

# 「OSS の品質管理」に対する市民共創方法： データ可視化プラットフォーム E2D3 の事例紹介

五十嵐 康伸<sup>1</sup> 一円 真治<sup>1</sup> 江畑 彩<sup>1</sup> 大曾根 圭輔<sup>1</sup> 小副川 健<sup>1</sup> 小野 恵子<sup>1</sup> 河野 麻衣子<sup>1</sup>  
佐藤 奈津紀<sup>1</sup> 澤 徳彦<sup>1</sup> 篠原 剛<sup>1</sup> 清水正行<sup>1</sup> 鈴木 典子<sup>1</sup> 竹内 秀行<sup>1</sup> 都築 祐善<sup>1</sup> 長久 武<sup>1</sup>  
西野 貴志<sup>1</sup> 林 美帆<sup>1</sup> 林 由佳<sup>1</sup> 久住 裕司<sup>1</sup> 平河 広輝<sup>1</sup> 槇田 直木<sup>1</sup> 松崎 剛<sup>1</sup> 蓑田 恭秀<sup>1</sup>  
宮内 元<sup>1</sup> 村上 康雄<sup>1</sup> 村瀬 真琴<sup>1</sup> 八木 啓<sup>1</sup> 山田 祐資<sup>1</sup> 山本 智<sup>1</sup> 山本 優<sup>1</sup> 綿貫 順一<sup>1</sup>

Yasunobu Igarashi<sup>1</sup> Shinji Ichien<sup>1</sup> Aya Ebata<sup>1</sup> Keisuke Osone<sup>1</sup> Takeshi Osoekawa<sup>1</sup> Keiko Ono<sup>1</sup>  
Maiko Kawano<sup>1</sup> Natsuki Sato<sup>1</sup> Norihiko Sawa<sup>1</sup> Takeshi Shinohara<sup>1</sup> Masayuki Shimizu<sup>1</sup>  
Noriko Suzuki<sup>1</sup> Hideyuki Takeuchi<sup>1</sup> Yuzen Tsuzuki<sup>1</sup> Takeshi Nagahisa<sup>1</sup> Takashi Nishino<sup>1</sup>  
Miho Hayashi<sup>1</sup> Yuka Hayashi<sup>1</sup> Hiroshi Hisazumi<sup>1</sup> Koki Hirakawa<sup>1</sup> Naoki Makita<sup>1</sup>  
Tsuyoshi Matsuzaki<sup>1</sup> Yasuhide Minoda<sup>1</sup> Hajime Miyauchi<sup>1</sup> Yasuo Murakami<sup>1</sup> Makoto Murase<sup>1</sup>  
Kei Yagi<sup>1</sup> Yusuke Yamada<sup>1</sup> Satoshi Yamamoto<sup>1</sup> Yamamoto Yu<sup>1</sup> Junichi Watanuki<sup>1</sup>

<sup>1</sup>E2D3.org

**Abstract:** 本論文では、E2D3(Excel to D3.js)というオープンソースソフトウェアのデータ可視化プラットフォーム上にあるデータ可視化テンプレート 64 種類に対して、市民 22 人が、9 日間かけて行った品質管理の共創過程を報告する。総テスト項目は 1216 項目で、テスト消化率は 45.1% であった。

## 1. はじめに

Science 2.0・Citizen Science・Open Academic Society・ニコニコ学会β・Civic Tech 等、研究者と市民、もしくは異なる組織に所属する市民が共創して研究する過程は、新しい研究手法として注目されている[1][2]。また、市民同士の共創の結果として生み出された研究成果がどのように更新されているかについても注目されている。本論文の著者である五十嵐は、ハッカソンで創出されたアイデア・プロトタイプを元に、企業がスマートフォンのアプリケーションとしてリリースする過程について研究した[3]。また五十嵐は、ハッカソンで市民が創出したデータ可視化テンプレートを素材にして、その後のハッカソンにおいて、先とは別の市民が新たなデータ可視化テンプレートを創出する過程についても研究した[4]。本論文では、E2D3(Excel to D3.js)というオープンソースソフトウェア(OSS)のデータ可視化プラットフォーム上にあるデータ可視化テンプレート群に対して、市民が品質管理を共創する過程について報告する[5, 6]。

OSS の品質管理は難しい。E2D3 を開発している E2D3.org(組織名称)は会社や NPO ではなく任意団体であり、大人の部活動的組織である。そのため、次に列挙する組織的な難しさも存在する。①E2D3 メンバー同士には会ったことがない人の対が存在する。②組織から各メンバーに給与は支払われない。各メンバーは自発的動機により、ボランティアとして参加している。③日本国内では北海道から九州まで、また海外も含めて様々な場所に住んでいるためオンライン上のコミュニケーションがメインになる。さらに今回報告する E2D3 の品質管理には、期間的な難しさも存在した。2016 年 4 月 4 日に行われる「統計データ利活用アプリケーション・アイデアコ

ンテスト:STAT DASH グランプリ 2016」の授賞式において E2D3 が総務大臣賞を受賞することが、2016 年 3 月 28 日に告知された[7]。そこで、告知から受賞の翌日までの 9 日間の間で品質管理のプロジェクトを立ち上げて、終了することを目的としたため、プロジェクトの期間が短かった。

OSS のコミッター(開発する人)とテスター(テストする人)が同一人物でありかつ一人の場合、自分が作成した設計書に基づいて、暗黙知も含めてテストを行うことが比較的容易である。しかしコミッターが複数人いる大きな OSS の場合、複数人で共同してテストを行うのが通常である。その場合、テスターによって、テストの手順の解釈が異なる問題が発生する場合がある。この問題に対応するため、誰が読んでも同じ手順を行える、ローコンテキストなテスト仕様書を作成する必要がある。また、テスターの力量がばらつく問題が発生する場合もある。さらにテスターによって、テスト結果の判定が異なる問題が発生する場合もある。これら問題に対応するため、テストの結果を集計したら、テスト責任者が結果の妥当性を統一した基準で判定し直す必要がある。本論文では、E2D3 に対して行ったテストとデバッグの設計方法・実施方法について報告する。

## 2. 材料と方法

### 2-1 材料

E2D3 は表計算ソフト Excel のシート情報に対して、JavaScript で書かれた可視化ライブラリ「D3.js」の機能を適用することで、ダイナミックかつインタラクティブなグラフテンプレートを現時点で約 100 種類用意している[8]。このグラフテンプレートを用いることで、プログラマでな

いと作成が難しい「D3.js」を用いたデータ可視化(グラフ)」を、表計算ソフト Excel の操作だけで簡単に実現できることが特徴である。Excel に E2D3 をインストールして作成したグラフとして図 1 を例示する。図1では、① A 列～D 列のデータを選択し、② データ範囲指定することで、③プログラムコードを書かずにデータをダイナミックかつインタラクティブに可視化できる。E2D3 で可視化できるグラフはデモ画面である <http://a.e2d3.org/> から簡単に見ることができる。

可視化テンプレートのカテゴリは選択できる。カテゴリごとに、数種類から 30 種類の可視化テンプレートがある(図2)。可視化テンプレートのサムネイル画像上のボタンを押下すると、該当の可視化テンプレートが Excel 上にグラフとして展開される。

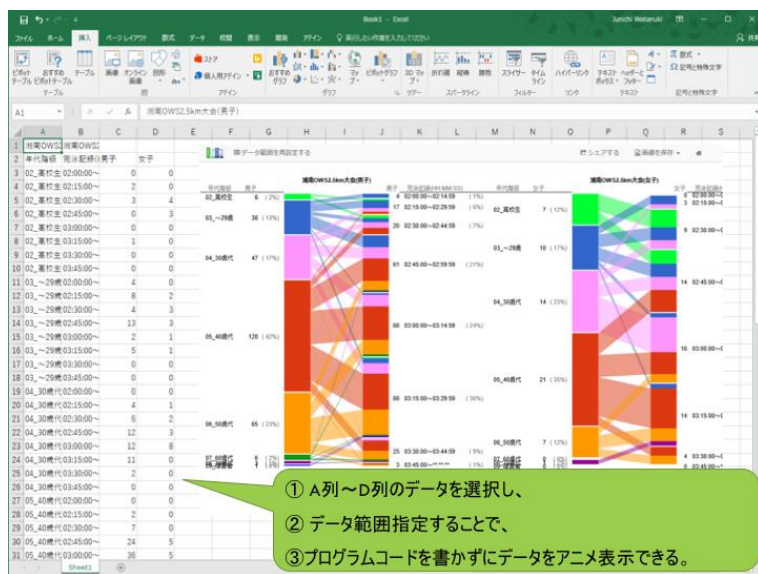


図1: 可視化テンプレートの一例

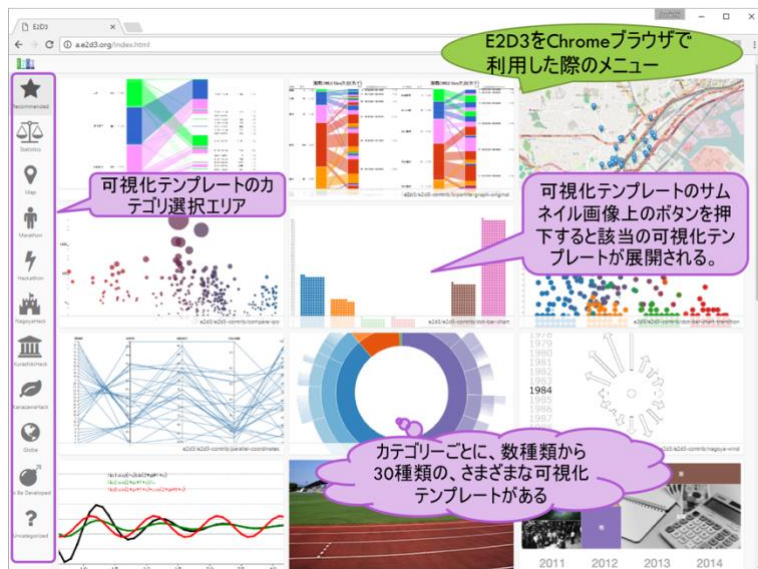


図2: 可視化テンプレートのカテゴリ

各グラフには可視化結果を、SNS・メールなどでシェアするためのボタンや、グラフを画像ファイルで保存するボタン、E2D3のトップメニューに戻るボタンが存在する(図3)。

## 2-2 品質管理方法の全体像

品質管理活動の全体像を図4にまとめた。不具合内容を記載した Issue に応じてコミッターがデバッグし、Github 上のマスターコードに Pull Request (Pull Req) を投稿した。マージャーが投稿されたコードを評価し、問題がなければマスターコードに Pull Req を Merge した。例外的に Issue 発行を経ずに、テスト責任者がテスト結果に基づき、コミッターへデバッグを依頼することもあった。テスト責任者が、テスト期間前に発行されていた Issue に基づき、コミッターへデバッグを依頼することもあった。どのテスト項目にどのテスターにアサインするかには優先順位を付けず、先入れ先出し (First-In First-Out) の方法でアサインした。

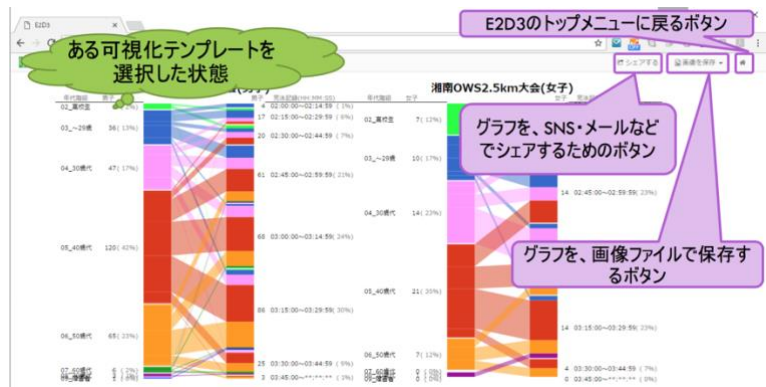


図3: グラフ上に存在するボタン

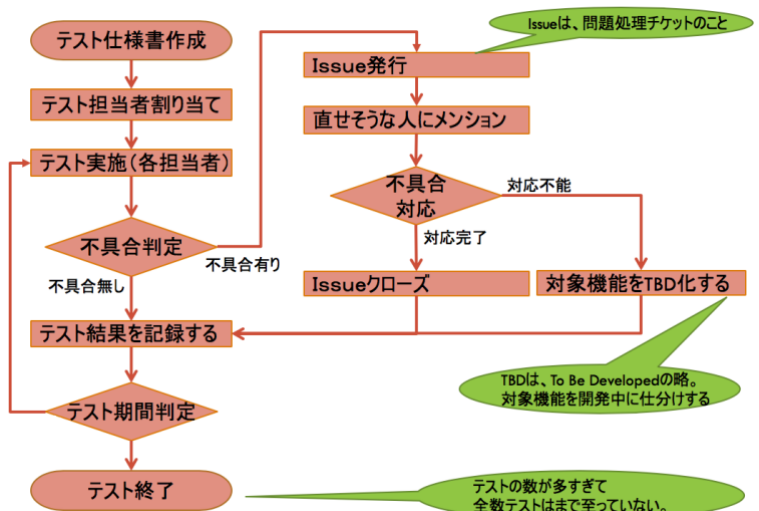


図4: 品質管理方法の全体像

2-3 Issue の発行方法、直せそうな人を探す方法、メンションする方法、修正状況の確認方法

発生された不具合に対して、GitHub 上で Issue を発行した(図5)。Issue 発行後のやりとりの本文中に、半角の@に続けて Github のアカウント名を記入することで、対象ユーザーに通知を届ける仕組みがある。

発行した Issue に対して、直せそうな人へメンションを送った(図6)。例えば、tags.yml というファイルを修正したいと思ったら、<https://github.com/e2d3/e2d3-contrib/blob/master/tags.yml> を開いた状態において、

- ① 対象ファイルのページに移動した
- ② 最後に修正した人にメンションした
- ③ 連絡つかない場合は、過去のコミッターの誰かにお願いした
- ④ 修正箇所が特定できる場合は Blame ページで該当者を探した
- ⑤ 不具合発生の時期が分かっている場合は、History ページから該当者を探した

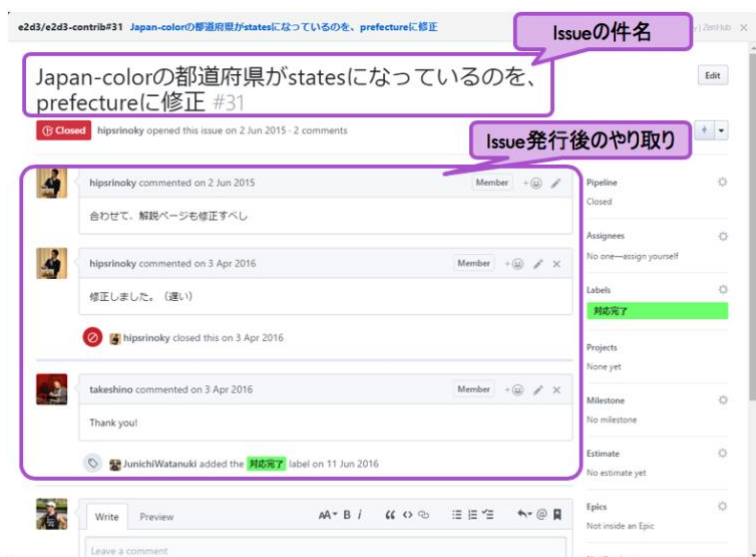


図5: Issue 発行の例

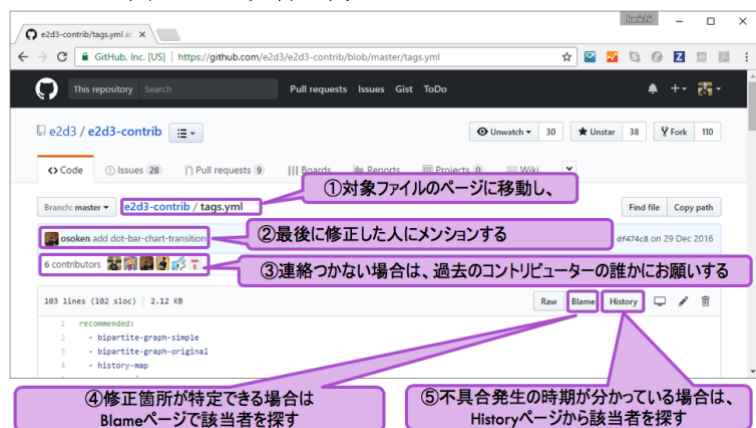


図6: 直せそうな人へのメンション方法

直せそうな人を探す方法 1 として、GitHub の Blame ページを用いた方法を図 7 にまとめた。  
<https://github.com/e2d3/e2d3-contrib/blame/master/tags.yml> を開いた状態において、

- ① 該当ファイルのソースコードを確認した
- ② 修正したい箇所を確認した
- ③ 修正箇所を投稿した人を特定した
- ④ 修正箇所を頼りに、修正対象の事情を知っている人にメンションした

直せそうな人を探す方法 2 として、GitHub の History ページを用いた方法を図 8 にまとめた。  
<https://github.com/e2d3/e2d3-contrib/commits/master/tags.yml> を開いた状態において、

- ① 修正を行った日付を確認した
- ② このタイミングで修正された内容を確認した (修正内容が差分表示で確認できる)
- ③ 修正を行った日付を確認した
- ④ 履歴情報を頼りに、修正対象の事情を知っている人にメンションした

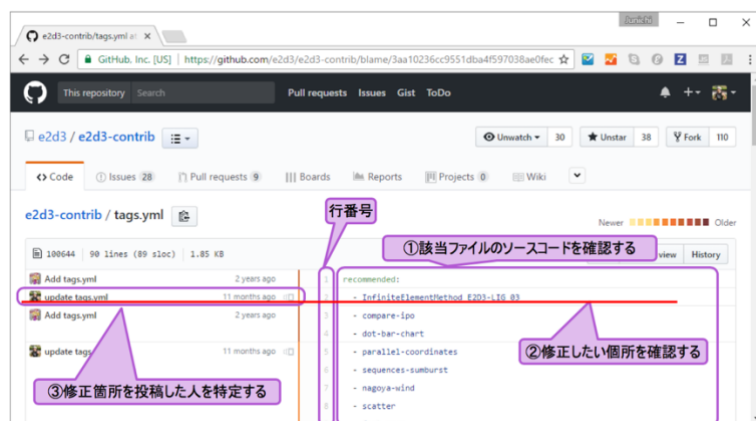


図7: Blame ページを用いて、直せそうな人を探す方法

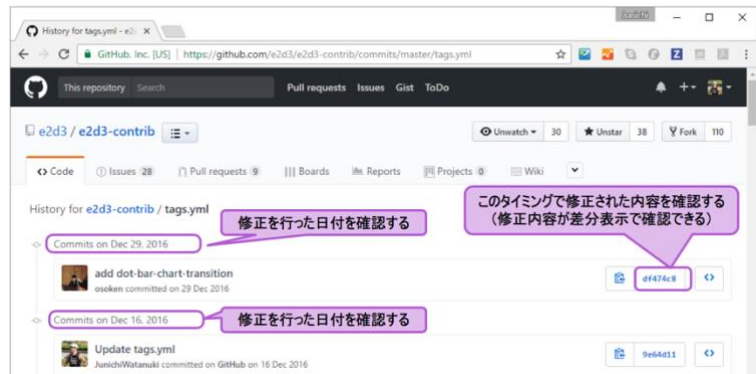


図8: History ページを用いて、直せそうな人を探す方法



修正状況は、GitHub の Network graph ページ上で確認した。図 9 は、<https://github.com/e2d3/e2d3-contrib/network> を開いた状態である。横方向の線の本数は 並行で修正してる人数であり、2016/3/31～2016/4/5 の期間を表示している。

## 2-4 テスト仕様書の作成方法

ソフトウェア品質特性の定義には ISO9126 及び JIS X 0129-1 等がある。本研究では JIS X 0129-1 この定義に沿い、以下の方法でテスト仕様書を作成した。ソフトウェアの主特性はの6つ(機能性・信頼性・使用性・効率性・保守性・移植性)に分けられる。この中から、我々は信頼性と移植性のテストを行った。信頼性の副特性は3つ(成熟性・障害許容性・回復性)に分けられる。この中から、我々は成熟性と回復性のテストを行った。移植性の副特性は4つ(環境適応性・設置容易性・共存性・置換性)に分けられる。この中から、我々は環境適応性のテストを行った。

テストフェーズは3つ(システムテスト・結合テスト・単体テスト)に分けられる。この中から、我々は「システムテスト」を行った。テスト実施者(テスター)の想定は2つ(ホワイトボックステスト・ブラックボックステスト)に分けられる。この中から、我々は「ブラックボックステスト」を行った。テスト技法は以下の2つ(動的テスト・静的テスト)に分けられる。この中から、我々は「動的テスト」を行った。テスト終了条件は「テスト期間の終了」とした。

## 2-5 品質管理の目標

品質管理の目標としては、受賞により期待される新たな利用者が品質面で残念に思わず利用を継続するレベルにまで改善することを目指した。そのため短期間で実施でき、最低限必要なテスト項目を検討した。より具体的には、E2D3 を初めて使う人が自由に操作しても、各種ボタンが機能する、Excel データが可視化される、表示が化けたりしない、フリーズしない、仕様と異なる動作をしないことを目標した。バグの中でデバッグが難しいものは、ワークアラウンド(回避方法)として readme にコメントするなど運用対処した。

# 3. 結果

## 3-1 集められたリソース

期間は2016年3月28日から4月5日の9日間であった。2016/3/28にE2D3.org ver. 0.7 のFB Group内(49人)で品質管理チームを新設を提案した。また過去のハッカソン参加者・勉強会参加者・過去のコミッタなど、つながりのある方々に対して協力を呼びかけた。その結果22人が本品質管理に協力してくれた。22人は以下の役割に分かれた(兼任を含む)。プロジェクトマネージャー:人、テスト責任者兼マージャー1人、テスター:17人、コミッター:15人。金銭報酬は支払わなかったが、仮に稼働率を考慮して、1日1万円の報酬を支払うとすると、 $22[\text{人}] \times 1[\text{万円/人日}] \times 9[\text{日}] = 198[\text{万円}]$ に相当する。協力者募集時のコミュニケーションツールには

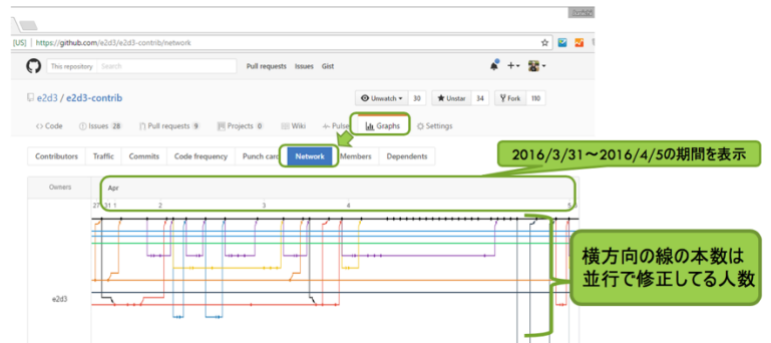


図9: Network graph ページを用いて、修正状況を確認する方法

Facebook、品質管理の実働時のコミュニケーションツールには Github・Slack・Google Drive を用いた。

## 3-2 スケジュール

3/28:テスト仕様書作成を開始した。

3/30:テスト仕様書を完成させた。

3/31:テスト開始した。Issue 発行、デバッグと Pull Req を始めた。

4/4 :テスト終了した。Issue 発行を終えた。授賞式があった。

4/5 :デバッグと Pull Req を終えた。

## 3-3 テスト仕様書の作成手順

はじめに、事前調査で判明した問題点を整理した。

1. (当時あった)64 個の可視化テンプレートの不具合傾向が一様でなかった。
2. ユーザーの利用環境により、不具合の有無が異なった。OS・Excel・ブラウザの version 依存が考えられた。
3. 開発時には考えていなかった操作方法で不具合が出るがあった。

また、以下の点に注意した。

1. テスト仕様書などは、ローコンテキストな表現で記述して、だれが読んでも同じ内容として理解可能となるよう心掛けた。
2. 専門用語は、着手前に説明資料を提供した。具体底には、専門分野外の人にとっては、表側、表頭、表体、頭注などと伝えても混乱を招いてしまうため、総務省統計局の統計表のみかたを提供した[9]。

(下)図11:作成したテスト仕様書

基本機能	動作確認したい機能	No	機能確認のためのE2D3での操作と、可視化エリアの状態
Reset Data Areaボタン 提供機能の調査  (E2D3グラフのデータ更新機能の実装及び動作不具合に関する、調査検証)	表体データの更新機能 (表体の最上行 & 最左列の値を更新)	1	[Reset Data Area]押下無しでも、可視化エリアが自動的に更新される
		2	[Reset Data Area]押下有り の操作の結果、可視化エリアが更新される
	表頭データの更新機能 (表頭の最左列の値を更新)	3	[Reset Data Area]押下無しでも、可視化エリアが自動的に更新される
		4	[Reset Data Area]押下有り の操作の結果、可視化エリアが更新される
	表側データの更新機能 (表側の最上行の値を更新)	5	[Reset Data Area]押下無しでも、可視化エリアが自動的に更新される
		6	[Reset Data Area]押下有り の操作の結果、可視化エリアが更新される
	表体 & 表側データの同時挿入対応 (表体 & 表側部分の最上行に、行データを挿入)	7	[Reset Data Area]押下無しでも、可視化エリアが自動的に更新される
		8	[Reset Data Area]押下有り の操作の結果、可視化エリアが更新される
	表体 & 表側データの同時削除対応 (表体 & 表側の最上行で、行データを削除)	9	[Reset Data Area]押下無しでも、可視化エリアが自動的に更新される
		10	[Reset Data Area]押下有り の操作の結果、可視化エリアが更新される
	表体 & 表頭データの同時挿入対応 (表体 & 表頭の最左列に列データを挿入)	11	[Reset Data Area]押下無しでも、可視化エリアが自動的に更新される
		12	[Reset Data Area]押下有り の操作の結果、可視化エリアが更新される
	表体 & 表頭データの同時削除対応 (表体 & 表頭の最左列で列データを削除)	13	[Reset Data Area]押下無しでも、可視化エリアが自動的に更新される
		14	[Reset Data Area]押下有り の操作の結果、可視化エリアが更新される
グラフ描画エリア機能	データ内容や描画機能が、維持されること	15	グラフ拡大縮小で、値や描画に本質的な変化を誘発しない
ShareChartボタン機能	グラフ共有機能が機能すること	16	Share URL をブラウザにコピー & Enter することで、作成したグラフと同じものが表示できる
Save Imgボタン機能	画像生成機能が機能すること	17	[Save img] → [Save SVG] → DL ファイルをダブルクリックで、作成したグラフと同じものが表示
		18	[Save img] → [Save PNG] → DL ファイルをダブルクリックで、作成したグラフと同じものが表示
Homeボタン機能	メニュー切り替え機能	19	[Home] ボタンを押下することでメインメニュー画面に遷移できること

	F	G	H	I	J	K	L	M	N	
1										
2										
3	E2D3Ver	グラフ識別子	最後のコメント	コメント	現カテゴリー	初回マージ	移行先カテゴリ	グラフ作成者	テスト担当者	修正担当者
4	0.6	virtual-toyamarathon1	thumbnail	オリジナル	marathon	2015/10/24	結貴	一円	一円	
5	0.3	japanmap-javascript	Add manifest.yml	example	2015/1/13 map	竹内	テスト対象外	内田	結貴	
6	0.6	virtual-toyamarathon2	no message	オリジナル	marathon	2015/10/24	結貴	竹内	結貴	
7	0.3	barchart-javascript	Add manifest.yml	example	2015/1/13 basic	竹内	テスト対象外	竹内	結貴	
8	0.3	react-javascript	Remove unnecessary files on t	example	2015/1/19 basic	竹内	テスト対象外	竹内	結貴	
9	0.3	treemap	Fix size of char	もくもく会で	example	2015/3/9	テストしないので、一旦なく	tbdに移す。いつか、recommen	結貴	
10	0.3	sumin-checker	added title line.	hackathon	2015/3/10	osoken	テスト対象外	山本	結貴	
11	0.4	foot-race	再描画時に前リ	オリジナル	hackathon	2015/5/13 recommended	happy2mylife	長久	長久	
12	0.5	dcsj-example	Follow new API	もくもく会で	statistics	2015/8/4	竹内	竹内→長久	竹内	
13	0.5	jhree-example	Add some char	オリジナル、アップロード、使いやすい	example	2015/9/12 recommended	結貴	竹内	結貴	
14	0.5	InfiniteElementMethod_E2D3L	Follow new API	有名	recommended	2015/9/12	recommended	竹内→長久	竹内	
15	0.5	compare-ipo	Update README	オリジナル	hackathon	2015/9/12 recommended	hpsrinoky (山本)	hpsrinoky (山本)	hpsrinoky (山本)	
16	0.5	hipartit-analyzer	Update data.csv	オリジナル	tbd	2015/9/19	結貴	テスト対象外	結貴	
17	0.6	dot-bar-chart	微調整	オリジナル	unclassified	2015/3/9	osoken	osoken	osoken	
18	0.6	multi-series-radar-chart	rename files	オリジナル	Uncategorized	2015/11/1	大曾根	山田	大曾根	
19	0.6	principal-component-analysis	Add principal c	もくもく会で	statistics	2015/11/3	竹内	Abe	結貴	
20	0.6	virtual-tsukubamaraonath	push	オリジナル	marathon	2015/11/8	結貴	山本	結貴	
21	0.3	japan-color	barchart2-javas	オリジナル	map	2015/3/9 recommended	chimerast (山本)	山本	山本	
22	0.6	radial-reingok-tiford-tree	Add tags.yml	オリジナル	example	2015/11/3 recommended	竹内	山田	山本	
23	0.3	sequences-sumburst	Fix size of char	使いやすい	recommended	2015/3/9	chimerast	chimerast→林	山本	
24	0.4	population-map-iran	Remove unnecessary files on t	recommended	2015/3/15	sawa	sawa→長久	sawa	sawa	
25	0.4	multi-series-line-chart	Fix size of charts	example	2015/3/17 basic	test	テスト対象外	山本	山本	
26	0.4	chaind-transitions	fix for Excel	もくもく会で	recommended	2015/3/24	テストしないので、一旦なく	tbdに移す。いつか、recommen	結貴	
27	0.4	tes-ismeries	Fix size of char	もくもく会で	recommended	2015/3/24	テストしないので、一旦なく	tbdに移す。いつか、recommen	結貴	
28	0.4	bubble-chart	Fix size of char	有名、ハンス	hackathon	2015/4/4 recommended	osoken	upsilon2	upsilon2	
29	0.4	bubblechart	scatter plot added	example	2015/4/4 basic	test	テスト対象外	江坂	江坂	
30	0.4	force-graph	add force-grapl	使いやすい	example	2015/4/15 recommended	tetsuyas1	江坂	江坂	



### 3-5 記録されたテスト仕様書

テスト仕様書に従いテストを行い、結果を記録した(図12)。テスト環境を記録した。環境(OS)1:Mac、Windows7、Windows8、Windows 10。環境(利用ソフト)2:Excel2013、Excel2016、Excel365、Excel Online。ブラウザの version 等、テスト環境の詳細点を記録した。

テスト実施結果を記録した。機能が仕様通りに動作したら、「○」を記入した。機能が仕様通りに動作しても気になることがあれば、「○」と共にコメントを記入した。不具合が見つかったら、「△」の記入と共に Issue 番号を記入した。修正対応が完了したら、「▲」もしくは「●」と共に、Merge 時の Pull Req 番号と結果を記録した。不具合が見つかったにも関わらず、修正対応ができなかった場合は、「×」を記入した。

各記号の意味は次の通りである。

- :合格
- △:不具合報告(Issue 対応済み)
- ▲:完璧ではないが修正した
- :最後まで直した
- ×:不合格(そのまま残っているのは、修正未対応)

テストの責任者がテスト実施結果の記録を見直すと、テスト結果の捉え方にテスト者の個人差が大きいところがあった。そこで本論文の報告のために、テスト終了後にテストの責任者が、システムの作りを考慮して、上記の結果を精査しなおして、ラベル付け直した。用いたラベルは次の通りである。

- テスト合格:テスト実施時期の条件で合格したもの。(条件付き合格も含んでいる)
- スコープ外:①可視化テンプレートに無い機能に対するテストであるためテスト対象外。②元のデータ形式特性のため、テストができないため、テストのスコープ外としたもの。
- テスト省略:セルの値変更によりグラフが自動で変更されるグラフは、「ResetDataArea」ボタンのテストを省略できるため、テストを省略したもの。
- 運用対処:Readme.md に利用上の制約事項を記載することで、課題解決したもの。機能を無効化することで対応したもの。
- 修正未達:デバッグに着手したが、期間内では修正が間に合わなかったもの。(期間後に修正完了したものも含む)
- 修正完了:テスト結果が不合格であったものを、期間内に修正したもの。

テスト環境を記録		テスト環境の詳細を記録		テスト実施結果を記録	
ID	E2D3Ver	グラフ識別子	詳細バージョンの記録	機能仕様通りに動作したら「○」を記入	機能仕様通りに動作しても気になることがあれば「○」に加えてコメントを記入
13	0.4	bubble-chart	M-O	○	Excel Online上で画面指定せずに押下すると、「Cannot read property '5' of undefined」(Chromeの場合)が表示されるため、エラーメッセージに置き換えた方が良いかと、除外
14	0.4	bubblechart	M-O	○	Excel Online上で画面指定せずに押下すると、「Cannot read property '5' of undefined」(Chromeの場合)が表示されるため、エラーメッセージに置き換えた方が良いかと、除外
15	0.4	force-graph	10-2016	○	Excel Online上で画面指定せずに押下すると、「Cannot read property '5' of undefined」(Chromeの場合)が表示されるため、エラーメッセージに置き換えた方が良いかと、除外
16	0.4	under-the-sea	10-2016	○	Excel Online上で画面指定せずに押下すると、「Cannot read property '5' of undefined」(Chromeの場合)が表示されるため、エラーメッセージに置き換えた方が良いかと、除外
17	0.4	wordcloud	Chromebook-0	○	Excel Online上で画面指定せずに押下すると、「Cannot read property '5' of undefined」(Chromeの場合)が表示されるため、エラーメッセージに置き換えた方が良いかと、除外
18	0.4	piechart-javascript	ChromeOS 49.0.2623.111 (64-bit)	○	Excel Online上で画面指定せずに押下すると、「Cannot read property '5' of undefined」(Chromeの場合)が表示されるため、エラーメッセージに置き換えた方が良いかと、除外
19	0.4	curry	10-2016	○	Excel Online上で画面指定せずに押下すると、「Cannot read property '5' of undefined」(Chromeの場合)が表示されるため、エラーメッセージに置き換えた方が良いかと、除外
20	0.4	foot-race	M-O(山本M-O)	○	Excel Online上で画面指定せずに押下すると、「Cannot read property '5' of undefined」(Chromeの場合)が表示されるため、エラーメッセージに置き換えた方が良いかと、除外
21	0.5	dcjs-example	M-O	○	Excel Online上で画面指定せずに押下すると、「Cannot read property '5' of undefined」(Chromeの場合)が表示されるため、エラーメッセージに置き換えた方が良いかと、除外

図12: 記録されたテスト仕様書

### 3-6 テストの結果

64 種類の可視化テンプレートに対してプロジェクトマネージャーとテスト責任者で優先順位を付け、32 種類をテスト対象、32 種類をテスト対象外とした。一つのテンプレートにテスト項目は 19 項目あった。そのため、全テスト項目は  $64 \times 19 = 1216$  項目となった。以下がテスト結果である。

1. 運用対処:20
2. テスト判定不能:1
3. 修正完了:3
4. 修正未達:5
5. スコープ外:87
6. テスト合格:349
7. テスト省略:22
8. 不合格(未対応):62
9. 未実施:663
10. 未実施(テスト方法不明):4

総テスト数:1216 項目 (=64 種類 x 19 項目)

テスト消化率は 549 項目 (実施したテスト項目 = 未実施以外のテスト項目) / 1216 項目 = 45.1% であった。テスト期間中に出された Issue は 4 件、Pull Req で 18 件であった。Closed の Issue は 2 件、Pull Req は 18 件であった。Open (In progress) な Issue は 2 件、Pull Req は 0 件であった。

## 4. 今後の課題

本研究で行えなかった課題を挙げる。以下のテストは、バリエーションが多すぎるため手動ではできないと判断し行わなかった。これらは手動ではなく、テストを自動化して行うべきである。

1. 複数のテストの組み合わせに関して「命令網羅表」「分岐網羅表」を作りテストする
2. テスト環境に関して「直交表」を作り網羅的にテストする
3. テスト環境において、OS の言語設定 2 種類: 日本語・英語を分ける

E2D3 はプラットフォームとしての本体と、その上でモジュール型開発されているグラフ可視化テンプレートに対して品質管理を行った。Cytoscape 等、E2D3 と同様のモジュール型開発の OSS の品質管理方法を調べることで有益な知識が発見できる可能性がある[10]。各グラフテンプレートで使われているライブラリーの version に関する品質管理は特に必要である。また、E2D3 のプラットフォーム部分の品質管理も今後の課題である。

また E2D3 は Waterfall 型ではなく、Ajail 型で開発されている。Ajail 型の品質管理手法体系と本論文の内容を比較する必要もある。さらに E2D3 はリリース版が常に更新される CI/CD (継続的インテグレーション・継続的デリバリー) 型である。そのため、システム試験終了後に行う「出荷判定会議」は行なっていない。CI/CD における品質管理を高めるツールとして Circle CI・Github Actions を取り入れたい。コミッターから送られてきた Pull Req に対して node.js で毎回自動テストすることも可能である。その場合、コミッターが設定したコンテナに、E2D3 の開発環境を自動的に作って、その環境でテストシナリオを実行して、自動判定して、結果が OK なら Merge する、もしくはコミッターにテスト判定を自動通知する方法が考えられる。さらに、Git flow、GitHub flow で Git の branch 管理にルール付けすることも有効である。

## 参考文献

- [1] 堀田竜士ら: 研究者と市民の共創を生み出す研究会の提案, 人工知能学会論文誌 34 巻, 4 号, D-I92\_1-8, (2019)
- [2] 白松俊ら: 市民共創知研究会(CCI): 地域課題に立ち向かう知を AI 研究者と市民が共創する場, 人工知能, 34 巻, 5 号, pp. 616-621, (2019)
- [3] 五十嵐康伸ら: ハッカソンを起点とした顧客との共創: 「企業と友だちになれる就活アプリ attache」の開発過程にみるオープンサービスイノベーション, 情報処理学会論文誌 デジタルプラクティス, vol. 7, no. 2, pp. 167-174, (2016)
- [4] 五十嵐康伸ら: 新しいデータ可視化表現が自発的かつ継続的に開発されるシビックテック活動の設計: E2D3 におけるアプリケーション・オープンソース・ハッカソンのデザイン, 情報処理学会論文誌 デジタルプラクティス, Vol. 8, No. 4, pp. 334-342, (2017)
- [5] <http://e2d3.org/>
- [6] 五十嵐康伸 監修: プロ直伝 伝わるデータ・ビジュアル術 ——Excel だけでは作れないデータ可視化レシピ, 技術評論社, (2019)
- [7] [https://www.e-stat.go.jp/api/event/result\\_statdash2016](https://www.e-stat.go.jp/api/event/result_statdash2016)
- [8] <https://d3js.org/>
- [9] <https://www.stat.go.jp/data/kokusei/2010/users-g/pdf/mikata.pdf>
- [10] <https://cytoscape.org/>