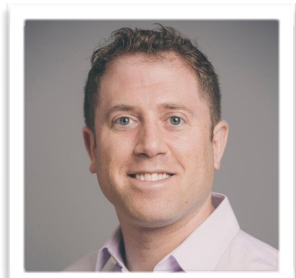


Analyze This! 145 Questions for Data Scientists in Software Engineering



Andrew Begel



Thomas Zimmermann

INTRODUCTION

- **ソフトウェア開発におけるビッグデータの必要性**
 - ソフトウェア開発の様々な局面における意思決定が支援されるべきである
 - データに裏付けられた実証的で信頼に足る役立つ知見を提供する必要がある
- **現時点では、データ利用／分析の目的が不明確**
 - 現場の技術者にとって本当に役立つ知見とは何なのかを、研究者が十分に理解していない

研究の目的：ソフトウェア技術者がデータサイエンティストに答えてもらいたい質問のリストを順位付けして作成すること

METHODOLOGY (1/2)

● 調査1回目

1,500人のMS社員に依頼。203人が回答。\$250の報酬。

① SURVEY

203 participants, 728 response items R1..R728

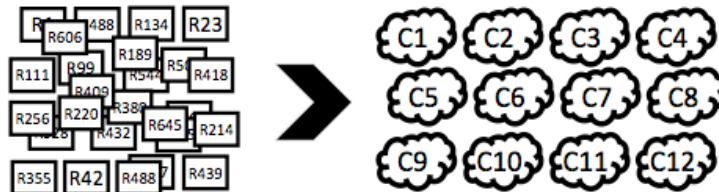
Suppose you could work with a team of data scientists and data analysts who specialize in studying how software is developed. Please list up to five questions you would like them to answer.

①ソフトウェアがどのように開発されているのかを専門に研究しているデータサイエンティストおよびデータアナリストに答えてもらいたい事柄を、一人5つまでリストアップしてもらう。

② CATEGORIES

679 questions in 12 categories C1..C12

We used an open card sort to group questions into categories.



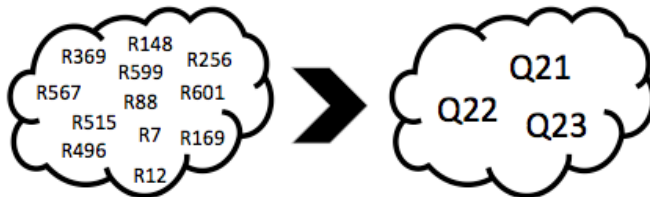
②重複や対象外の質問を取り除いた後、計679個の質問を、オープンカードソート法を用いて12個のカテゴリに分類する。



③ DESCRIPTIVE QUESTIONS

145 questions Q1..Q145

We summarized each category with a set of descriptive questions.



③各カテゴリをいくつかの質問事項として要約し、合計**145**の質問を作成する。

METHODOLOGY (2/2)

● 調査1回目

1回目の調査と異なる2,500人のMS社員に依頼。
607人が回答。\$250の報酬。

④ SURVEY

16,765 ratings by 607 participants

We used a split questionnaire design. Each participant received a subset of the questions Q1..Q145 (on average 27.6) and was asked:

In your opinion, how important is it to have a software data analytics team answer this question?

[Essential | Worthwhile | Unimportant | Unwise | I don't understand]

⑤ TOP/BOTTOM RANKED QUESTIONS

⑥ DIFFERENCES IN DEMOGRAPHICS

④ 145の質問の重要度を決めるために、各質問を技術者に評価してもらう。

- ・ Essential (極めて重要)
- ・ Worthwhile (価値のある/役立つ)
- ・ Unimportant (役立たない)
- ・ Unwise (愚かな/浅はかな)
- ・ I don't understand (理解できない)

すべての質問を評価してもらうのは大変なので、カテゴリに基づいて、145の質問をいくつかのまとまりに分割する (平均27.6個の質問)

⑤ どの質問が技術者にとって重要なのか (重要でないのか) を分析する

⑥ 役職の違い (開発者 vs. テスター vs. 管理者) によって質問の重要度がどのように異なるのかを分析する

RESULTS:

Questions Categories (1/2)

- **Bug Measurements (BUG)**

- バグに関する質問：コードや設計書のどこにバグがあるのか？どんなバグが作られがちなのか？ソフトウェアライフサイクルのどの時点でバグが見つかるのか？バグ修正のコストは？など

- **Development Practices (DP)**

- 開発の実務に関する（コードに関係する）質問：デバック、パフォーマンスチューニング、リファクタリング、リポジトリのブランチ作成、コードレビュー、コメントの方法、レガシーコードによる開発コスト、メトリクス計測、不正確な工数予測の影響、コード変更によるプロダクトリクスなど

- **Development Best Practices (BEST)**

- ソフトウェア開発におけるベストプラクティスに関する質問

- **Testing Practices (TP)**

- テストに関する質問：テストの自動化、テスト戦略、ユニットテスト、テスト駆動開発、カバレッジ、テストケースの削減方法、テストケースの作成方法、インパクトバグを検出するためのテスト方法など

- **Evaluating Quality (EQ)**

- ソフトウェア品質に関する質問：出荷可能かを知るためのメトリクス、複雑度メトリクス、重複コード、テストカバレッジと顧客の利用機能、レガシー&エイジングコード、低品質ソフトウェアの共通原因など

- **Services (SVC)**

- 主にクラウドベースのソフトウェアサービスに関する質問：開発・テスト手法の変更方法、運用とパフォーマンスの計測方法、ラピッドリリースサイクルによる顧客の獲得維持と収益化など

RESULTS:

Questions Categories (2/2)

- **Customers and Requirements (CR)**

- 顧客の関心に関する質問：顧客が最も好む重要な機能，後方互換の必要性，テストの重要性など

- **Software Development Lifecycle (SL)**

- ソフトウェアライフサイクルに関する質問：各開発工程への工数配分の方法，工数配分とソフトウェア（品質）の関係など

- **Software Development Process (PROC)**

- 開発プロセスに関する質問：アジャイルvs.ウォーターフォール，ペアプログラミングの利点，クラウドサービスに特化した方法論，社内統一の開発プロセスやツールを用いる方法とその利点など

- **Productivity (PROD)**

- 開発者の生産性に関する質問：ソフトウェア開発者の生産性を計測する様々なメトリクス，個人の生産性とチームの成功との関係，自分自身／自分のチームの生産性をモニタする方法（新しいコードベースを取り入れた場合，新しいチームメンバが加わった場合，チームメンバーが去った際に起きる問題に対処する場合）など

- **Teams and Collaboration (TC)**

- 開発チーム内のコラボレーションに関する質問：様々な役割を持ったメンバーがどれくらいチームにいるべきか，開発チームのサイズ，チーム内／間での情報共有やコラボレーションの方法など

- **Reuse and Shared Components (RSC)**

- コードの再利用に関する質問：スクラッチvs.再利用のタイミング，正しいコードを検索するためのコスト，リンクvs.コピペによる影響など

RESULTS :

重要質問ベスト20



顧客が望む機能や顧客の利益に関する質問



開発方法やプロセスの効果に関する質問



プロダクトの品質に関する質問

Question	Category	Percentages			Rank		
		Essential	Worthwhile+	Unwise	Essential	Worthwhile+	Unwise
👤 Q27 How do users typically use my application?	DP	80.0%	99.2%	0.83%	1	1	128
👤 Q18 What parts of a software product are most used and/or loved by customers?	CR	72.0%	98.5%	0.00%	2	2	130
★ Q50 How effective are the quality gates we run at checkin?	DP	62.4%	96.6%	0.85%	3	6	125
✂ Q115 How can we improve collaboration and sharing between teams?	TC	54.5%	96.4%	0.00%	4	8	130
★ Q86 What are best key performance indicators (KPIs) for monitoring services?	SVC	53.2%	93.6%	0.92%	5	12	106
★ Q40 What is the impact of a code change or requirements change to the project and tests?	DP	52.1%	94.0%	0.00%	6	10	130
✂ Q74 What is the impact of tools on productivity?	PROD	50.5%	97.2%	0.92%	7	4	106
✂ Q84 How do I avoid reinventing the wheel by sharing and/or searching for code?	RSC	50.0%	90.9%	0.91%	8	19	108
👤 Q28 What are the common patterns of execution in my application?	DP	48.7%	96.6%	0.84%	9	5	127
👤 Q66 How well does test coverage correspond to actual code usage by our customers?	EQ	48.7%	92.0%	0.00%	10	16	130
★ Q42 What tools can help us measure and estimate the risk associated with code changes?	DP	47.8%	92.2%	0.00%	11	15	130
★ Q59 What are effective metrics for ship quality?	EQ	47.8%	96.5%	1.77%	12	7	91
✂ Q100 How much do design changes cost us and how can we reduce their risk?	SL	46.6%	94.8%	0.86%	13	9	123
👤 Q19 What are the best ways to change a product's features without losing customers?	CR	46.2%	92.3%	1.54%	14	14	103
👤 Q131 Which test strategies find the most impactful bugs (e.g., assertions, in-circuit testing, A/B testing)?	TP	44.5%	91.8%	0.91%	15	18	108
✂ Q83 When should I write code from scratch vs. reuse legacy code?	RSC	44.5%	84.5%	3.64%	15	45	37
Q1 What is the impact and/or cost of findings bugs at a certain stage in the development cycle?	BUG	43.1%	87.9%	2.59%	17	28	68
👤 Q92 What is the tradeoff between releasing more features or releasing more often?	SVC	42.5%	79.6%	0.00%	18	64	130
✂ Q2 What kinds of mistakes do developers make in their software? Which ones are the most common?	BUG	41.7%	98.3%	0.00%	19	3	130
👤 Q25 How important is a particular requirement?	CR	41.7%	87.4%	2.36%	20	32	73
★ Q60 How should we use metrics to help us decide when a feature is good enough to release (or poor enough to cancel)?	EQ	41.1%	90.2%	3.57%	23	20	41
👤 Q17 What is the best way to collect customer feedback?	CR	39.8%	93.0%	1.56%	26	13	101
✂ Q3 In what places in their software code do developers make the most mistakes?	BUG	35.0%	94.0%	0.00%	41	10	130
★ Q102 What kinds of problems happen because there is too much software process?	PROC	30.1%	92.0%	0.88%	57	16	116

RESULTS :

重要質問ワースト10

Question	Category	Percentages			Rank		
		Essential	Worthwhile+	Unwise	Essential	Worthwhile+	Unwise
Q72 Which individual measures correlate with employee productivity (e.g., employee age, tenure, engineering skills, education, promotion velocity, IQ)?	PROD	7.3%	44.5%	25.5%	142	141	1
Q71 Which coding measures correlate with employee productivity (e.g., lines of code, time it takes to build the software, a particular tool set, pair programming, number of hours of coding per day, language)?	PROD	15.6%	56.9%	22.0%	126	131	2
Q75 What metrics can be used to compare employees?	PROD	19.4%	67.6%	21.3%	110	112	3
Q70 How can we measure the productivity of a Microsoft employee?	PROD	19.1%	70.9%	20.9%	113	104	4
Q6 Is the number of bugs a good measure of developer effectiveness?	BUG	16.4%	54.3%	17.2%	122	136	5
Q128 Can I generate 100% test coverage?	TP	15.3%	44.1%	14.4%	127	142	6
Q113 Who should be in charge of creating and maintaining a consistent company-wide software process and tool chain?	PROC	21.9%	55.3%	12.3%	93	134	7
Q112 What are the benefits of a consistent, company-wide software process and tool chain?	PROC	25.2%	78.3%	10.4%	79	72	8
Q34 When are code comments worth the effort to write them?	DP	7.9%	41.2%	9.6%	141	143	9
Q24 How much time and money does it cost to add customer input into your design?	CR	15.9%	68.2%	8.3%	124	111	10

RESULTS :

重要質問 (部門別/役割別)

Question		Category	Response	Discipline		
				Dev	Test	PM
Q5	How many new bugs are introduced for every bug that is fixed?	BUG	Essential	27.3%	41.9%	12.5%
Q10	When should we migrate our code from one version of a library to the next?	BEST	Essential	32.6%	16.7%	5.1%
Q20	How much value do customers place on backward compatibility?	CR	Essential	14.3%	47.1%	18.3%
Q21	What is the tradeoff between frequency and high-quality when releasing software?	CR	Essential	22.9%	48.5%	14.5%
Q42	<i>What tools can help us measure and estimate the risk associated with code changes? (Essential #11)</i>	DP	Essential	34.5%	70.6%	40.4%
Q121	How can we make it easier for people to find and use commonly used tools?	TC	Essential	27.3%	48.6%	20.0%
Q24	<i>How much time and money does it cost to add customer input into your design? (Unwise #10)</i>	CR	Worthwhile+	62.9%	88.2%	60.3%
Q113	<i>Who should be in charge of creating and maintaining a consistent company-wide software process and tool chain? (Unwise #7)</i>	PROC	Worthwhile+	60.0%	71.9%	40.4%
Q132	How should we handle test redundancy and/or duplicate tests?	TP	Worthwhile+	48.6%	81.1%	47.4%
Q134	Should we develop a separate test suite for servicing a product after we ship it?	TP	Worthwhile+	32.4%	67.6%	67.6%

				Management Role	
				Manager	Ind. Contributor
Q29	How much legacy code is in my codebase?	DP	Worthwhile+	36.7%	65.2%
Q31	When in the development cycle should we test performance?	DP	Worthwhile+	63.3%	81.4%
Q70	<i>How can we measure the productivity of a Microsoft employee? (Unwise #4)</i>	PROD	Worthwhile+	57.1%	77.3%
Q120	What are the most commonly used tools on a software team?	TC	Worthwhile+	95.8%	67.8%

RESULTS :

重要質問（地域別／経験別／勤務年数別）

				Region		
				Asia	Europe	North America
Q70	How can we measure the productivity of a Microsoft employee? (Unwise #4)	PROD	Essential	52.9%	30.0%	11.0%
Q104	How do software methodologies affect the success and customer satisfaction of shrinkwrapped and service-oriented products?	PROC	Essential	52.9%	10.0%	24.7%
Q128	Can I generate 100% test coverage? (Unwise #6)	TP	Essential	60.0%	0.0%	9.0%
Q129	What is the effectiveness, reliability, and cost of automated testing?	TP	Essential	71.4%	12.5%	23.6%

				Years as Manager	
				change in odds per year	
Q85	How much cloned code is ok to have in a codebase?	RSC	Essential	36%	
Q69	How does the age of code affect its quality, complexity, maintainability, and security?	EQ	Worthwhile+	-28%	

				Years at Company	
				change in odds per year	
Q16	What criteria should we decide when to use managed code or native code (e.g., speed, productivity, functionality, newer language features, code quality)?	BEST	Essential	-23%	
Q119	What are the best tools and processes for sharing knowledge and task status?	TC	Essential	-18%	
Q142	Should we do Test-Driven Development?	TP	Essential	-19%	
Q95	How much time went into testing vs. into development?	SL	Worthwhile+	-12%	
Q123	How much distinction should there be between developer/tester roles?	TC	Worthwhile+	-14%	
Q136	Who should write unit tests, developers or testers?	TP	Worthwhile+	-13%	
Q142	Should we do Test-Driven Development?	TP	Worthwhile+	-13%	
Q144	How should we do Test-Driven Development while prototyping?	TP	Worthwhile+	-14%	
Q113	Who should be in charge of creating and maintaining a consistent company-wide software process and tool chain? (Unwise #7)	PROC	Unwise	15%	

IMPLICATIONS

- **研究**

- 145の質問は、企業が解決を必要としている問題点（＝研究者が取り組むべき問題）を提示している
- 他の企業が同様のサーベイを行うことで、研究者が真に取り組むべき問題点を明確に定義することができる

- **実務**

- 145の質問のすべてが未解決の問題を示している訳ではない
- 解決可能なツールや手法の存在に気付けない事が問題である
- 産学がより密に協力してこの問題に取り組むべきである

- **教育**

- 145の質問は将来ソフトウェア業界で働く学生にとって役に立つ（将来どのような問題に直面するのかを理解できる）
- ソフトウェア工学の講義内容を作成するのにも役に立つ

CONCLUSION

- **ソフトウェアエンジニアがデータサイエンティストに聞きたい145の質問をリストアップした**
 - 145の質問を12個のカテゴリに分類した
 - 145の質問をランク付けして重要な質問を明らかにした
- **replication study (再現実験) が多方面で行われる事を希望する**
 - データセットを用意した
 - <http://research.microsoft.com/apps/pubs/?id=200784>