時間 (4.0日)	<ul style="list-style-type: none"> <li>● 埋設物図の重ねせ平面図・断面図作成</li> <li>● 各社体裁の異なる図面の読み取り、位置合わせの作業</li> </ul>
埋設物事業者確認 (地下埋設物重ね図)	28時間 (3.5日)	<ul style="list-style-type: none"> <li>● 協議1社当たり0.5日と想定=2.5日</li> <li>● 1社から修正指示を受けたことを想定し、修正作業+再協議=1.0日</li> </ul>
地下構造物計画 平面図・断面図作成	8時間 (1.0日)	<ul style="list-style-type: none"> <li>● 地下通路を想定</li> <li>● 作図のみ。形状・配置の検討は含まない</li> <li>● 地下埋設物の図面に追記</li> </ul>
地下埋設物との 近接度合いの確認	2時間	<ul style="list-style-type: none"> <li>● 計画地下構造物の影響範囲にかかる地下埋設物の確認、最小離隔の確認</li> <li>● 平面図・断面図で影響範囲の確認+最小離隔の読み取り</li> </ul>
計画修正	3時間	<ul style="list-style-type: none"> <li>● 近接度合い確認の結果、地下埋設物との離隔を取るために地下通路位置をスライドする変更を1回実施したことを想定</li> <li>● 2D図面(平面図・断面図)で修正</li> </ul>

### (4) 2023年度に実装されたシステム及び実証会の様子



上空から見た図  
既存の建物BIMと地下埋設物を重ね合わせ



下方から見た図  
BIMデータと地下埋設物モデルを重ね合わせ



立ち合いシステム  
※計画している地下構造物と既設の埋設物の  
離間距離を確認できる



実証実験の様子


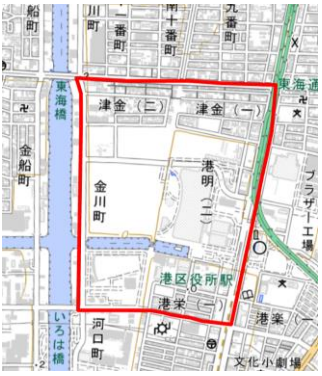

## (5) 2023年度の検証結果 ～改善された作業効率(比較表)～

業務フロー	既存フロー(2D図面での検討)		実証システムフロー(3Dでの検討)	
	所要時間	備考	所要時間	備考
地下埋設物資料収集	20時間 (2.5日)	<ul style="list-style-type: none"> <li>● 埋設物情報の収集1社当たり0.5日と想定(0.5日/社×5社=2.5日) ※埋設物情報:水道、下水道、電気、ガス、通信</li> </ul>	0.5時間	<ul style="list-style-type: none"> <li>● 実証システム立上げ</li> <li>● 計画情報入力・申請</li> <li>● 3D都市モデル閲覧</li> <li>● ダウンロード</li> </ul>
地下埋設物平面図・断面図作成	32時間 (4.0日)	<ul style="list-style-type: none"> <li>● 埋設物図の重ね合わせ平面図・断面図作成</li> <li>● 各社体裁の異なる図面の読み取り、位置合わせの作業</li> </ul>	1.5時間	<ul style="list-style-type: none"> <li>● ダウンロードしたデータをBIMアプリケーションへのインポート、編集</li> </ul>
埋設物事業者確認 (地下埋設物重ね図)	28時間 (3.5日)	<ul style="list-style-type: none"> <li>● 協議1社当たり0.5日と想定=2.5日</li> <li>● 1社から修正指示を受けたことを想定し、修正作業+再協議=1.0日</li> </ul>	1時間	<ul style="list-style-type: none"> <li>● 埋設物事業者にもメールで通知</li> <li>● 申請範囲の地下埋設物を確認</li> </ul>
地下構造物計画平面図・断面図作成	8時間 (1.0日)	<ul style="list-style-type: none"> <li>● 地下通路を想定</li> <li>● 作図のみ(形状・配置の検討は含まない)</li> <li>● 地下埋設物の図面に追記</li> </ul>	16時間 (2.0日)	<ul style="list-style-type: none"> <li>● 地下通路を想定</li> <li>● BIMソフトでの作図のみ(形状・配置は含まない)</li> <li>● 地下埋設物がインポートされた図面に追記</li> </ul>
地下埋設物との近接度合いの確認	2時間	<ul style="list-style-type: none"> <li>● 計画地下構造物の影響範囲にかかる地下埋設物の確認、最小離隔の確認</li> <li>● 平面図・断面図で影響範囲の確認+最小離隔の読み取り</li> </ul>	1時間	<ul style="list-style-type: none"> <li>● BIMソフト上で、計画地下構造物の影響範囲にかかる地下埋設物の確認、最小離隔の確認</li> </ul>
計画修正	3時間	<ul style="list-style-type: none"> <li>● 近接度合い確認の結果、地下埋設物との離隔を取るために地下通路位置をスライドする変更を1回実施したことを想定</li> <li>● 2D図面(平面図・断面図)で修正</li> </ul>	1時間	<ul style="list-style-type: none"> <li>● 近接度合い確認の結果、地下埋設物との離隔を取るために地下通路位置をスライドする変更を1回実施したことを想定</li> <li>● BIMソフトで3D都市モデルを修正</li> </ul>
合計時間	93時間(11.625日)		21時間(2.625日)	
時間短縮効果	▲約78%			

## (6) 2024年度の実証の目的

地下埋設物モデルの標準的なデータ更新手法は確立されておらず、インフラ設備の維持管理のための設備点検結果の管理方法はメンテナンス事業者ごとに異なっている。そこで、本実証では、インフラの維持管理を支援するシステムを開発することで、さまざまな事業者が関わっている都市開発やインフラの維持管理に関して、その基盤となるインフラ(ここでは地下埋設物モデル)を継続的に活用するための標準的なデータ更新手法を確立した。また、設備点検結果の標準的な管理方法やデータ項目を整理し、3D都市モデルを活用した効率的な点検作業の推進を図った。

## (7) 2024年度の実証・モデル整備の対象地

項目	内容		
実証地	東京都中央区 日本橋エリア	愛知県名古屋市	大阪府大阪市
面積	0.37km <sup>2</sup>	0.61km <sup>2</sup>	0.87km <sup>2</sup>
マップ (対象 エリア は赤枠 内)			

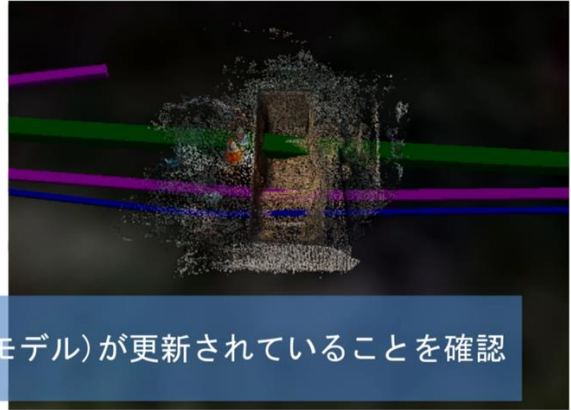
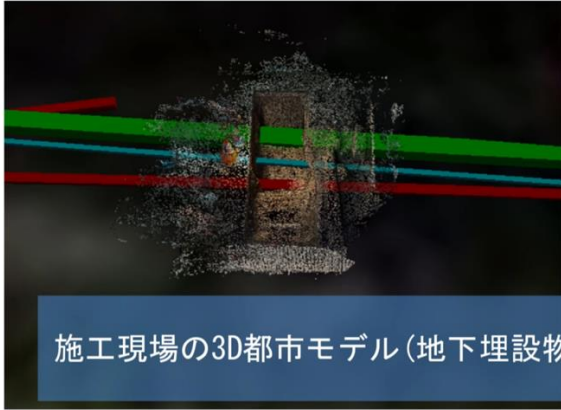
## (8) 2024年度の実証・モデル整備のターゲット

	評価指標・KPI	目標値
施工記録を活用 した更新手法の 開発	事務所内作業の 負担軽減率	実証システムによる業務フローの所要時間(min) ÷ 既存業務フローによる所要時間(min) < 1
	更新した3D都市モデル (地下埋設物モデル)の 位置精度	正確な位置とデータ更新後の位置の誤差が 地図情報レベル500相当
各種設備点検の 省力化	点検入力作業の軽減率	実証システムによる業務フローの所要時間(min) ÷ 既存業務フローによる所要時間(min) < 1

## (9)2024年度に実装されたシステムイメージ・実証の様子

### 設備データ更新前後の比較

■ : ガス管 (更新前)   ■ : ガス管 (更新後)  
■ : 水道管 (更新前)   ■ : 水道管 (更新後)  
■ : 通信管 (更新前)   ■ : 通信管 (更新後)



施工現場の3D都市モデル(地下埋設物モデル)が更新されていることを確認

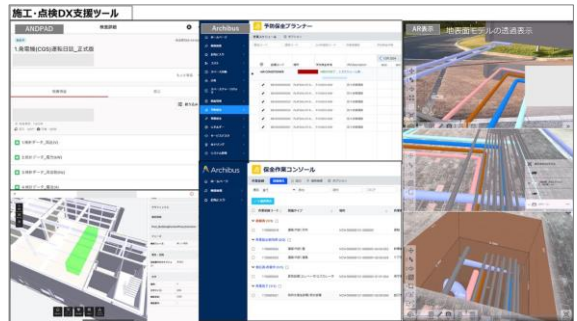
設備データ更新前

設備データ更新後

本システムでは、実際に作業した場所のデータが正しく更新されているかを検証することが求められていた。



実際に更新作業を行っている際に  
管路を上から撮影し、3D化をしている



各種設備点検の省力化が行えるように  
結果の利活用をすることができる



ワークショップにて  
3D化した管路を確認している様子



ワークショップにて  
上記ツールの有用性を検証した

## (10)2024年度の検証結果 ～作業時間効率化・データ精度の確保～

	評価指標・KPI	目標値	検証結果	結果概要
施工記録を活用した更新手法の開発	事務所内作業の負担軽減率	実証システムによる業務フローの所要時間(min) ÷既存業務フローによる所要時間(min)<1	・62%の時間短縮	<ul style="list-style-type: none"> <li>事務所内作業を約62%軽減でき、竣工図作成以降の工程で大きな時間短縮効果が確認された</li> <li>後続の図面作成とデータ更新で大幅な業務効率化を実現した。</li> </ul>
	更新した3D都市モデル(地下埋設物モデル)の位置精度	正確な位置とデータ更新後の位置の誤差が地図情報レベル500相当	・地図情報レベル500の許容誤差基準(標準偏差25cm以内)を達成	<ul style="list-style-type: none"> <li>地下埋設物3次元計測ツールは相対位置精度1mm以内で計測し、マンホール位置情報を基準とした補正により地図情報レベル500相当の位置精度を確保した。</li> </ul>
各種設備点検の省力化	点検入力作業の軽減率	実証システムによる業務フローの所要時間(min) ÷既存業務フローによる所要時間(min)<1	<ul style="list-style-type: none"> <li>名古屋: 24%の時間短縮</li> <li>大阪: 56%の時間短縮</li> </ul>	<ul style="list-style-type: none"> <li>施工管理・点検DX支援ツールの使用により、コジェネレーション(発電機のみ)の点検入力作業で約25%、ターボ冷凍機の点検入力作業で約56%の時間短縮が見込まれることを確認した。</li> </ul>

## 1.2.2 経済産業省:デジタルライフライン全国総合整備計画

経済産業省が推進している本計画では、主にデータのマネジメント及び運用管理について実施検討を行いつつ、埋設物照会に関するアプリケーションや災害時設備状況共有アプリケーション等の開発・実証を行っている。

### デジタルライフライン全国総合整備計画の全体像

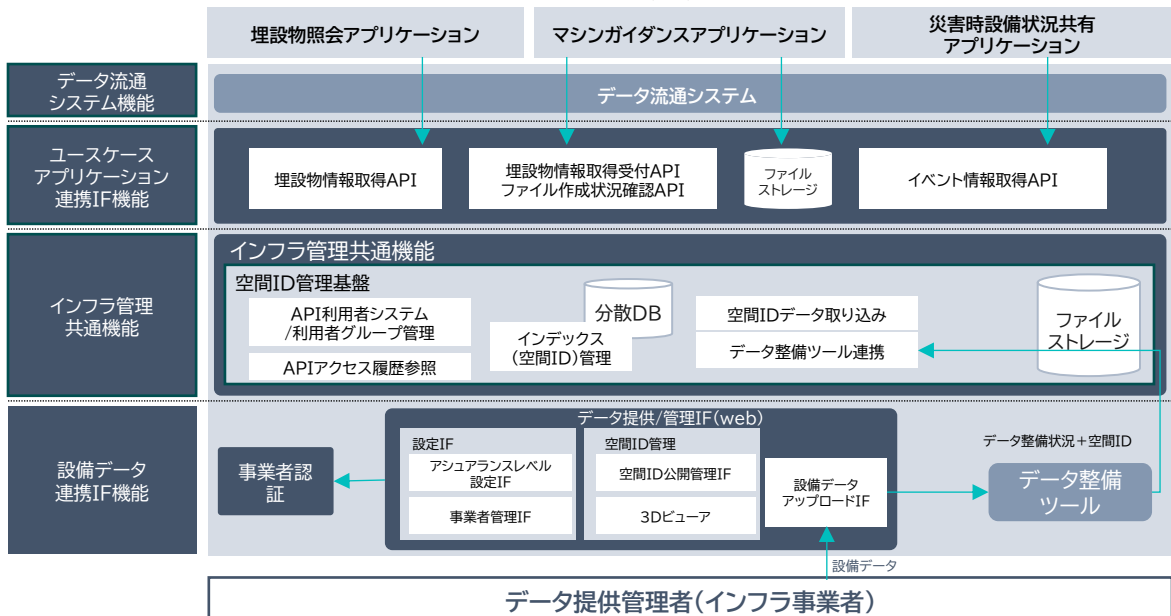


出典:「産業DXのためのデジタルインフラ整備事業」を基に作成

本事業の中で開発されたシステムと、これらの連携方法は下図のようになっており、開発成果はセキュリティ上の懸念があるもの以外はOSSとして公開されている。

【参照:ODS-IS-IMDX・GitHub】

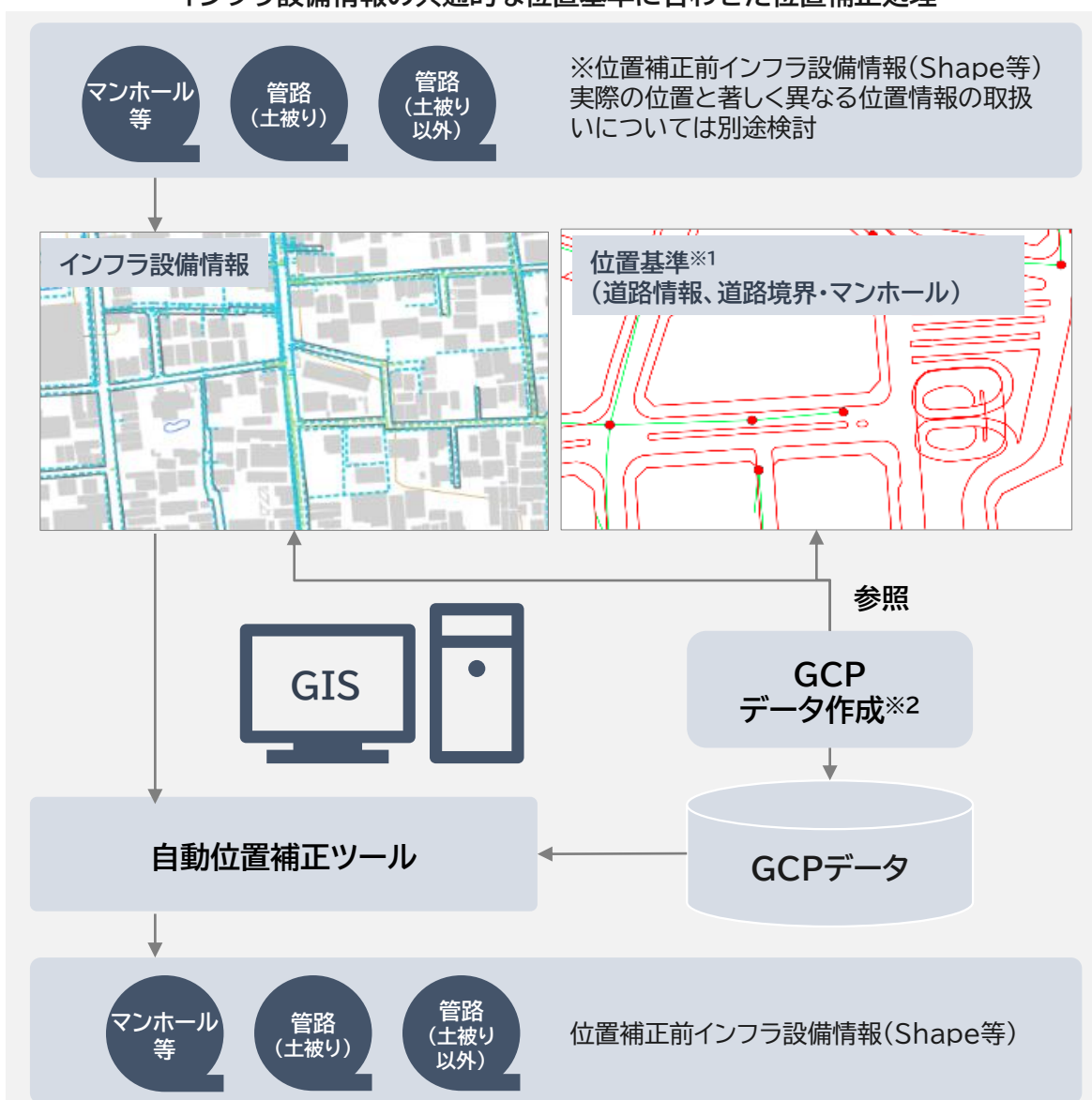
### OSSの全体提供概要



出典:「産業DXのためのデジタルインフラ整備事業」を基に作成

開発されたツールの中には、GISデータの管理などを平易に行うことができるプロダクトもある。例えば、GISのデータを保有している自治体であれば、インフラ関連のデータを一つ位置合わせすることによって、ほかのデータについても自動的に位置補正できるようになる(下水道の情報を位置合わせすれば、当該データを基盤としてマンホール等の情報についても付随して位置合わせすることができる)。

### インフラ設備情報の共通的な位置基準に合わせた位置補正処理

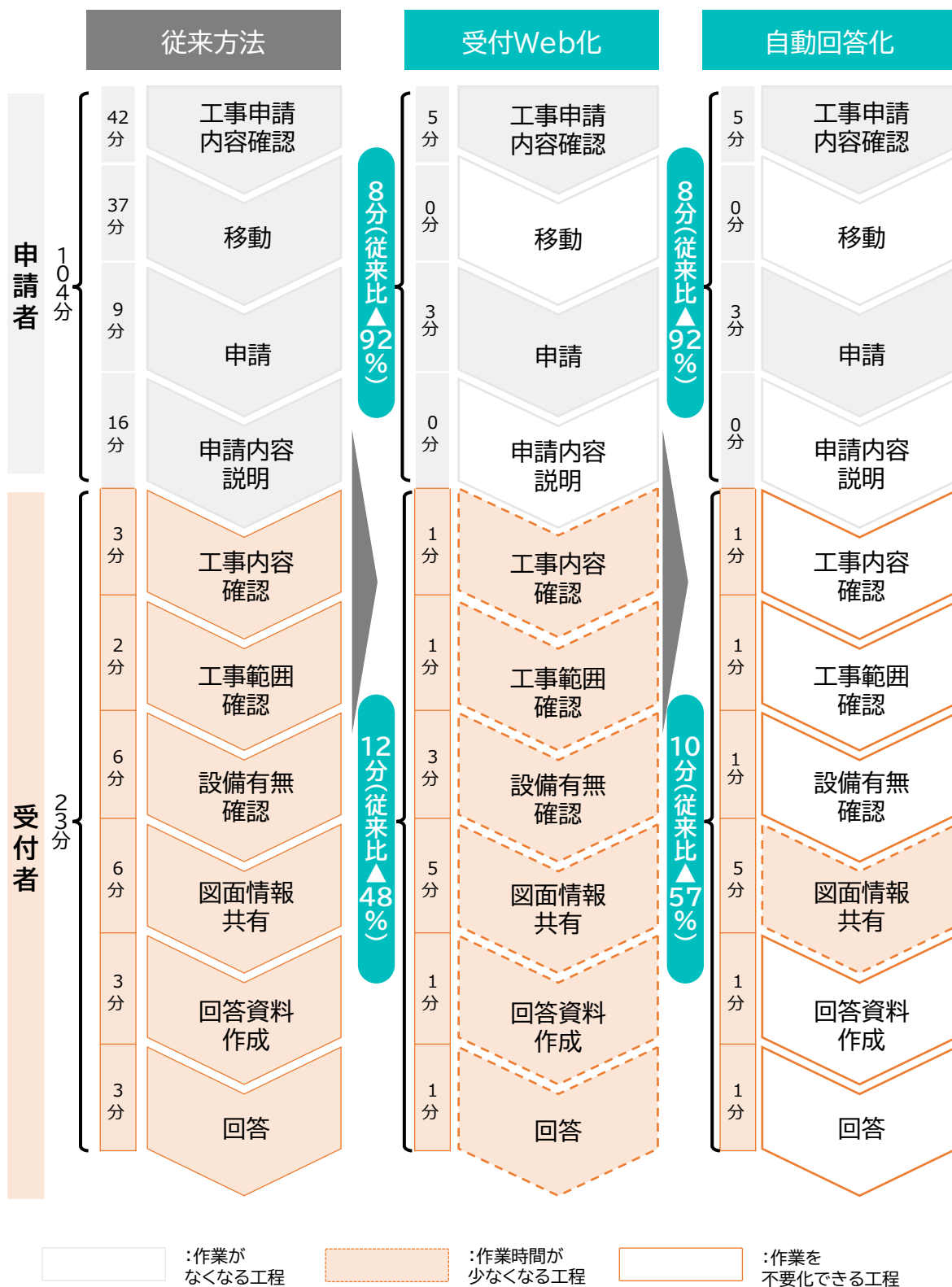


※1位置基準としては、地図情報レベル500相当の位置精度が担保されている、道路境界・マンホールの位置等を想定する。基本的には道路台帳付図などが考えられる。

※2 GCP(Ground Control Point)データ作成は、インフラ設備情報のマンホールと、位置基準のマンホールで同一と推定するマンホールを関係づけたデータを作成する。マンホールの形状が小さく位置基準にマンホールが収録されていない場合は、インフラ設備情報で利用している背景地図の道路境界で代用することも可能。GCPは元のインフラ設備情報の位置精度にもよるが、一定の面積ごとに数か所作成する。

出典:「産業DXのためのデジタルインフラ整備事業」を基に作成

本実証実験では、埋設物照会アプリケーションの活用により、申請者の業務を大幅に削減できることを確認した。詳細は下図のとおりである。また、受付者の業務量も半分以上に低減できることが明らかになった。これらの結果から、地下埋設物に関する3Dデータ活用への需要が見込まれる。



出典:「産業DXのためのデジタルインフラ整備事業」を基に作成

# CONTENTS

0. はじめに
1. 地下埋設物等の現状課題と関連取組
  - 1.1 地下埋設物等を取り巻く現状と課題
  - 1.2 国の最新取組
2. 使用するデータの準備
  - 2.1 既設管路の整備に必要なデータ
  - 2.2 新設・移設管路の整備に必要なデータ
3. 地下埋設物等の3D都市モデルデータ作成
  - 3.1 管路データを作る際の基本的な考え方
  - 3.2 マンホール等の道路付属物の基本的な考え方
  - 3.3 データ更新時の作業フロー
4. システム運用の考え方・体制
  - 4.1 既往システム等の運用方法・体制
  - 4.2 市町村における先行事例
5. 今後の取組展開に向けて

付録 データ作成ツールマニュアル

## 2.1 既設管路の整備に必要なデータ

3D都市モデル標準製品仕様書では、地下埋設物に関する地物が以下のように定義されており、本書では、このうち赤枠の四つの項目を対象として解説する。

### 2.1.1 対象とする地下埋設物等の項目

地下埋設物等の項目	
水道管	マンホール
下水道管	消火栓等の弁系列
熱供給管	U字溝(トラフ)
ガス管	ハンドホール
通信ケーブル	電気ケーブル

### 2.1.2 自治体が保有する原典情報(玉名市の例:2Dデータ)

各自治体においては、以下に例示する庁内型GIS(2DのGISデータ)や関連図面・台帳を保有しており、これらの情報を活用して3D都市モデルを構築する。なお、本書では、熊本県玉名市の保有データを例に、原典情報をGISデータで保有しているケースを想定して解説する(具体のモデル整備・更新方法は次ページ以降を参照)。



熊本県玉名市の庁内GISに保存されているデータ(参考)	
管路データ	マンホール等
管路のGISデータ	マンホールの位置データ
管厚	人孔の内径
口径/管径	外径
外径	人孔キー(ID)(主題属性の一部)
竣工年度(主題属性の一部)	地盤高(主題属性の一部)
管路形状(主題属性の一部)	メーター口径(主題属性の一部)
管路形状(主題属性の一部)	竣工年度(主題属性の一部)
上流人孔管ID/下流人孔管ID(主題属性の一部)	人孔種別(主題属性の一部)
上流管底高/下流管底高(主題属性の一部)	メーター番号(主題属性の一部)
材質(主題属性の一部)	消火栓の型式ID(主題属性の一部)
勾配(主題属性の一部)	消火栓の種別(主題属性の一部)
管径/土被り	弁の区分(主題属性の一部)
管路の区間延長(主題属性の一部)	弁の種類(主題属性の一部)
処分区間の名称(主題属性の一部)	上流土被/下流土被

## 2.1.3 水道管及び下水道関連の必要情報

水道管及び下水道関連で必要な情報は以下のとおりであり、データの取扱いやすさなどの観点から、本ガイドラインではLOD2までのモデルを対象とする。

対象 LOD	空間属性の型	取得基準	取得方法	補足	作成時に必要な情報
LOD0	Multi Curve	<ul style="list-style-type: none"> <li>水道管</li> <li>下水道管</li> <li>指定された管径よりも小さい管路</li> </ul>	<ul style="list-style-type: none"> <li>管路の中心線を取得する</li> <li>高さは0とする</li> </ul>	—	<ul style="list-style-type: none"> <li>管路のGISデータ (前ページのシステムイメージ参照)</li> <li>主題属性データ (前ページの図面イメージ参照)</li> </ul>
LOD0	Multi Surface	<ul style="list-style-type: none"> <li>水道管</li> <li>下水道管</li> <li>指定された管径よりも大きい管路</li> </ul>	<ul style="list-style-type: none"> <li>管路の外縁を取得する</li> <li>高さは0とする</li> </ul>	—	<ul style="list-style-type: none"> <li>口径/管径</li> <li>管厚</li> </ul>
LOD1	Solid	<ul style="list-style-type: none"> <li>水道管</li> <li>下水道管</li> </ul>	<ul style="list-style-type: none"> <li>管路の上方からの正射影の外周を取得する</li> <li>外周を地表面から一律の高さで下向きに押し出した立体を作成する</li> </ul>	<ul style="list-style-type: none"> <li>一律の高さは、地表面から管路の下端までとする。</li> </ul>	<ul style="list-style-type: none"> <li>外径</li> </ul>
LOD2	Solid	<ul style="list-style-type: none"> <li>水道管</li> <li>下水道管</li> </ul>	<ul style="list-style-type: none"> <li>管路の上方からの正射影の外周を取得し、管路が埋設された深さから一律の高さで下向きに押し出した立体を作成する</li> </ul>	<ul style="list-style-type: none"> <li>管路が埋設された深さは、管路の土被り量とする。一律の高さは、管路の外径とする。</li> </ul>	<ul style="list-style-type: none"> <li>土被り</li> </ul>
LOD3	Composite Surface 又は Multi Surface	<ul style="list-style-type: none"> <li>水道管</li> <li>下水道管</li> </ul>	<ul style="list-style-type: none"> <li>管路の外形(外側から見える形)を構成する面を取得する</li> <li>面の各頂点に管路の高さを与える</li> </ul>	<ul style="list-style-type: none"> <li>曲面の場合は、データセットが採用する地図情報レベルの水平及び高さの誤差の標準偏差に収まるよう平面に分割する。</li> <li>面の位相が必要な場合、Composite Surfaceを使用する。</li> </ul>	—
LOD4	Composite Surface 又は Multi Surface	<ul style="list-style-type: none"> <li>水道管</li> <li>下水道管</li> </ul>	<ul style="list-style-type: none"> <li>管路の外形(外側から見える形)を構成する面を取得する</li> <li>管路の内空を構成する面を取得する</li> <li>面の各頂点に管路の高さを与える</li> </ul>	<ul style="list-style-type: none"> <li>曲面の場合は、データセットが採用する地図情報レベルの水平及び高さの誤差の標準偏差に収まるよう平面に分割する。</li> <li>面の位相が必要な場合、Composite Surfaceを使用する。</li> </ul>	—

## 2.1.4 水道管及び下水道関連のモデルイメージ(LOD2)

管路の用途(単体又は複数)に応じてLOD2のモデルは以下のように大別される。

地下埋設物モデル(LOD2)の取得イメージ	
取得例	
用途	<p>管路(単体)</p> <p>管路(複数)</p>
説明	<p>管路の上方からの正射影の外周を、管路が埋設された深さから、管径の大きさで下向きに押し出した立体として表現する</p> <p>複数まとめて埋設されている管路の最外縁を外周とし、その上方からの正射影を、管路が埋設された深さから管径の合計の大きさで、下向きに押し出した立体として表現する</p>

## 2.1.5 水道管及び下水道関連の属性情報

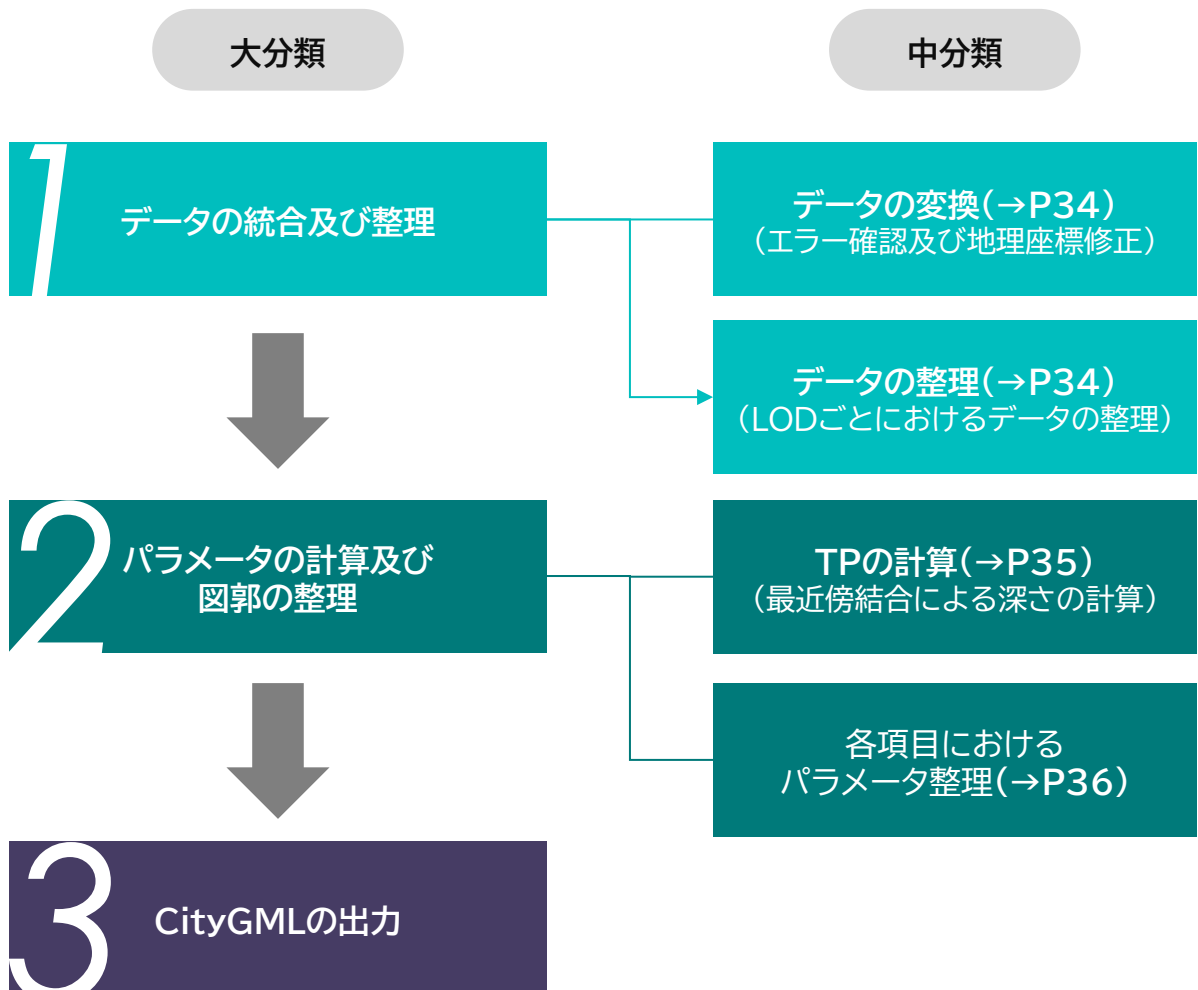
対象とする管路モデルには、以下の主題属性情報を付与することが望ましく、各種シミュレーションや分析等への活用が期待できる。

管路における主題属性情報(参考)	
上流人孔管ID	下流人孔管ID
上流管底高(m)	下流管底高(m)
竣工年度	材質
勾配	管路形状
管路の区間延長	処分区間の名称

## 2.1.6 水道管及び下水道関連におけるデータ作成フロー

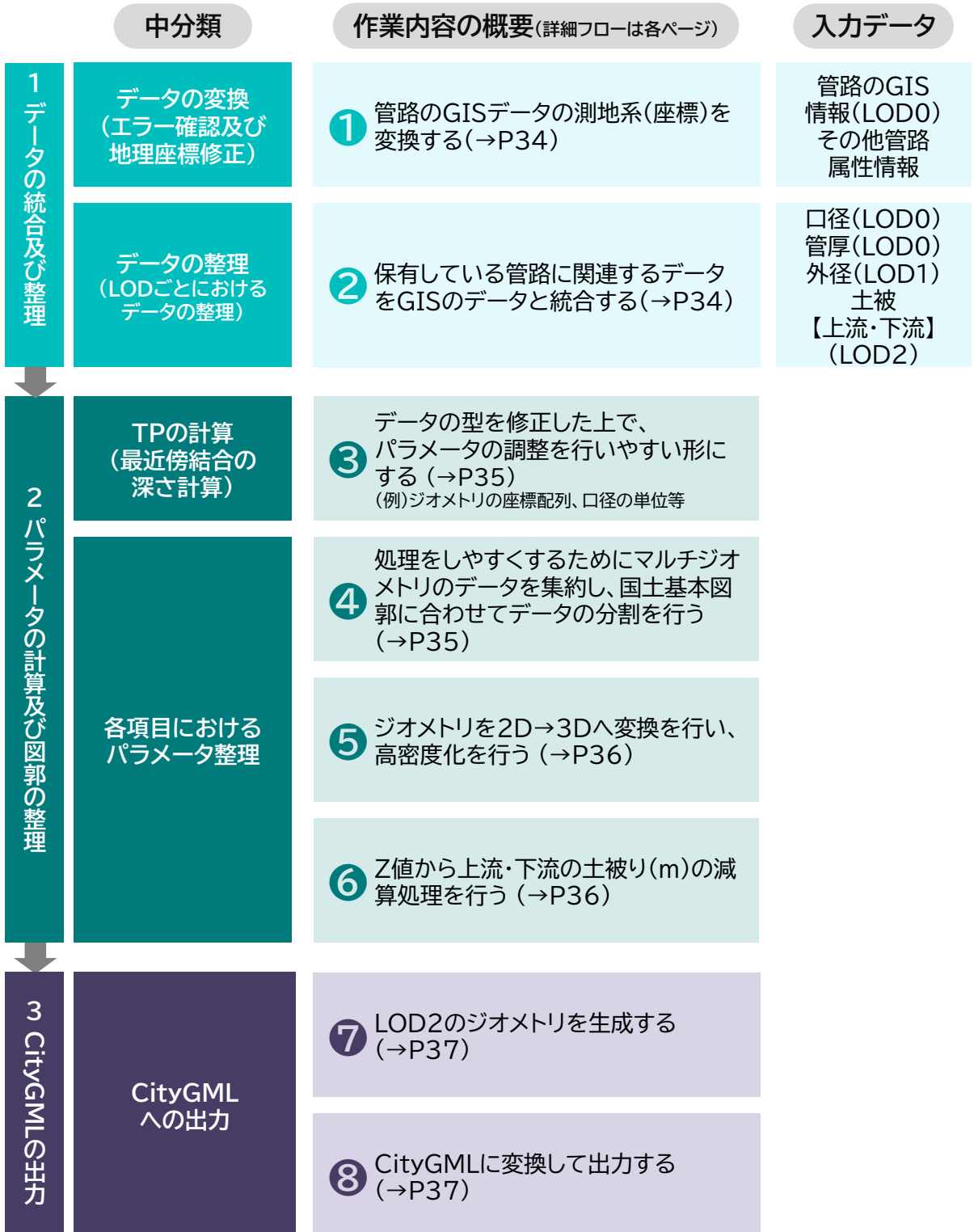
管路のデータ整備に関する作業手順は、以下に示すフローのように概略を整理することができ、大分類の作業概要は中分類の内容に区分される。

### 地下埋設物を構築する際の作業フロー(概略を整理)



## 2.1.7 水道管及び下水道関連の実施項目とデータの関係

さらに中分類の内容を細分化すると、以下のようになり、モデル整備に関連するデータは初期段階においてデータを統合・整理することによって、作業の効率化につなげることが可能になる。



## 2.1.8 マンホール及び消火栓等の必要情報

マンホールのモデル作成に必要とされるデータは以下のとおりであり、データの取扱いやすさなどの観点から、本ガイドラインではLOD2までのモデルを対象とする。

対象LOD	空間属性の型	取得基準	取得方法	補足	作成時に必要な情報
LOD0	Point	<ul style="list-style-type: none"> <li>マンホール</li> </ul>	<ul style="list-style-type: none"> <li>マンホールの蓋の中心位置を取得する</li> <li>高さは0とする</li> </ul>	—	<ul style="list-style-type: none"> <li>マンホールの位置データ</li> <li>主題属性データ</li> </ul>
LOD1	Solid	<ul style="list-style-type: none"> <li>マンホール</li> </ul>	<ul style="list-style-type: none"> <li>マンホールの蓋の上方からの正射影を包含する矩形を取得する</li> <li>矩形を地表面から一律の高さで下向きに押し出した立体を作成する</li> </ul>	<ul style="list-style-type: none"> <li>一律の高さは、地表面からマンホールの最下端までとする</li> </ul>	<ul style="list-style-type: none"> <li>人孔の内径</li> </ul>
LOD2	Solid	<ul style="list-style-type: none"> <li>マンホール</li> </ul>	<ul style="list-style-type: none"> <li>マンホールの規格により定まる外径とマンホールの中心を基に推定したマンホール本体の上方からの正射影の外周を、地表面からマンホールの深さで下向きに押し出した立体を作成する</li> </ul>	<ul style="list-style-type: none"> <li>一律の高さは、マンホールの高さとする</li> </ul>	<ul style="list-style-type: none"> <li>外径 (以下二つは管路から取得) 上流土被 下流土被</li> </ul>
LOD3	Composite Surface又はMultiSurface	<ul style="list-style-type: none"> <li>マンホール</li> </ul>	<ul style="list-style-type: none"> <li>マンホールの外形(外側から見える形)を構成する面を取得する</li> <li>面の各頂点にマンホールの高さを与える</li> </ul>	<ul style="list-style-type: none"> <li>曲面の場合は、データセットが採用する地図情報レベルの水平及び高さの誤差の標準偏差に収まるよう平面に分割する</li> <li>面の位相が必要な場合、Composite Surfaceを使用する</li> </ul>	—
LOD4	Composite Surface又はMultiSurface	<ul style="list-style-type: none"> <li>マンホール</li> </ul>	<ul style="list-style-type: none"> <li>マンホールの外形(外側から見える形)を構成する面を取得する</li> <li>マンホールの内部を構成する面を取得する</li> <li>面の各頂点にマンホールの高さを与える</li> </ul>	<ul style="list-style-type: none"> <li>曲面の場合は、データセットが採用する地図情報レベルの水平及び高さの誤差の標準偏差に収まるよう平面に分割する</li> <li>面の位相が必要な場合、Composite Surfaceを使用する</li> </ul>	—

## 2.1.9 マンホール及び消火栓等のモデルイメージ(LOD2)

マンホール及び消火栓等に関して、LOD2のモデルは以下のように大別される。

地下埋設物モデル(LOD2)の取得イメージ	
取得例	
用途	<div style="display: flex; justify-content: space-around;"> <div>消火弁</div> <div>マンホール</div> </div>
説明	<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>弁栓類、ガバナ等の設備を包含する矩形の正射影を、設備が埋設された深さから設備の高さで下向きに押し出した立体として表現する</p> </div> <div style="width: 45%;"> <p>マンホール本体の上方からの正射影の外周を、地表面からマンホールの深さで下向きに押し出した立体を作成する</p> </div> </div>

## 2.1.10 マンホール及び消火栓等における主題属性

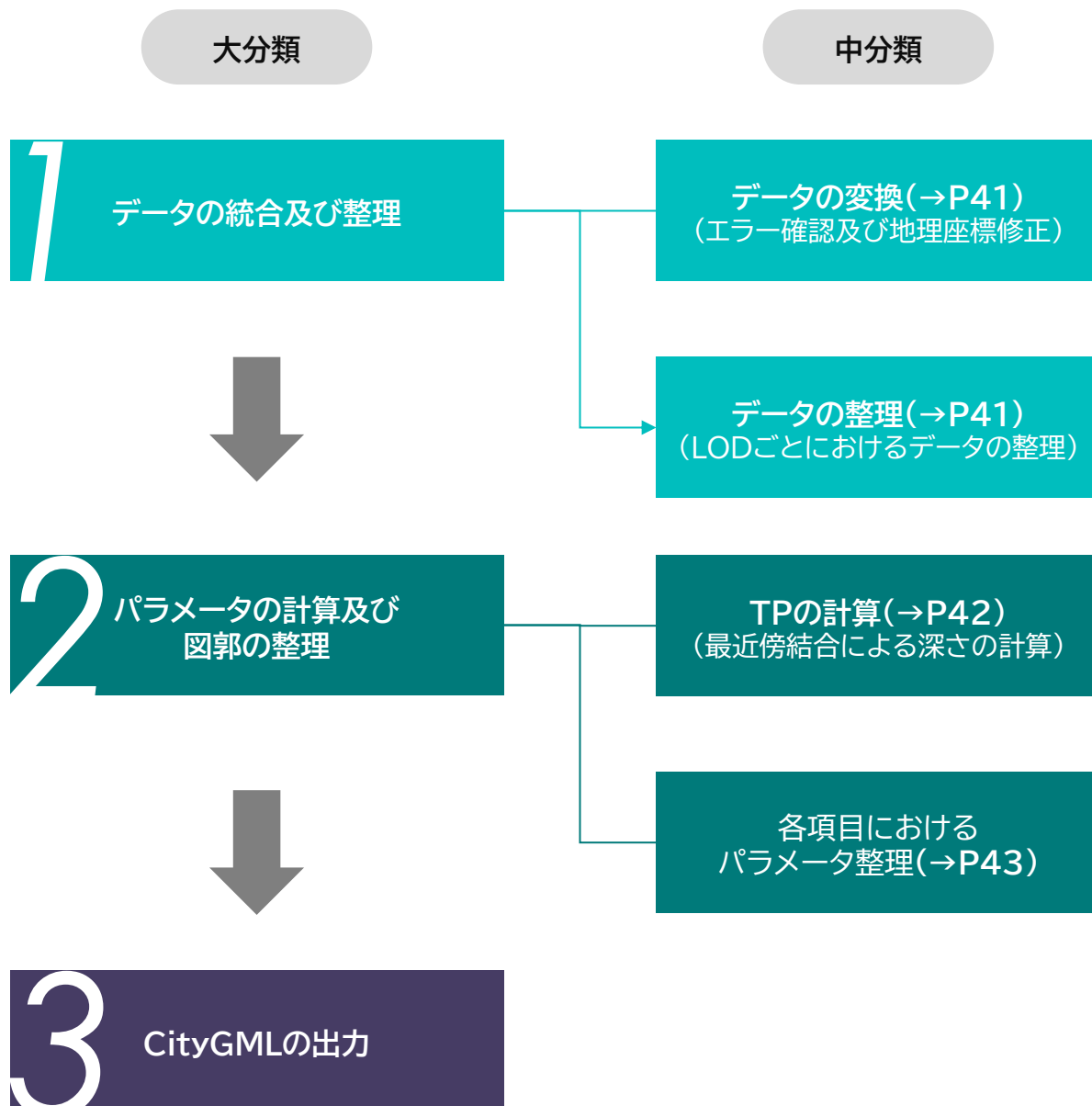
マンホール及び消火栓等には、以下の主題属性情報を付与することが望ましく、各種シミュレーションや分析等への活用が期待できる。

マンホール及び消火栓等における主題属性情報(参考)	
人孔キー(ID)	人孔種別
竣工年度	地盤高(m)
メーター番号	メーター口径
消火栓の型式ID	消火栓の種別
弁の区分	弁の種類

## 2.1.11 マンホール及び消火栓等のデータ作成フロー

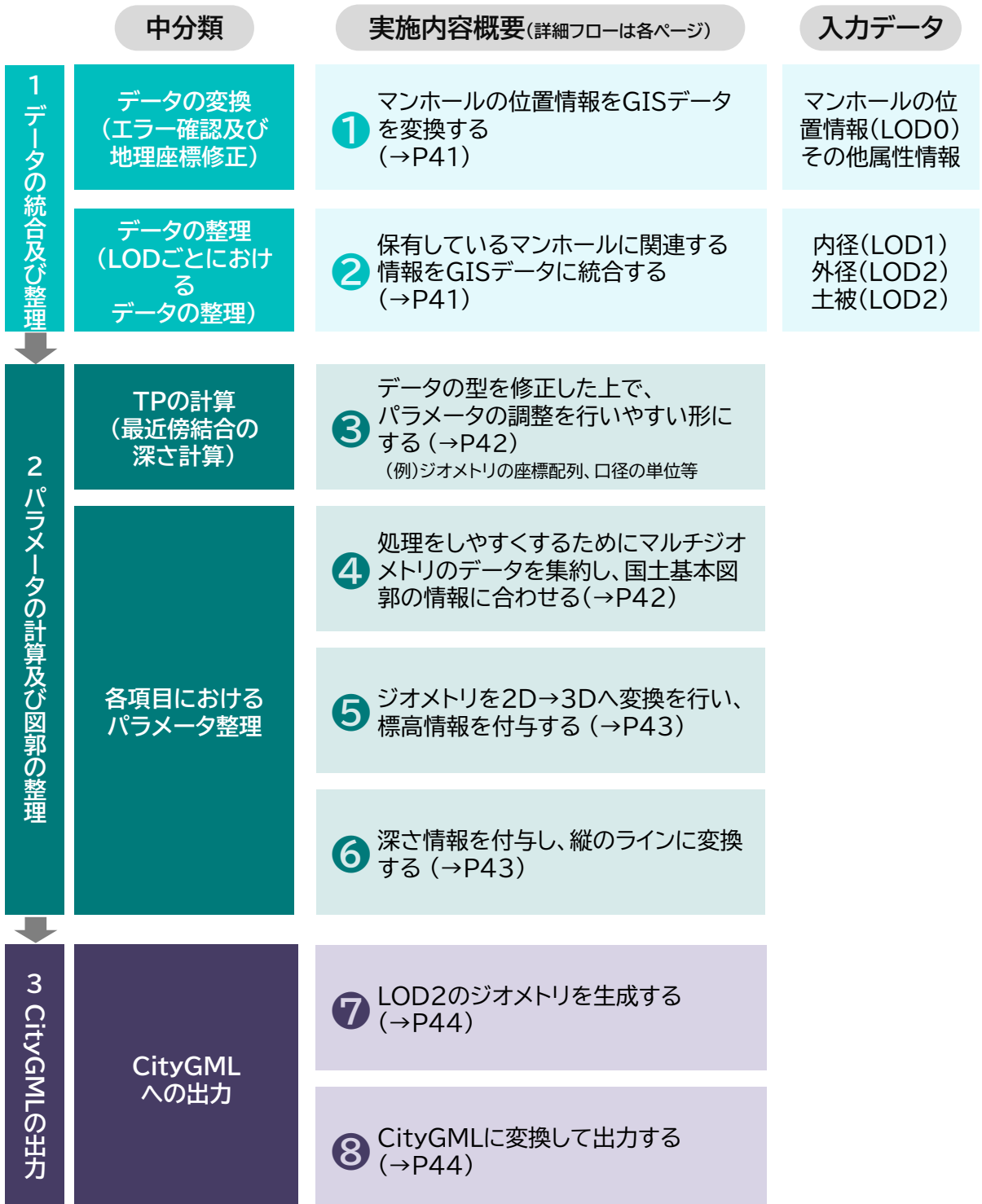
マンホール及び消火栓等のデータ整備に関する作業手順は、以下に示すフローのように概略を整理することができ、大分類の作業概要は中分類の内容に区分される。

マンホール及び消火栓等のデータを構築する際の作業フロー(概略を整理)



## 2.1.12 マンホール及び消火栓等の作業項目と入力データの関係

さらに中分類の内容を細分化すると、以下のようになり、モデル整備に関連するデータは初期段階においてデータ統合・整理することによって、作業の効率化につなげることが可能になる。

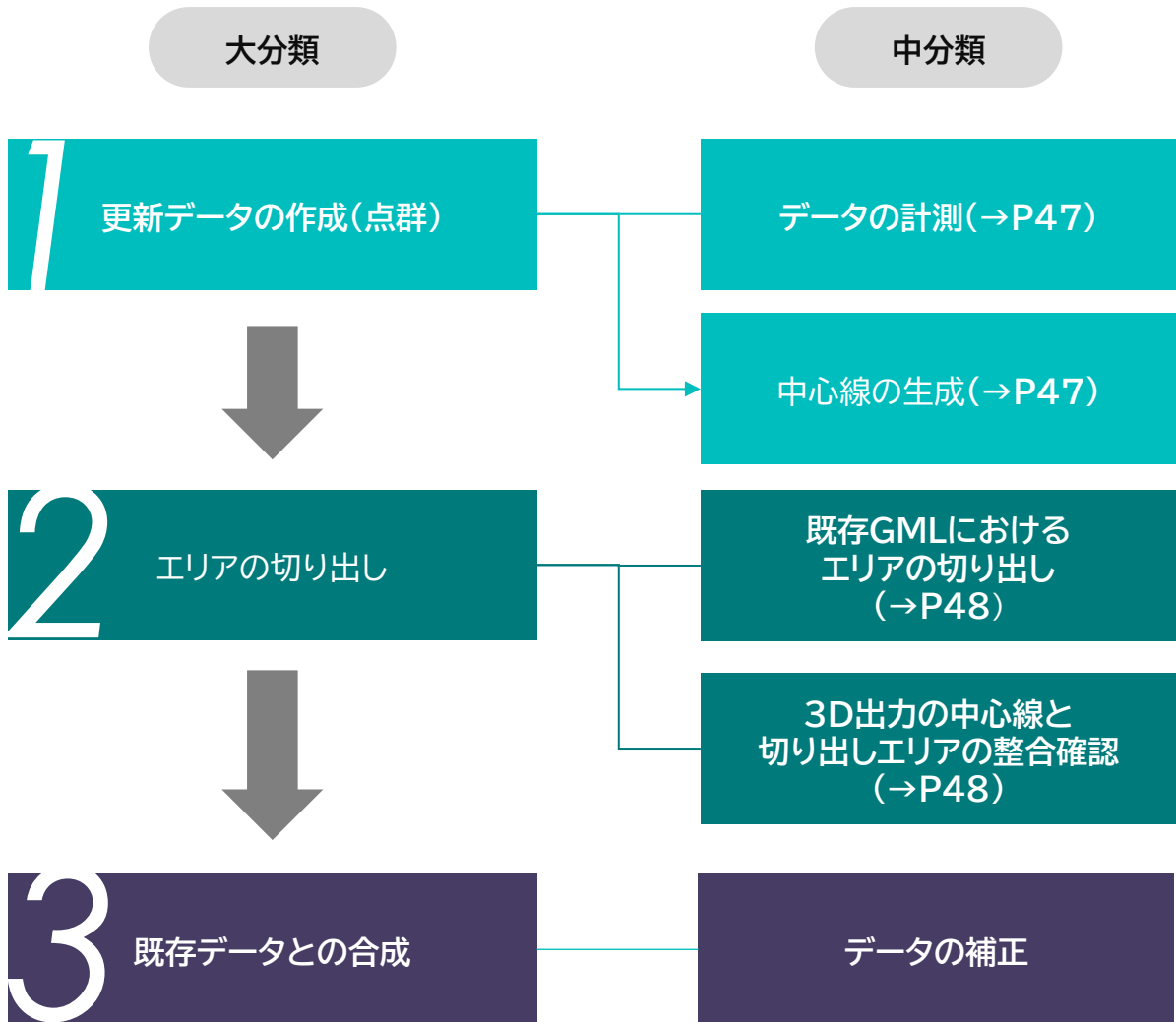


## 2.2 新設・移設管路の整備に必要なデータ

### 2.2.1 既設の上水道・下水道のデータ更新

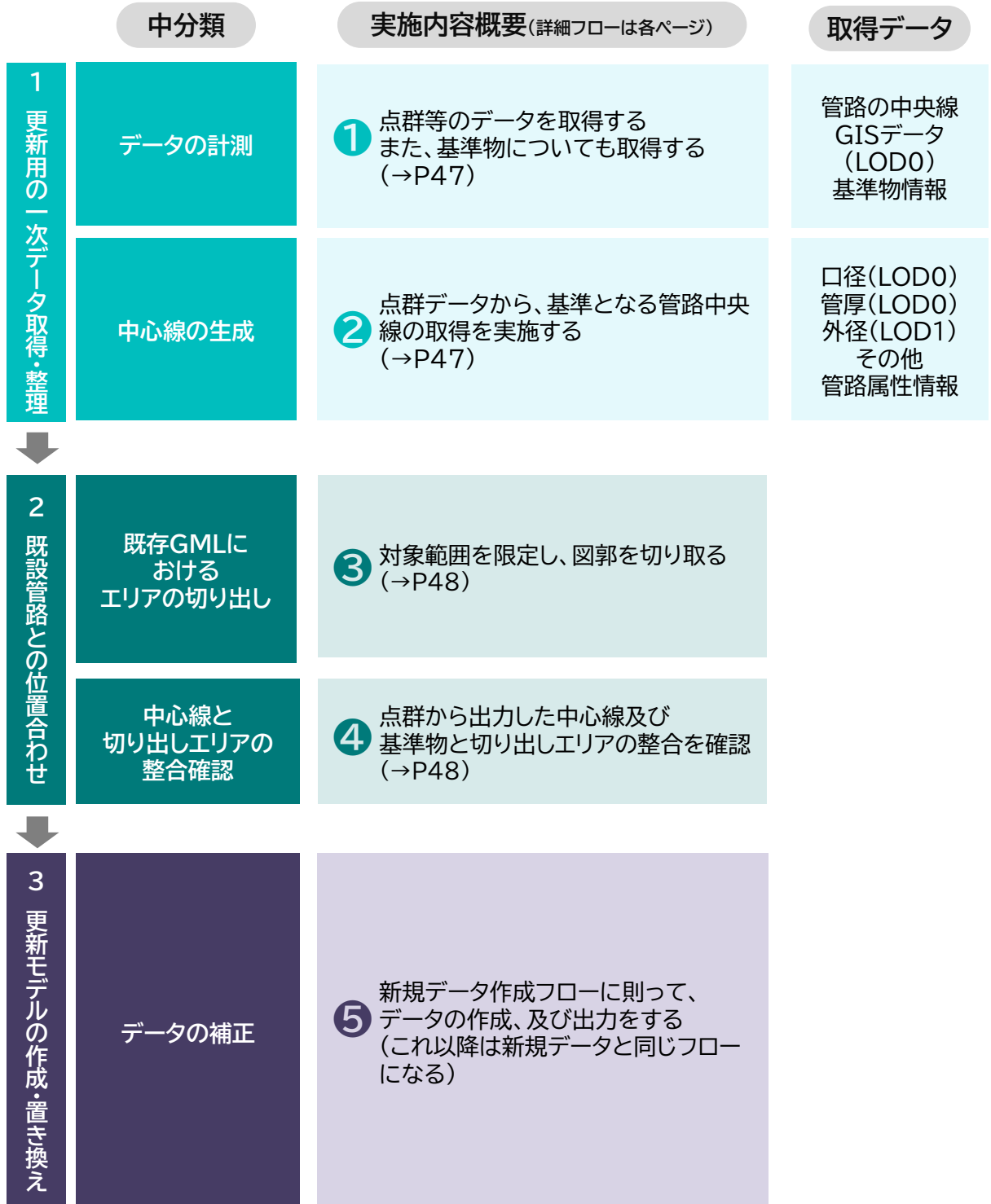
新設・移設管路の場合には、現在整備されているデータに追加を行う形になる。基本的にはLOD2のデータを基準として構築を行い、3Dの取得方法によってデータの取扱い方は異なるものの、LOD0に該当する口径・管厚データに関しては保有している、又はこれまで使用しているデータと同等とみなして事前に連携を行う必要がある。

地下埋設物を更新する際のフロー(概観)



## 2.2.2 データ更新時の作業フロー一覧

さらに中分類の内容を細分化すると、以下のようになり、モデル更新に関連するデータは初期段階においてデータを統合・整理することによって、作業の効率化につなげることが可能になる。



# CONTENTS

0.

はじめに

1.

地下埋設物等の現状課題と関連取組

1.1 地下埋設物等を取り巻く現状と課題

1.2 国の最新取組

2.

使用するデータの準備

2.1 既設管路の整備に必要なデータ

2.2 新設・移設管路の整備に必要なデータ

3.

地下埋設物等の3D都市モデルデータ作成

3.1 管路データを作る際の基本的な考え方

3.2 マンホール等の道路付属物の基本的な考え方

3.3 データ更新時の作業フロー

4.

システム運用の考え方・体制

4.1 既往システム等の運用方法・体制

4.2 市町村における先行事例

5.





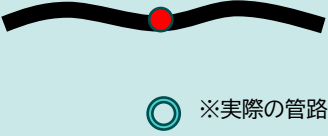
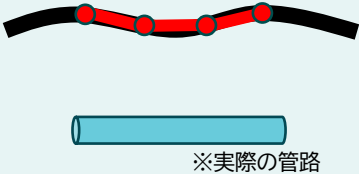
今後の取組展開に向けて

付録 データ作成ツールマニュアル

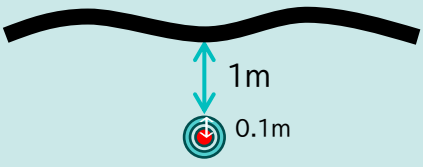
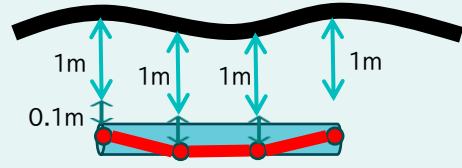
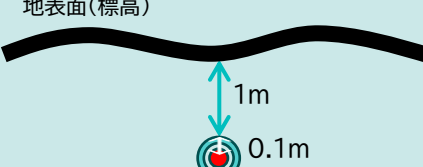
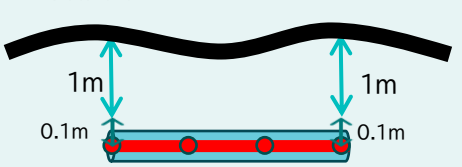
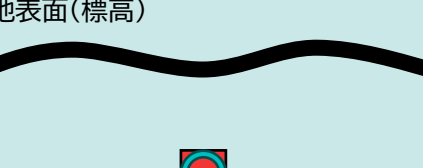
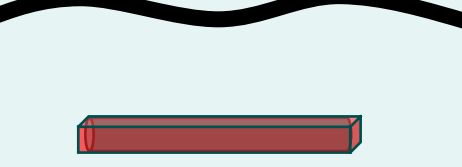
## 3.1 管路データを作る際の基本的な考え方

地下埋設物、特に管路データは以下に示すイメージでLOD2モデルを構築する。基本的な概略は下表のように、ポイント情報から管路を作成し、高さを与えて3D化することが想定される。

### 3.1.1 管路LOD2のデータを構築する基本概念(作業イメージ)

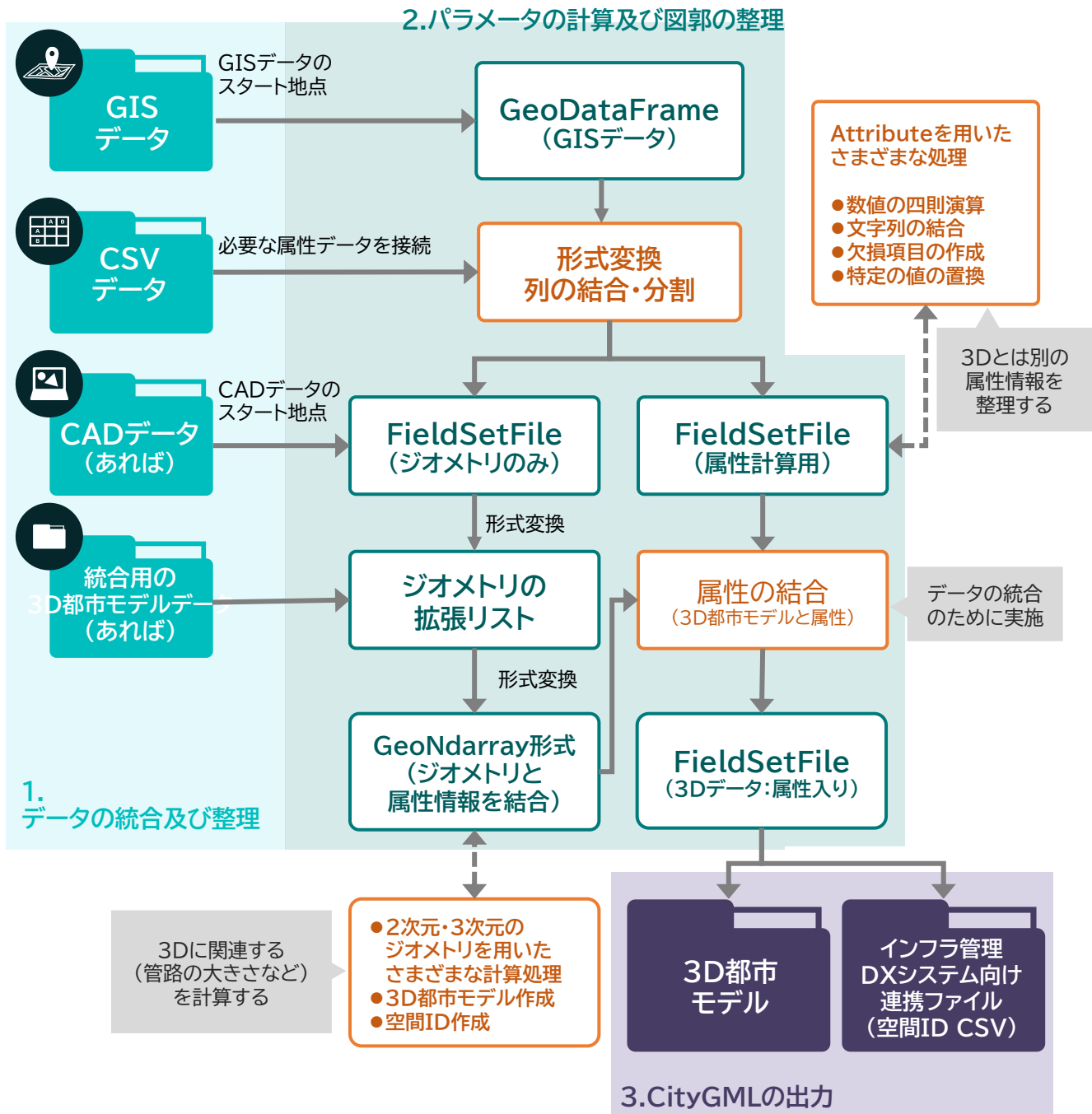
手順	参考図表	
<p>1. 初期状態としては以下のとおり パラメータとしては以下のように保持していると想定する</p> <p>ジオメトリ:3Dライン 属性 ①土被り 1m ②外径 0.2m</p> <p>最終目標: 管路の中心線を構築し、3D都市モデルを作る</p>	横断面図	
	縦断面図	
<p>2. 処理の名前: ConvertLineStringDensification やっていること: 【ラインの高密度化】</p> <p>ラインデータに対して頂点数を増やすことで、区間の埋め合わせを行う ラインに対してのポイントの密度を高めることで、正確に3D都市モデルを表現することができるようになる</p>	横断面図	
	縦断面図	
<p>3. 処理の名前: ConvertCoordinatesToCoordinates WithDrape やっていること: 【ドレープ処理・標高情報を与える】</p> <p>ラインデータと頂点データに対して、標高(Z軸)の情報を与える。これが、どの場所にあるか、どのくらいの高さにあるか把握するための前処理に該当する</p>	横断面図	<p>地表面(標高)</p>  <p>※実際の管路</p>
	縦断面図	<p>地表面(標高)</p>  <p>※実際の管路</p>

### 3.1.1 管路LOD2のデータを構築する基本概念(続き)

手順	参考図表
<p>4. 処理の名前: ConvertCoordinatesZByEarthCovering やっていること: 【実際の管路の高さに合わせる】</p> <p>管路の標高の高さ(Z値)から <b>属性①土被り(1m)</b>と <b>属性②の外径の半分(0.2÷2=0.1m)</b> の値を引き算する</p> <p>最終的な目標は、管路の中心線を作成するため、管路の半径分埋まっているところに中心があると想定し、外径の半分の数値を採用する</p>	<p>横断面</p>  <p>※実際の管路</p> <p>縦断面</p>  <p>※実際の管路</p>
<p>5. 処理の名前: GetEstimationZ やっていること: 【中間の頂点が直線的になるように調整】</p> <p>先の状態だと、地形に沿って上下をしてしまいます。この場合、現実と勾配が異なってしまうため、特定の位置(例:マンホール等)の間の高さを基準として勾配を算出し、管路の中心線を合わせる</p>	<p>横断面</p>  <p>※実際の管路</p> <p>縦断面</p>  <p>※実際の管路</p>
<p>6. 処理の名前: ConvertCoordinatesToMultipatch やっていること: 【LOD2の出力】</p> <p>実際に管路を出力する円形にしてしまうと、データ容量が大きくなってしまうため、四角柱の形でデータの出力を行う</p>	<p>横断面</p>  <p>※実際の管路</p> <p>縦断面</p>  <p>※実際の管路</p>

### 3.1.2 管路データを作るための基本概念図(作業の関係性)

管路データの作業フローは以下のとおりであり、作業手順(1. ~3.)が背景の色の範囲と合致しており、これらの手順を実行することで、適正な処理を行うことができる。



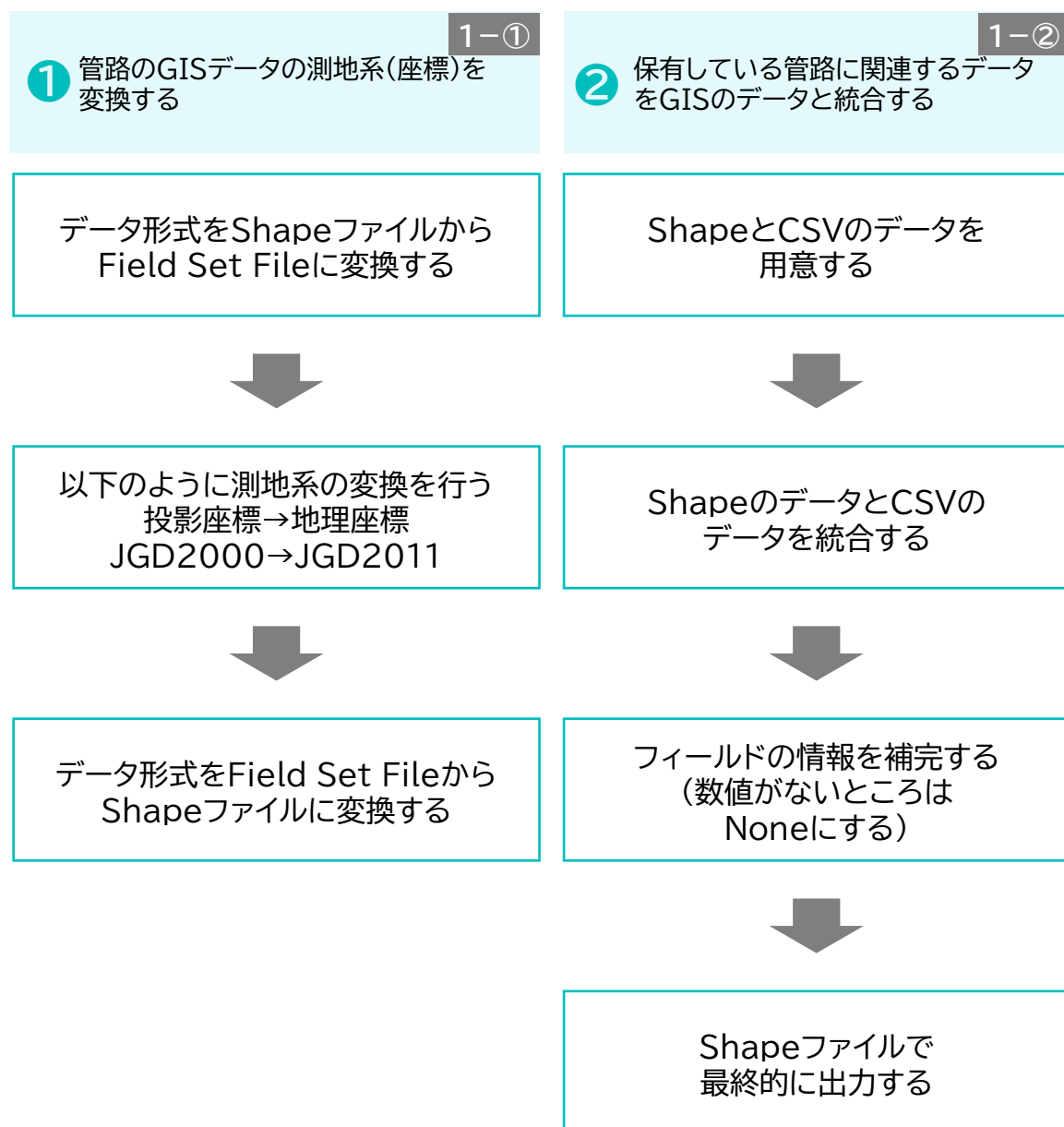
### 3.1.3 管路データの作成に必要な作業手順

実際に行う手順としては以下のとおりになり(詳細フローは後述)、データの整理を行った上で、図郭を整理し、LOD2のジオメトリを作成していく工程となる。



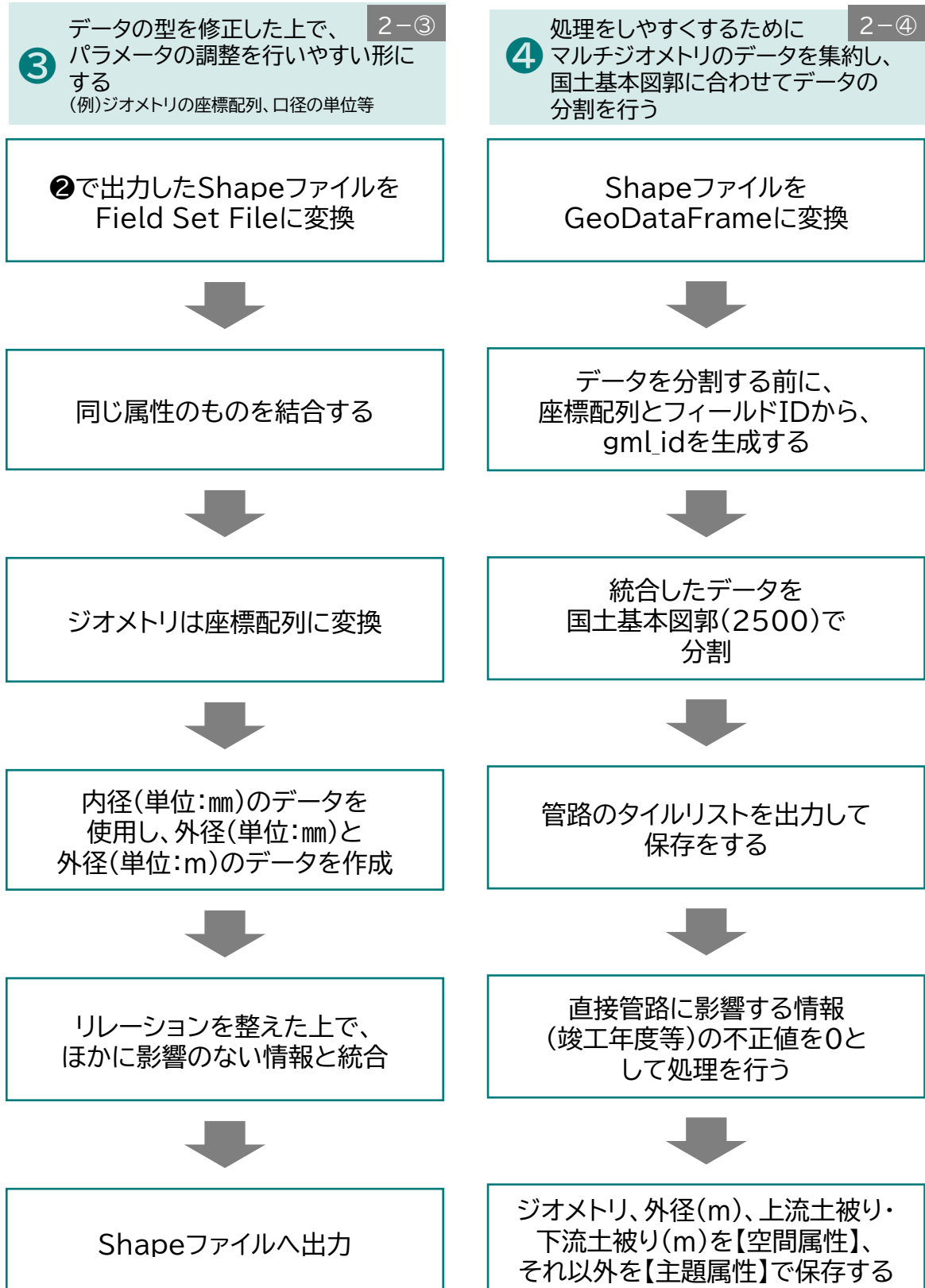
### 3.1.4 管路のデータ構築詳細実施フロー

本フローでは、保有しているGISのデータを結合して、ばらばらであった情報を一つのGISデータとして統合していく作業を行っている。



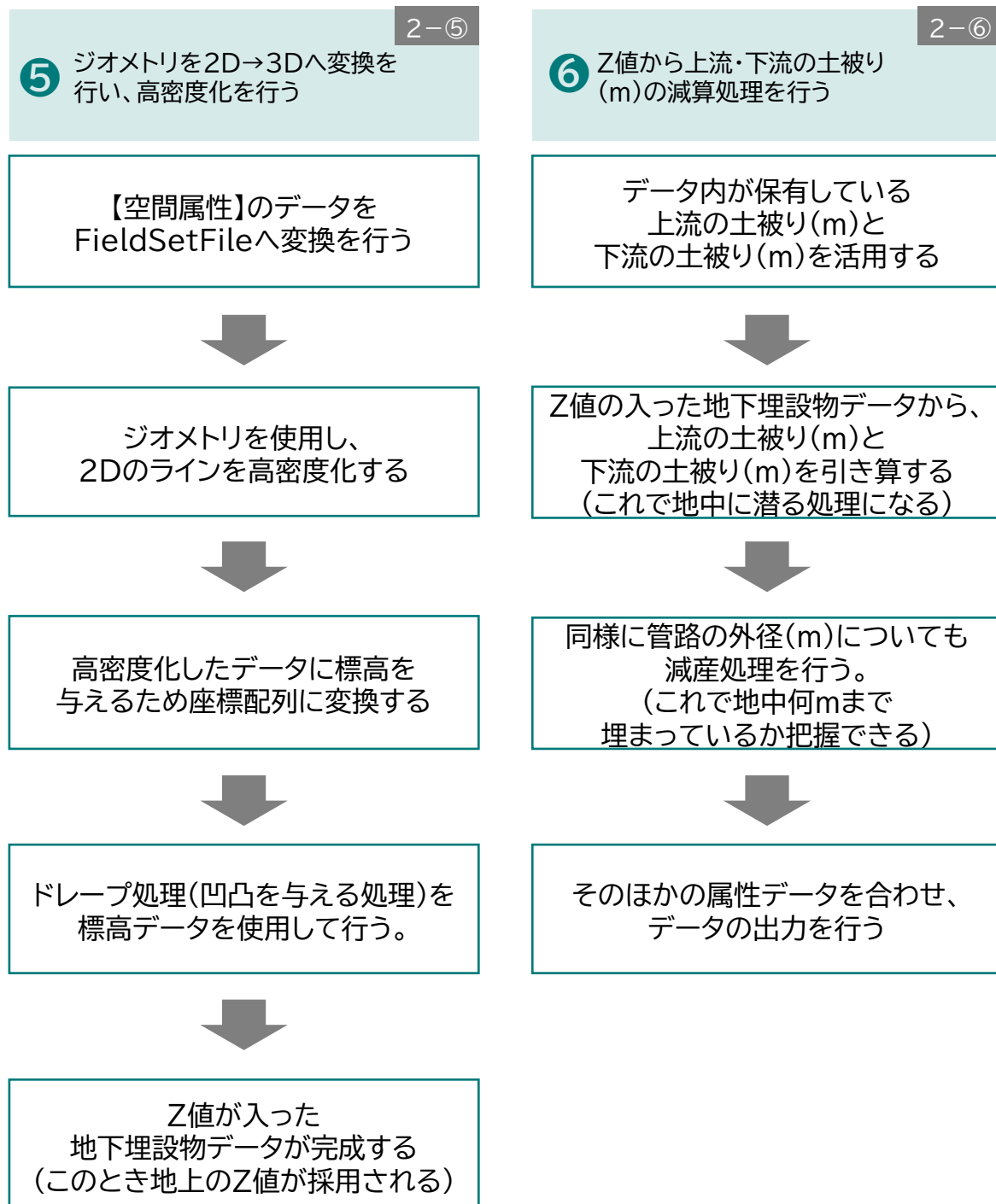
### 3.1.4 管路のデータ構築詳細実施フロー(続き)

ここでは、この後のLOD2のデータを作る際にエラーの値が出ないように、データの型をそろえて、欠損データの排除を行う。



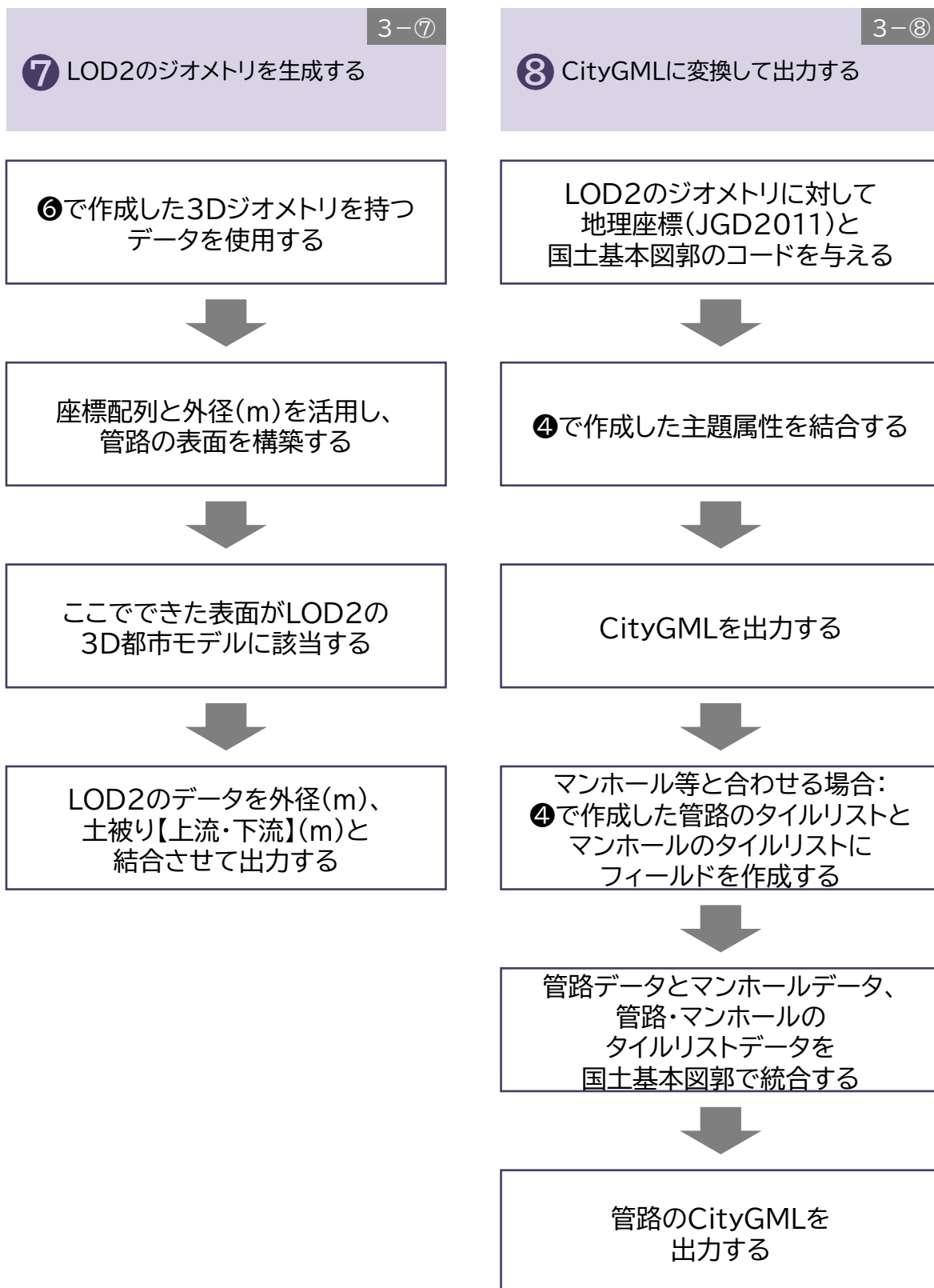
### 3.1.4 管路のデータ構築詳細実施フロー(続き)

ここでは、高さの情報を付与して、2Dのデータを3Dのモデルに変換していく。これによって、管路が立体的に正しい位置に配置されるようになる。



### 3.1.4 管路のデータ構築詳細実施フロー(続き)


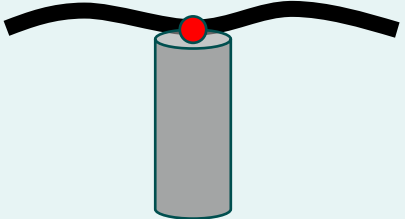
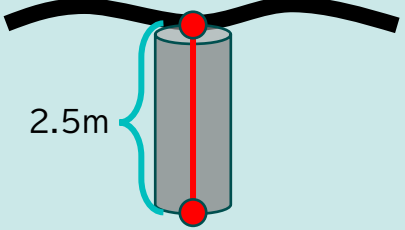
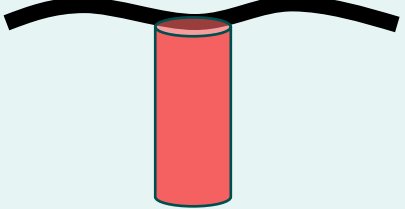
ここでは、LOD2の実際のジオメトリデータを作成し、CityGML形式にデータ変換して出力する



## 3.2 マンホール等の道路付属物の基本的な考え方

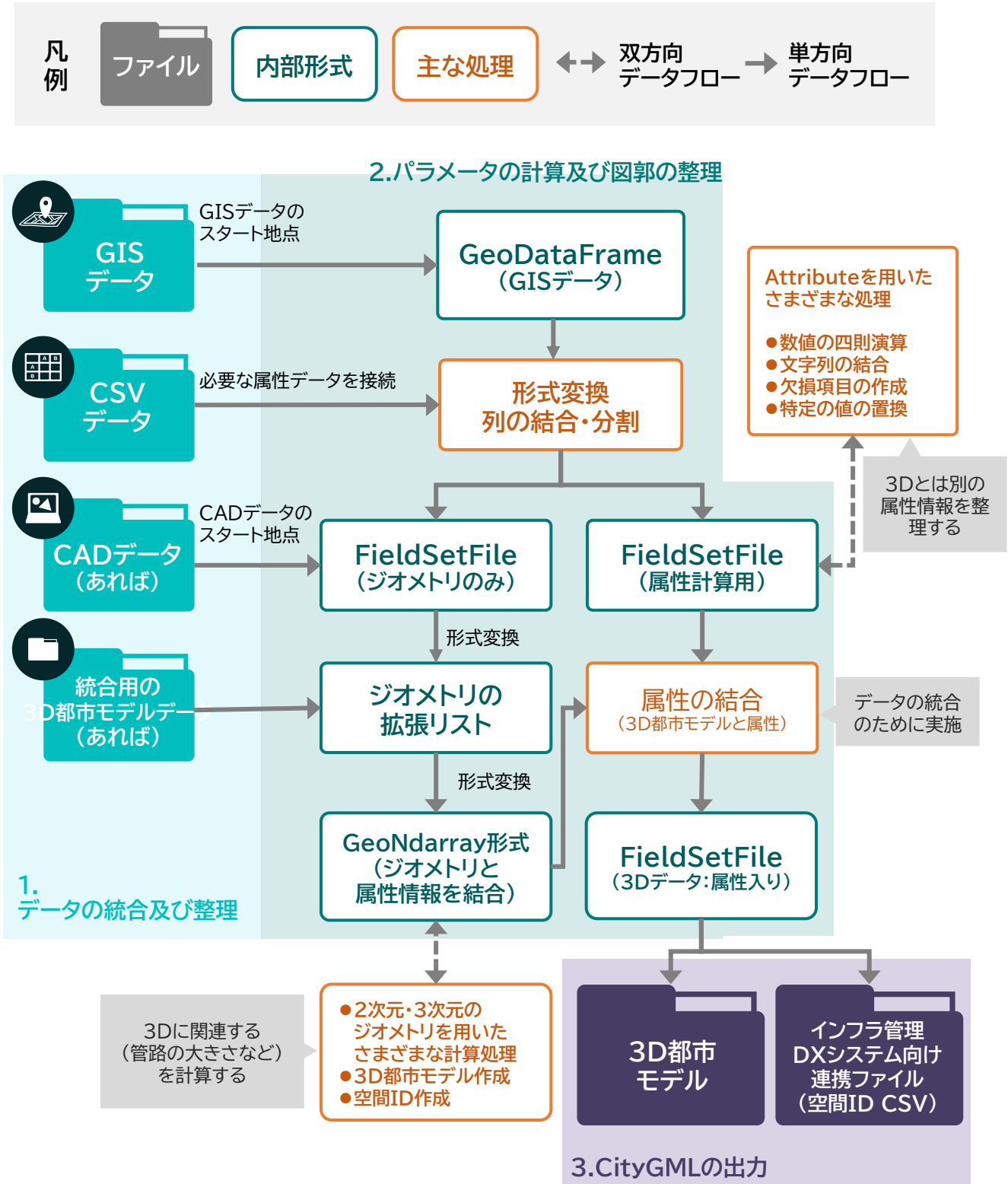
マンホール等は、以下に示すような流れでLOD2を構築していく。基本的な概略は、下表のように、ポイント情報から管路を作成し、高さを与えて3D化するという流れを解説している。

### 3.2.1 マンホール等のLOD2のデータを構築する基本概念

手順	参考図表
<p>1. 初期状態としては以下のとおり パラメータとしては以下のように保持していると想定する</p> <p>ジオメトリ:3Dポイント 属性 ①深さ 2.5m ②短辺外径 1m、長辺外径 1m</p>	<p>横断面図</p> 
<p>2. 処理の名前: ConvertCoordinatesToCoordinates WithDrape やっていること: 【ポイントに地表面の高さデータを与える】</p> <p>ポイントデータに地表面の標高を与える これによりマンホールがどの場所にあるのか 特定をすることができる</p>	<p>横断面図</p> <p>地表面(標高)</p> 
<p>3.処理の名前: ConvertPointToLineFromDepth やっていること: 【ポイントを縦方向に伸ばして、深さにする】</p> <p>ポイントのデータを縦方向に伸ばして、マン ホールの深さと同等になるように伸ばす</p>	<p>横断面図</p> <p>地表面(標高)</p> 
<p>4. 処理の名前: ConvertCoordinatesToMultipatch やっていること: 【線を太らせてLOD2のデータにする】</p> <p>マンホールの形状になるようにするために、 半径分を左右に太らせて、3Dでマンホールに 見えるように設定をする</p>	<p>横断面図</p> <p>地表面(標高)</p> 

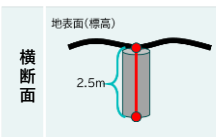

## 3.2.2 マンホール等のLOD2のデータを構築する基本概念

基本概念は管路とマンホールで基本的には同じ内容となる。管路では横に伸ばしていたものをマンホールにおいては縦方向に伸ばし、連動させることになる。



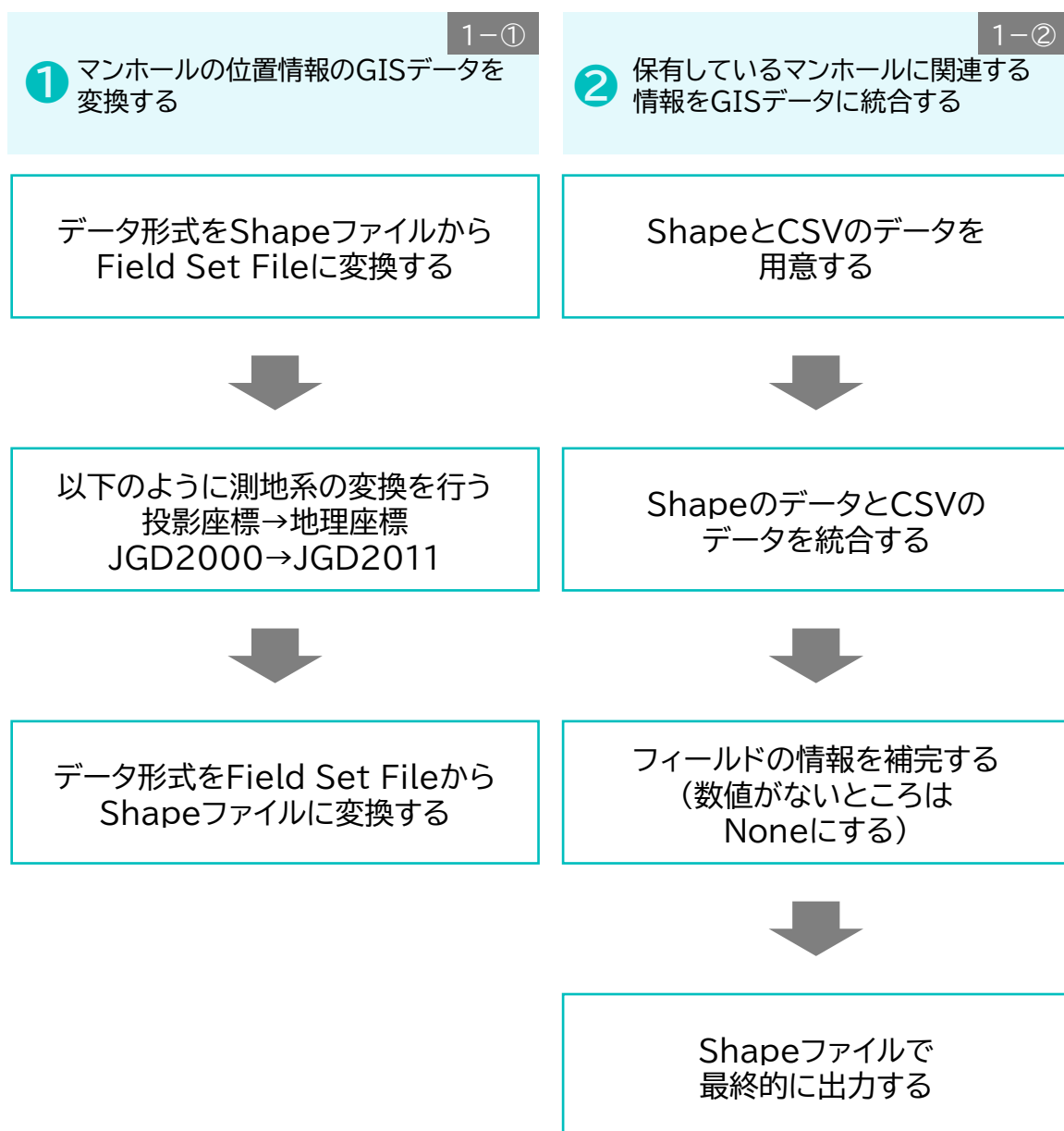
### 3.2.3 マンホール等における実施項目とデータの関係

実際に行う手順は以下に示すとおりである。詳細フローについては後述するが、データの整理を行った上で、図郭を整理し、LOD2のジオメトリを作っていく作業となる。

	中分類	実施内容概要(詳細フローは各ページ)	該当するイメージ	
1 データの統合及び整理	データの変換 (エラー確認及び地理座標修正)	① マンホールの位置情報のGISデータを変換する(→P41)	1-①	
	データの整理 (LODごとにおけるデータの整理)	② 保有しているマンホールに関連する情報をGISデータに統合する(→P41)	1-②	
2 パラメータの計算及び図郭の整理	TPの計算 (最近傍結合の深さ計算)	③ データの型を修正した上で、パラメータの調整を行いやすい形にする(→P42) (例)ジオメトリの座標配列、口径の単位等	2-③	
		④ 処理をしやすいするためにマルチジオメトリのデータを集約し、国土基本図郭の情報に合わせる(→P42)	2-④	
	各項目におけるパラメータ整理	⑤ ジオメトリの情報に対して標高情報を付与する 【4.2.1の2.の処理に該当】(→P43)	2-⑤	
		⑥ 深さ情報を付与し、縦のラインに変換する 【4.2.1の3.の処理に該当】(→P44)	2-⑥	
		⑦ LOD2のジオメトリを生成する 【4.2.1の4.の処理に該当】(→P45)	3-⑦	
		⑧ CityGMLに変換して出力する(→P46)	3-⑧	
3 CityGML出力	CityGMLへの出力			

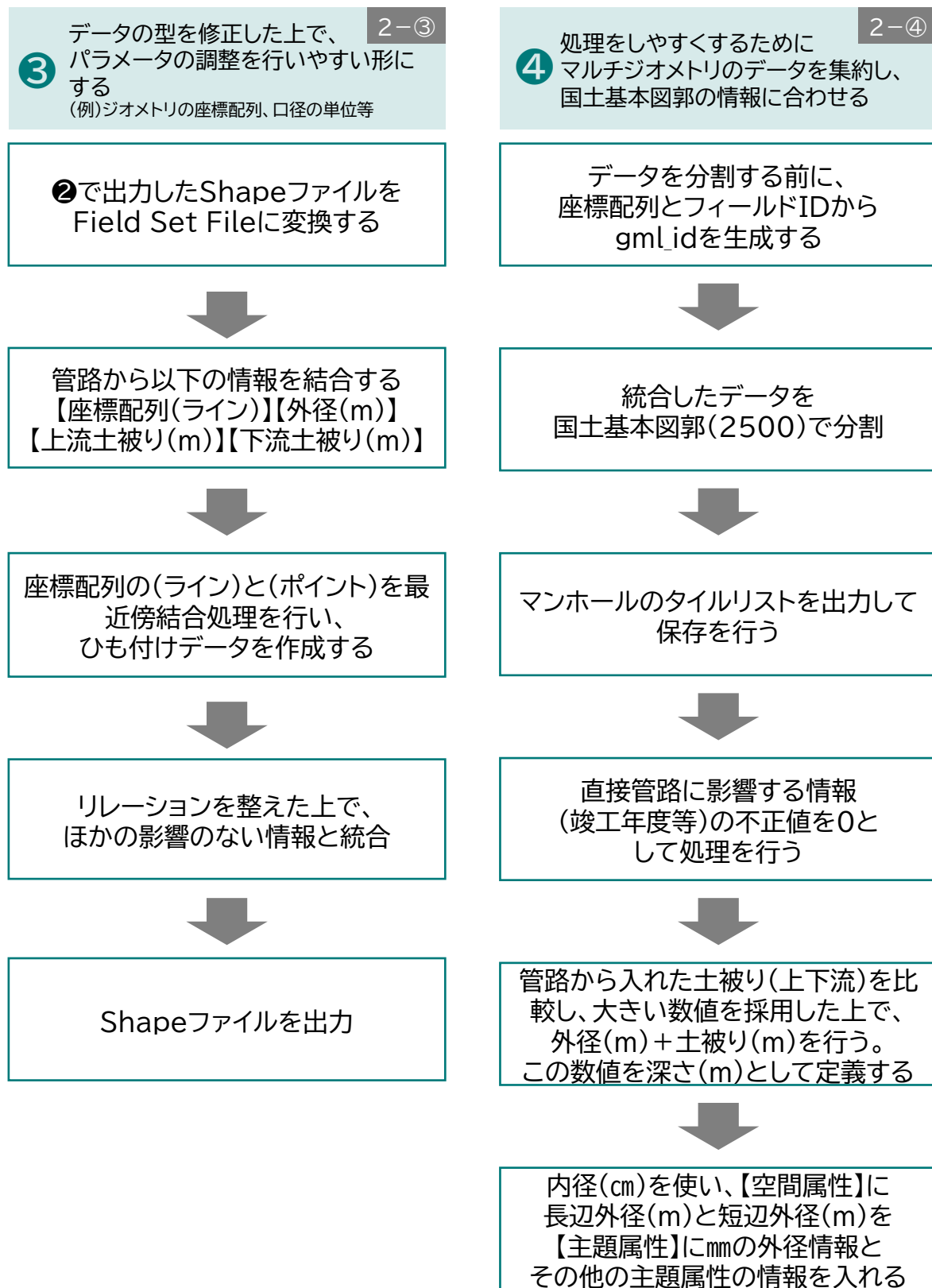
### 3.2.4 マンホール等のデータ構築詳細実施フロー

ここでは、保有しているGISのデータを結合して、ばらばらであった情報を一つのGISデータとして統合していく作業を行っている。



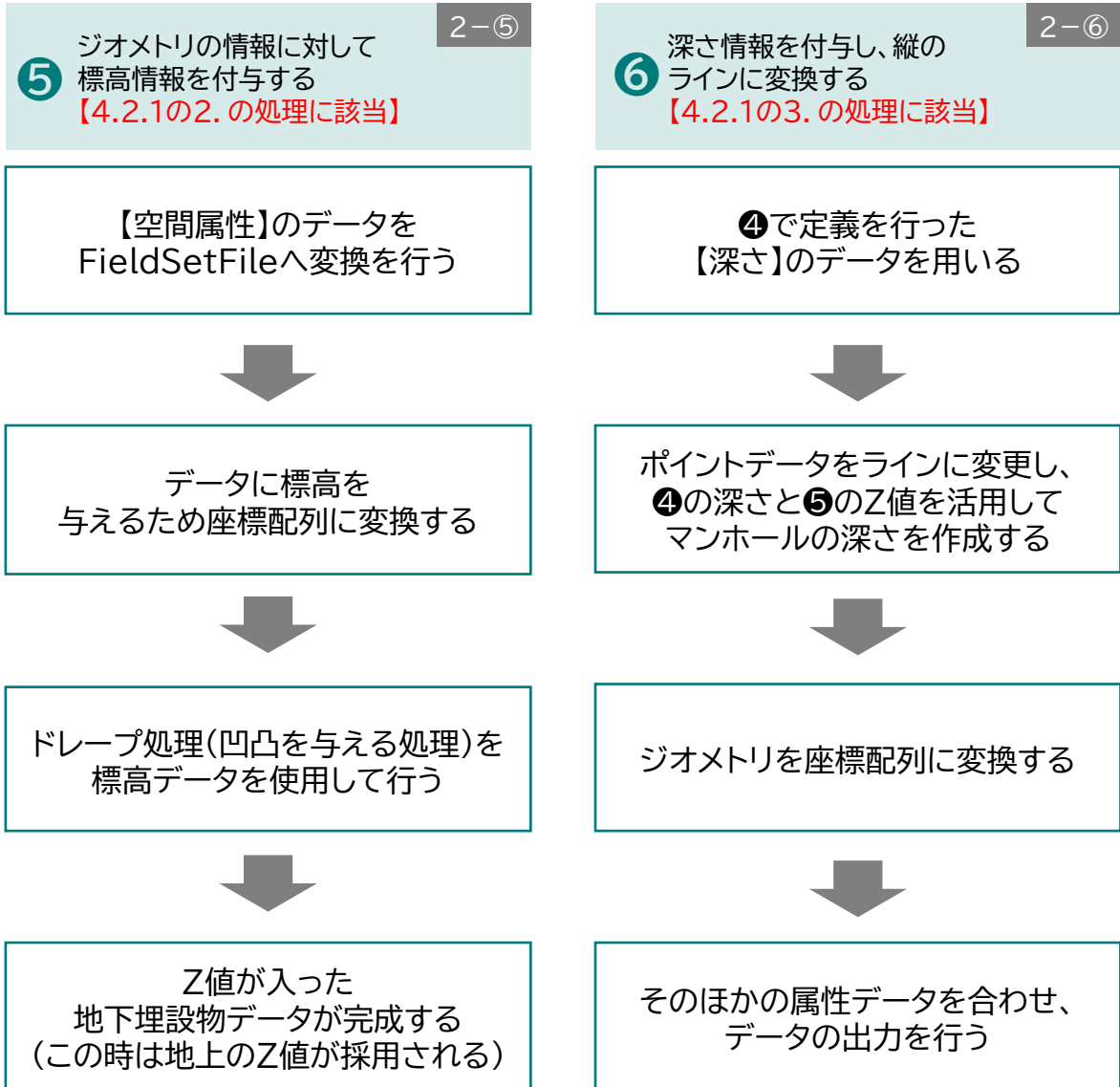
### 3.2.4 マンホール等のデータ構築詳細実施フロー(続き)

ここでは、この後のLOD2のデータを作る際にエラーの値が出ないように、データの型をそろえて、欠損データの排除を行う。



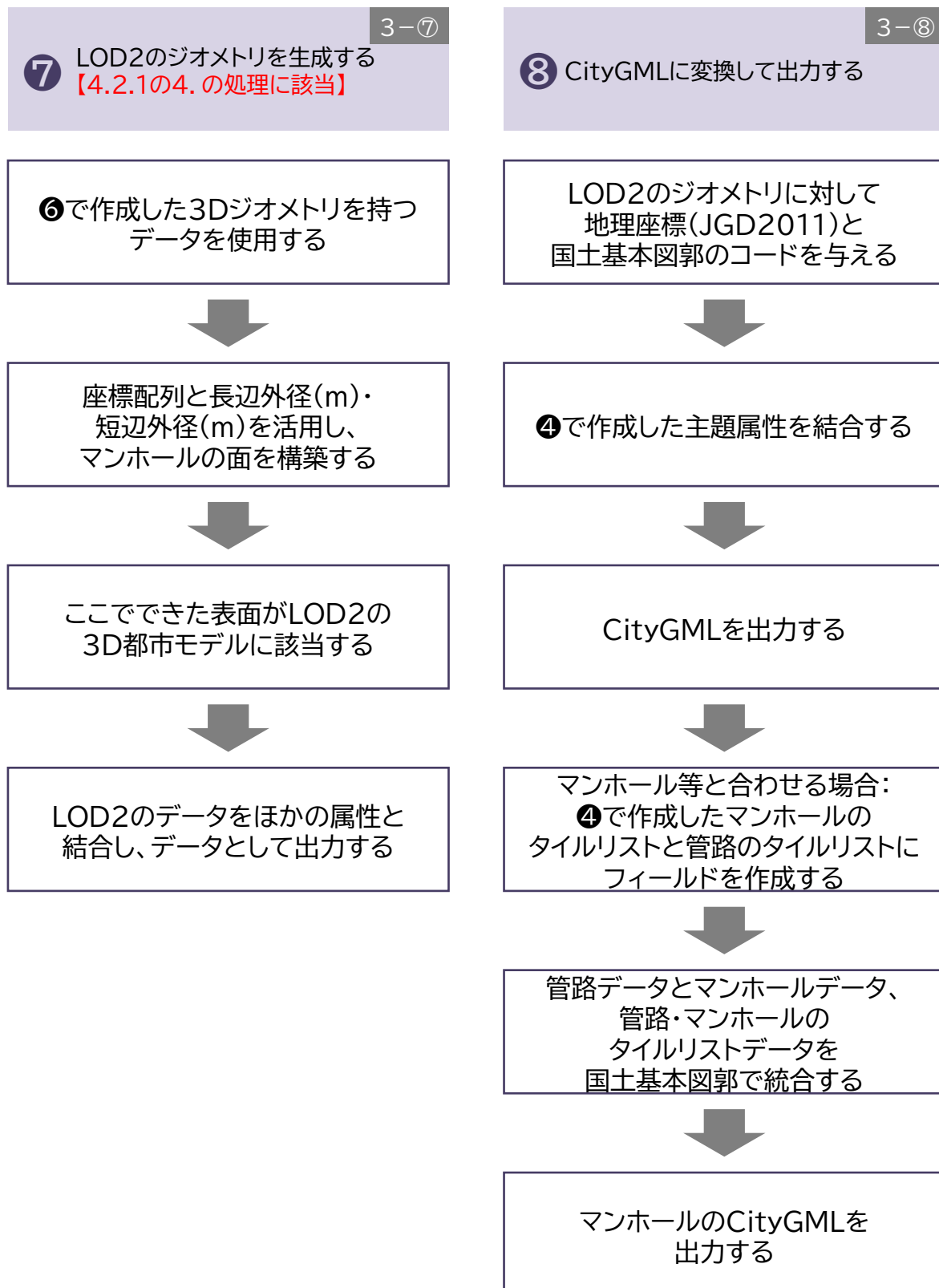
### 3.2.4 マンホール等のデータ構築詳細実施フロー(続き)

ここでは、高さの情報を付与して、2Dのデータを3Dのモデルに変換していく。これによって、管路が立体的に正しい位置に配置されるようになる。



### 3.2.4 マンホール等のデータ構築詳細実施フロー(続き)

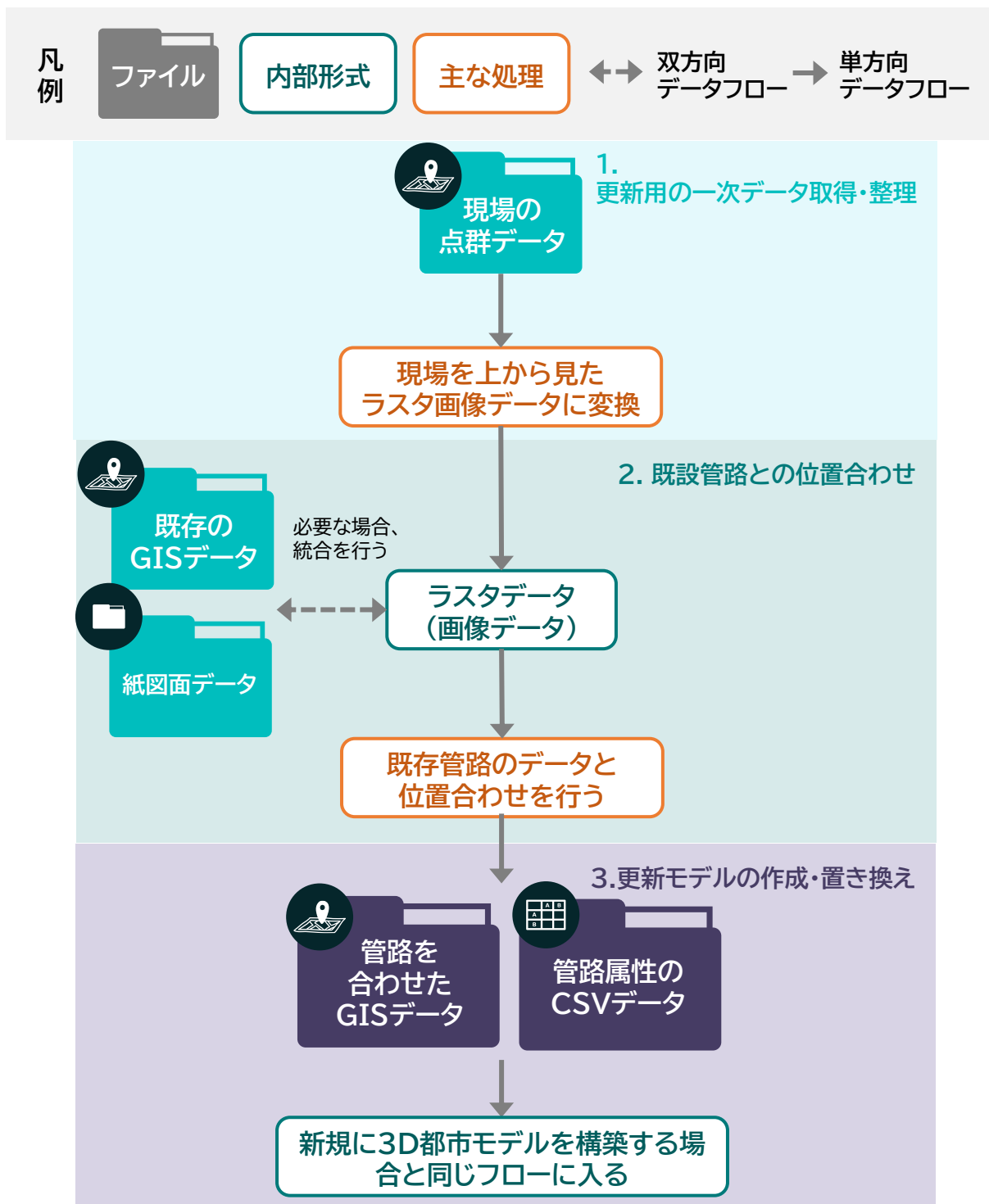
ここでは、LOD2の実際のジオメトリデータを作成し、CityGML形式にデータ変換して出力する



## 3.3 データ更新時の作業フロー

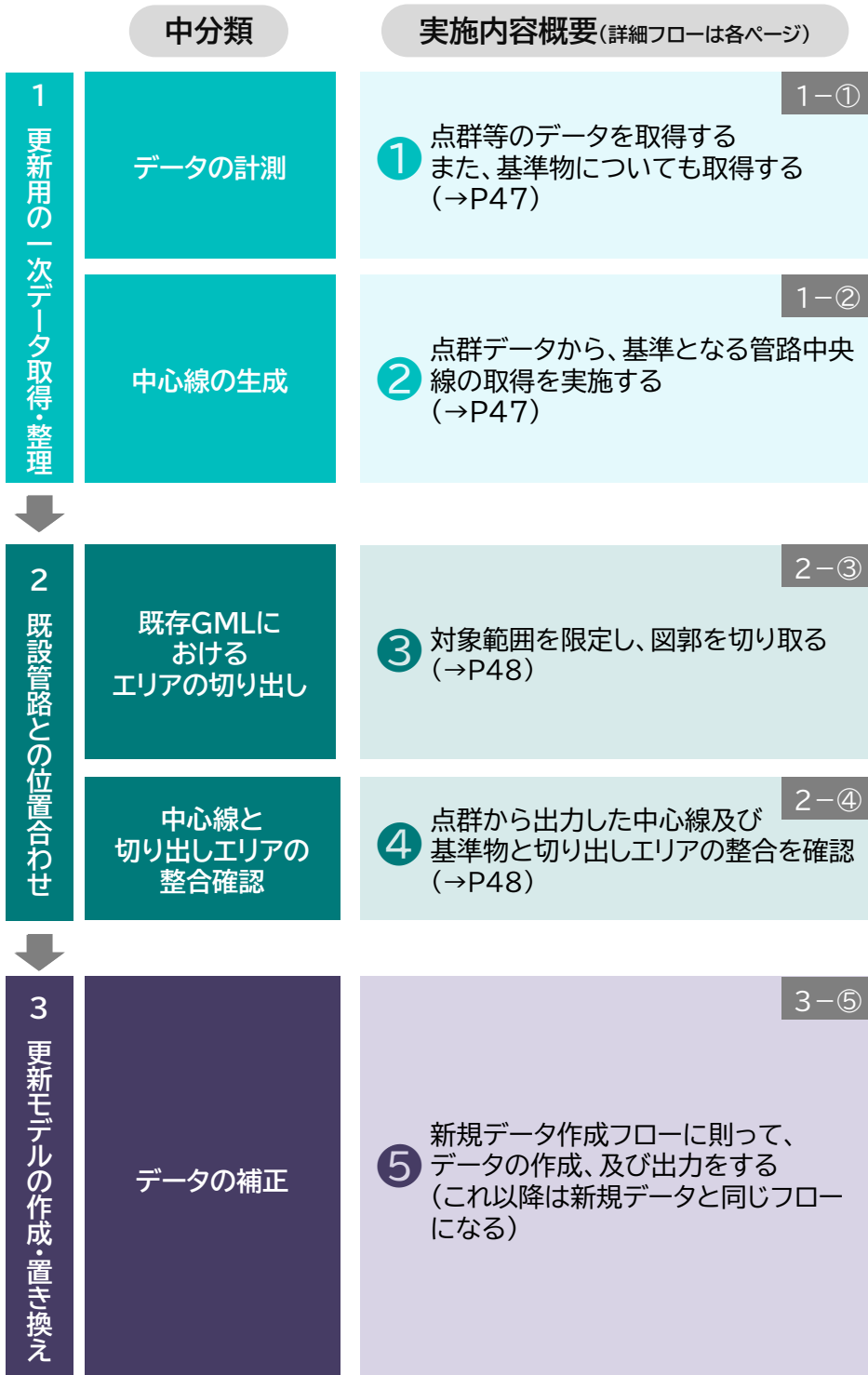
### 3.3.1 データ更新時の基本概念図

データ(モデル)更新を行う際には、留意すべき点がいくつかある。基本フローとしては新たにデータを作る場合と同じフローになることが想定されるが、点群データを用いてモデルを作成することにより、さらに精度の高いデータを構築することができる。ただし、元のモデルとの整合を図るためにも、過去データの測地系と現在のデータの測地系のどちらに依拠するのかなどについて留意する必要がある。



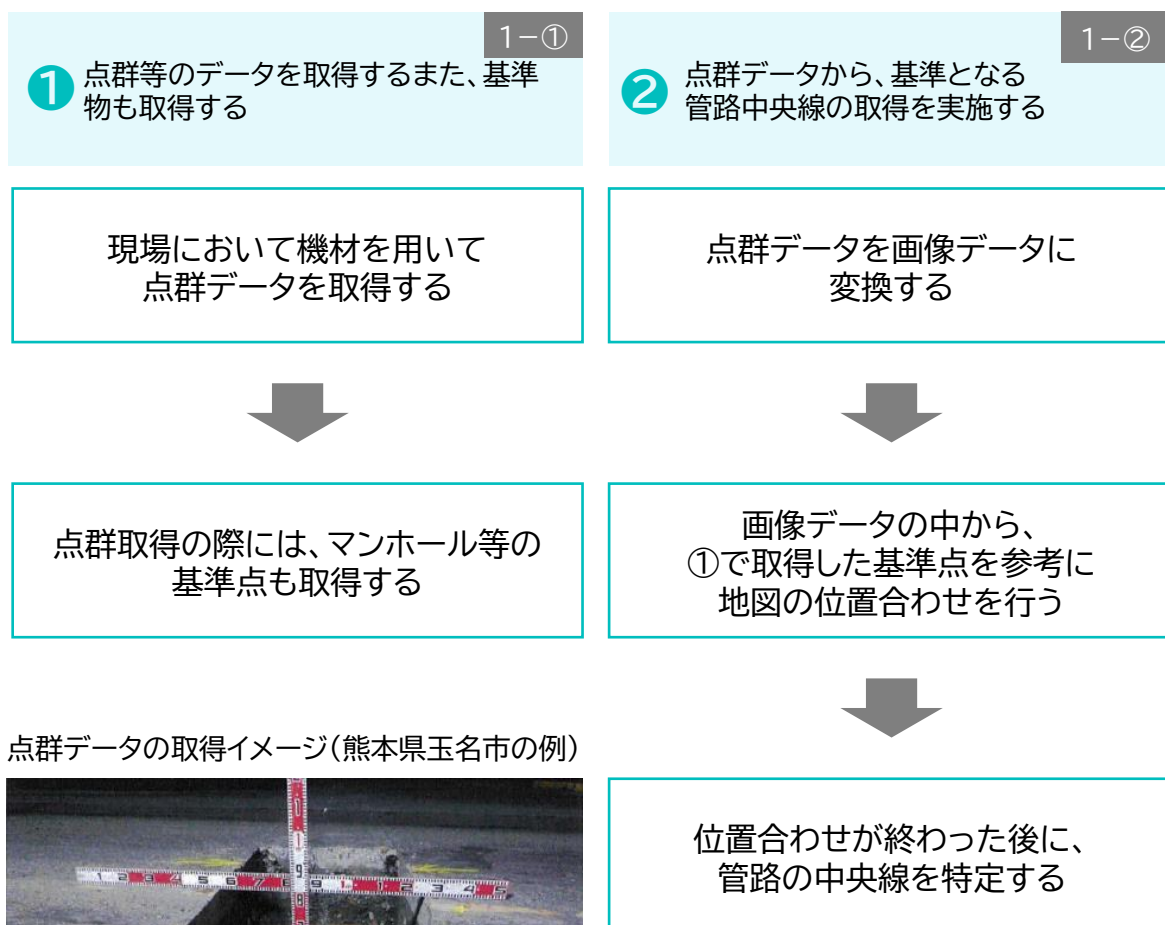
### 3.3.2 3Dデータ更新時の業務フロー一覧

各フェーズごとに細分化を行うと、以下のようになる。基本的にLODに応じたモデルを作成するための作業はほかのデータと同じであるが、基盤として用意するためのプロセスと、以下の項目が必要となる。



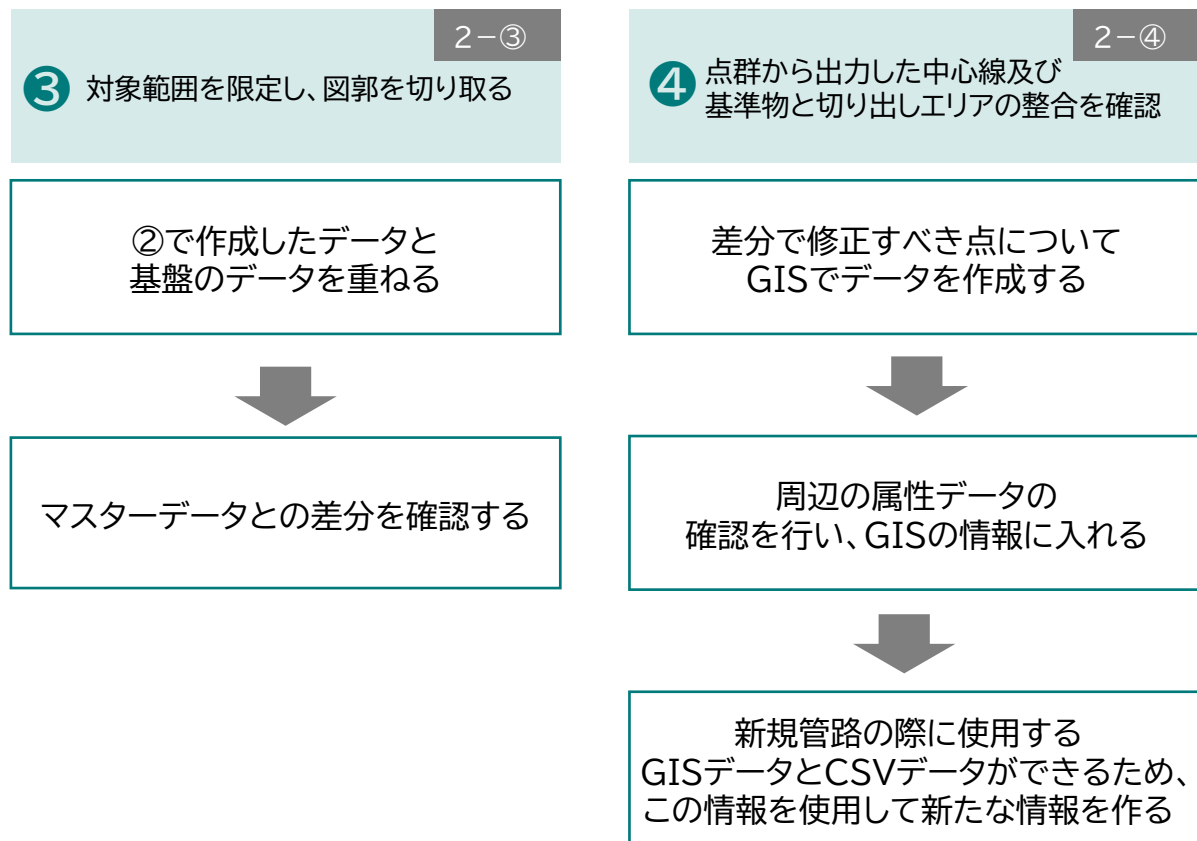
### 3.3.3 3Dデータ更新時の詳細フロー

ここでは点群データから画像データを生成し、管路の中央線を特定するところまでをおこなう。



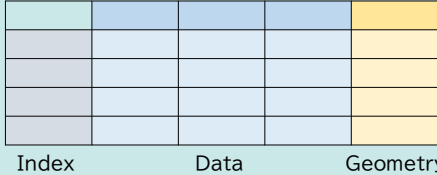
### 3.3.3 3Dデータ更新時の詳細フロー(続き)

ここでは対象範囲を限定した上で、3Dの管路データを作るために必要なGISのデータ及びCSVのデータを生成する。

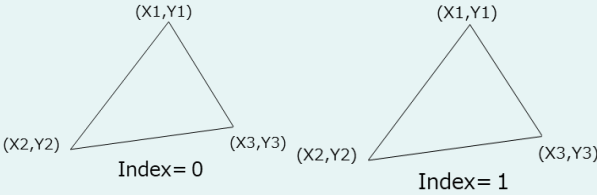
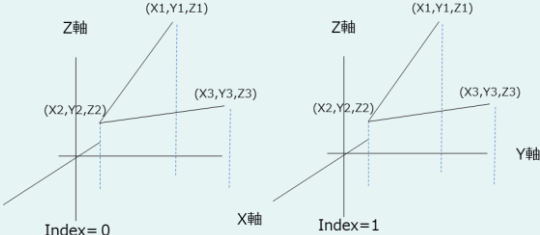


# 3.4 用語集

## ◆データの型の種類と処理内容に関する解説

番号	データの型の種類	中身の処理の内容	イメージ図																									
1	GeoDataFrame	<p>GeoDataFrameのデータ形式は、右図のとおり</p> <p><b>Index:</b> 地物のキーとなる値が格納</p> <p><b>Data:</b> Attributeの値が格納</p> <p><b>Geometry:</b> ジオメトリの形状、座標の値が格納</p> <p>地理空間データを扱うためのデータ構造となっており、その中でもデータに Point(点)、 LineString(線)、 Polygon(面)等の空間的な情報が格納される</p>	 <p>Index                  Data                  Geometry</p> <table border="1" data-bbox="768 576 939 768"> <thead> <tr> <th>Geometry</th> </tr> </thead> <tbody> <tr> <td>Point(~)</td> </tr> <tr> <td>LineString(~)</td> </tr> <tr> <td>Polygon(~)</td> </tr> <tr> <td>Point(~)</td> </tr> </tbody> </table> <p>Collection(Point(),LineString())と記述することで、列に対して2種類以上の要素を記載することは可能。ただし、同じ行に2種類記載することはできない</p> <p>例:GeoDataFrame <span style="border: 1px solid red; padding: 2px;">ジオメトリリストのこと。</span></p> <table border="1" data-bbox="772 816 1200 919"> <thead> <tr> <th>City</th> <th>Latitude</th> <th>Longitude</th> <th>geometry</th> </tr> </thead> <tbody> <tr> <td>0 tokyo</td> <td>35.68</td> <td>-139.69</td> <td>Point(-139.69 35.68)</td> </tr> <tr> <td>1 osaka</td> <td>34.69</td> <td>-135.5</td> <td>Point(-135.5 34.69)</td> </tr> <tr> <td>2 hokaido</td> <td>43.06</td> <td>-141.35</td> <td>Point(-141.35 43.06)</td> </tr> <tr> <td>3 okinawa</td> <td>26.21</td> <td>-127.67</td> <td>Point(-127.67 26.21)</td> </tr> </tbody> </table>	Geometry	Point(~)	LineString(~)	Polygon(~)	Point(~)	City	Latitude	Longitude	geometry	0 tokyo	35.68	-139.69	Point(-139.69 35.68)	1 osaka	34.69	-135.5	Point(-135.5 34.69)	2 hokaido	43.06	-141.35	Point(-141.35 43.06)	3 okinawa	26.21	-127.67	Point(-127.67 26.21)
Geometry																												
Point(~)																												
LineString(~)																												
Polygon(~)																												
Point(~)																												
City	Latitude	Longitude	geometry																									
0 tokyo	35.68	-139.69	Point(-139.69 35.68)																									
1 osaka	34.69	-135.5	Point(-135.5 34.69)																									
2 hokaido	43.06	-141.35	Point(-141.35 43.06)																									
3 okinawa	26.21	-127.67	Point(-127.67 26.21)																									
2	DataFrame	<p>DataFrameは基本的に上記のGeoDataFrameと同様だが、一般的なデータを扱うためのデータ構造となっている</p>	<p>例:DataFrame</p> <table border="1" data-bbox="843 955 1210 1106"> <thead> <tr> <th></th> <th>Name</th> <th>age</th> <th>born</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>佐藤</td> <td>10</td> <td>東京</td> </tr> <tr> <td>1</td> <td>鈴木</td> <td>38</td> <td>富山</td> </tr> <tr> <td>2</td> <td>高橋</td> <td>42</td> <td>静岡</td> </tr> <tr> <td>3</td> <td>田中</td> <td>25</td> <td>鳥取</td> </tr> </tbody> </table>		Name	age	born	0	佐藤	10	東京	1	鈴木	38	富山	2	高橋	42	静岡	3	田中	25	鳥取					
	Name	age	born																									
0	佐藤	10	東京																									
1	鈴木	38	富山																									
2	高橋	42	静岡																									
3	田中	25	鳥取																									
3	FieldSetFile	<p>GeoDataFrameのように地理空間データを扱いますが、データ形式が異なり、右図のとおり3列で構成されるCSVファイルになる</p> <p>データイメージとしては、GeoDataFrameを<b>横形式</b>から<b>縦形式</b>に変換したイメージである</p> <p><b>Dwh:</b> データ名/Attribute名の値が格納されている</p> <p><b>Type:</b> Dwhに対するデータの形式・型の値が格納されている</p> <p><b>Value:</b> Dwhに対する文字列の値(base64エンコードした)が格納されている</p>	<table border="1" data-bbox="782 1156 1290 1483"> <thead> <tr> <th>Dwh</th> <th>Type</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>index</td> <td>int</td> <td>~</td> </tr> <tr> <td>data1</td> <td>str</td> <td>~</td> </tr> <tr> <td>data2</td> <td>...</td> <td>~</td> </tr> <tr> <td>data3</td> <td>...</td> <td>~</td> </tr> <tr> <td>data4</td> <td>...</td> <td>~</td> </tr> <tr> <td>geometry</td> <td>...</td> <td>~</td> </tr> </tbody> </table>	Dwh	Type	Value	index	int	~	data1	str	~	data2	...	~	data3	...	~	data4	...	~	geometry	...	~				
Dwh	Type	Value																										
index	int	~																										
data1	str	~																										
data2	...	~																										
data3	...	~																										
data4	...	~																										
geometry	...	~																										
4	ジオメトリリスト	<p>一つ以上のジオメトリを格納したリスト形式のデータのことである</p> <p>geometryで構成されており、要素としてはPoint(点)、LineString(線)、Polygon(ポリゴン)などの空間情報を持つ</p>	<pre data-bbox="839 1570 1210 1790">[ [Point (~) ]   [Point (~) ]   [Point (~) ]   [Point (~) ] ]</pre>																									

## 3.4 用語集(続き)

番号	データの型の種類	中身の処理の内容	イメージ図
5	GeoNdarray	ジオメトリ情報を格納するための統一されたデータ構造となっており、全てのジオメトリ情報は [index, X軸, Y軸, Z軸] 又は [index, X軸, Y軸] の形式で統一されている	<p>GeoNdarrayデータイメージ</p> <pre>「 [0, X1, Y1]   [0, X2, Y2]   [0, X3, Y3]   [0, X1, Y1]   [1, X1, Y1]   [1, X2, Y2]   [1, X3, Y3]   [1, X1, Y1] 」</pre> <p>↓</p> <p>表現されるデータ</p>  <p>GeoNdarrayデータイメージ</p> <pre>「 [0, X1, Y1, Z1]   [0, X2, Y2, Z2]   [0, X3, Y3, Z3]   [1, X1, Y1, Z1]   [1, X2, Y2, Z2]   [1, X3, Y3, Z3] 」</pre> <p>↓</p> <p>表現されるデータ</p> 

# CONTENTS

0. はじめに
1. 地下埋設物等の現状課題と関連取組
  - 1.1 地下埋設物等を取り巻く現状と課題
  - 1.2 国の最新取組
2. 使用するデータの準備
  - 2.1 既設管路の整備に必要なデータ
  - 2.2 新設・移設管路の整備に必要なデータ
3. 地下埋設物等の3D都市モデルデータ作成
  - 3.1 管路データを作る際の基本的な考え方
  - 3.2 マンホール等の道路付属物の基本的な考え方
  - 3.3 データ更新時の作業フロー
4. システム運用の考え方・体制
  - 4.1 既往システム等の運用方法・体制
  - 4.2 市町村における先行事例
5. 今後の取組展開に向けて

付録 データ作成ツールマニュアル

## 4.1 既往システム等の運用方法・体制

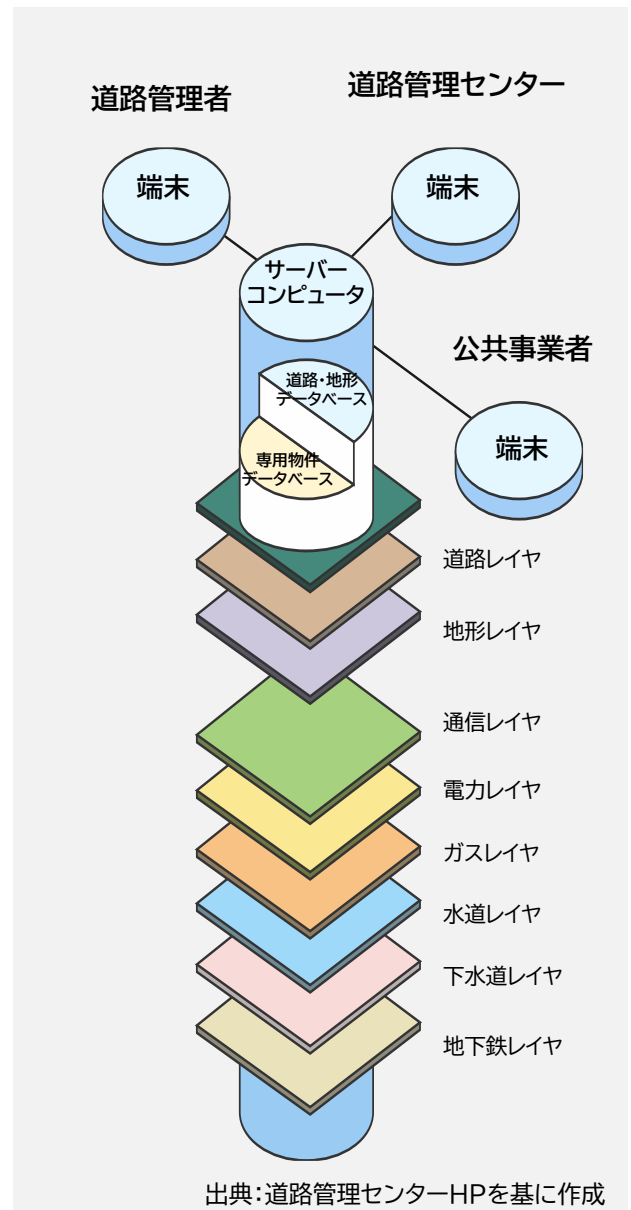
地下埋設物に関するデータが完成したとしても、どのようにデータを運用すべきか見当がついていなければ有効に生かすことはできない。そこで、そのほかの既存インフラにおけるデータの運用・更新を参考に見ていく。本章では道路管理システム(ROADIS)のシステム構造を取り上げ、地下埋設物データの更新・運用に応用できるポイントについて整理する。

### 4.1.1 道路管理システム(ROADIS)

都市の道路においては、地下埋設で対応しているような水道、下水道であったり通信、電力、ガス等の多数の公共物件が輻輳(ふくそう)して収容されている。これらを体系的に管理するためにGIS(地理情報システム)を活用してROADISは開発されている。これによって道路管理に関する仕組みが構築されている。

#### (1) 仕組みと機能

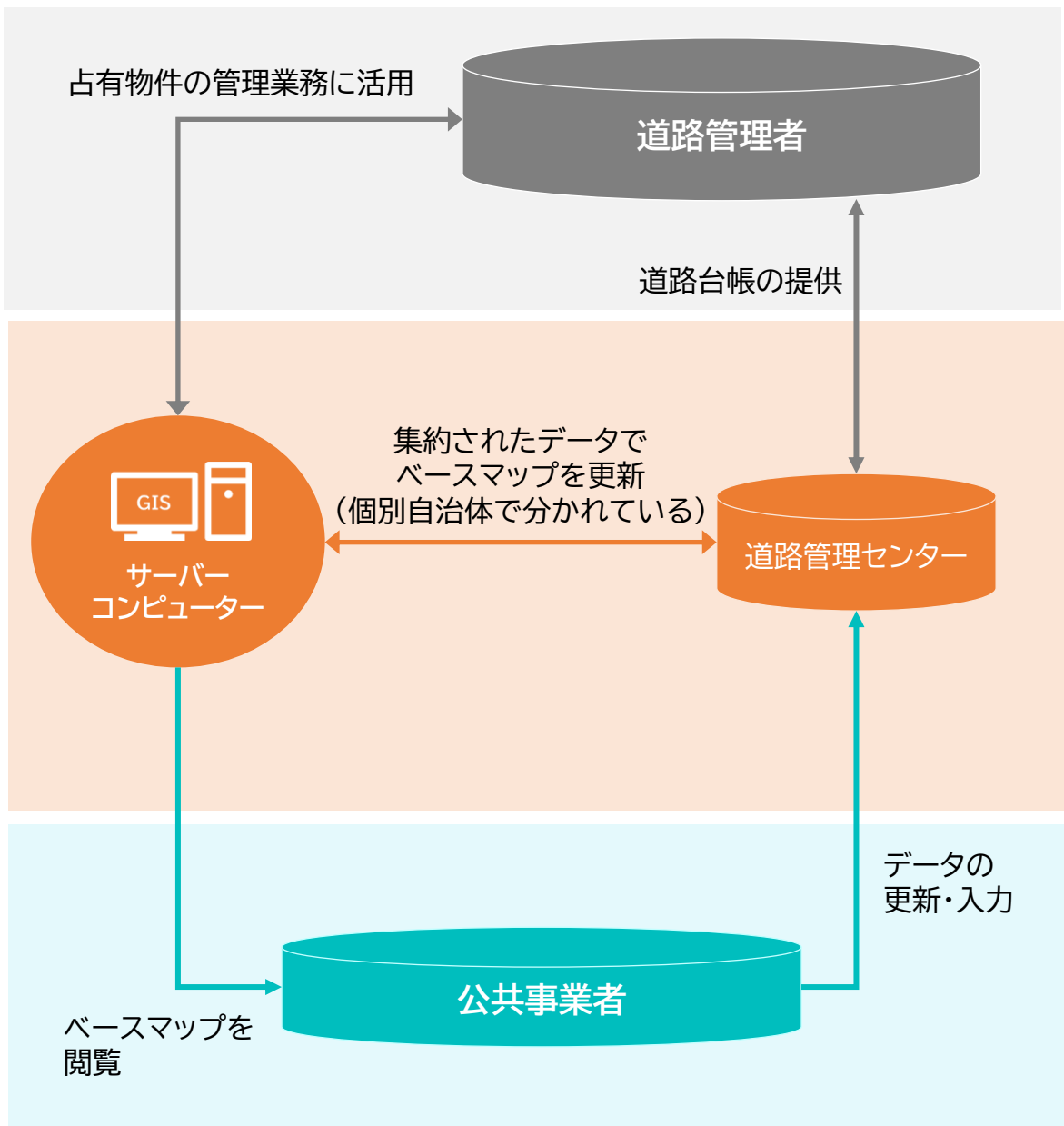
- 道路と占有物件に関する各種情報をGISを利用して総合的に管理をする。
- このシステム自体は、通信回線などを通じて、道路管理者や公益事業者に情報を正確かつ迅速に提供をするシステムである。
- システムには地形・道路の情報と埋設物に関する図形情報、構造や管路の材質などの属性情報が入っており、情報の検索や更新が平易にできる。



道路管理システム全体組織の関連図

## (2) データの更新体制

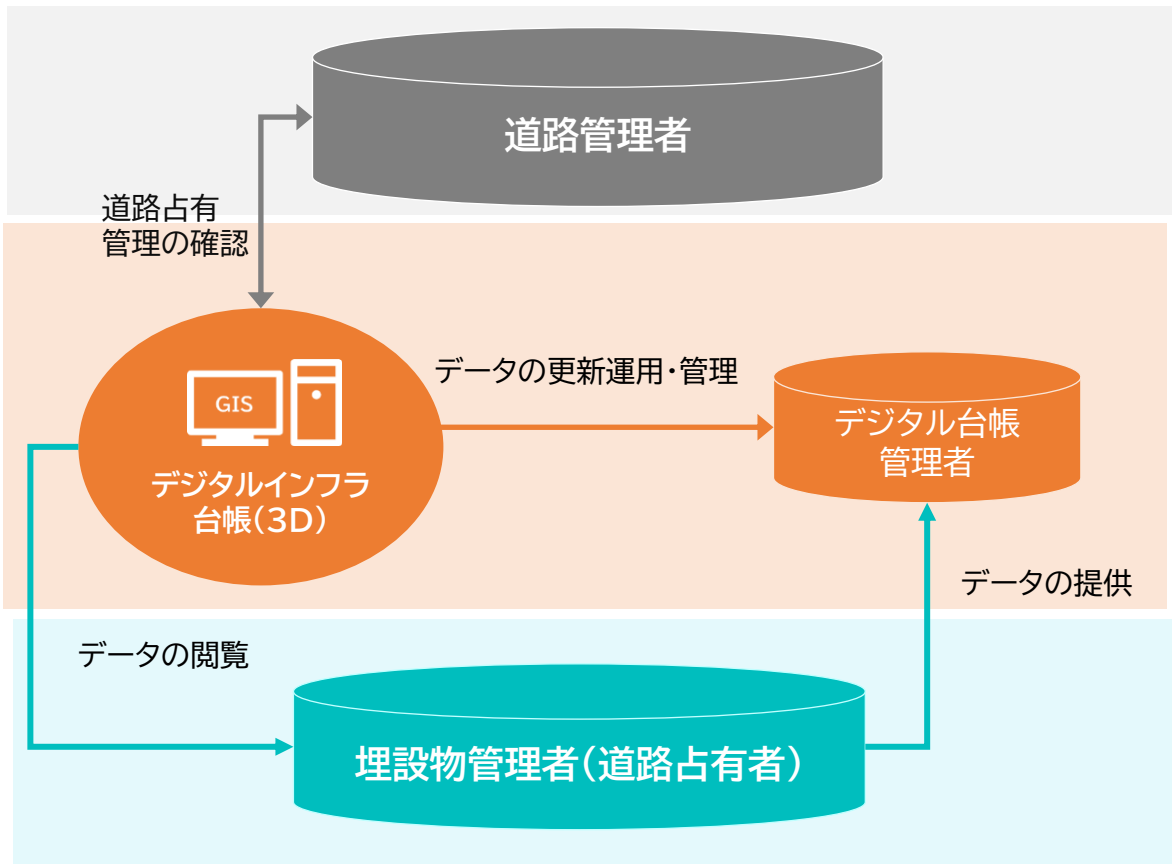
- 市区町村ごとに独立型のシステムとなっており、道路管理者や公益事業者はパソコンを道路管理センターのサーバーに接続することで、道路や占有物件の現状データを手に入れることができるようになっている。
- データの更新は、道路管理者や公益事業者から提供されたものを道路管理センターが更新をしている。



出典:道路管理センターHPを基に作成

## 4.1.2 地下埋設物のデータプラットフォーム構築イメージ

道路データと同様に地下埋設物に関するデータを管理していくためには、デジタルインフラ台帳を基準として、その周辺における管理を行っていく必要がある。理想的な図としては以下のような形になる。



- 道路管理者  
道路管理者においては、道路の占有業務の効率化をデジタルインフラ台帳を用いて行っていくことになる。ROADISの道路管理者と同様の動きになると想定され、主に市役所などの道路管理課などが担うことが多くなる。
- デジタル台帳管理者  
デジタルインフラ台帳を管理する組織になる。この組織に関しては第三者機関が担う場合もあるが、道路管理者が代行して実施する場合もある。主に、道路インフラデータの運用・更新を担い、ROADISの中で道路管理センターの役割を果たす。
- 埋設物管理者(道路占有者)  
埋設物管理者においては、施工計画や申請手続の効率化を図ることができるようになる。また、埋設物情報を提供することによって、デジタル台帳管理者が最新のデータに更新を行うことができるよう支援をする。ROADISの中では、公共事業者が担っていた部分に該当する。

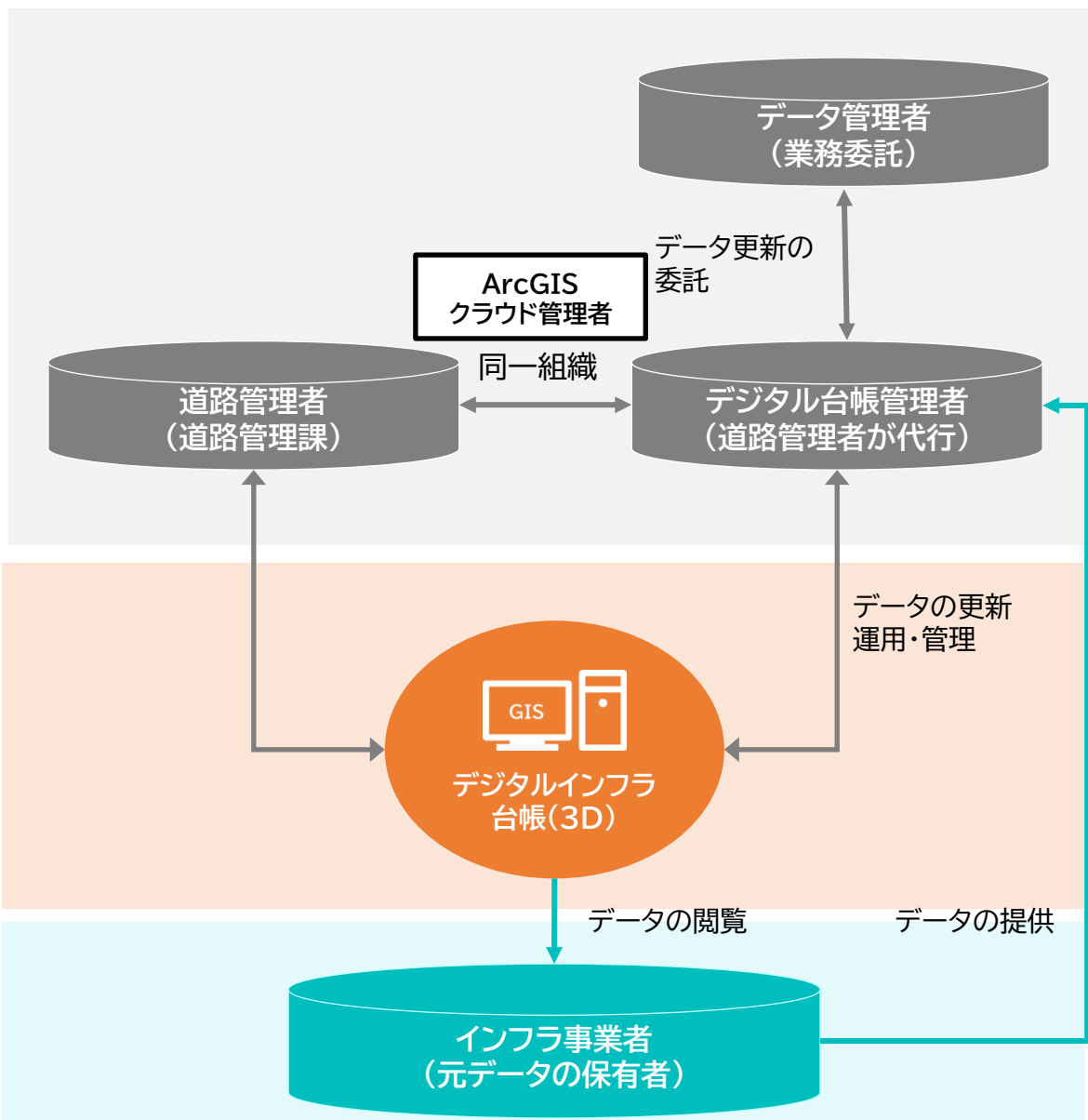
## 4.2 自治体における先行事例

自治体の先行事例として、三重県四日市市及び熊本県玉名市の取組を紹介する。また、その際の課題についても併せて整理する。

### 4.2.1 三重県四日市市の取組事例

#### (1) 四日市市のデジタルインフラ台帳の運用体制

- 四日市市においては、市の関与する割合を高めてデジタルデータの運用を行っていく方針としている。アクセス環境としてはArcGISクラウドを採用しており、利用者がArcGISを利用できる環境であれば、情報を手に入れることができるようになっている。



## (2)利用想定

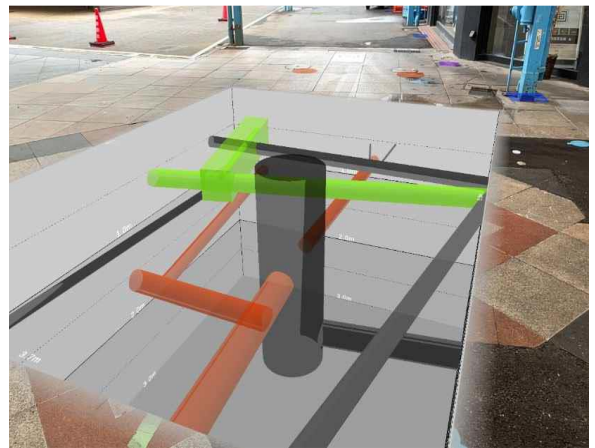
案	内容	利用者の負担	課題
1. 市役所の端末を利用	市役所の専用端末を利用して、閲覧する	無	最新情報を確認するために、データが更新される度に市役所への訪問が必要
2. 自社の端末を利用	利用者がArcGISを契約して、自社の端末からオンライン環境で閲覧する	30,000円/年程度	年ごとの費用負担が必要

## (3)課題

- 現時点の運用体制では、前述のとおり、道路管理者がデジタル台帳管理者の役割を果たすことになっている。ただし、道路管理者がデータを管理すること自体が難易度が高い。そのため、データ管理については、業務委託者を入れている。
- 市における費用負担が大きくなることが想定される。



デジタルインフラ台帳のイメージ



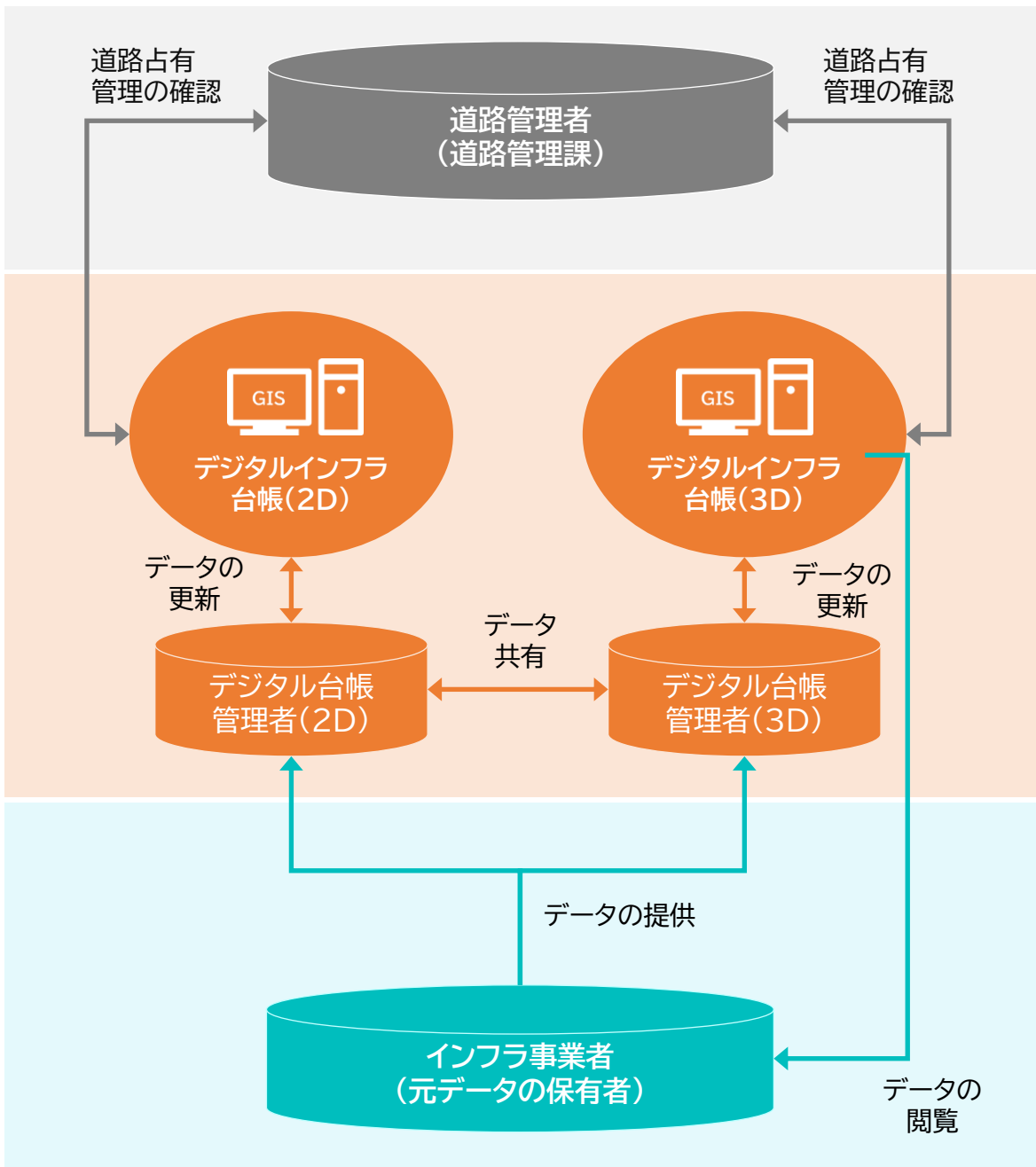
仮想掘削穴による地下埋設物3D都市モデルのAR表示例

※上図は参考用のイメージであり、実際の地下埋設物の状況を示すものではありません。

## 4.2.2 熊本県玉名市の事例(構想検討イメージ)

### (1) 玉名市のデジタルインフラ台帳の運用体制(イメージ)

- 熊本県玉名市においては、既存の2Dで管理を行っているGIS基盤が存在している。そこに新たに3Dによる台帳管理を行うことを想定し、既存の2DのGISについても同時進行で更新していくことによって、更に詳細なデータのやり取りを行うことができるように設計を行っていく方針である。



## (2)今後の利用想定イメージ

- 玉名市では、以下のように段階的に利活用を広げていくことを想定している。  
F8VPSを活用しながら展開をしていくイメージである。

	実施内容
STEP1	市役所内で閉じた環境を維持しながら利活用していくことを想定している
STEP2	公開範囲を限定しながら、施設事業者等に対して見せることを想定する また、その際には建物などの情報と連携することによって、図面上の整合などを取っていくことを想定している
STEP3	公開範囲については留意をしながら、一般市民に対しても閲覧できるようにする また、メタバース空間上での利活用についても想定をしていく

- 玉名市においては既に以下のようなメタバース空間を用意しており、これらの空間に接続していくことを目的としている。



玉名市メタバース



GOSHUIN RUMBLE -tamana-



玉名市水害避難シミュレーションVR



スマート防災ポータル

## (3)課題

- 既存の台帳と並行で管理を行う必要があるため、二重に管理を行うことが求められる。
- 内製化を行っていく上では、習熟に一定時間がかかる一方で、更新作業は頻繁に発生しないため、どのように運用していくかが課題である。

# CONTENTS

0.

はじめに

1.

地下埋設物等の現状課題と関連取組

1.1 地下埋設物等を取り巻く現状と課題

1.2 国の最新取組

2.

使用するデータの準備

2.1 既設管路の整備に必要なデータ

2.2 新設・移設管路の整備に必要なデータ

3.

地下埋設物等の3D都市モデルデータ作成

3.1 管路データを作る際の基本的な考え方

3.2 マンホール等の道路付属物の基本的な考え方

3.3 データ更新時の作業フロー

4.

システム運用の考え方・体制

4.1 既往システム等の運用方法・体制

4.2 市町村における先行事例

5.

今後の取組展開に向けて

付録 データ作成ツールマニュアル

## 5.1 想定されるユースケース(利用シーン)

地下埋設物関連のデータ整備によって、今後想定される利活用の方法としては以下のユースケースが期待できる。

### 5.1.1 都市開発事業者の効率的な情報収集支援

#### 1. 従来のお取組と課題

開発行為や工事申請の際、都市開発事業者は地下埋設物の位置を確認するため、自治体の窓口で膨大な紙図面を閲覧する必要がある。複数の管路種別が複雑に交錯する地下空間において、該当する図面を探し出し、正確な位置関係を把握することは、申請者にも窓口職員にも大きな負担となっている。

#### 2. モデルの活用シーン(概要)

3次元データによるデジタル化により、申請者はオンライン上で必要な範囲の地下埋設物情報を検索・閲覧できるようになる。住所や地番を入力するだけで該当箇所の管路配置が立体的に表示され、詳細情報も即座に確認可能となる。窓口職員にとっても、工事予定箇所の周辺状況や既存工事との重複確認がシステム上で完結するため、申請内容の審査や事前協議が効率化される。

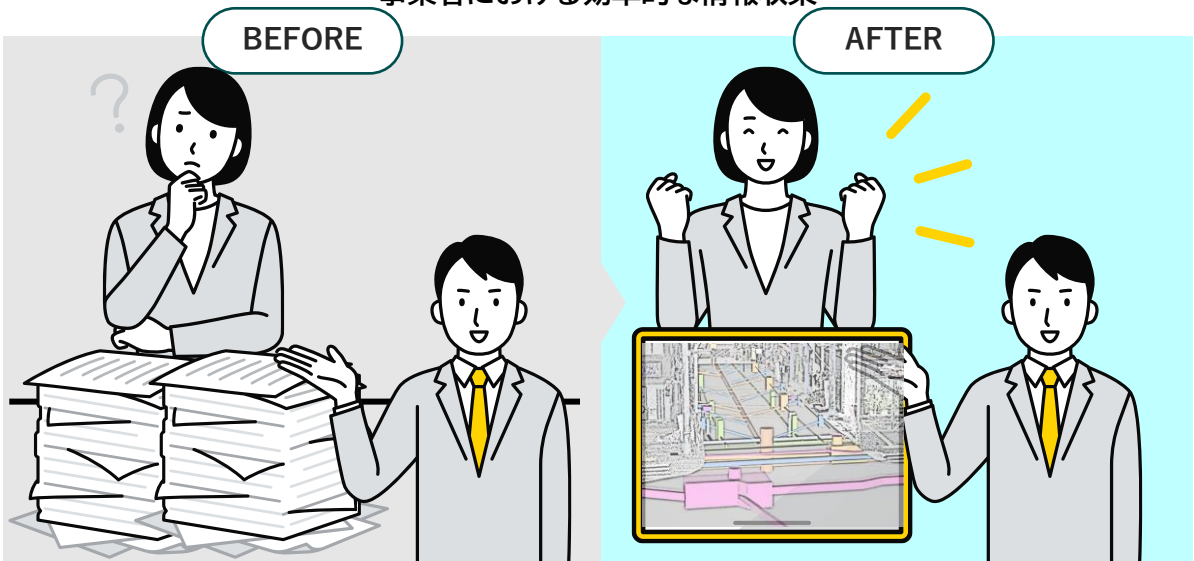
#### 3. 期待される効果

窓口訪問回数の削減や問合せ対応時間の短縮により、申請者の負担軽減と窓口業務の効率化が実現する。また、申請から許可までのリードタイムが短縮され、事業者の工程管理も改善される。

#### 4. 想定される検討事項とその対応方針

想定される検討事項	対応方針
情報公開範囲	セキュリティ上の理由から公開可能な情報レベルの整理
利用者の認証携帯	マイナンバー連携／事業者登録による認証の導入検討
利用規約	データ精度の免責事項／データの二次利用に関する規定の整備

#### 事業者における効率的な情報収集



## 5.1.2 関係者間での工事計画調整の効率化と情報共有支援

### 1. 従来取組と課題

従来の工事では、紙図面の情報のみに基づいて施工計画を立てるため、実際に試掘してみると管路の位置が図面と異なっているケースが頻発している。これにより、再設計や工程の見直しが必要となり、工期の遅延やコスト増加が発生している。また、複数の事業者が同じエリアで工事を行う際、地下埋設物の情報共有が不十分なため、調整に時間を要する。

### 2. モデルの活用シーン(概要)

進行中の水道管工事において、竣工時の本設位置を反映した3次元モデルを構築し、デジタルインフラ台帳として蓄積する。施工前には既存埋設物の正確な位置を3次元で確認でき、他事業者との工事調整もシステム上で行うことが可能となる。また、設計段階で施工方法や仮設計画の妥当性を検証できるため、現場での手戻りを最小化できる。

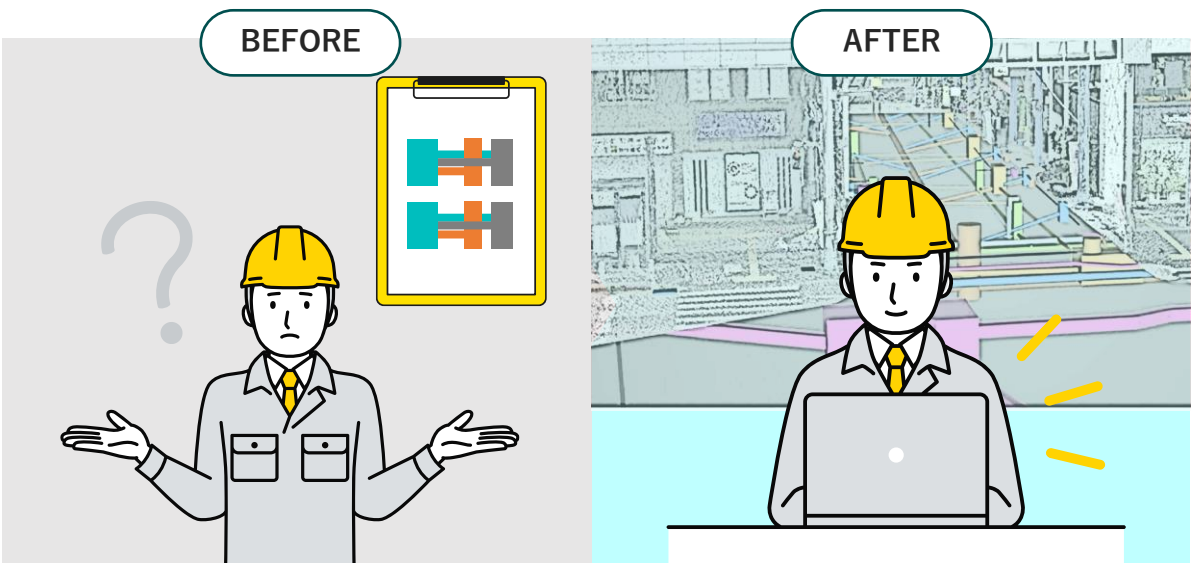
### 3. 期待される効果

試掘回数の削減に、より工期短縮とコスト削減が期待できる。また、事前の干渉チェックにより、施工中のトラブルや事故リスクが低減され、安全性が向上する。さらに、複数事業者間での工事スケジュールの調整が効率化され、道路掘削の重複や近接工事による住民負担の軽減にもつながる。

### 4. 想定される検討事項とその対応方針

想定される検討事項	対応方針
データ更新体制	竣工情報の迅速な反映フロー／更新責任部署の明確化
公開範囲	工事関係者と庁内利用の区分／事業者間での情報共有ルールの構築
データ精度	竣工測量精度の基準設定／図面との整合性確認方法の検討
システム連携	既存の工事調整／道路占用管理システムとの連携方法の検討

#### 関係者間における工事計画の調整



## 5.1.3 埋設設備の長寿命化計画及び維持管理業務への活用

### 1. 従来取組と課題

紙図面を基にした管路管理では、過去の補修履歴や更新工事の記録が散在しており、管路の現況把握に多大な労力を要していた。敷設年度や補修箇所を特定するためには、膨大な工事台帳を遡る必要があり、老朽化が進む管路の優先順位付けが困難であった。また、部分的な更新が繰り返された結果、どの区間がどのように継ぎ接ぎされているのか全体像を把握できず、計画的に進まない状況があった。

### 2. モデルの活用シーン(概要)

3次元データに敷設年度、管種、過去の補修履歴を統合することで、管路の経年劣化状況を一元的に可視化できる。老朽度や重要度に基づいて更新優先順位を自動的にランク付けし、限られた予算の中で効果的な更新計画を立案できる。また、点検記録や劣化診断結果をデジタルデータとして蓄積することで、予防保全型の維持管理に移行し、突発的な事故や漏水のリスクを低減できる。

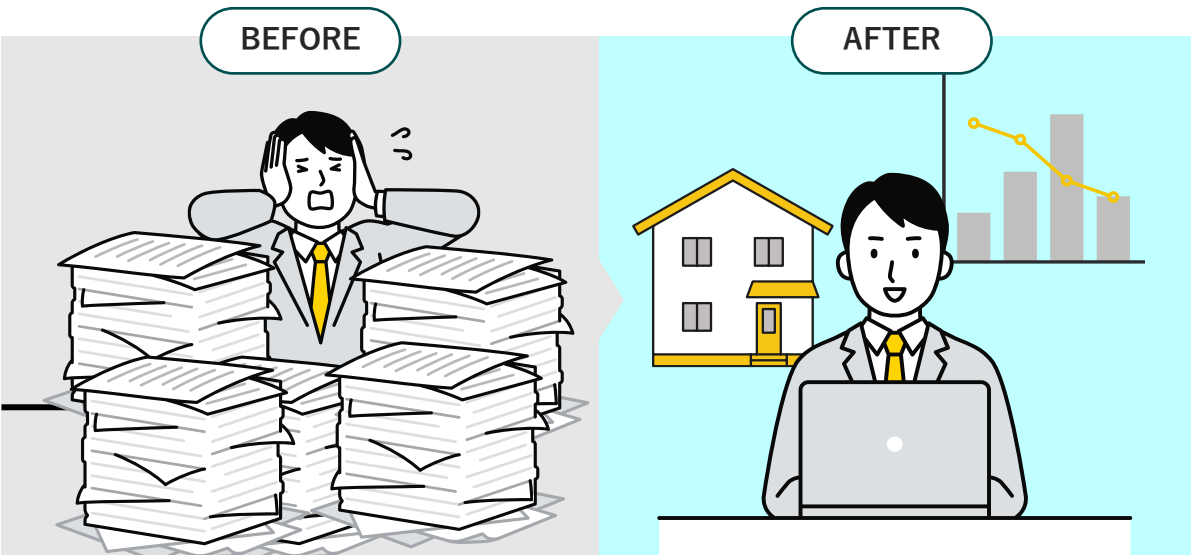
### 3. 期待される効果

計画的な更新により、突発的な事故対応や緊急工事が削減され、維持管理コストの平準化が図れる。管路の健全度を定量的に評価できるため、議会や住民への説明資料としても活用でき、予算確保の根拠が明確になる。さらに、点検業務の効率化により、職員の業務負担が軽減され、限られた人員でも適切なアセットマネジメントが実現する。

### 4. 想定される検討事項とその対応方針

想定される検討事項	対応方針
データ統合	既存の維持管理台帳や点検記録システムとの連携検討
劣化診断手法の標準化	健全度評価基準の設定、目視点検やデジタルデータと突合
更新計画への反映プロセス	長寿命化計画や財政計画との連動を検討
データ更新体制	点検結果や工事完了情報の迅速な反映フローの構築

#### 長寿命化計画や維持管理業務への活用



## 5.1.4 点群データ取得によるモデル更新を通じた行政検査の効率化・品質確保

### 1. 従来取組と課題

工事現場では、施工後の管路などを写真で記録する必要があるが、撮影角度や距離によっては必要な部位が写らず、記録の抜け漏れや判断に必要な情報不足が生じるリスクがある。特に、埋め戻し後は撮り直しが難しく、写真撮影は一回で失敗できない高負荷な作業となっている。また、2次元写真では奥行き方向の位置関係や他設備との離隔など、空間的な状況の把握に限界がある。

### 2. モデルの活用シーン(概要)

工事中・完了時の状況を点群データとして360度・高精度に取得し、3D都市モデルとして保存することで、任意の視点から後で必要な箇所を確認できる。設計データや既存の3D都市モデルと重ね合わせることで、設計との差異や他インフラとの離隔・干渉状況を行政側の机上でも確認可能となる。

### 3. 期待される効果

完了検査の抜け漏れ防止から点群から任意の角度で再確認できるため、不足カットや見落としを防ぐことができるようになる。また、3D上で位置関係や寸法を示せるため、設計との差異や安全上のクリアランスを客観的に説明できる。加えて、検査のための点群が、将来の更新工事や資産管理の基盤データとしても活用可能となる。

### 4. 想定される検討事項とその対応方針

想定される検討事項	対応方針
点群データの保管／ 長期アーカイブ化	大規模データを保存する際のルール決め(保存期間などの設定)
データの取扱い	セキュリティ範囲の特定／プライバシーポリシーの整備

#### 行政検査への活用



## 5.1.5 災害時における復旧優先順位の迅速な判断への活用

### 1. 従来取組と課題

災害時におけるライフラインの迅速な復旧は、住民生活の早期回復に不可欠である。しかしながら、従来の紙図面による管理では、被災箇所の特定に時間を要し、被害の根本原因の究明にも多大な労力がかかる。また、複数の管路が複雑に交錯する地下空間において、2次元図面のみでは被害範囲の全体像を把握することが困難である。

### 2. モデルの活用シーン(概要)

3次元データ化により、管路の配置や接続関係を視覚的かつ正確に把握できるようになる。災害発生時には、被害情報をデジタルデータとして集約し、被災箇所の特定から応急措置の判断まで、一連の対応をシステム上で迅速に行うことが可能となる。

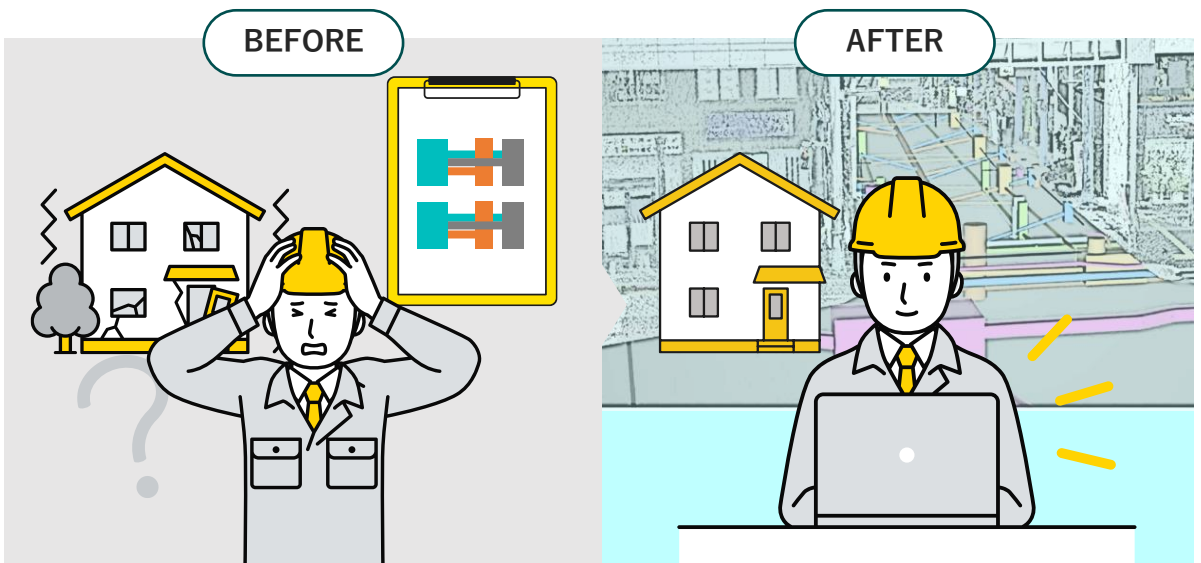
### 3. 期待される効果

重要度の高い管路(病院や避難所への供給ルート等)を事前に設定することで、災害時の点検箇所を優先順位付けできるようになる。これにより、限られたリソースを効果的に配分し、復旧作業の最適化が期待できる。さらに、過去の被害データを蓄積することで、脆弱性の高い箇所の事前補強や予防保全にもつながる。

### 4. 想定される検討事項とその対応方針

想定される検討事項	対応方針
データ精度	既存図面のデジタル化における許容誤差と検証方法の構築
システム連携	防災システムや被害報告システムとのデータ連携方式の検討
運用体制	平常時の更新フローと災害時の利用体制の整備
セキュリティ対策	重要インフラ情報の適切な管理とアクセス制限

#### 災害時の復旧優先順位の判断



## 5.1.6 事前防災検討のための内水氾濫等シミュレーション活用

### 1. 従来のお取組と課題

気候変動の影響により、都市型水害である内水氾濫が頻発している。豪雨時に排水管のキャパシティを超えた雨水が地表に溢れ出す内水氾濫では、被災範囲の特定や浸水原因の究明に時間を要している。

### 2. モデルの活用シーン(概要)

排水管路の3次元データ化により、管径や勾配、接続関係を含む排水能力を可視化できるようになる。また、浸水シミュレーションと組み合わせることで、豪雨時にボトルネックとなる箇所を事前に特定することができる。加えて、都市全体の排水能力についても推定を行うことができるようになり、地形データと組み合わせることでより高精度なシミュレーションを行うことができる。

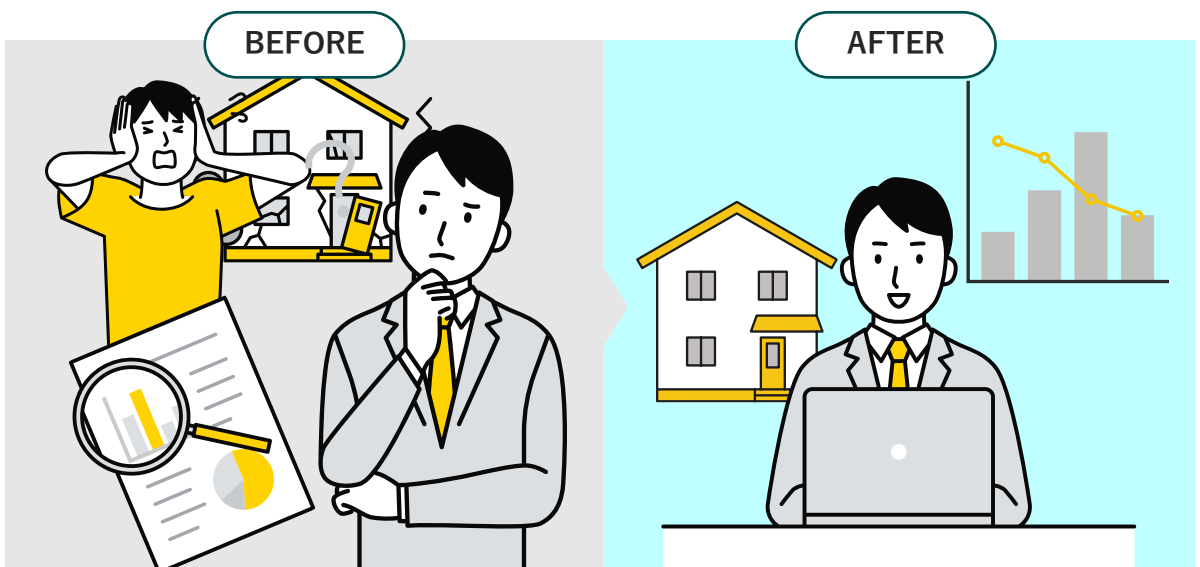
### 3. 期待される効果

内水氾濫のシミュレーションを行うことができるようになると、優先的に補修をすべき管路を特定することができるようになる。また、都市全域における排水能力についてもアップデートをかけることができ、都市の防災性向上が期待できる。

### 4. 想定される検討事項とその対応方針

想定される検討事項	対応方針
浸水シミュレーションとの連携	降雨データや地形データとの統合方式の検討
データ精度	管径・勾配・接続情報の正確性／老朽化状況の反映方法の構築
他部署との情報連携	関連部署間におけるデータ共有体制の構築
住民への情報提供範囲	浸水リスク情報の公開／ハザードマップへ反映を検討

### 内水氾濫シミュレーションによる防災対策



## 5.2 今後の取組展開に向けて

- これまでProject PLATEAUでは、地下埋設物モデルに関する実証を通じて、効率的な整備・更新手法や、計画検討・関係者協議のためのシステム等の開発、関連業務効率化の検証などが進められてきた。
- 本ハンドブックでは、その知見等も踏まえつつ、熊本県玉名市を例に、行政が保有している水道関連のGISデータから地下埋設物の3D都市モデルを構築するための具体方法や、その運用方法等について解説を行った。関係者間での安全な施設管理等の観点を踏まえたデータの共有範囲や運用方法、データ整備・更新手法等の整理といった課題はあるものの、現時点において地方公共団体やインフラ事業者における情報共有のツールとしての活用が期待できる。
- 今後は、本モデルの利点を活かしつつ、駅周辺等の拠点エリアにおける都市開発プロジェクト等、具体的な活用場面を想定しながら、都市開発等における3D都市モデルの有用性および管理運用体制の検証を段階的に進めていく必要がある。その先に、都市レベルでの統一的な仕様に基づくモデル整備・活用や、中長期的なまちづくりや空間整備の検討への活用といったことも期待できるであろう。
- さらに、BIM等の多様なデータとの連携を含む、地上・地下モデルの一体的な活用により、複数関係者が同一プラットフォーム上で移設計画等を調整するなど、プロジェクトの効率化・高度化を見据えた効果検証を通じて、実装地域の横展開の可能性を探っていくことが求められる。
- これらの取組を通じて、都市のデジタルツインの社会実装やデジタル技術を活用したまちづくりの推進が期待される。

# CONTENTS

0. はじめに
1. 地下埋設物等の現状課題と関連取組
  - 1.1 地下埋設物等を取り巻く現状と課題
  - 1.2 国の最新取組
2. 使用するデータの準備
  - 2.1 既設管路の整備に必要なデータ
  - 2.2 新設・移設管路の整備に必要なデータ
3. 地下埋設物等の3D都市モデルデータ作成
  - 3.1 管路データを作る際の基本的な考え方
  - 3.2 マンホール等の道路付属物の基本的な考え方
  - 3.3 データ更新時の作業フロー
4. システム運用の考え方・体制
  - 4.1 既往システム等の運用方法・体制
  - 4.2 市町村における先行事例
5. 今後の展望

付録 データ作成ツールマニュアル

## CONTENTS

1. データ整備のオリエンテーション
2. データ設計方針の立て方
3. 各ツールの設定
4. 3Dデータの確認方法
5. 外径を作成する処理

# 1. データ整備の考え方

## 1.1 データ作成における問題の把握

データ整備の本質は問題解決と同じアプローチで考えることが可能であり、「5W1H」に置き換えて状況を整理することができる。3D都市モデルを作るための「無い・できない」を「ある・できる」に変える(問題を解決する)ことが、「モデル化支援」の主旨となる。

#	要素 \ 状況	3D都市モデルを作ることができない (現在: AsIs)	3D都市モデルを作ることができる (将来: ToBe)
1	When (いつ)	3D都市モデルが欲しい時(要望発生時)に	3D都市モデルが欲しい時(要望発生時)に
2	Where (どこで)	庁舎内(事務所内)で	庁舎内(事務所内)で
3	Who (だれが)	一般的な自治体(インフラ事業者)職員が	データ整備手法を習得した自治体(インフラ事業者)職員が
4	What (なにを)	3D都市モデルを	3D都市モデルを
5	How (どのように)	作る(整備する)ことができない	作る(整備する)ことができる
6	Why(なぜ)	庁舎内(事務所内)に材料は足りず、 <b>道具が無い</b> 。また、 <b>作り方も分からない</b>	庁舎内(事務所内)に <b>材料・道具・作り方の三つとも揃っている</b>

## 1.2 データ作成における問題の把握

「5W1H」で整理した状況を、知っている事例で単語を置き換えると問題点をイメージしやすくなる。状況がイメージできた後は、元の単語に戻すことで解決方法の糸口を見つけることができる。

#	要素 \ 状況	目玉焼きを作ることができる (ToBe)	目玉焼きを作ることができない (AsIs)
1	When (いつ)	目玉焼きが欲しい(食べたい)時に	目玉焼きが欲しい(食べたい)時に
2	Where (どこで)	台所で	台所で
3	Who (だれが)	調理経験のある人が	調理経験のない人が
4	What (なにを)	目玉焼きを	目玉焼きを
5	How (どのように)	作ることが <b>できる</b>	作ることが <b>できない</b>
6	Why(なぜ)	台所に材料・道具は揃っており、調理経験のある人は <b>道具の使い方と目玉焼きの作り方を知っている</b>	台所に材料・道具は揃っているが、調理経験のない人は <b>道具の使い方と目玉焼きの作り方を知らない</b>

元の単語に戻すと・・・

庁舎内に材料と道具は揃っているが、**データ整備経験のない人はデータ整備ツール(道具)の使い方と3D都市モデル(作業結果)の作り方を知らない**

## 1.3 解決方法の検討

具体的にイメージできる事例を基に、不足しているモノ(材料・道具)と情報(使い方・作り方)を把握する。足りないモノや情報を確認したら、欲しい成果を外部から調達するのではなく、材料や道具を外部調達や代替品の作成で対応できないか考える。

#	比較項目	目玉焼き	3D都市モデル
1	要求事項	目玉焼きが食べたい	GISデータ(3D都市モデル)が作りたい
2	作業材料(必須)	自家製:卵	自家製:上下水道の設備データ(シェープファイル) 調達:地表面の標高値を示すデータ(DEM)(国土地理院提供)
3	作業材料(任意)	調達:油、塩、味の素、しょうゆ、砂糖、トマト、キャベツ等	調達:上下水道の設備データを作成する際に使用した背景地図データ(例:都市計画図・道路台帳附図など)
4	作業道具	フライパン、コンロ、材料を入れる器、菜箸、包丁、盛り付ける皿等	<新手法> データ整備ツール <従来手法> プログラミング言語、データベースソフトウェア等
4	作業人	料理人	データ整備主幹
5	作業成果	目玉焼き(及第点) 定義:卵をフライパンで割り入れて焼いた料理  <b>美味しい目玉焼き(成功作)</b> 定義:調理方法や工程の工夫と下味や彩り等に工夫を利用した、卵をフライパンで割り入れて焼いた料理	GISデータのファイル 定義:GIS分野の概念(※)に準じたデータ構造を持つデータ  <b>3D都市モデルのデータファイル</b> 定義:3D都市モデルの標準製品仕様書に準じたデータ構造を持つ、GIS分野の概念(※)に準じたデータ構造を持つデータ

GISデータを言い換えると…  
後続のシステムが要求するデータ仕様と合致しないため、  
後続のシステムでエラーが発生  
=目玉焼き(及第点)

当プロジェクトにおける「モデル化支援」の主旨を具体的に表現すると、「美味しい目玉焼きを作るための材料・道具や手法」について、「無い・できない」状況を「ある・できる」状況に変えるための学習機会設けて、習得を支援するをすること、となる。

3D都市モデルを言い換えると…  
後続のシステムが要求するデータ仕様と合致するため、  
後続のシステムが期待通りに動作  
=美味しい目玉焼き

※データソース、レイヤ、地物、ジオメトリ、属性

先のページでは、不足している材料・道具と情報(使い方の把握・分析)をおこなった。ここでは、手順について確認を行い実際にどのようなものが不足するか、実現するためにどのようなものが必要かを検討する。

#	比較項目	目玉焼き	3D都市モデル
6-1	レシピ (及第点)	<p>美味しくないとレシピ(定義とおりの基本動作のみ)</p> <ol style="list-style-type: none"> <li>卵を割る</li> <li>割った卵の中身をフライパンへ入れる</li> <li>フライパンを加熱</li> <li>完成</li> </ol> <p>→細部について疑問点を残しつつ、とりあえず料理は完成する</p>	<p>GISデータのファイル</p> <ol style="list-style-type: none"> <li>ファイル/ツールを開く</li> <li>属性項目を編集する</li> <li>ジオメトリ項目を編集する</li> <li>新しいファイルに編集結果を書き込む</li> <li>完成</li> </ol> <p>→何をどのように使えばいいかわからないGISのデータができてあがる</p>
6-2	レシピ (成功)	<p>美味しいレシピ (定義通りの基本動作に加えて、アレンジあり)</p> <ol style="list-style-type: none"> <li>フライパンに油をひく(準備) 何故油が必要? →熱した面が焦げ付かないようにするため(要望の実現)</li> <li>フライパンを熱する(準備) 何度まで加熱する? →〇〇度。これ以上は卵が焦げ付く</li> <li>卵を割って中身に下味をつける。(準備+基本動作) 何故下味をつける? →美味しく食べたいので(要望の実現)</li> <li>卵の中身を熱したフライパンへ入れる。(基本動作) 何故卵の中身を熱したフライパンへ入れる? →目玉焼きを作る工法</li> <li>盛り付ける。彩りを考慮して、トマトやキャベツを添える →見た目も重視</li> <li>完成</li> </ol>	<p>3D都市モデルのファイル</p> <ol style="list-style-type: none"> <li>ファイル/ツールを開く <b>なぜ開く必要があるか?</b> →WordやExcelのように編集を行うための作業スペースが必要であるため。</li> <li>属性項目を編集する <b>どんな項目を入れる必要があるか?</b> →地下埋設物に関連する管経等が必要 →管路の情報を確認するために必要(要望の実現のため)</li> <li>2Dのジオメトリ項目を3Dのジオメトリ項目にする <b>なぜそうするのか?</b> →3Dで位置関係を把握するために必要(要望の実現)</li> <li>3Dのジオメトリ項目を3D都市モデルのジオメトリ項目にする。 <b>なぜこれをするのか?</b> →管路の情報を確認するときが必要</li> <li>3D都市モデルのジオメトリ項目と編集した属性項目を作法に則ってファイルに書き込む <b>なぜこれをするのか?</b> →データ更新の際に、書き込むため →精度を保つことで正確なデータを使用することができる。</li> <li>完成</li> </ol>



用語や手法がイメージできないものや行動の必要性が説明できないものは、**自身で作業結果の良し悪しを判断できない。**



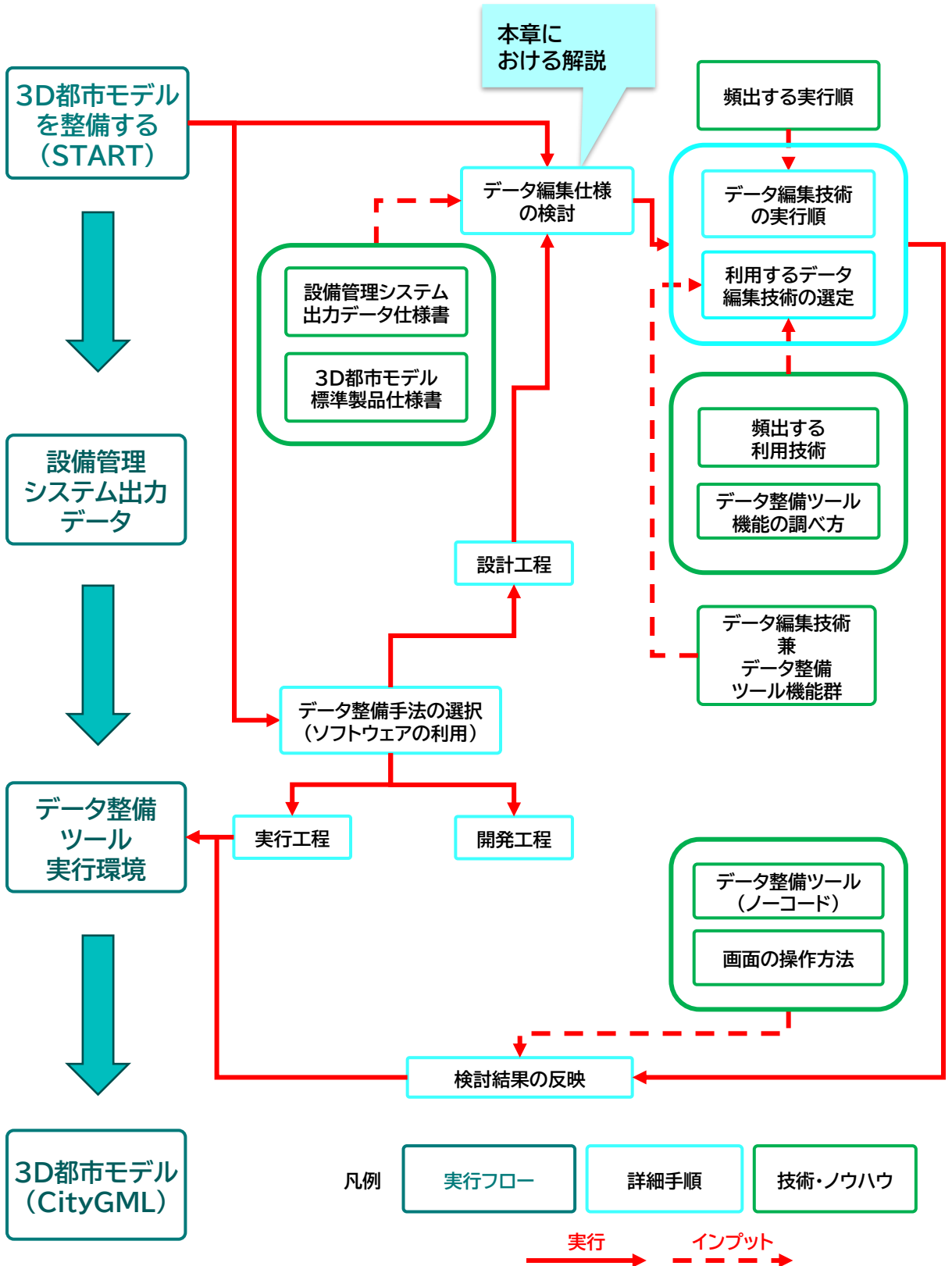
## CONTENTS

1. データ整備のオリエンテーション
2. データ設計方針の立て方
3. 各ツールの設定
4. 3Dデータの確認方法
5. 外径を作成する処理

## 2. データ設計方針の立て方

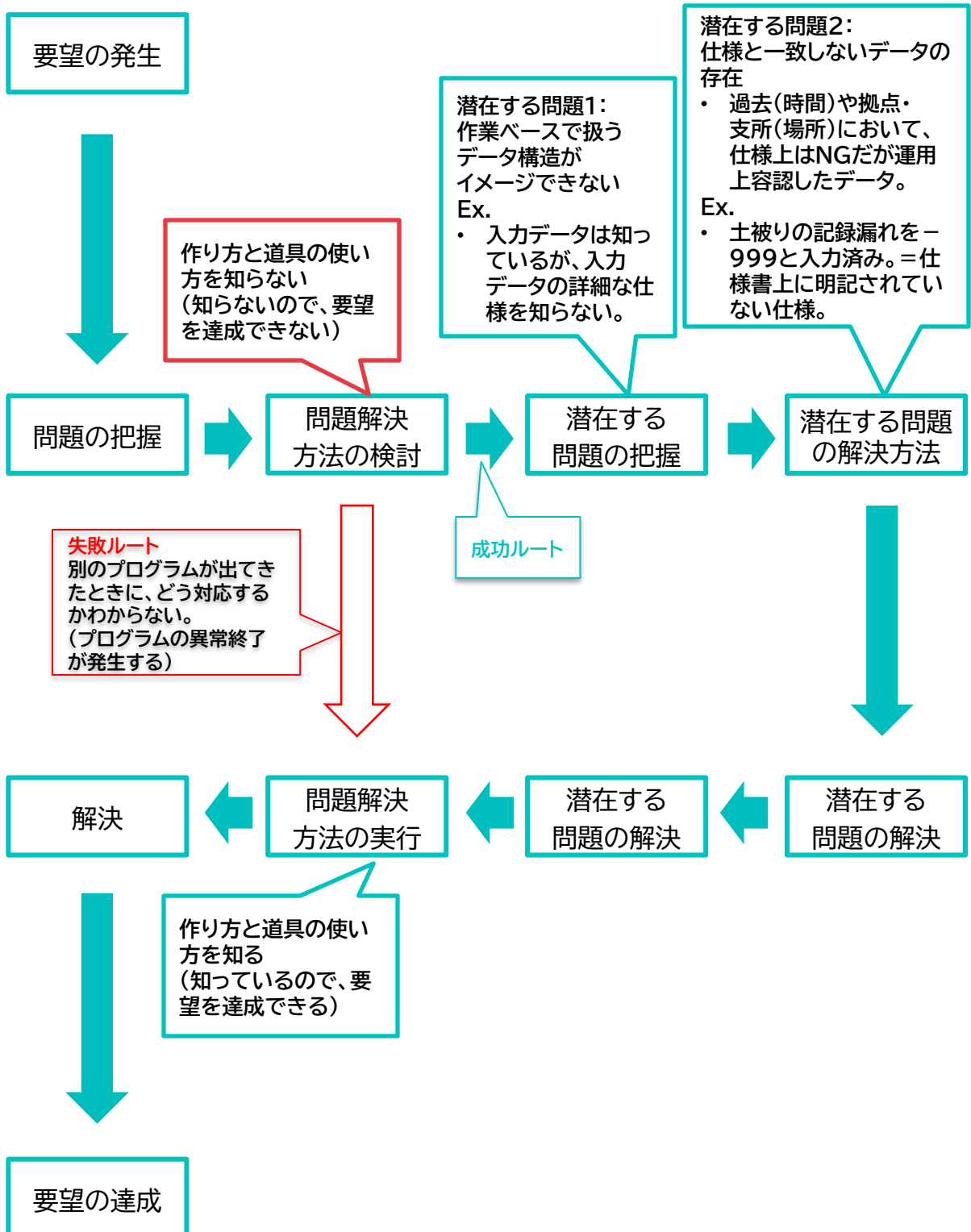
### 2.1 解決に向けた工程の体系化

本性における解説は、下図の「データ編集仕様の検討」を対象とする。



## 2.2 データ整備のプロセスと、潜在する問題の把握・解決

データ整備の本質は問題解決と同じアプローチで考えることが可能であり、「5W1H」に置き換えて状況を整理することができる。3D都市モデルを作るための「無い・できない」を「ある・できる」に変える(問題を解決する)ことが、「モデル化支援」の主旨となる。



## 2.3 潜在する問題1:仕様の理解(3D都市モデル)

「予め決まった項目に対し自分で付け足す」ことでユースケースで利用する3D都市モデルを定義する。各モデルの定義は以下の通りになっている。

定義名	説明
XML	独自のタグを定義してデータに意味や構造を持たせ、システム間でのデータ交換や管理を容易にする拡張することができる言語
GML	地理空間データに関する標準化団体であるOpen Geospatial Consortium(OGC)が策定した、地理情報として保管が必要な要素を含むXMLのデータ
CityGML	地理空間データに関する標準化団体であるOpen Geospatial Consortium(OGC)が策定した3D都市モデルのためのオープンデータモデル及びデータ形式の国際標準
i-UR	CityGMLの拡張規則に基づき、内閣府地方創生推進事務局が日本の都市で必要なデータを拡張した技術仕様、CityGMLだけでは不足する項目については、i-URとして追加されている
3D都市モデル標準製品仕様書	国土交通省都市局がCityGML及びi-URに基づき策定した、日本における3D都市モデルの標準仕様 例:地下埋設物モデル(水道管、下水道管等)
3D都市モデル拡張製品仕様書	各データ整備者(例:玉名市)が3D都市モデル標準製品仕様書から必要な地物及び属性を抽出・追加した製品仕様

当ユースケースの  
ツール作成マニュアルの  
範囲

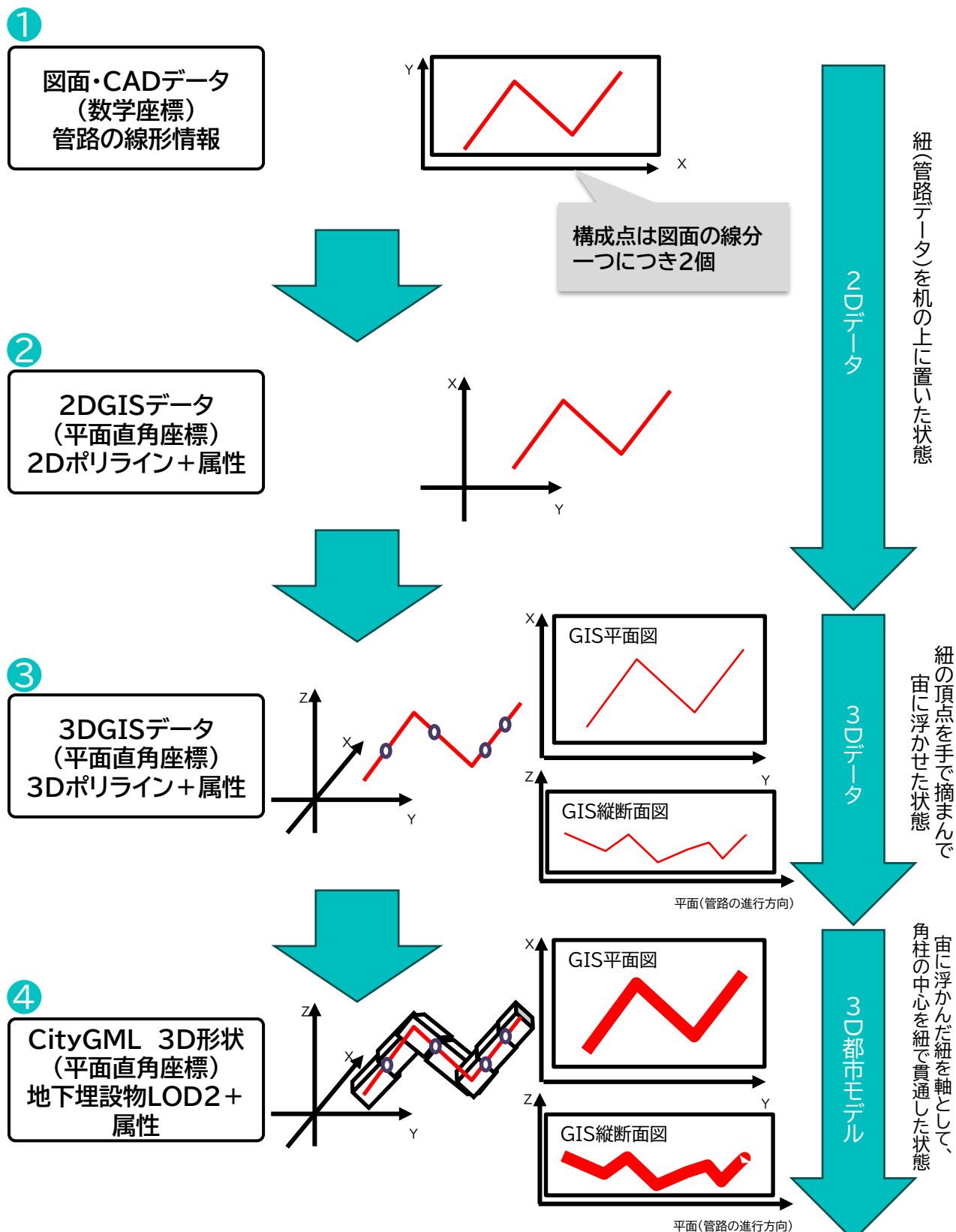
原則として、  
継承した定義の項目は  
変更・削除ができない

標準製品仕様書にて、  
継承した定義(※)の  
記載あり  
※i-UR・CityGML・  
GML=入門として読む  
資料はこれ一つで済むよ  
うにまとめられている

3D都市モデルで実現し  
たい内容に含まれるデー  
タ項目が標準製品仕様書  
に不足している場合、拡  
張製品仕様書で追加

## 2.4 潜在する問題1:仕様の理解(3D都市モデルのジオメトリ)

当ユースケースで出力する3D都市モデルのジオメトリ項目(空間属性)は、LOD2とする2DのジオメトリからLOD2のジオメトリを作るまでの変遷は以下のとおり。



## 2.5 潜在する問題2:仕様の理解(3D都市モデルの属性)

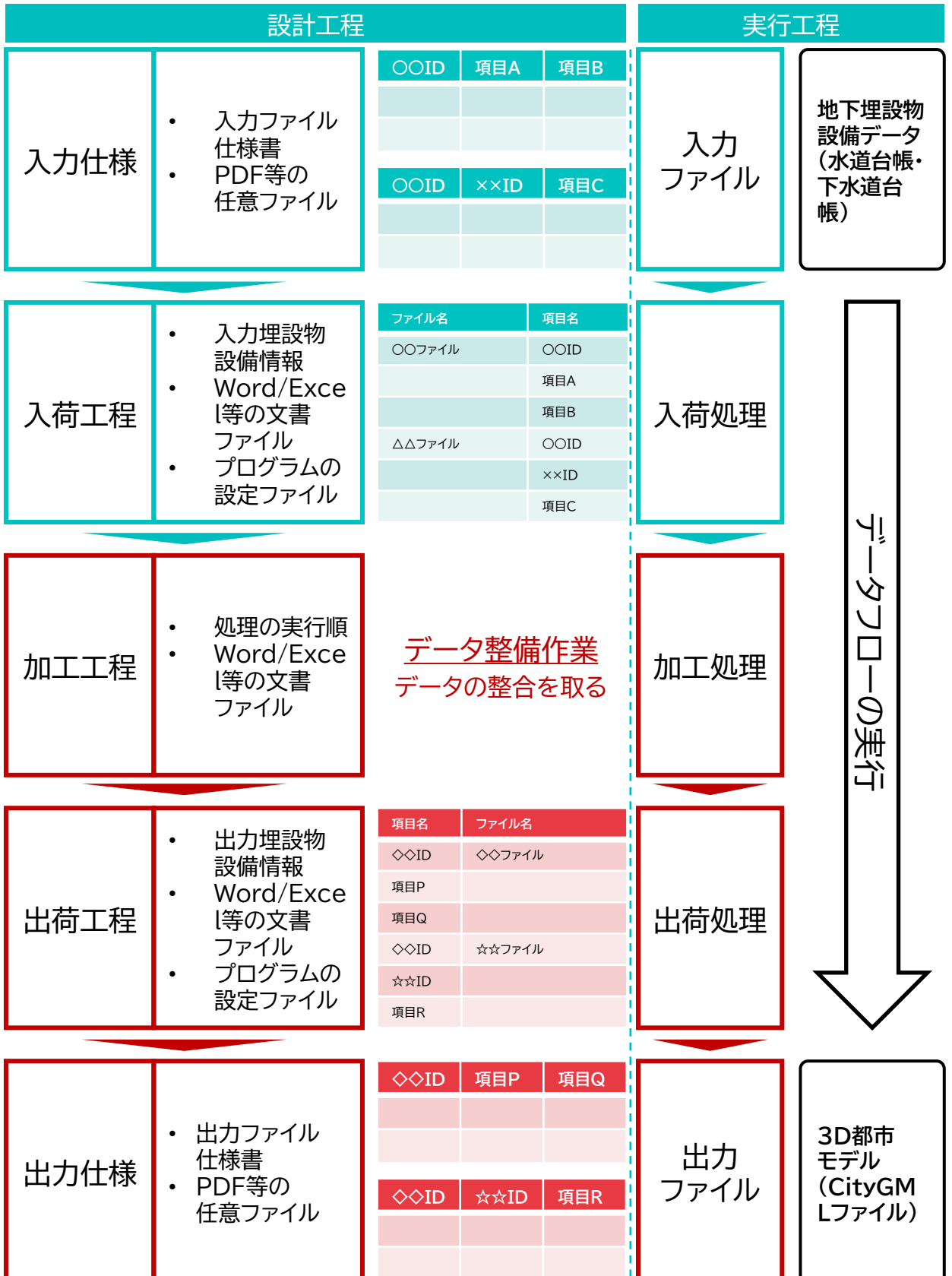
水道台帳として管理している項目で、3D都市モデルと合致すると考えられる属性(主題属性)は次のとおり。

#	主題属性属性名	属性の型及び多重度	定義
1	<code>gml:description [ _Feature ]</code>	<code>gml:StringOrRefType [0..1]</code>	都市オブジェクトの概要。
2	<code>gml:name [ _Feature ]</code>	<code>gml::CodeType [0..1]</code>	都市オブジェクトを識別する名称。文字列とする。
3	<code>core:creationDate [ _CityObject ]</code>	<code>xs::date [0..1]</code>	データが作成された日。運用上必須とする。
4	<code>uro:year [ UtilityNetworkElement ]</code>	<code>xs::gYear [0..1]</code>	埋設された年度。
5	<code>uro:yearType [ UtilityNetworkElement ]</code>	<code>gml::CodeType [0..1]</code>	埋設された年度の確からしさ。コードリスト ( <a href="#">UtilityNetworkElement_yearType.xml</a> )より選択する。
6	<code>uro:depth [ UtilityLink ]</code>	<code>gml::LengthType [0..1]</code>	土被りの深さ。単位はm。
7	<code>uro:material [ UtilityLink ]</code>	<code>gml::CodeType [0..1]</code>	材質の種類。コードリスト ( <a href="#">UtilityNetworkElement_material.xml</a> )より選択する。
8	<code>uro:innerDiameter [ Pipe ]</code>	<code>gml::LengthType [0..1]</code>	内径。単位はmm又はinchとする。
9	<code>uro:outerDiameter [ Pipe ]</code>	<code>gml::LengthType [0..1]</code>	外径。単位はmm又はinchとする。
10	<code>uro:sleeveType [ Pipe ]</code>	<code>gml::CodeType [0..1]</code>	被覆の有無。コードリスト ( <a href="#">Pipe_sleeveType.xml</a> )より選択する。

上記以外にも3D都市モデルの属性項目は多数存在する詳細は標準製品仕様書の「4.15.3 地下埋設物モデルの応用スキーマ文書」を参照

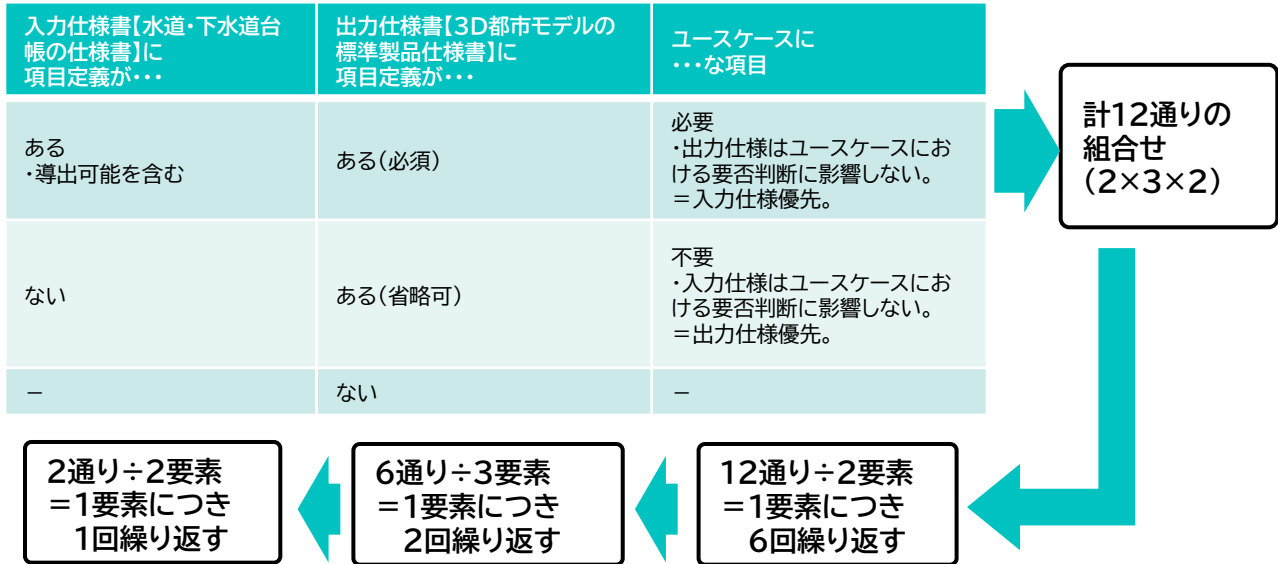
## 2.6 問題解決方法(データ整備工程)の検討について

3D都市モデルのデータ作成方法は、加工工程の「データ整備作業」で行う組合せを決めることが重要。



## 2.7 データ整備設計方針の検討(バリエーション網羅)

特定の条件によって項目(列)を出力する・しないの判断が変わる場合を、網羅的に検討する。



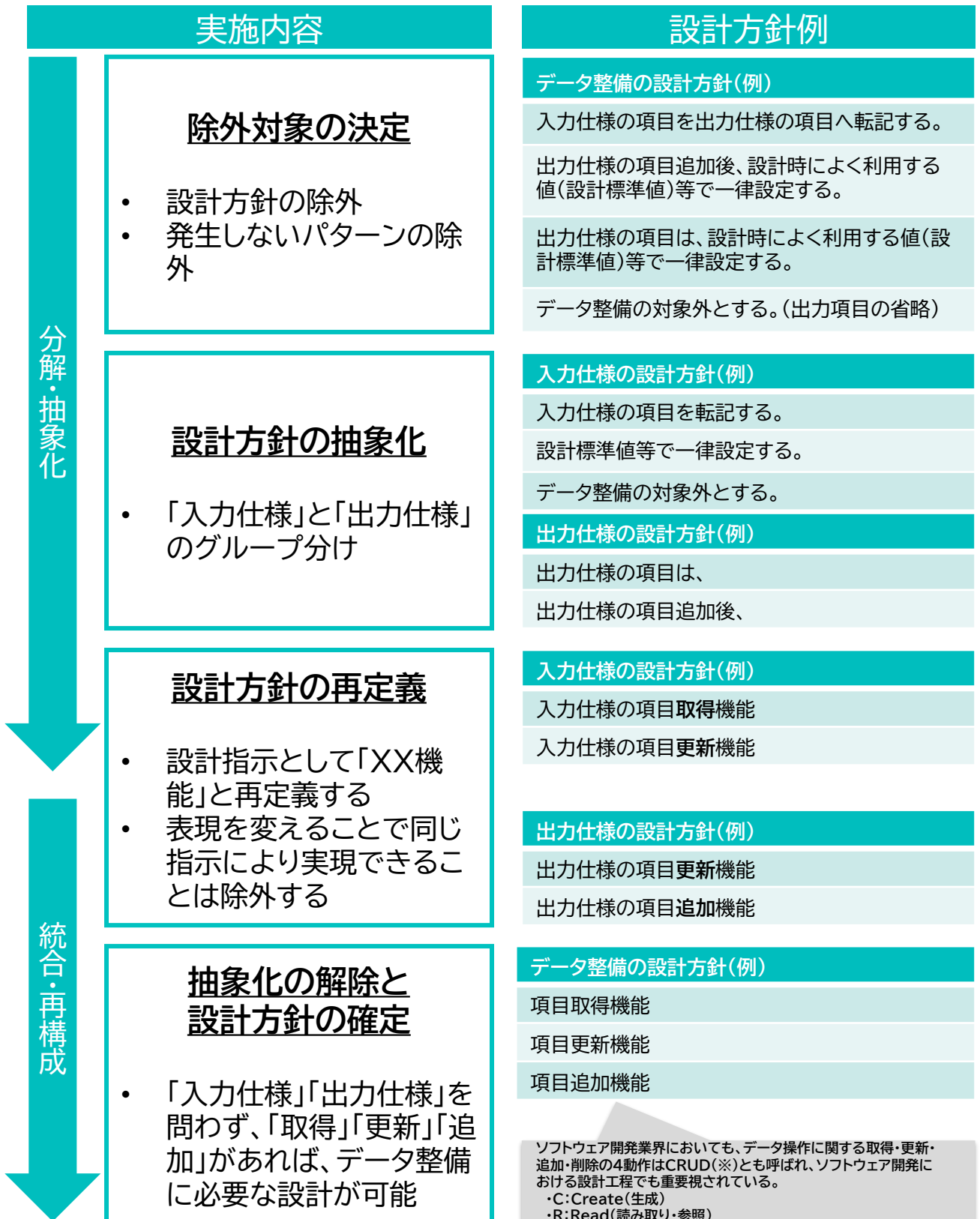
## 2.8 データ整備の設計方針検討(シミュレーション)

項目の要素(場合分け)が示す状況が意味するものをシミュレーションし、データ整備の設計方針を定める。

#	入力仕様書に項目定義が...	出力仕様書に項目定義が...	ユースケースに...な項目	左記が示す状況のシミュレーション(例)	データ整備の設計方針(例)
1	ある	ある(必須)	必要(入力仕様優先)	入力仕様があり、出力仕様もある(必須)。入力仕様を優先	入力仕様の項目を出力仕様の項目へ転記する。
2	ある	ある(省略可)	必要(入力仕様優先)	入力仕様があり、出力仕様はある(省略可)。入力仕様を優先	入力仕様の項目を出力仕様の項目へ転記する。
3	ある	無い	必要(入力仕様優先)	入力仕様があり、出力仕様は無い。入力仕様を優先	出力仕様の項目追加後、設計時によく利用する値(設計標準値)等で一律設定する。
4	ある	ある(必須)	不要(出力仕様優先)	入力仕様があり、出力仕様もある(必須)。出力仕様を優先	出力仕様の項目は、設計時によく利用する値(設計標準値)等で一律設定する。
5	ある	ある(省略可)	不要(出力仕様優先)	入力仕様があり、出力仕様もある(省略可)。出力仕様を優先	データ整備の対象外とする。(出力項目の省略)
6	ある	無い	不要(出力仕様優先)	入力仕様があり、出力仕様は無い。出力仕様を優先	データ整備の対象外とする。(出力項目の省略)
7	無い	ある(必須)	必要(入力仕様優先)	入力仕様が無く、出力仕様はある(必須)。入力仕様を優先	出力仕様の項目は、設計時によく利用する値(設計標準値)等で一律設定する。
8	無い	ある(省略可)	必要(入力仕様優先)	入力仕様が無く、出力仕様はある(省略可)。入力仕様を優先	出力仕様の項目は、設計時によく利用する値(設計標準値)等で一律設定する。
9	無い	無い	必要(入力仕様優先)	入力仕様が無く、出力仕様も無い。入力仕様を優先	出力仕様の項目追加後、設計時によく利用する値(設計標準値)等で一律設定する。
10	無い	ある(必須)	不要(出力仕様優先)	入力仕様が無く、出力仕様はある(必須)。出力仕様を優先	出力仕様の項目は、設計時によく利用する値(設計標準値)等で一律設定する。
11	無い	ある(省略可)	不要(出力仕様優先)	入力仕様が無く、出力仕様はある(省略可)。出力仕様を優先	データ整備の対象外とする。(出力項目の省略)
12	無い	無い	不要(出力仕様優先)	入力仕様が無く、出力仕様も無い。出力仕様を優先	何も無い(理論上のみ存在するパターンであり、実際に発生することはない。)

## 2.9 設計方針の検討と確定(分解・抽象化と統合・再構成)

場合分けが示す状況から要素へ分解後、設計要素に再構成することで、データ整備の設計方針を定める。



ソフトウェア開発業界においても、データ操作に関する取得・更新・追加・削除の4動作はCRUD(※)とも呼ばれ、ソフトウェア開発における設計工程でも重要視されている。

- ・C: Create(生成)
- ・R: Read(読み取り・参照)
- ・U: Update(更新)
- ・D: Delete(削除)

## CONTENTS

1. データ整備のオリエンテーション
2. データ設計方針の立て方
3. **各ツールの設定**
4. 3Dデータの確認方法
5. 外径を作成する処理

## 3.1 OpenJDKの設定

本章では、モデルの整備・更新のための各ツールのセットアップ方法を解説する。

### 3.1.1 OpenJDKのダウンロード

- ①こちら(<https://jdk.java.net/25/>)にWebブラウザでアクセスする
- ②Windows/x64欄のzipリンクをクリックする
  - ※利用環境に適したものをダウンロードする
  - ※最新のバージョンを利用する

#### OpenJDK JDK 25.0.2 General-Availability Release

This page provides production-ready open-source builds of the Java Development Kit, version 25, an implementation of the Java SE 25 Platform under the GNU General Public License, version 2, with the Classpath Exception.

Commercial builds of JDK 25.0.2 from Oracle, under a non-open-source license, can be found [here](#).

#### Documentation

- [Features](#)
- [Release notes](#)
- [API Javadoc](#)

#### Builds

<b>Linux/AArch64</b>	<a href="#">tar.gz</a>	220266156 bytes	(sha256)
<b>Linux/x64</b>	<a href="#">tar.gz</a>	222522097	(sha256)
<b>macOS/AArch64</b>	<a href="#">tar.gz</a>	215449592	(sha256)
<b>macOS/x64</b>	<a href="#">tar.gz</a>	217744020	(sha256)
<b>Windows/x64</b>	<a href="#">zip</a>	221671696	(sha256)

#### Notes

- If you have difficulty downloading any of these files please contact [download-help@openjdk.org](mailto:download-help@openjdk.org).
- These builds reflect the current state of development. Each build is accompanied by release notes that describe the features and changes in that build. A [JDK Enhancement Proposal \(JEP\)](#) recently targeted to this release might not be in the current build, but will appear in a future build.

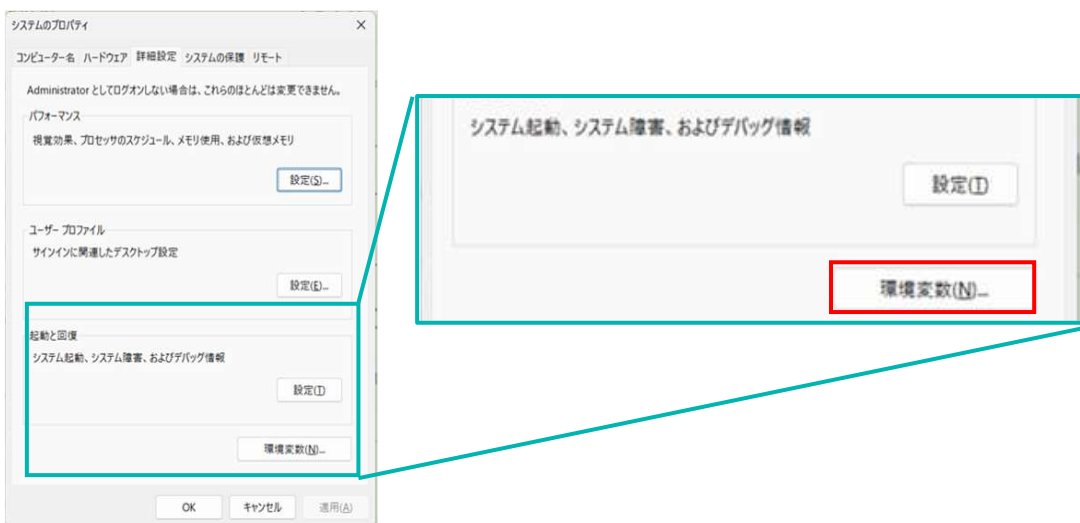
- ③ZIPファイルがダウンロードできたことを確認する

## 3.1.2 OpenJDKの初期設定

- ①ダウンロードしたZIPファイルを、任意の場所に解凍する
- ②Windowsアイコンを右クリックして、「システム」をクリックする
- ③「システム>バージョン情報」画面の中段にある「システムの詳細設定」をクリックする

関連リンク [ドメインまたはワークグループ](#) [システムの保護](#) [システムの詳細設定](#)

- ④「システムのプロパティ」画面の「環境変数」ボタンをクリックする



- ⑤「システム環境変数」側の新規ボタンをクリックする

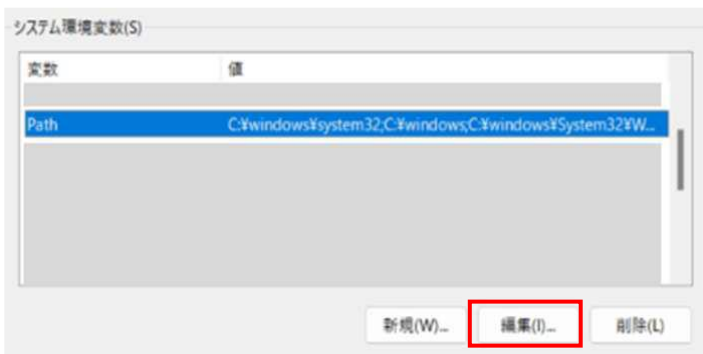


⑥「システム変数の編集」画面が表示されたら、以下内容を設定する  
設定完了後にOKボタンをクリックする

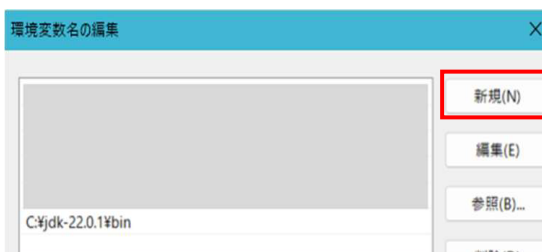
- 変数名: JAVA\_HOME
- 変数値: 1. で解凍したOpenJDKのフォルダパス  
例)Cドライブ直下に解凍した場合:  
C:¥jdk-22.0.1



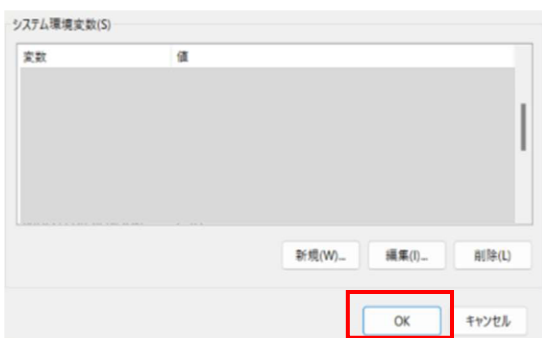
⑦「システム環境変数」に戻り、次に「Path」のボタンをクリックする  
その後に「編集」ボタンをクリックする



⑧「システム環境変数」側の新規ボタンをクリックする



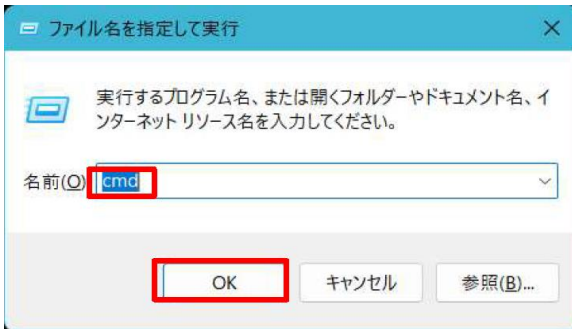
⑨最後に、「環境変数」画面のOKボタンをクリックすれば、初期設定は完了



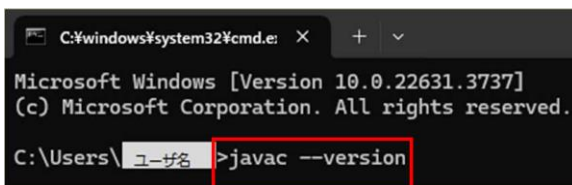
### 3.1.3 OpenJDKの動作確認

①Windowsキー+Rキーを押下する

②「ファイル名を指定して実行」画面に「cmd」を入力して、OKボタンをクリックする

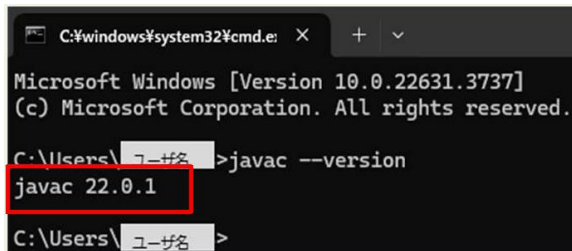


③コマンドプロンプトに「javac --version」を入力してEnterキーを押下する



④「javac 22.0.1」が表示されたことを確認する

※設定をしたバージョンが表示されていることを確認する



## 3.2 Pythonの設定

### 3.2.1 Pythonのダウンロード

- ①こちら(<https://www.python.org/downloads/>)にWebブラウザでアクセスする
- ②Windows installer(64bit)のリンクをクリックする  
※利用環境に適したものをダウンロードする  
※バージョンはPython 3.11.9を利用する

Version	Operating System	Description	MDS Sum	File Size	GPG	Sigstore
Gzipped source tarball	Source release		bfd4d3bfeac4216c35d7a503b02d5c	25.3 MB	SIG	sigstore
XZ compressed source tarball	Source release		22ea467e7d915477152e9d5da156ddc	19.2 MB	SIG	sigstore
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	fa29f4561eb4b5c4f52456a8b8ba347b	42.8 MB	SIG	sigstore
<b>Windows installer (64-bit)</b>	Windows	Recommended	e9dc0502e34932eebcf2be056d5cbcd	25.0 MB	SIG	sigstore
Windows installer (ARM64)	Windows	Experimental	32bd93f71cb078965e4fa2eb2663fa1	24.3 MB	SIG	sigstore
Windows embeddable package (64-bit)	Windows		6d9aa08531d48cc261ba667e2df17c4	10.7 MB	SIG	sigstore
Windows embeddable package (32-bit)	Windows		31e7648158376e92a4463aa022a78e1	9.6 MB	SIG	sigstore
Windows embeddable package (ARM64)	Windows		0611b6aa35483ab1c61d45e0d9f2de0d	10.0 MB	SIG	sigstore
Windows installer (32-bit)	Windows		2a5d1ac2d8a0aa847515f9dd121ccbb7	23.8 MB	SIG	sigstore

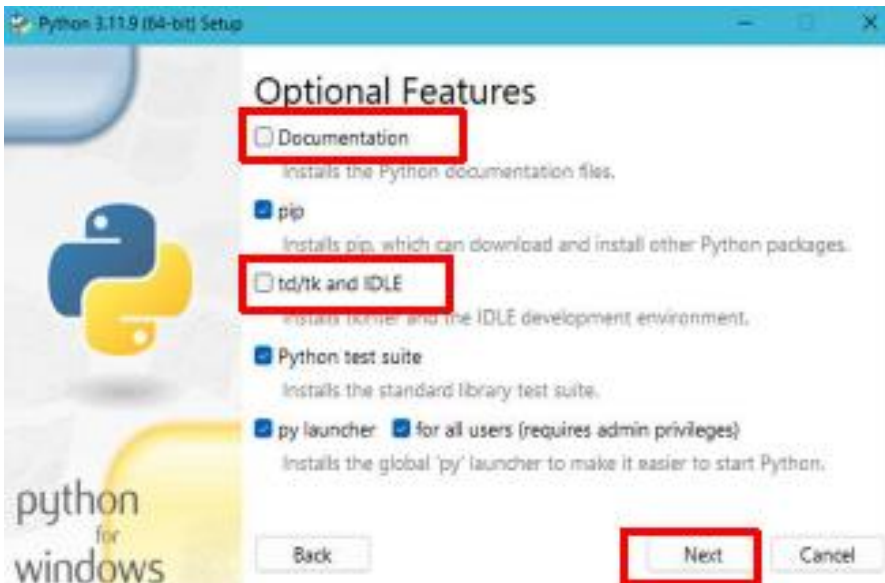
- ③ZIPファイルがダウンロードできたことを確認する

### 3.2.2 Pythonの初期設定

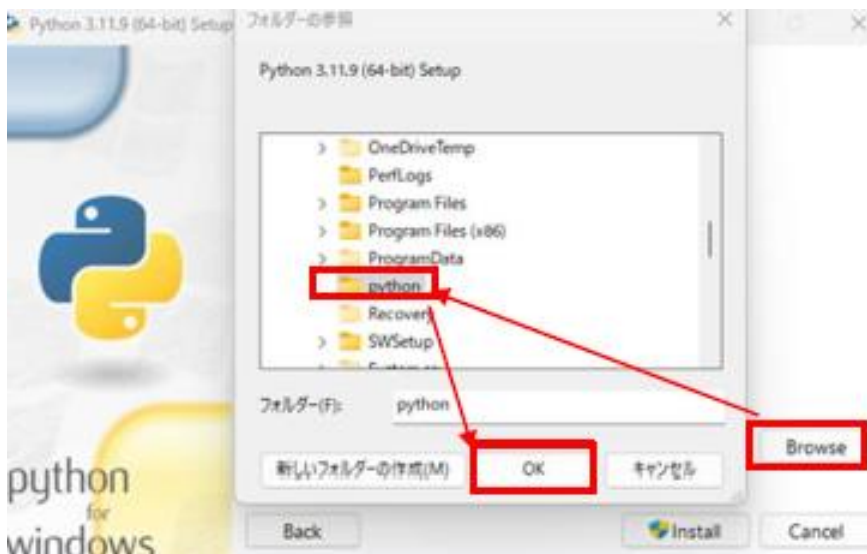
- ①ダウンロードしたpythonのインストーラをダブルクリックする
- ②pythonの環境変数設定を省略するために、「Add python.exe to PATH」にチェックを入れてから、「Customize installation」をクリックする



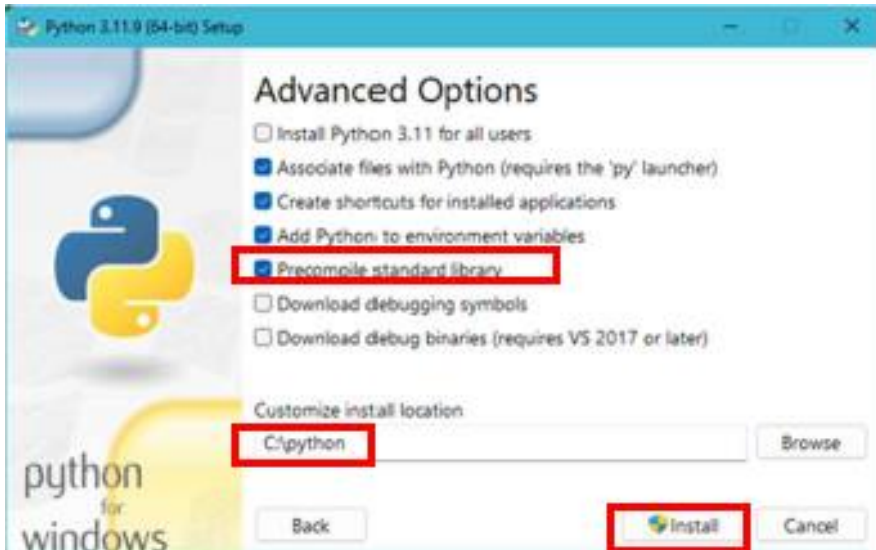
- ③「Optional Features」画面が表示されたら、「Documentation」と「td/tk and IDLE」のチェックを外してから、「Next」ボタンをクリックする



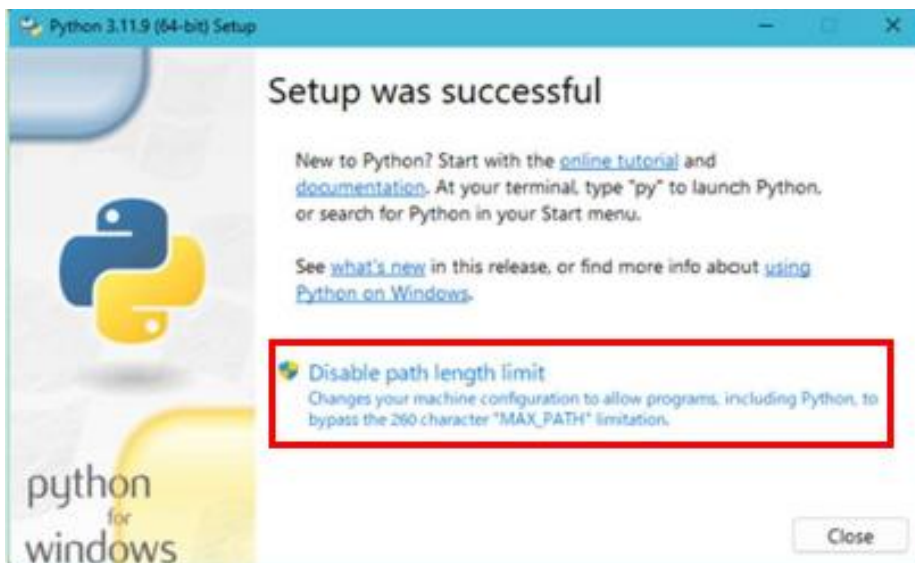
- ④「Advanced Options」画面が表示されたら、Browseボタンをクリックして、任意の場所にpythonフォルダを作成後、OKボタンをクリックする  
※新しいフォルダーの作成ボタンでフォルダを作成できる



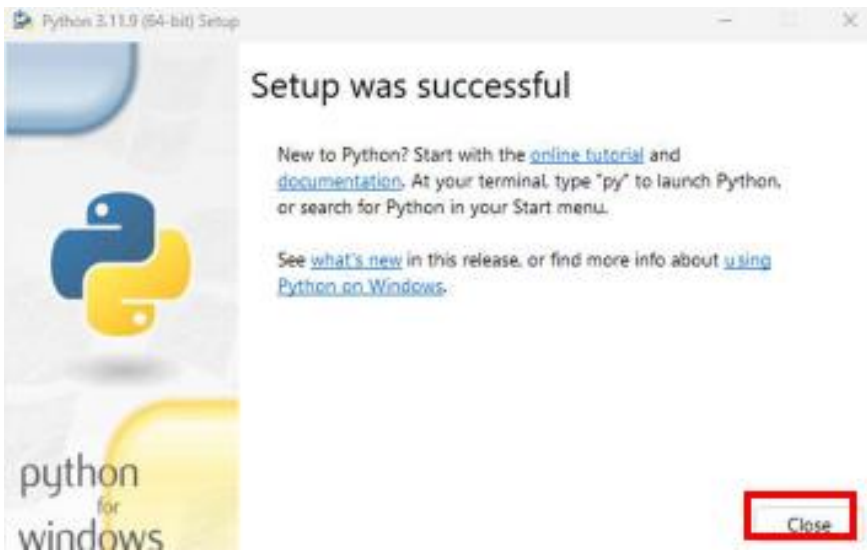
- ⑤「Precompile standard library」にチェックを入れ、「Customize install location」が「④」で作成したpythonフォルダになっていることを確認後、Install ボタンをクリック



- ⑥「Setup was successful」画面が表示されたら、「Disable path length limit」をクリック  
※ユーザアクセス制御画面が表示されたら「はい」をクリック

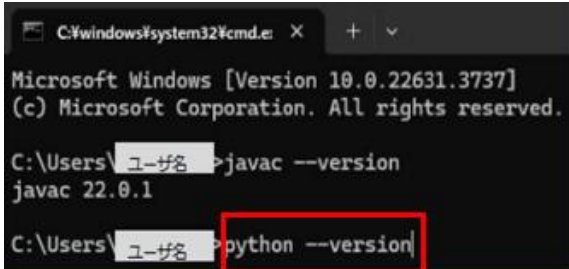


⑦右図の画面に変わったら、Closeボタンをクリック



### 3.2.3 Pythonの動作確認

- ①コマンドプロンプトに「python --version」を入力してEnterキーを押下する  
※コマンドプロンプトの表示方法は、「3.1.3 OpenJDKの動作確認の1、2」を参照してください

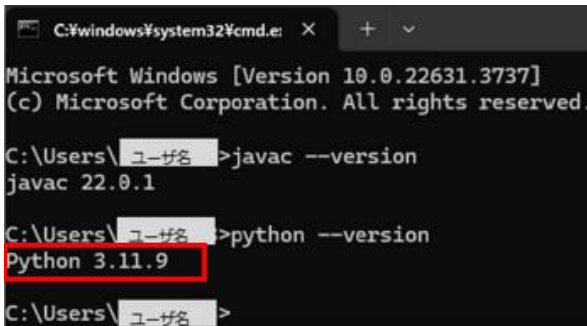


```
C:\Windows\system32\cmd.exe X + v
Microsoft Windows [Version 10.0.22631.3737]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ユーザー>javac --version
javac 22.0.1

C:\Users\ユーザー>python --version
```

- ②「Python 3.11.9」が表示されることを確認する  
※設定したバージョンが表示されていることを確認する



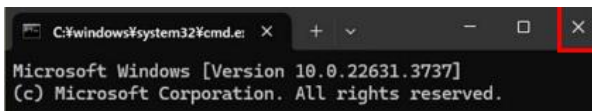
```
C:\Windows\system32\cmd.exe X + v
Microsoft Windows [Version 10.0.22631.3737]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ユーザー>javac --version
javac 22.0.1

C:\Users\ユーザー>python --version
Python 3.11.9

C:\Users\ユーザー>
```

- ③コマンドプロンプトの右上の×ボタンをクリックして、コマンドプロンプトを閉じる



```
C:\Windows\system32\cmd.exe X + v - □ X
Microsoft Windows [Version 10.0.22631.3737]
(c) Microsoft Corporation. All rights reserved.
```

## 3.3 Apache Nifiの設定

### 3.3.1 Apache Nifiのダウンロード

①こちら(<https://nifi.apache.org/download/>)にWebブラウザでアクセスする

②Apache NiFiのBinaries欄にあるStandardアイコンをクリックする

※バージョンは2.0.0を利用する

※バージョンが存在しない場合は、以下URLを使用する

URL:

<https://www.apache.org/dyn/closer.lua?path=/nifi/2.0.0/nifi-2.0.0-bin.zip>

(このバージョンでないと、うまく動作しない可能性があるため、必ずこのバージョンをダウンロードする)

Documentation Development Community Projects Apache

# Download

Previous releases are available in release archives. Archive downloads are subject to rate limiting.

OpenPGP Project Keys can be used for download verification.

NIFI MINIFI REGISTRY FDS

## Apache NiFi

NiFi 2.0.0-M4 [Release Notes](#)

- Released: 2024-07-01
- [Migration Guidance](#)

Sources

Source 2.0.0-M4 OpenPGP SHA-256 SHA-512

Binaries

NiFi Standard 2.0.0-M4 OpenPGP SHA-256 SHA-512

NiFi Stateless 2.0.0-M4 OpenPGP SHA-256 SHA-512

NiFi Toolkit 2.0.0-M4 OpenPGP SHA-256 SHA-512

③右図の赤枠のリンクをクリックして、ダウンロードが開始されることを確認する



④ZIPファイルがダウンロードされたことを確認する

⑤コマンドプロンプトを起動して、下記コマンドを入力をEnterキーを押下する  
<コマンド>

例)certUtil -hashfile “ダウンロード先“¥“ダウンロードファイル名(拡張子あり)“ SHA512

```
C:\NiFi>certUtil -hashfile C:\NiFi\nifi-2.0.0-M4-bin.zip SHA256
SHA256 ハッシュ (対象 C:\NiFi\nifi-2.0.0-M4-bin.zip):
fe9b07e2148a2468da4c86ec406308c7965c79b32c1076fcald2c83b17caf04b
CertUtil: -hashfile コマンドは正常に完了しました。

C:\NiFi>
```

⑥右図の画面に変わったら、Closeボタンをクリック



⑦コマンドプロンプトに表示されているコードと、ブラウザ上に表示されたコードが一致していることを確認する

※ファイルが正しくダウンロードされていることを確認できる

```
C:\NiFi>certUtil -hashfile C:\NiFi\nifi-2.0.0-M4-bin.zip SHA256
SHA256 ハッシュ (対象 C:\NiFi\nifi-2.0.0-M4-bin.zip):
fe9b07e2148a2468da4c86ec406308c7965c79b32c1076fca1d2c83b17caf04b
CertUtil: -hashfile コマンドは正常に完了しました。

C:\NiFi>
```

```
fe9b07e2148a2468da4c86ec406308c7965c79b32c1076fca1d2c83b17caf04b
```

## 3.3.2 Apache NiFiのフォルダ構成

フォルダ階層	各フォルダの概要
¥NIFI-2.0.0 └─ assets └─ bin └─ conf └─ content_repository └─ database_repository └─ docs └─ extensions └─ flowfile_repository └─ lib └─ logs └─ nar_repository └─ provenance_repository └─ python └─ run └─ state └─ work	[assets] NiFiが使用するリソースファイルが格納される場所 [bin] NiFiの起動やステータス確認用のバッチファイルなどが保存されている場所 [conf] NiFiのコンフィグファイルなどが保存されている場所 [content_repository] システム内の全てのFlowFilesのコンテンツを保持する場所 [database_repository] H2データベースリポジトリの保存場所で、フロー設定履歴を追跡する内部データベースの設定の定義ファイルがある [docs] 各ドキュメントが保存されている場所 [extensions] JAVA製カスタムプロセッサを保存する場所 [flowfile_repository] 現在システムに存在する各FlowFileのメタデータの「先行書き込みログ」が保存される場所 [lib] narライブラリが保存される場所 [logs] 標準ログファイルが保存される場所 [nar_repository] narファイルが格納される場所 [provenance_repository] 各FlowFileの履歴が格納される場所 [python] Python製のカスタムプロセッサを保存する場所 [run] NiFiのプロセスIDやステータスを保存する場所 [state] コンポーネントの状態を保存する場所 [work] Python製のカスタムプロセッサの実行環境のこと データ整備ツールの機能群として使用されている

## 3.3.3 ApacheNiFiのセットアップ

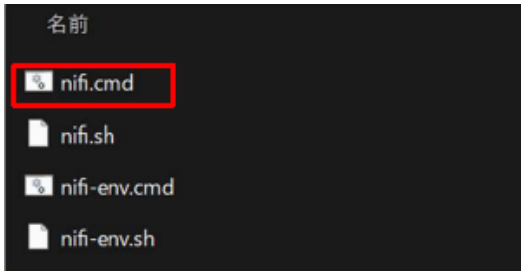
- ①ダウンロードしたZIPファイルを任意の場所に解凍する
- ②「(解凍先)¥nifi-2.0.0¥conf¥nifi.properties」をダブルクリックして開封する
- ③45行目辺りに以下の記述を修正して保存する  
 修正前)nifi.python.command=python3  
 修正後)nifi.python.command=python  
 ※「=」の右側は、「3.2 Pythonのインストールの4)」で指定したフォルダ内にあるpython.exeを指定する

※Pythonカスタムプロセッサを有効にするために、この行のコメントを外してPythonを呼び出すために使用するコマンドに設定する

```
#####
# Python Extensions #
#####
# Uncomment in order to enable Python Extensions.
nifi.python.command=python
nifi.python.framework.source.directory=./python/framework
```

### 3.3.4 Apache NiFi の起動方法

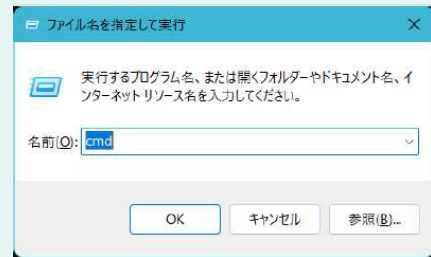
- ①ダウンロードしたZIPファイルを任意の場所に解凍する
- ②「(解凍先)¥nifi-2.0.0¥bin」にある「nifi.cmd」を起動する  
ただし、コマンドプロンプトによる起動が必要。起動方法を2種類記載する



#### 手法①

1. Windowsキー+Rキーを押下する

2. 「ファイル名を指定して実行」画面の名前欄に、「cmd」を入力してOKボタンをクリックする



3. コマンドプロンプトに下記コマンドを入力して、Enterキーを押下する  
コマンド:cd (解凍先)¥nifi-2.0.0¥bin  
例:cd C:¥nifi-2.0.0-bin¥nifi-2.0.0¥bin

入力コマンド

```
cd C:\nifi-2.0.0-bin\nifi-2.0.0\bin
```

コマンド実行後、binフォルダに移動したことを確認する

実行後コマンド

```
C:\nifi-2.0.0-bin\nifi-2.0.0\bin>
```

4. コマンドプロンプトに下記コマンドを入力して、Enterキーを押下する  
コマンド:nifi.cmd start

すると、以下のようにコマンドプロンプトにメッセージが表示される

```
これで起動していることが確認できる  
<メッセージ>  
INFO [main]  
org.apache.nifi.bootstrap.Command  
Application  
Process [xxxx] started  
※xxxxは起動の都度、変化する  
INFO [main]  
org.apache.nifi.bootstrap.Command  
Bootstrap  
Process Running
```

### 3.3.4 Apache NiFi の起動方法(続き)

#### 手法②

1. テキストに以下コマンドを入力して、保存名前はrun.batなどとする <コマンド> set "filePath=%~dp0\nifi.cmd" start cmd /k "%filePath% start"	-
2. 1. で保存したファイルを「nifi.cmd」と同様のフォルダ内に格納し、実行する	
3. コマンドプロンプトに以下のようなメッセージが表示されていることを確認する <メッセージ> INFO [main] org.apache.nifi.bootstrap.Command Application Process [xxxx] started ※xxxxは起動の都度、変化する INFO [main] org.apache.nifi.bootstrap.Command Bootstrap Process Running	-

### 3.3.5 Apache NiFi の停止方法

- ①起動したコマンドプロンプトをアクティブにして、Ctrl+Cキーを押下する
- ②「バッチジョブを終了しますか(Y/N)?」のメッセージが表示されたら、「Y」を入力してEnterキーを押下する

※すぐに表示されない場合は、表示されるまで待つ

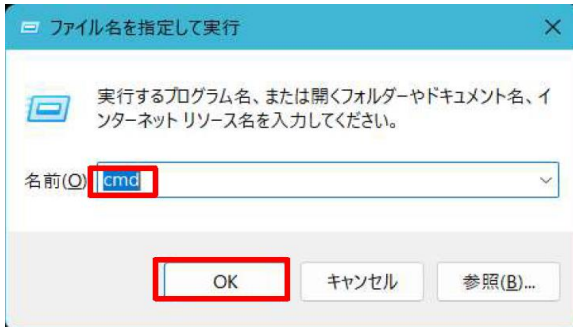
※停止に時間がかかった場合に「NiFi has not finished shutting down after 20 seconds. Killing process.」とメッセージが表示される場合があるが、「Y」を入力してEnterキーを押下で問題ない

```
2025-02-03 15:07:37,550 INFO [main] org.apache.nifi.bootstrap.Command
2025-02-03 15:07:37,553 INFO [main] org.apache.nifi.bootstrap.Command
バッチ ジョブを終了しますか (Y/N)?
```

## 3.3.6 Apache NiFi のユーザID/パスワード変更

①Windowsキー+Rキーを押下する

②「ファイル名を指定して実行」画面の名前欄に、「cmd」を入力してOKボタンをクリックする



③コマンドプロンプトに下記コマンドを入力して、Enterキーを押下する

コマンド:cd (解凍先)¥nifi-2.0.0¥bin

例:cd C:¥NiFi¥nifi-2.0.0¥bin

コマンド実行後、binフォルダに移動したことを確認する

その後、コマンドプロンプトに下記コマンドを入力して、Enterキーを押下する

コマンド:dir

その際に、「nifi-env.cmd」と「nifi.cmd」が存在することを確認する

```
C:\nifi-2.0.0-bin\nifi-2.0.0\bin>dir
ドライブ C のボリューム ラベルがありません。
ボリューム シリアル番号は 7A2D-5636 です

C:\nifi-2.0.0-bin\nifi-2.0.0\bin のディレクトリ

2025/02/03  14:18    <DIR>          .
2025/02/03  15:07    <DIR>          ..
2024/11/08  12:00           1,206 nifi-env.cmd
2024/11/08  12:00           2,231 nifi-env.sh
2024/11/08  12:00           2,476 nifi.cmd
2024/11/08  12:00          11,592 nifi.sh
2025/02/03  15:07             66 run.bat
               5 個のファイル              17,571 バイト
               2 個のディレクトリ  393,419,657,216 バイトの空き領域

C:\nifi-2.0.0-bin\nifi-2.0.0\bin>
```

④コマンドプロンプトに下記コマンドを入力して、Enterキーを押下

※パスワードは12文字以上必要

パスワードのみ変更する場合は、newUsernameに既存のユーザ名を指定する

※Apache NiFiは起動していなくても本作業は可能

<コマンド>

nifi.cmd set-single-user-credentials newUsername newPassword

<例>

nifi.cmd set-single-user-credentials tanakataro 1234testname

⑤以下のメッセージが表示されれば、ユーザ名とパスワードの変更が完了  
次回のApacheNiFiログイン時から、設定したユーザ名とパスワードでログインする  
<メッセージ>

Login Identity Providers Processed

### 3.3.7 Apache NiFi の文字コードの設定方法

①Apache NiFiをインストール後、「ProcessorInspection.py」に2点の変更を行う

※作業する際は、ApacheNiFiを停止後に作業を行う

フォルダ:(解凍先)¥nifi-2.0.0¥python¥framework

②赤字で記載している内容に修正する(get\_module\_string\_constants)

【ソースコード】

def

get\_module\_string\_constants(module\_file: str) -> dict:

with open(module\_file, encoding="utf-8") as file:

・変更前

```
def get_module_string_constants(module_file: str) -> dict:
    with open(module_file) as file:
```

・変更後

```
def get_module_string_constants(module_file: str) -> dict:
    with open(module_file, encoding="utf-8") as file:
```

③赤字で記載している内容に修正する(get\_processor\_class\_nodes)

【ソースコード】

def get\_processor\_class\_nodes(module\_file: str) -> list:

with open(module\_file, encoding="utf-8") as file:

・変更前

```
def get_processor_class_nodes(module_file: str) -> list:
    with open(module_file) as file:
```

・変更後

```
def get_processor_class_nodes(module_file: str) -> list:
    with open(module_file, encoding="utf-8") as file:
```

## 3.3.8 Apache NiFi の必要機能の追加

①Apache NiFiが停止していることを確認する

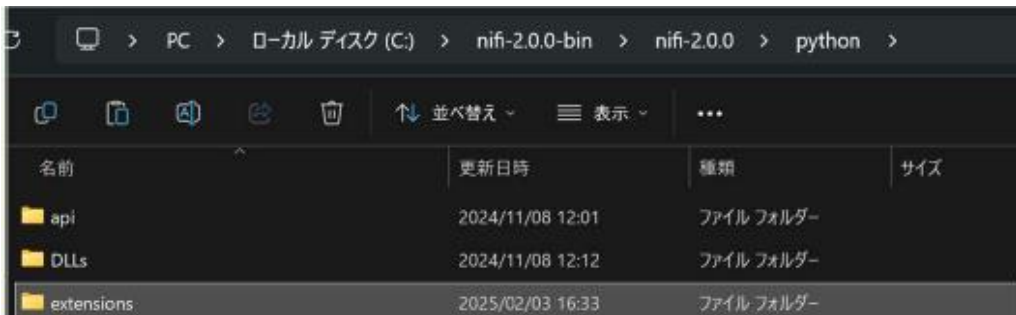
※停止していない場合は、「4.3 停止方法」をご参照の上、停止操作を実施する

②Githubに公開されているフォルダ「api」「extensions」を、右図フォルダに格納する

取得元:Github

取得パス→¥api¥extensions

例:フォルダ:(解凍先)¥nifi-2.0.0¥python



③Apache NiFiでPythonカスタムプロセッサが使用できるか確認する

ApacheNiFiを起動してログインする

※ログイン方法は、Appendixに記載している「データ整備ツールの利用者向けマニュアル」を参照

④Processorアイコンをキャンバス上にドラッグアンドドロップして、「Add Processor」画面のFilter typesに2. で追加したプロセッサのいずれかを入力し、リストに表示されるか確認する



⑤NiFi Flowのキャンバスに、対象のPythonカスタムプロセッサを追加する

※プロセッサ追加の手順は、別紙「データ整備ツールの利用者向けマニュアル」を参照

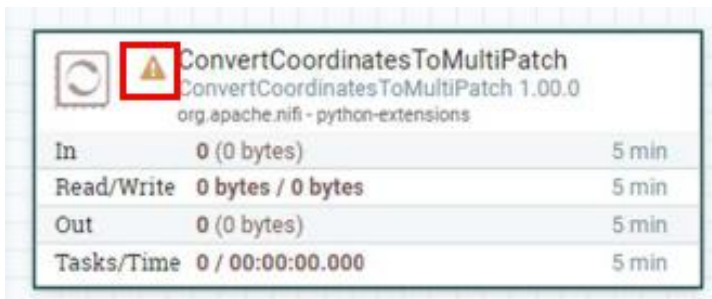
※選択肢に表示されるまで時間を要する場合があります



⑥追加・変更時は、Pythonの仮想環境の作成と依存関係のあるサードパーティのダウンロードが開始される

進捗状況はにカーソルを合わせることで確認可能で、メッセージ①⇒メッセージ②の順に変化し、両メッセージが表示されなくなれば使用可能

※追加・変更にはインターネット接続が必要



ConvertCoordinatesToMultiPatch		
ConvertCoordinatesToMultiPatch 1.00.0		
org.apache.nifi - python-extensions		
In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

メッセージ①:ランタイムの初期化

- 'Processor' is invalid because Initializing runtime environment for the Processor.

メッセージ②:依存関係のあるサードパーティのダウンロード

- 'Processor' is invalid because In the process of downloading third-party dependencies required by the Processor.

## CONTENTS

1. データ整備のオリエンテーション
2. データ設計方針の立て方
3. 各ツールの設定
4. **3Dデータの確認方法**
5. 外径を作成する処理

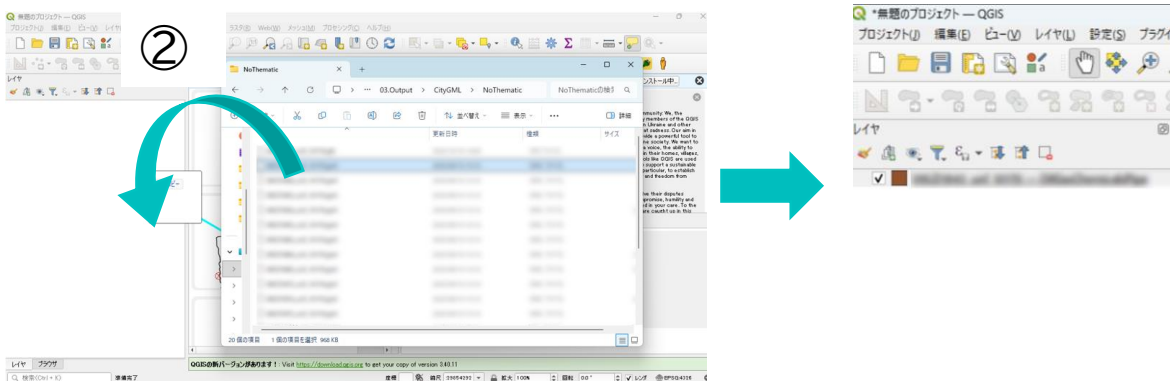


## 4. 3Dデータの確認方法

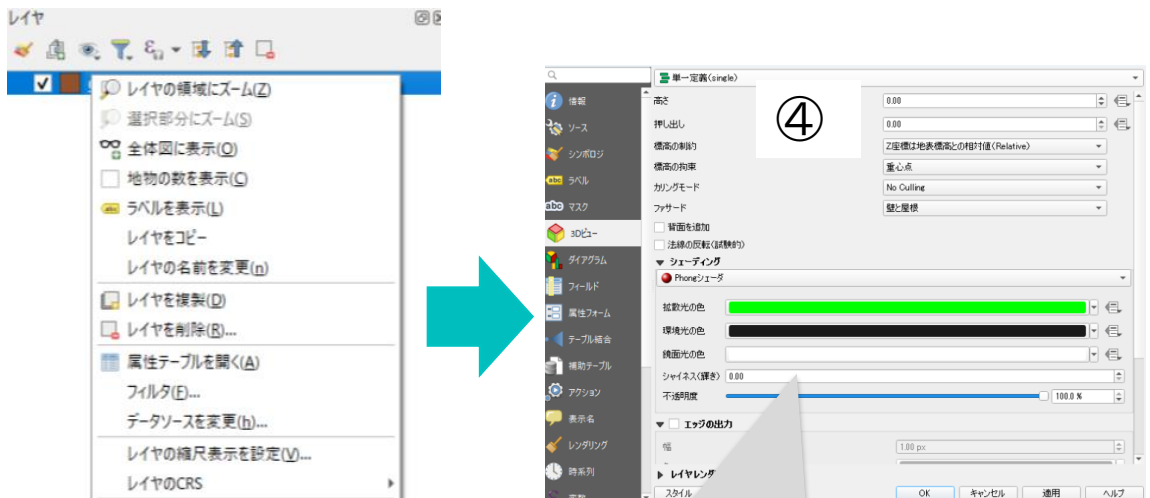
今回確認する方法として使用するソフトウェアはQGISを使用する。

### 4.1 QGISにおけるデータの確認方法

- ① QGISがインストールされている端末にて、QGISを開く
- ② CityGML(Thematicなし)のデータをQGISの画面上にドラッグアンドドロップし、読み込む



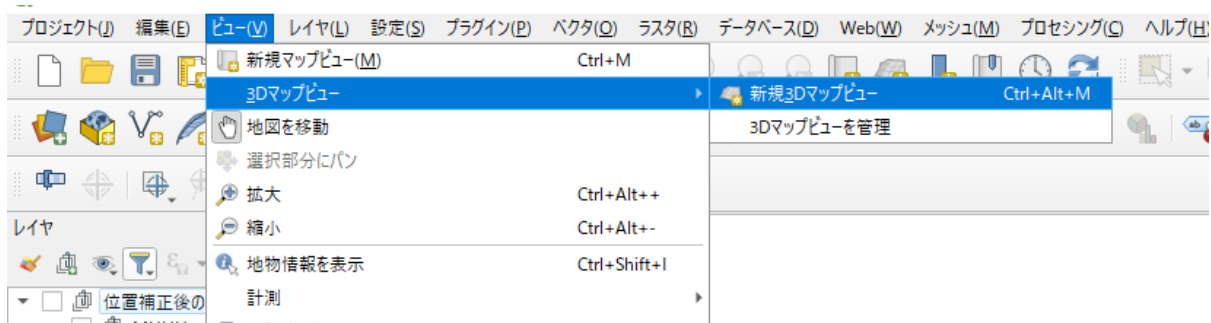
- ③ 読み込んだデータのプロパティを開く
- ④ プロパティの「3Dビュー」タブを開き、「単一定義」に設定する



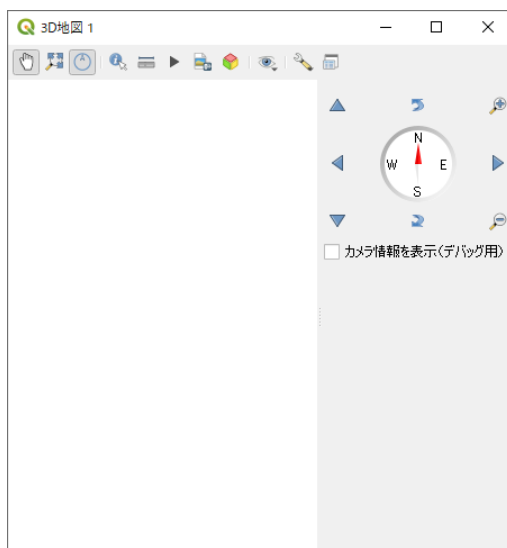
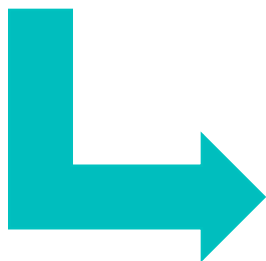
③

「拡散光の色」を見やすい色に設定することを推奨

⑤画面上部のメニューバーから、「ビュー → 3Dマップビュー → 新規3Dマップビュー」をクリックすると、3D表示が開始される。



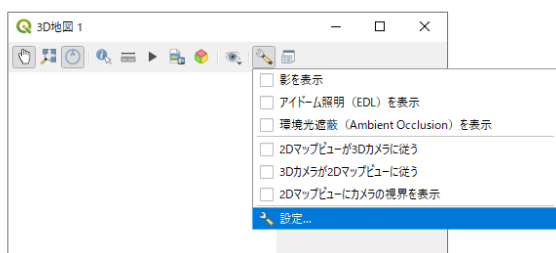
⑤



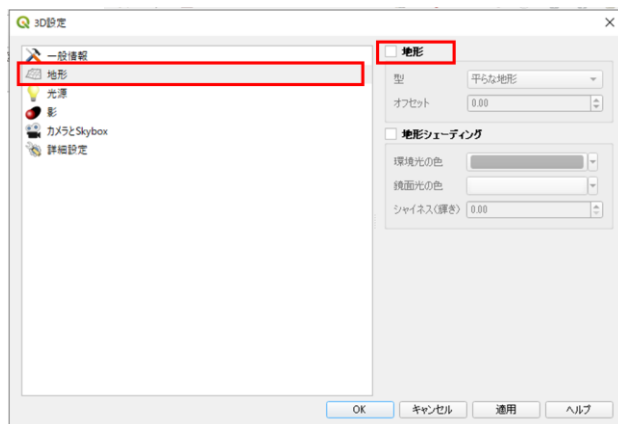
⑥画面上部のメニューバーから、「設定アイコン → 設定」をクリックすると、3D設定画面が表示されます。

⑦3D設定画面の地形メニューから地形のチェックを外してください。

⑥



⑦



## CONTENTS

1. データ整備のオリエンテーション
2. データ設計方針の立て方
3. 各ツールの設定
4. 3Dデータの確認方法
5. 外径を作成する処理



## 5. 外径を作成する処理

外形を計算するためには以下のような想定を行い、実際に計算を行う。

