

All Spreadsheets must Die

Robert Schadek

May 8, 2019

Getting started

A random list of languages we love to hate



C++ 4.4*Million* (2015)

C 1.9*Million* (2015)

Java 9*Million* (2009)

JS 10*Million* (2018)

These are all small fish

These are all small fish

Excel

≈ 750 Million (2016)





2001FinancialStatements.xlsx - Microsoft Excel

Home Insert Page Layout Formulas Data Review View Load Test Tools

Clipboard Font Paragraph Alignment Number Styles Cells Editing

AP52

1 Consolidated Statements of Shareholders' Equity
2 (DOLLAR IN THOUSANDS)
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52

Consolidated Statements of Shareholders' Equity
(DOLLAR IN THOUSANDS)

Balance, January 1, 1999

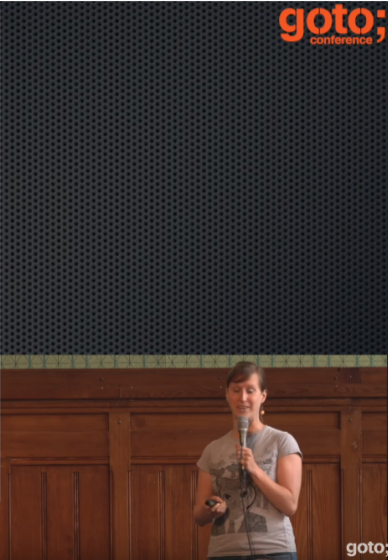
Net income
Translation adjustment
Pensions
Unrealized loss on investment securities
Other comprehensive income
Comprehensive income
Stock options exercised
Unearned compensation
Performance shares
Procamp and flexus acquisitions
Dividends declared and paid
Treasury shares
Balance, December 31, 1999

Net income
Translation adjustment
Pensions
Unrealized loss on investment securities
Other comprehensive loss
Comprehensive income
Stock options exercised
Unearned compensation
Performance shares
Dividends declared and paid
Treasury shares
Balance, December 31, 2000

Net income
Translation adjustment
Pensions
Unrealized gain on investment securities
Other comprehensive loss
Comprehensive income
Stock options exercised
Unearned compensation
Dividends declared and paid
Treasury shares
Balance, December 31, 2001

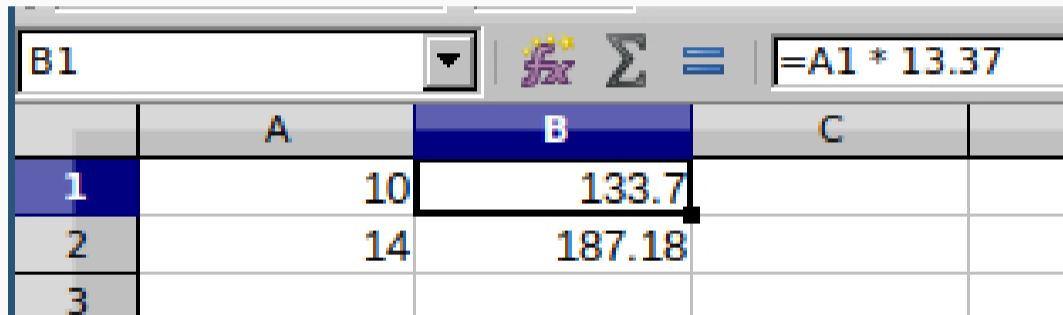
	A	B	C	D	E	F	G	H	I	J	K	L			
Balance, January 1, 1999	69,431,481	\$	89,862	\$	89,862	\$	271,902	\$	(12,802)	\$	245	\$	899,123		
Net income													130,816		
Translation adjustment													9,510		
Pensions													614		
Unrealized loss on investment securities													(3,235)		
Other comprehensive income															
Comprehensive income													137,695		
Stock options exercised	188,104		134		1,918								2,052		
Unearned compensation	145,799		188		3,933				(3,485)				636		
Performance shares	20,397		26		686								712		
Procamp and flexus acquisitions	1,710,214		2,138		37,351		9,487						48,976		
Dividends declared and paid					(81,668)		(1,229)						(81,668)		
Treasury shares													(1,229)		
Balance, December 31, 1999	71,482,997	\$	89,354	\$	87,169	\$	691,415	\$	(13,644)	\$	(5,865)	\$	(8,034)	\$	844,395
Net income													136,919		
Translation adjustment													(7,904)		
Pensions													1,507		
Unrealized loss on investment securities													(296)		
Other comprehensive loss													(8,793)		
Comprehensive income													130,126		
Stock options exercised	273,238		343		5,444								5,787		
Unearned compensation	247,635		308		5,583				(3,915)				1,976		
Performance shares	15,335		19		334								353		
Dividends declared and paid					(84,271)		(2,380)						(84,271)		
Treasury shares													(2,380)		
Balance, December 31, 2000	536,208	\$	90,024	\$	90,530	\$	784,063	\$	(15,544)	\$	(12,458)	\$	(7,949)	\$	936,866
Net income													66,893		
Translation adjustment													(47,375)		
Pensions													(1,628)		
Unrealized gain on investment securities													1,213		
Other comprehensive loss													(47,788)		
Comprehensive income													18,195		
Stock options exercised	176,395		221		4,860								5,081		
Unearned compensation													1,412		
Dividends declared and paid													(45,774)		
Treasury shares													(12,780)		
Balance, December 31, 2001	712,603	\$	90,245	\$	103,390	\$	805,182	\$	(28,724)	\$	(8,444)	\$	(8,537)	\$	903,110

Functional programming in Excel
Feliene Hermans
@Feliene



A little bit of Spreadsheet bashing

Seeing the code is difficult

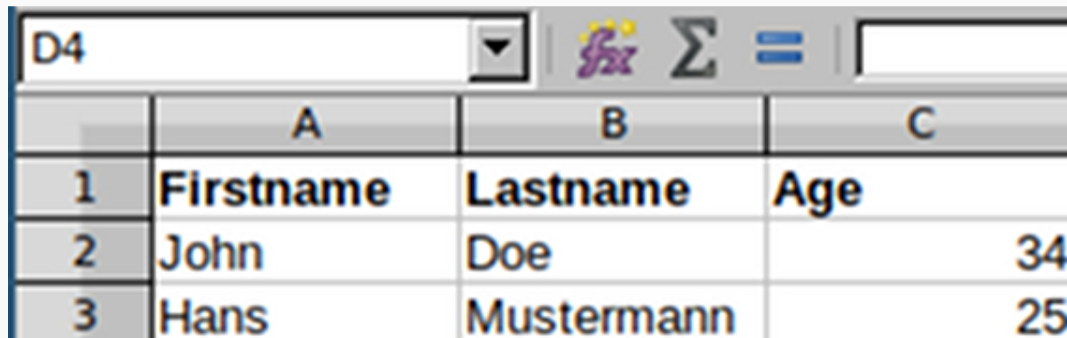


The image shows a screenshot of an Excel spreadsheet. The formula bar at the top displays the formula `=A1 * 13.37`. Below the formula bar, the spreadsheet grid is visible with columns labeled A, B, and C, and rows labeled 1, 2, and 3. The cell B1 is highlighted, showing the value 133.7. The cell A1 contains the value 10, and the cell A2 contains the value 14. The cell B2 contains the value 187.18.

	A	B	C
1	10	133.7	
2	14	187.18	
3			

	A	B	C
1	Firstname	Lastname	Age
2	John	Doe	34
3	Hans	Mustermann	twenty-five
4			

Dynamic Types



The image shows a spreadsheet interface. At the top, the formula bar displays 'D4' and contains icons for undo, redo, and a summation symbol. Below the formula bar is a table with three columns labeled A, B, and C, and three rows numbered 1, 2, and 3. The table contains the following data:

	A	B	C
1	Firstname	Lastname	Age
2	John	Doe	34
3	Hans	Mustermann	25

Dynamic Types

The image shows a spreadsheet interface. At the top, the formula bar displays 'C3' and '= 25'. Below it is a table with three columns: A, B, and C. The first row contains headers: 'Firstname', 'Lastname', and 'Age'. The second row contains 'John', 'Doe', and '34'. The third row contains 'Hans', 'Mustermann', and '25'. The cell containing '25' is highlighted with a thick black border.

	A	B	C
1	Firstname	Lastname	Age
2	John	Doe	34
3	Hans	Mustermann	25



git blame

git blame

lets not go there

- =SUM(1,2)

- `=SUM(1,2)`
- `equal, identifier, lparen, int(1), comma, int(2), rparen`

- =SUM(1,2)
- equal, identifier, lparen, int(1), comma, int(2), rparen
- set Excel locale to de_DE

- =SUM(1,2)
- equal, identifier, lparen, int(1), comma, int(2), rparen
- set Excel locale to de_DE
- =SUM(1,2)

- =SUM(1,2)
- equal, identifier, lparen, int(1), comma, int(2), rparen
- set Excel locale to de_DE
- =SUM(1,2)
- equal, identifier, lparen, float(1.2), rparen

- Knowledge silos
- Slow
- No separation between data and code
- Access management ...

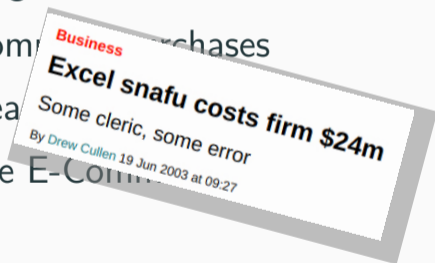
- Knowledge silos
- Slow
- No separation between data and code
- Access management ... anybody?

(Typical) Spreadsheet Lifecycle

1. Create private shopping spreadsheet
2. Show spreadsheet to college
3. Use spreadsheet for all company purchases
4. Put web frontend on spreadsheet backend
5. Pivot company to become E-Commerce company

(Typical) Spreadsheet Lifecycle

1. Create private shopping spreadsheet
2. Show spreadsheet to college
3. Use spreadsheet for all computer purchases
4. Put web frontend on spreadsheet
5. Pivot company to become E-Commerce



(Typical) Spreadsheet Lifecycle

1. Create private shopping spreadsheet
2. Show spreadsheet
3. Use spreadsheet to purchase
4. Put web frontend on spreadsheet
5. Pivot company to become E-Commerce

Quicktake
The London Whale

By [Patricia Hurtado](#)
Updated on 23 February 2016, 22:04 GMT

The trader known as the London Whale lost at least \$6.2 billion for

SHARE THIS ARTICLE

Business
Excel snafu costs firm \$24m

Some cleric, some error

By [Drew Cullen](#) 19 Jun 2003 at 09:27

(Typical) Spreadsheet Lifecycle

1. Create private shopping spreadsheet
2. Show spreadsheet
3. Use spreadsheet
4. Put web frontend on spreadsheet
5. Pivot company to become E-Commerce



(Typical) Spreadsheet Lifecycle

1. Create private shopping spreadsheet
2. Show spreadsheet
3. Use spreadsheet
4. Put web frontend on spreadsheet
5. Pivot company to become E-Commerce

Quicktake
The London Whale
By Patricia Hurtado
Updated on 23 February 2016, 22:04 GMT
The trader known as the London Whale

Forget Excel: The Mistake
Correlation is not causation

Excel error leaves Barclays with unwanted Lehman assets
15 October 2008

sts firm \$24m

hart and Rogoff's Biggest

(Typical) Spreadsheet Lifecycle

1. Create private shopping
2. Show spreadsheet
3. Use spreadsheet
4. Put web front end on spreadsheet
5. Pivot company to become E-Comm

Quick London 2012 synchronised swimming tickets oversold

The Olympics organisers admit they had to ask purchasers to swap events after selling 10,000 too many tickets

By Patrick...
Updated on 23 February 2016, 22:04 GMT

The trader known as the London W...

SHARE THIS ARTICLE

FORGET
Forget Excel
Mistake

Contribution is not causation
BY THE AUTHOR
BY DREW CUTLER

By Drew Cutler

BY THE AUTHOR

Excel or leaves
Barclays with unwanted
Lehman assets

15 October 2008

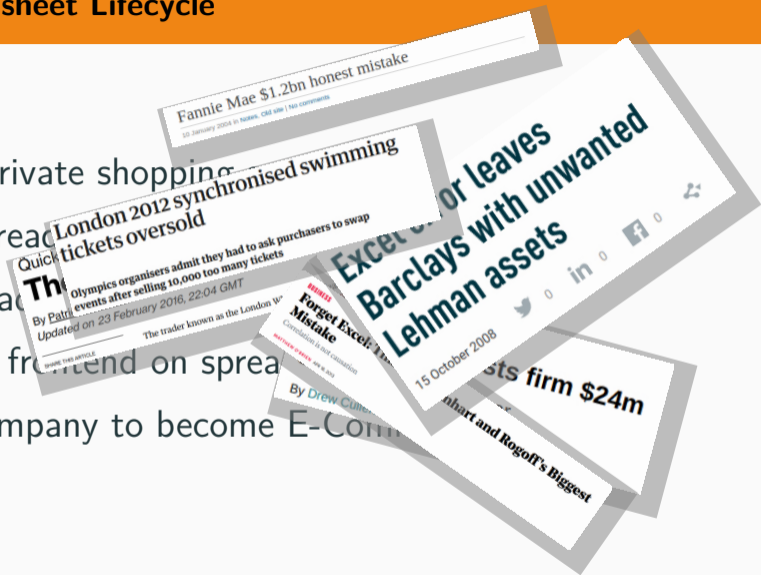


...sts firm \$24m

...hart and Rogoff's Biggest

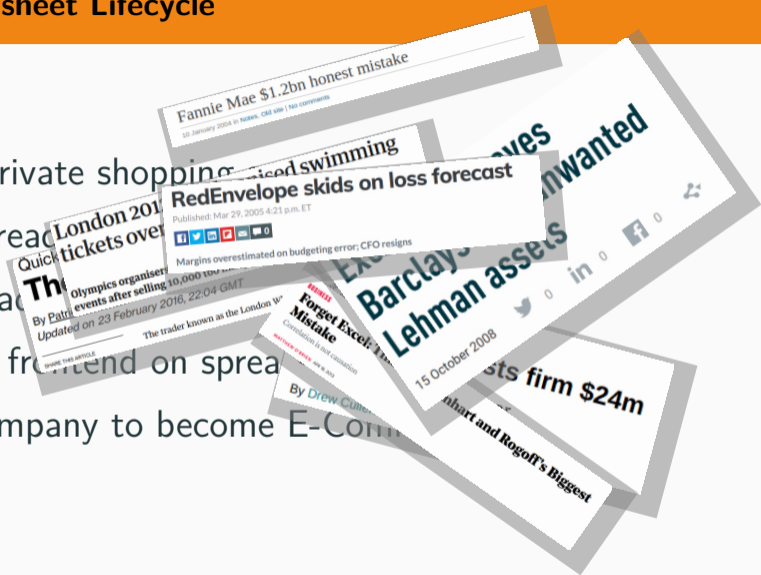
(Typical) Spreadsheet Lifecycle

1. Create private shopping
2. Show spreadsheet
3. Use spreadsheet
4. Put web front end on spreadsheet
5. Pivot company to become E-Comm



(Typical) Spreadsheet Lifecycle

1. Create private shopping list
2. Show spreadsheet to friends
3. Use spreadsheet to track progress
4. Put web front end on spreadsheet
5. Pivot company to become E-Commerce



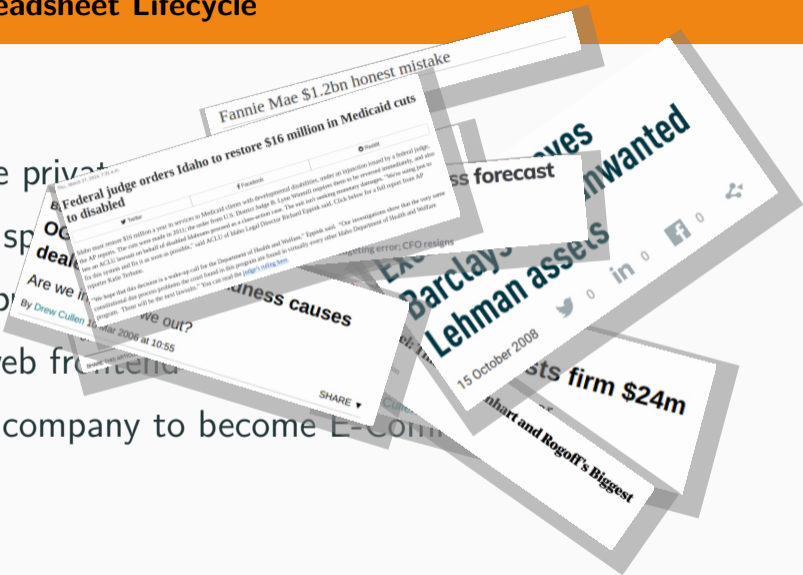
(Typical) Spreadsheet Lifecycle

1. Create private spreadsheet
2. Show spreadsheet
3. Use spreadsheet
4. Put web front end
5. Pivot company to become E-Commerce



(Typical) Spreadsheet Lifecycle

1. Create private spreadsheet
2. Show spreadsheet to deal
3. Use spreadsheet to work out business causes
4. Put web front end on spreadsheet
5. Pivot company to become E-commerce



(Typical) Spreadsheet Lifecycle

1. Create pr
2. Show sp
3. Use sp
4. Put web
5. Pivot cor



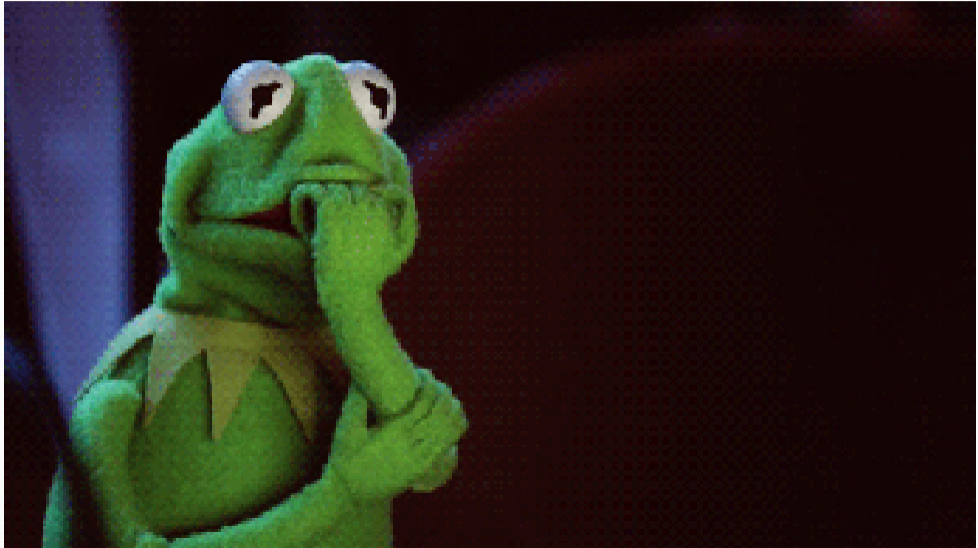
The screenshot shows the EuSpRIG website interface. The top navigation bar includes links for Home, About EuSpRIG, EuSpRIG 2019 Annual Conference, EuSpRIG 2019 Call for Papers & Presentations, Basic Research, Best Practice, Horror Stories, Regulations' Presentations, Conference Abstracts, Papers & Indexes, Conference Reports & Videos, and Delegates. The main content area is titled 'EuSpRIG ORIGINAL HORROR STORIES' and lists two stories in reverse chronological order.

089) Which way around do you calculate a percentage change?
<http://www.startribune.com/587/story/367953.html>
Making a political plus of math error. Star Tribune, 12 Apr 06
DFler Rebecca Otto is accusing Auditor Pat Anderson, a Republican, of sloppy work. In a column reporting the percentage changes in unreserved fund balances from 2003 to 2004, instead of dividing the difference by the 2003 figure, the auditor's office divided it by the 2004 figure. Deputy Auditor Tony Sutton said "The researcher who worked on that report just made a mistake in the formula in the spreadsheet. He feels bad about it."
Risk: Giving political opponents an opportunity to comment adversely on the state auditor
Avoidance: Check not just the data but the formulas too.
Advice: Register Now! for the EuSpRIG 2012 Conference

088) Scoring an own goal
http://www.channelregister.co.uk/2005/03/10/logic_spreadsheet_snaul/
The Register (UK) OGC spreadsheet madness 10 Mar 06
The Office of Government Commerce is blaming a spreadsheet error for a foul-up over accrediting suppliers for its new Catalyst procurement programme. In a letter to suppliers, they confessed "Unfortunately, [we are] not, as we had hoped, in a position to accept your tender at this time. This is because an error in the original evaluation spreadsheet has been identified. necessitatina

Spreadsheets rule the world!

How you should be feeling right now



Two assumptions going forward

1. You believe that spreadsheets rule the world.
2. You want D to rule the world instead.

How are we going to win this?

How are we going to win this?

We are not!

Lets draw up a battle plan

Lets take stock of what we have

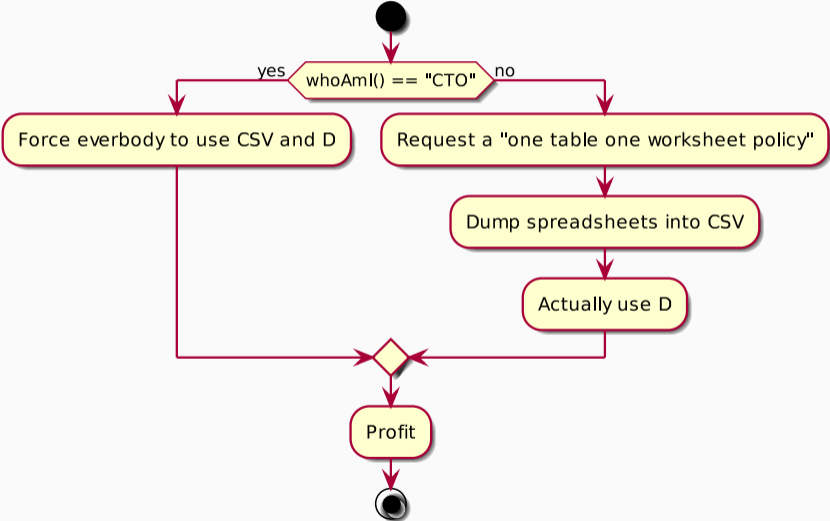
- Too many spreadsheets
- Too many tasks
- Too little man-power

Lets take stock of what we have

- Too many spreadsheets
- Too many tasks
- Too little man-power

- Millions of lines of source in different languages
- D

Possible Attack Vectors



How to work with limited man-power

Leveraging existing libraries

Writing data to spreadsheets

Leveraging existing libraries

Writing data to spreadsheets

- It is required, people will ask for that
- Writing a somewhat feature complete xlsx writer is a huge task

Leveraging existing libraries

Writing data to spreadsheets

- It is required, people will ask for that
- Writing a somewhat feature complete xlsx writer is a huge task

- libxlsxwriter is a feature rich xlsx writer
- Wrapping it by hand,

Leveraging existing libraries

Writing data to spreadsheets

- It is required, people will ask for that
- Writing a somewhat feature complete xlsx writer is a huge task

- libxlsxwriter is a feature rich xlsx writer
- Wrapping it by hand, no way (+78000 lines of structs, enums and functions)

Leveraging existing libraries

Writing data to spreadsheets

- It is required, people will ask for that
- Writing a somewhat feature complete xlsx writer is a huge task

- libxlsxwriter is a feature rich xlsx writer
- Wrapping it by hand, no way (+78000 lines of structs, enums and functions)

- dpp to the rescue
- libxlsxwriter.d (+4000 lines)
- But it is still a C api

```
1 void chart_axis_set_name(lxw_chart_axis*, const(char)*)
2 void chart_axis_set_name_font(lxw_chart_axis*, lxw_chart_font*)
3 void chart_axis_set_num_font(lxw_chart_axis*, lxw_chart_font*)
4 void chart_axis_set_num_format(lxw_chart_axis*, const(char)*)
5 void chart_axis_set_line(lxw_chart_axis*, lxw_chart_line*)
6 void chart_axis_set_fill(lxw_chart_axis*, lxw_chart_fill*)
7 ...
```

Semi-automatic refactoring

```
1  struct ChartAxis {
2      lxw_chart_axis* handle;
3
4      void setName(string name) {
5          chart_axis_set_name(this.handle, toStringz(name));
6      }
7
8      void setNameRange(string n, lxw_row_t row,
9                          lxw_col_t col)
10     {
11         chart_axis_set_name(this.handle, toStringz(n), row, col);
12     }
13     ...
14 }
```

Creating fake data

Problem to solve: We needed fake data with a variety of attributes.

- Name
- Address
- i18n
- ...



faker.js

- +160 attributes
- 39 languages

```
1 module["exports"] = [  
2   "#{prefix} #{first_name} #{last_name}",  
3   "#{first_name} #{nobility_title_prefix} #{last_name}",  
4   "#{first_name} #{last_name}",  
5   "#{first_name} #{last_name}",  
6   "#{first_name} #{last_name}",  
7   "#{first_name} #{last_name}"  
8 ];
```

Listing 1: locales/de/name/name.js


```
1  override string nameName() {
2      switch(uniform(0, 6, this.rnd)) {
3          case 0:
4              return format! "%s %s %s" (namePrefix(), nameFirstName(),
5                  nameLastName());
6          case 1:
7              return format! "%s %s %s" (nameFirstName(), nameNobilityTitlePrefix(),
8                  nameLastName());
9          case 2:
10             return format! "%s %s" (nameFirstName(), nameLastName());
11          case 3:
12             return format! "%s %s" (nameFirstName(), nameLastName());
13          case 4:
14             return format! "%s %s" (nameFirstName(), nameLastName());
15          case 5:
16             return format! "%s %s" (nameFirstName(), nameLastName());
17          default: assert(false);
18      }
```

```
1 import faked;
2
3 auto f = new Faker(1337);
4 writeln(f.nameName());
5
6 // localized to german
7 f = new Faker_de(1338);
8 writeln(f.nameName());
```

- Input:
 - Parser and Generator \approx 1500 lines of D
 - A day of boring work

- Input:
 - Parser and Generator \approx 1500 lines of D
 - A day of boring work
- Output:
 - Output feature equivalent \approx 70000 lines faker.js clone
 - Most changes in faker.js just require a rerun of the tool to update

- Input:
 - Parser and Generator \approx 1500 lines of D
 - A day of boring work
- Output:
 - Output feature equivalent \approx 70000 lines faker.js clone
 - Most changes in faker.js just require a rerun of the tool to update
- Bonus:
 - Created two PRs to faker.js fixing wrong template expansion

Taking a step back

The Wanted Output: Salary Table

Firstname	Lastname	Amount	Currency	CreatedBy
Hans	Meier	73331	USD	Ruth Ember
John	Doe	83431	GPB	Ruth Ember
Ruth	Ember	103431	EUR	Hans Meier

The starting point

```
1  class Employee {
2      long id;
3      DateTime createdAt;
4
5      EmployeeInfo info;
6      long infoId;
7  }
8
9  class EmployeeInfo {
10     long id;
11     string firstname;
12     string lastname;
13
14     Salary salary;
15     long salaryId;
16 }

17 class Salary {
18     long id;
19     Employee createdBy;
20     long createdById;
21
22     CurrencyAmount amount;
23     long amountId;
24 }

25
26 class CurrencyAmount {
27     long id;
28     double amount;
29
30     Currency currency;
31     long currencyId;
32 }

33 class Currency {
34     long id;
35     string name;
36 }
```


The vibe.d REST interface

```
1 interface Backend {
2     Employee[] getAllEmployees();
3     Employee getEmployee(long empId);
4     EmployeeInfo getEmployeeInfo(long empInfoId);
5     Salary getSalary(long salaryId);
6     CurrencyAmount getCurrencyAmount(long amountId);
7     Currency getCurrency(long currencyId);
8 }
```

The Frontend Code: Types

```
1 interface Employee {
2   id: number;
3   createdAt: number;
4
5   info?: EmployeeInfo;
6   infoId: number;
7 }
8
9 interface EmployeeInfo {
10  id: number;
11  firstname: string;
12  lastname: string;
13
14  salary?: Salary;
15  salaryId: number;
16 }
```

```
17 interface Salary {
18   id: number;
19   createdBy?: Employee;
20   createdById: number;
21
22   amount?: CurrencyAmount;
23   amountId: number;
24 }
25
26 interface CurrencyAmount {
27   id: number;
28   amount: number;
29
30   currency?: Currency;
31   currencyId: number;
32 }
```

The Frontend Code: Backend Service

```
1 import {
2     Employee, EmployeeInfo, Salary, CurrencyAmount, Currency
3 } from "model";
4
5 class Backend {
6     getAllEmployees() : Employee[] { ... }
7     getEmployee(empId: number): Employee { ... }
8     getEmployeeInfo(empInfoId: number): EmployeeInfo { ... }
9     getSalary(salaryId: number): Salary { ... }
10    getCurrencyAmount(amountId: number): CurrencyAmount { ... }
11    getCurrency(currencyId: number): Currency { ... }
12 }
```

The Frontend Code: Calling the Backend

```
1  this.backend.getAllEmployees().pipe(  
2    mergeMap((emps: Employee[]) => {  
3      const obs = [];  
4      for(const emp of emps) {  
5        obs.push(this.backend.getEmployeeInfo(emp.infoId)  
6          .pipe(map((empInfo: EmployeeInfo) => {  
7            const ne: Employee = {...emp, info : empInfo};  
8            return ne;  
9          })))  
10       }  
11     });  
12   }  
13   return forkJoin(obs);  
14 },  
15 mergeMap((emps: Employee[]) => {  
16   const obs = [];
```

The Communication



The Takeaways

- Clearly, this is unworkable
- Not plastic at all
- Just a lot of boring work

The real Takeway



So, what do we/I want?

Declare how we want data to get, when we're asking for it.

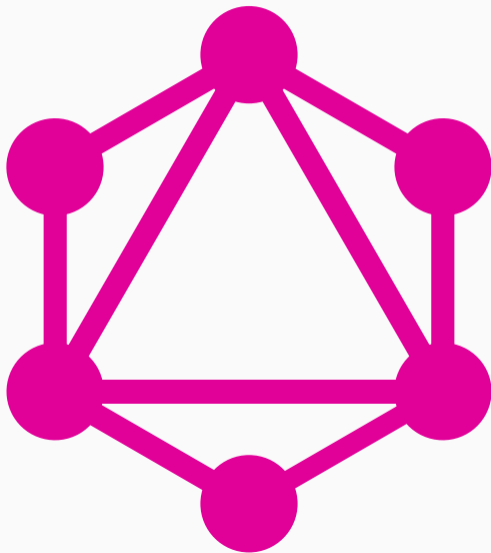
Why can't we write this?

```
1 {
2   allEmployees {
3     info {
4       firstname
5       lastname
6       salary {
7         amount
8         currency {
9           name
10        }
11       createdBy {
12         info {
13           firstname
14           lastname
15         }
16       }
17     }
18   }
19 }
20 }
```

Why can't we write this?

```
1 {
2   allEmployees {
3     info {
4       firstname
5       lastname
6       salary {
7         amount
8         currency {
9           name
10        }
11       createdBy {
12         info {
13           firstname
14           lastname
15         }
16       }
17     }
18   }
19 }
20 }
```

```
1 {
2   allEmployees: [ {
3     info: {
4       firstname: "Hans",
5       lastname: "Meier",
6       salary: {
7         amount: 73331,
8         currency: {
9           name: "USD"
10        }
11       createdBy: {
12         info: {
13           firstname: "Ruth",
14           lastname: "Ember"
15         }
16       }
17     }
18   },
19   ...
20 ]
21 }
```



GraphQL

```
1  schema {
2    query: Backend
3  }
4
5  type Backend {
6    getAllEmployees: [Employee]
7  }
8
9  type Employee {
10   id: number!;
11   createdAt: number!;
12
13   info: EmployeeInfo;
14   infoId: number!;
15 }
16
17 type EmployeeInfo {
18   id: number!;
19   firstname: String!;
20   lastname: String!;
21   salaryId: number!;
22   salary: Salary;
23 }
24
25 type Salary {
26   id: number!;
27   createdBy: Employee;
28   amountId: number!;
29   amount: CurrencyAmount;
30 }
31
32 type CurrencyAmount {
33   id: number!;
34   amount: number!;
35   currencyId: number!;
36   currency: Currency;
37 }
38
39 type Currency {
40   id: number!;
41   name: string!;
42 }
```

Less code is better

```
1 query one {
2   allEmployees {
3     ...deep
4   }
5 }
6
7 fragment names on EmployeeInfo {
8   firstname
9   lastname
10 }
11
12 fragment empInfo on Employee {
13   info {
14     ...names
15   }
16 }
```

```
1 fragment deep on Employee {
2   info {
3     ...names
4     salary {
5       amount
6       currency {
7         name
8       }
9       createdBy {
10        ...empInfo
11      }
12    }
13  }
14 }
```

Introspecting Types

```
1 {
2   __type(name: "Employee") {
3     name
4     fields {
5       name
6       type {
7         name
8         kind
9         ofType {
10          name
11        }
12      }
13    }
14  }
15 }
```

```
1 {
2   "data": {
3     "__type": {
4       "name": "Employee",
5       "fields": [
6         {
7           "name": "id",
8           "type": {
9             "name": null,
10            "kind": "NON_NULL",
11            "ofType" {
12              "name": "INT"
13            }
14          }
15        },
16        {
17          "name": "info",
18          "type": {
19            "name": "EmployeeInfo",
20            "kind": "OBJECT"
21          }
22        }
23      ]
24    }
25  }
```

[Packages](#)[Documentation ▾](#)[About ▾](#)[Download](#)[Lo](#)

Search results for: *graphql*

Package	Latest version
---------	----------------

Found 0 packages.

JUST DO IT




```
1 import graphql;
2
3 interface Query {
4     Employee[] getAllEmployees();
5 }
6
7 class Schema {
8     Query queryType;
9 }
```

```
1 import graphql;
2
3 interface Query {
4     Employee[] getAllEmployees();
5 }
6
7 class Schema {
8     Query queryType;
9 }
```

```
1 auto graphql = new GraphQL!(Schema)();
2
3 graphql.setResolver(
4     "queryType", "getAllEmployees",
5     delegate(string name, Json parent,
6         Json args, ref Context context) @safe
7     {
8         Employee[] employees = getAllEmployees();
9         Json ret = Json.emptyObject();
10        ret["data"] = toGraphQLJson(employees);
11        return ret;
12    });
```

GraphQLD

```
1 void graphqlEndpoint(HTTPRequest req,
2     HTTPServerResponse res)
3 {
4     string toParse = extractQuery(req);
5     auto p = Parser(Lexer(toParse));
6
7     Document d = p.parseDocument();
8     auto fv = new QueryValidator(d);
9     auto sv = new SchemaValidator!Schema(d, graphqlD.schema);
10    fv.accept(d);
11    sv.accept(d);
12
13    Context con = buildContext(req);
14    Json ret = graphqlD.execute(d, extractVariables(req), con);
15    res.writeJsonBody(ret);
16 }
```

```
1 graphqlD.setResolver("Employee", "info",
2     delegate(string name, Json parent, Json args,
3         ref Context context)
4     {
5         const id = parent["infoId"].get!long();
6         EmployeeInfo ei = getEmployeeInfo(id);
7         Json ret = Json.emptyObject();
8         ret["data"] = toGraphQLJson(ei);
9         return ret;
10    });
```

- Mostly feature complete
- Some validations are missing
- \approx 17000 lines
- \approx 9000 lines are generated by darser
- ready for use now

Homework

Write a GraphQL backend that uses an excel spreadsheet as a database.

Conclusion

Conclusion

- Spreadsheets are a terrible programming language.
- C++ and Rust are not our main competition on our path to world domination.

Conclusion

- Spreadsheets are a terrible programming language.
- C++ and Rust are not our main competition on our path to world domination.

- We need to learn to use what is there.
- Use D to work smart not hard.
- Do not write the code, write the code that writes the code.
- Look at JS for inspiration.
- GraphQL 👍

The End

- [1] *Infographic: C/C++ facts we learned before going ahead with CLion.*
<https://blog.jetbrains.com/clion/2015/07/infographics-cpp-facts-before-clion/>. (Accessed on 04/08/2019).
- [2] Developer Economics. *Developer Economics: State of the Developer Nation 15th Edition.* <https://www.developereconomics.com/reports/state-of-the-developer-nation-15th-edition>. (Accessed on 04/08/2019).
- [3] Microsoft. *Build 2016 Keynote (Day 2).*
<https://www.youtube.com/watch?v=bf0Rr81is6U>. (Accessed on 04/08/2019).
- [4] Irish Tech News. *Seven reasons why Excel is still used by half a billion people worldwide.* <https://irishtechnews.ie/seven-reasons-why-excel-is-still-used-by-half-a-billion-people-worldwide/>. (Accessed on 04/08/2019).

- [5] Felienne Hermans. *GOTO 2016: Pure Functional Programming in Excel by Felienne Hermans*. <https://www.youtube.com/watch?v=0yKf8TrLU0w>. (Accessed on 04/08/2019).
- [6] *Marak/faker.js: generate massive amounts of realistic fake data in Node.js and the browser*. <https://github.com/marak/Faker.js/>. (Accessed on 05/03/2019).
- [7] *kaleidicassociates/faked: D library to create real fake data*. <https://github.com/kaleidicassociates/faked>. (Accessed on 05/03/2019).
- [8] *GraphQL | A query language for your API*. <https://graphql.org/>. (Accessed on 05/03/2019).

- [9] *burner/graphqld: A vibe.d library to handle the GraphQL Protocol written in the D Programming Language*. <https://github.com/burner/graphqld>. (Accessed on 05/03/2019).
- [10] *burner/Darser: LL1 Parser Generator for D*. <https://github.com/burner/Darser>. (Accessed on 05/03/2019).

Encore

Reappearing UDA Pattern

```
1 struct Employee {
2     @GQLD(
3         Description("The social security number of an employee"),
4         Deprecated(IsDeprecated.yes, "To complex")
5     )
6     SocialSecurityNumber number;
7 }
```


Reappearing UDA Pattern

```
1  struct Employee {
2      @GQLD(
3          Description("The social security number of an employee"),
4          Deprecated(IsDeprecated.yes, "To complex")
5      )
6      SocialSecurityNumber number;
7  }

9  enum IsDeprecated {
10     undefined,
11     no,
12     yes
13 }

14
15 struct GQLDData {
16     Description desc;
17     Deprecated depre;
18 }
```

Reappearing UDA Pattern

```
1  struct GQLDData {
2      Description desc;
3      Deprecated depre;
4  }
5
6  GQLDData GQLD(Args...)(Args args) {
7      GQLDData ret;
8      static foreach(mem; __traits(allMembers, GQLDData)) {
9          static foreach(arg; args) {
10             static if(is(typeof(__traits(getMember, ret, mem)) ==
11                 typeof(arg)))
12                 {
13                     __traits(getMember, ret, mem) = arg;
14                 }
15             }
16         }
17     return ret;
18 }
```

Static Foreach Switch Case

```
1  Json ret = Json.emptyObject();
2  string typename = ...;
3  1: switch(typename) {
4      static foreach(type; collectTypes!(T)) {{
5          case typeToTypeName!(type): {
6              ret["data"] = typeToJson!(type)();
7              break 1;
8          }
9      }}
10     default: break;
11 }
12 return ret;
```

Collecting all Referenced Types

```
1  alias allTypes = collectTypes!Schema;
2
3  template collectTypesImpl(Type) {
4      import graphql.uda;
5      static if(is(Type : GraphQLCustomLeaf!F, F)) {
6          alias collectTypes Impl= AliasSeq!(Type);
7      } else static if(is(Type == interface)) {
8          alias RetTypes = AliasSeq!(collectReturnType!(Type,
9              __traits(allMembers, Type)));
10         alias ArgTypes = AliasSeq!(collectParameterTypes!(Type,
11             __traits(allMembers, Type)));
12         alias collectTypesImpl = AliasSeq!(Type, RetTypes,
13             ArgTypes, InterfacesTuple!Type);
14         ....
15     } else static if(is(Type == union)) {
16         alias collectTypesImpl = AliasSeq!(Type, InheritedClasses!Type);
17     } else static if(is(Type : Nullable!F, F)) {
18         alias collectTypesImpl = .collectTypesImpl!(F);
19     } ...
```

Type trinary expression

```
1  template InheritedClass(T) {
2      import std.meta : staticMap, AliasSeq, NoDuplicates;
3      import std.traits : Select;
4
5      alias getInheritedFields() = staticMap!(.InheritedClass, FieldTypeTuple!T);
6      alias ftt = Select!(is(T == union), getInheritedFields, AliasSeq);
7
8      alias getBaseTuple() = staticMap!(.InheritedClass, BaseClassesTuple!T);
9      alias cls = Select!(is(T == class), getBaseTuple, AliasSeq);
10
11     alias getInter() = staticMap!(.InheritedClass, InterfacesTuple!T);
12     alias inter = Select!(is(T == class) || is(T == interface),
13         getInter,
14         AliasSeq
15     );
16
17     alias InheritedClass = NoDuplicates!(AliasSeq!(ftt!(), cls!(), inter!()));
18 }
```

Compile-Time are long as

- \approx 7000 lines
- \approx 8 seconds build time

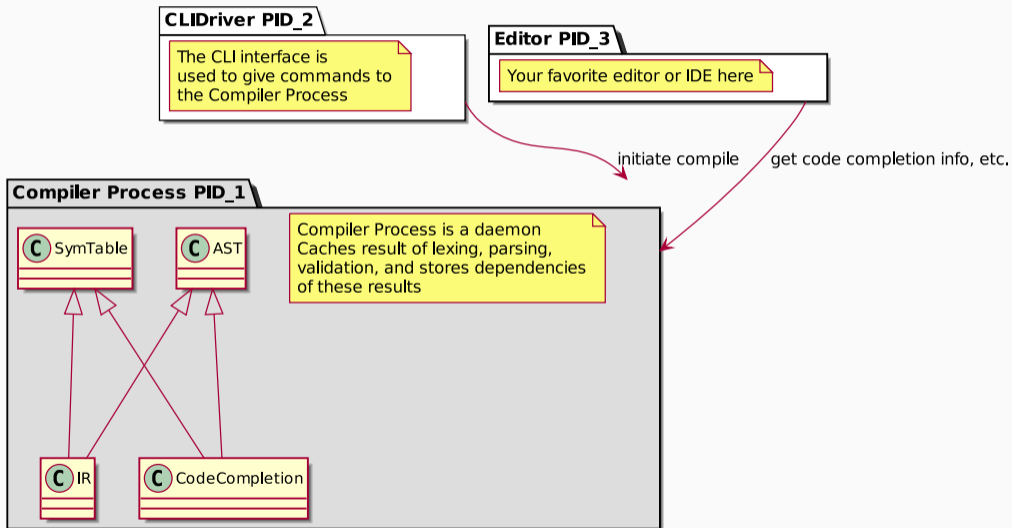
Overall Legacy Architecture

- Traditional compiler pipeline design is dated

Overall Legacy Architecture

- Traditional compiler pipeline design is dated
- We have practically unlimited memory
- Recreating the AST, IR, and ASM on every compile is extremely wasteful
- Why does code-completion and the compiler use different frontends

Compiler Re-arch idea



- Darser is a recursive descent parser generator for LL(1) grammars
- It also generates the AST and a default Visitor
- Not at CT, but as a pre-build step

- Darser is a recursive descent parser generator for LL(1) grammars
- It also generates the AST and a default Visitor
- Not at CT, but as a pre-build step

- It generates “good” error messages
- Not just a generic Node types, but names that reflect the grammar
- Inheriting from the default Visitor is trivial and powerful
- Used right now by GraphQLD

I am out of Slides
