


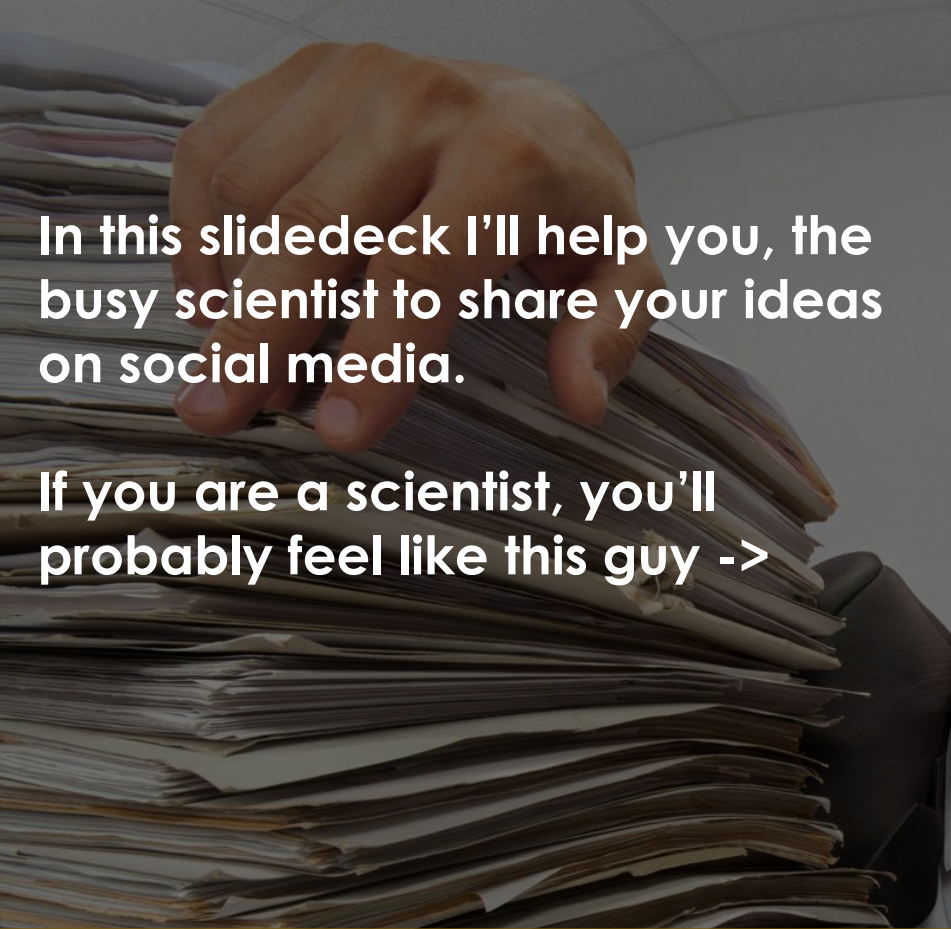


Social Media for the Busy Scientist Felienne Hermans (@felienne)



In this slidedeck I'll help you, the busy scientist to share your ideas on social media.

Social Media for the Busy Scientist Felienne Hermans (@felienne)




In this slidedeck I'll help you, the busy scientist to share your ideas on social media.

If you are a scientist, you'll probably feel like this guy ->



Social Media for the Busy Scientist Felienne Hermans (@felienne)



In this slidedeck I'll help you, the busy scientist to share your ideas on social media.

If you are a scientist, you'll probably feel like this guy ->

Papers, reviewing, grant proposals, teaching. No time for anything, let alone Twitter.

Social Media for the Busy Scientist

Felienne Hermans (@felienne)



Social Media for the Busy Scientist Felienne Hermans (@felienne)

**Felienne Hermans**[View my profile page](#)**5,116**

TWEETS

832

FOLLOWING

1,499

FOLLOWERS

Who to follow · Refresh · View all

**Joanneke v.d. Bosch** @Joanneke

Followed by Marina Noordegraaf ...

Follow

**Dmitri Petrov** @PetrovADmitri

Follow

**Eveline** @Evevreede

Followed by Tom Verhoeff and oth...

Follow

Popular accounts · Find friends

Trends · Change

Ideal

#vacature

#heartbreaker

#politie

#Amsterdam

Klopt

Goedemorgen

Bedankt

Tweets

**Frank Geerlings** @FrankGeerlings

2m

My 5 minutes of fame. Or as we say in dutch: 'Mijn 5 minutes of fame.' — our family on local TV news. at5.nl/artikelen/1122...

Expand

**NewRelic** @newrelic

23 Sep

Does a FREE 15" Mac Book Pro sound good to you? Only a few more days left to win one! Check it out: newrelic.com/lp/bundle?utm_...

Promoted by NewRelic

Followed by Thijs Kroesbergen, kate matsudaira and George Lee.

Expand

**Yanina Romero** @Yanina_Romero

3m

Feeling the iPhone 5Cs for the first time.. so playful! [#iPhone5C](#)

from Munich, Munich

**Paul Klint** @PaulKlint

4m

Wisdom of the crowd in practice: Gamers solve decade old HIV puzzle in ten days buff.ly/GGS01q

**Dirk Fahland** @dfahland

12m

Every* building in the Netherlands. Color coded by age. citysack.wag.org/buildings/#51... /centered on [#eindhoven](#)/via m.theatlanticcities.com/housing/2013/1...

Retweeted by Hajo Reijers

Expand

**Sam Aaron** @samaaron


7m

Here's a video of a talk I gave at AgileJKL in Finland this year entitled: HyperAgile Development: vimeo.com/73278809

Twitter is not the easiest social network to start with



**By the end of this deck, you'll know
3 easy ways to start on social media**

The background image shows three King penguins standing on a snowy, rocky terrain. The penguins are black and white with a distinctive yellow patch on their heads. They are facing right. The background features a range of snow-covered mountains under a grey sky. The image is split vertically: the left half is in full color, and the right half is in grayscale.

And the best thing about it these tips: you already did most of the work anyway.

**By the end of this deck, you'll know
3 easy ways to start on social media**

A woman in a grey dress is standing at the front of a room, gesturing with her hands while presenting to a group of people seated at tables. A large projection screen behind her displays a presentation slide. The room has a drop ceiling with fluorescent lights and a projector mounted on it. The audience is seen from behind, seated at long tables with water bottles and papers. The overall atmosphere is professional and educational.

Tech4Good CRM Software

What You'll See - Top Features - Core Values
How to Get Started - Getting Started - Getting Started

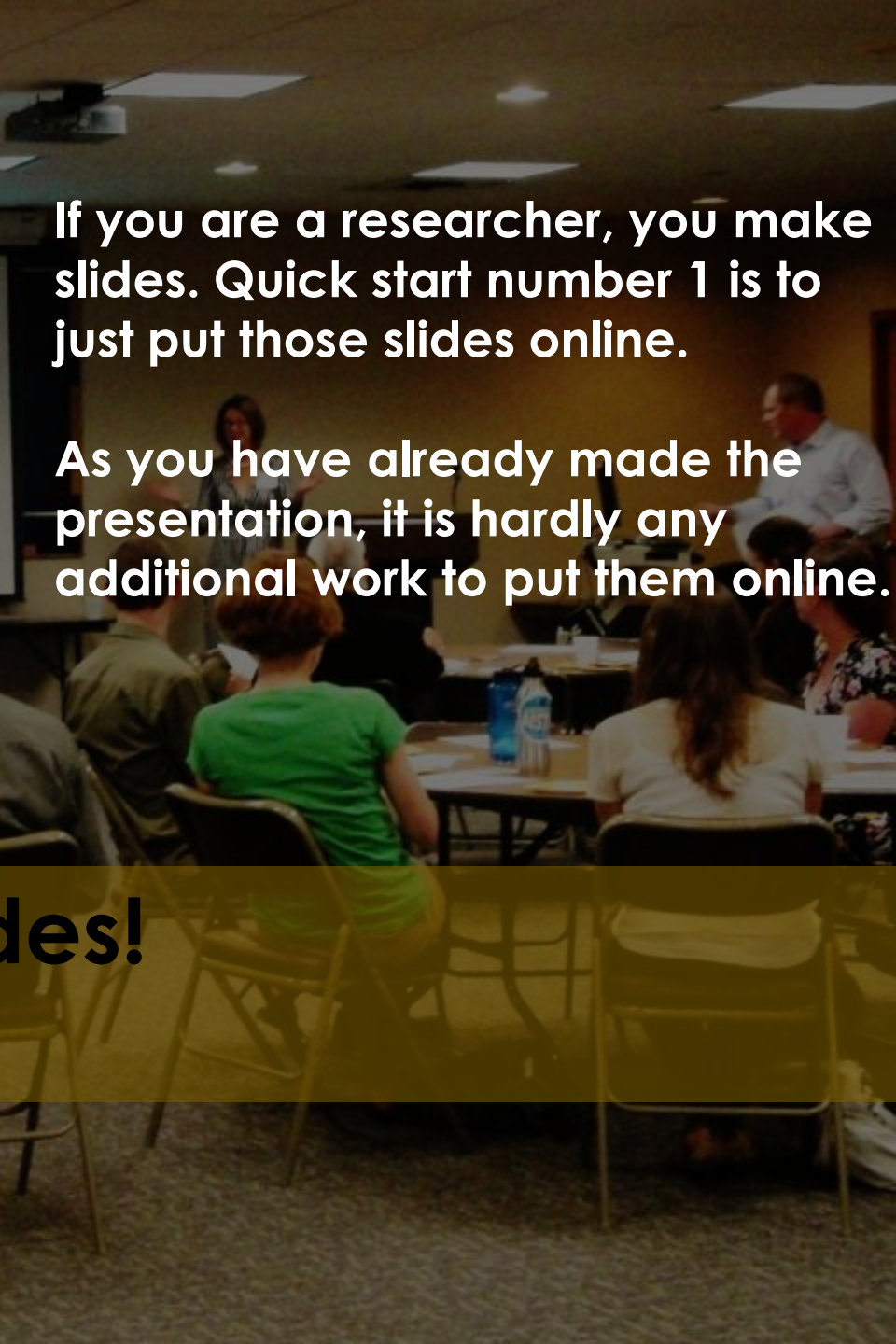
Quickstart 1) Slides!



If you are a researcher, you make slides. Quick start number 1 is to just put those slides online.

As you have already made the presentation, it is hardly any additional work to put them online.

Quickstart 1) Slides!



2001financialstatements.xlsx - Microsoft Excel

Home Insert Page Layout Formulas Data Review View Load Test Team

Clipboard Font Alignment Number Styles Cells Editing

AF52

	A	B	C	D	E	F	G	H	I	J	K	L
1	Consolidated Statements of Shareholders' Equity											
2	[DOLLARS IN THOUSANDS]											
3												
4												
5												
6												
7		Common Shares		Additional	Retained	Treasury	Comprehensive	Comprehensive	Accumulated			
8		Number	Par Value	Capital	Earnings	Shares	Income (Loss)	Income (Loss)	Other	Total		
9	Balance, January 1, 1999	69,494,483	\$ 86,868	\$ 43,281	\$ 604,227	\$ (21,902)		\$ (12,802)	\$ (549)	\$ 699,123		
10												
11	Net income				128,856		\$ 128,856			128,856		
12	Translation adjustment						9,558			9,558		
13	Pensions						614			614		
14	Unrealized loss on investment securities						(3,235)			(3,235)		
15	Other comprehensive income						6,937	6,937				
16	Comprehensive income						\$ 135,793					
17	Stock options exercised	108,104	134	1,918						2,052		
18	Unearned compensation	149,799	188	3,933					(3,485)	636		
19	Performance shares	20,397	25	686						712		
20	Procomp and Nexus adjustments	10,214	2,138	37,351		9,487				49,976		
21	Dividends declared and paid											
22	Treasury shares											
23												
24	Balance, December 31, 1999	71,482,997	\$ 89,354	\$ 87,169	\$ 691,416	\$ (13,614)	\$ (6,865)	\$ (4,034)	\$ 844,395			
25	Net income									136,919		
26	Translation adjustment									(7,904)		
27	Pensions									1,507		
28	Unrealized loss on investment securities									(396)		
29	Other comprehensive loss									(6,793)		
30	Comprehensive income						\$ 130,126					
31	Stock options exercised	273,238	343	5,444						5,787		
32	Unearned compensation	247,635	308	5,583					(3,915)	1,976		
33	Performance shares	15,335	19	334						353		
34	Dividends declared and paid				(44,271)					(44,271)		
35	Treasury shares					(2,300)				(2,300)		
36												
37	Balance, December 31, 2000	536,208	\$ 90,024	\$ 98,530	\$ 784,063	\$ (15,944)	\$ (12,658)	\$ (7,949)	\$ 936,066			
38	Net income				66,893		\$ 66,893			66,893		
39	Translation adjustment						(47,373)			(47,373)		
40	Pensions						(1,628)			(1,628)		
41	Unrealized gain on investment securities						1,213			1,213		
42	Other comprehensive loss						(47,788)	(47,788)				
43	Comprehensive income						\$ 19,105					
44	Stock options exercised	176,395	221	4,860						5,081		
45	Unearned compensation								1,412	1,412		
46	Dividends declared and paid				(45,774)					(45,774)		
47	Treasury shares					(12,780)				(12,780)		
48												
49	Balance, December 31, 2001	712,603	\$ 90,245	\$ 103,390	\$ 805,182	\$ (28,724)	\$ (60,446)	\$ (6,537)	\$ 903,110			
50												
51												

Analyzing & visualizing spreadsheets

Felienne Hermans (@felienne)

Here are the slides that summarize my PhD dissertation. I have put what I say during the presentation in a grey overlay (just as with this presentation)

financialstatements.xlsx - Microsoft Excel

Home Insert Page Layout Formulas Data Review View Load Test Team

Clipboard Font Alignment Number

Conditional Formatting Styles Cells Styles Cells Styles

Autosum Fill Sort & Find & Select

Clear

Editing

AF52

A B C D E

1 Consolidated Statements of Shareholders' Equity

2 (DOLLARS IN THOUSANDS)

3

4

5

6

7

8 Common Shares Additional Retained

9 Number Par Value Capital Earnings

10

11 Balance, January 1, 1999 69,494,483 \$ 86,868 \$ 43,281 \$ 604,227 \$ (21,902) \$ (1,002) \$ (549) \$ 899,123

12

13 Net income 128,856

14 Translation adjustment 9,558

15 Pensions 614

16 Unrealized loss on investment securities (3,235)

17 Other comprehensive income 6,937

18 Comprehensive income 135,793

19 Stock options exercised 108,104 134 1,918

20 Unearned compensation 149,799 188 3,933

21 Performance shares 20,397 26 600

22 Procomp and Nexus positions 210,214 2,138 37,351 9,487

23 Dividends declared and paid 13

24 Treasury shares 71,482,997 \$ 89,354 \$ 87,169 \$ 698,513 \$ (13,600) \$ (6,905) \$ (1,034) \$ 844,395

25 Balance, December 31, 1999

26 Net income 136,819

27 Translation adjustment (7,904)

28 Pensions 1,507

29 Unrealized loss on investment securities (200)

30 Other comprehensive income (6,793)

31 Comprehensive income 130,126

32 Stock options exercised 273,238 343 5,444

33 Unearned compensation 247,635 308 5,583

34 Performance shares 15,335 19 334

35 Dividends declared and paid (44,271)

36 Treasury shares (2,300)

37 Balance, December 31, 2000 536,208 \$ 90,024 \$ 98,530 \$ 784,063 \$ (15,944) \$ (12,658) \$ (7,949) \$ 936,066

38 Net income 66,893

39 Translation adjustment (47,373)

40 Pensions (1,628)

41 Unrealized gain on investment securities 1,213

42 Other comprehensive income (47,786)

43 Comprehensive income 19,105

44 Stock options exercised 176,395 221 4,860

45 Unearned compensation

46 Dividends declared and paid (45,774)

47 Treasury shares

48 Balance, December 31, 2001 712,603 \$ 90,245 \$ 103,390 \$ 805,182 \$ (28,724) \$ (60,446) \$ (6,537) \$ 83,110

49

50

51

In this slidedeck I present an overview of my PhD research. I recently defended my dissertation titled: Analyzing and visualizing spreadsheets'

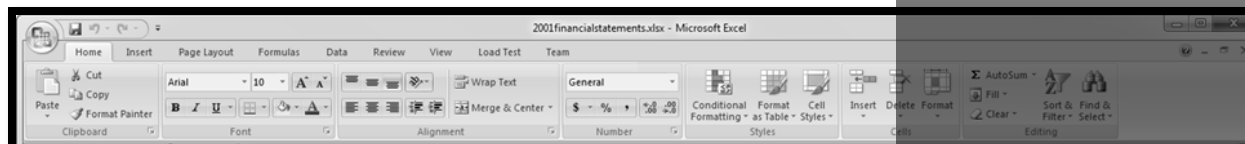
Analyzing & visualizing spreadsheets

Felienne Hermans (@felienne)

Here are the slides from my PhD dissertation.

Analyzing & visualizing spreadsheets

Here are the slides that summarize my PhD dissertation. I have put what I say during the presentation in a grey overlay (just as with this presentation)



1	Consolidated Statements of Shareholders'
2	(DOLLARS IN THOUSANDS)
3	
4	
5	
6	
7	
8	
9	Balance, January 1, 1999
10	
11	Net income
12	Translation adjustment
13	Pensions
14	Unrealized loss on investment securities
15	Other comprehensive income
16	Comprehensive income
17	Stock options exercised
18	Unearned compensation
19	Performance shares
20	Procomp and Nexus adjustments
21	Dividends declared and paid
22	Treasury shares
23	
24	Balance, December 31, 1999
25	Net income
26	Translation adjustment
27	Pensions
28	Unrealized loss on investment securities
29	Other comprehensive loss
30	Comprehensive income
31	Stock options exercised
32	Unearned compensation
33	Performance shares
34	Dividends declared and paid
35	Treasury shares
36	
37	Balance, December 31, 2000
38	Net income
39	Translation adjustment
40	Pensions
41	Unrealized gain on investment securities
42	Other comprehensive loss
43	Comprehensive income
44	Stock options exercised
45	Unearned compensation
46	Dividends declared and paid
47	Treasury shares
48	
49	Balance, December 31, 2001
50	
51	

1	Consolidated Statements of Shareholders'
2	(DOLLARS IN THOUSANDS)
3	
4	
5	
6	
7	
8	
9	Balance, January 1, 1999
10	
11	Net income
12	Translation adjustment
13	Pensions
14	Unrealized loss on investment securities
15	Other comprehensive income
16	Comprehensive income
17	Stock options exercised
18	Unearned compensation
19	Performance shares
20	Procomp and Nexus adjustments
21	Dividends declared and paid
22	Treasury shares
23	
24	Balance, December 31, 1999
25	Net income
26	Translation adjustment
27	Pensions
28	Unrealized loss on investment securities
29	Other comprehensive loss
30	Comprehensive income
31	Stock options exercised
32	Unearned compensation
33	Performance shares
34	Dividends declared and paid
35	Treasury shares
36	
37	Balance, December 31, 2000
38	Net income
39	Translation adjustment
40	Pensions
41	Unrealized gain on investment securities
42	Other comprehensive loss
43	Comprehensive income
44	Stock options exercised
45	Unearned compensation
46	Dividends declared and paid
47	Treasury shares
48	
49	Balance, December 31, 2001
50	
51	



Funny story: I wasn't hired to research spreadsheets at all. When I started my PhD project, I was supposed to research the gap between business users and programmers.

Programmers

Bridging the gap

Here are the slides that summarize my PhD dissertation. I have put what I say during the presentation in a grey overlay (just as with this presentation)

So now you might wonder: does this work?

[illegible]

This presentation has been viewed almost 7,500 times in 6 months.

7,427
views

1.144

3

63 <http://www.felienne.com>

5 <http://www.markpack.org>

[More...](#)

This presentation has been viewed almost 7,500 times in 6 months.

41
Like

67
Tweet

46
Share

+1

Pin it

WordPress

After you have added the notes, just upload the presentation to slideshare (or speakerdeck) and you are good to go.

So now you might wonder: does this work?

Analyzing & visualizing spreadsheets
Felienne Hermans (@felienne)

This presentation has been viewed almost 7,500 times in 6 months.

An overview of my PhD research
 by Felienne Hermans on Mar 25, 2013 [Edit](#)

7,427 views

Statistics		Actions		13 EMBEDS	
VIEW	Count	Action	Count	Embed URL	Count
Total Views	7,427	Likes	3	http://blogs.technet.com	339
Views on SlideShare	6,283	Downloads	63	http://www.felienne.com	263
Embed Views	1,144	Comments	5	http://www.markpack.or...	243

[More...](#)

So now you might wonder: does this work?

This presentation has been viewed almost 7,500 times in 6 months.

Also, many people have liked, tweeted and shared it.

After you have added the presentation to slideshare (or speakerdeck) you are good to go.

So now you might wonder how this work?

Analyzing & visualizing spreadsheets

Felienne Hermans (@felienne)

This presentation has been viewed almost 7,500 times in 6 months.

Also, many people have tweeted and shared it.

A photograph of King Willem-Alexander and Queen Maxima of the Netherlands during their coronation ceremony. King Willem-Alexander is on the left, wearing a black tuxedo with a white bow tie and a sash, and is raising his right hand in a gesture. Queen Maxima is on the right, wearing a blue dress with a sash and a tiara. The background is dark with gold lights.

So, you see: content is king!

**Don't worry about tweets and likes.
If you share nice content, 'social'
will follow.**

**<- this is our actual king, King
Willem-Alexander during his
coronation last April.**

Content is king

Introduction

1.1 Spreadsheets

Spreadsheets can be considered the most successful programming paradigm in the history of computers. End-user programmers outnumber software programmers [Sca05] and it has been estimated that 90% of computers have Excel installed [Bra09].

Their use is diverse, ranging from inventory administration to educational applications and from scientific modeling to financial systems, a domain in which their use is particularly prevailing. Panko [Pan06] estimates that 95% of U.S. firms, and 80% in Europe, use spreadsheets in some form for financial reporting. One of the success factors of spreadsheets is their easy-to-use interface and great flexibility. Users do not have to think about a design of their spreadsheet programs: they can just start entering data and formulas. However, there are some scenarios in which a spreadsheet user wants to have information about the underlying design of the spreadsheet. For instance, when debugging, when reading a spreadsheet created by someone else or when making big changes to the spreadsheets. Given the fact that the design is hidden 'behind' the spreadsheet and inaccessible for the user, difficulties can arise in those scenarios where a user does need to understand the spreadsheet's design.

This research aims at providing spreadsheet users with information about their spreadsheet's design, whenever they need and want that information.

1.2 A brief history of spreadsheets

Although the electronic spreadsheet was first conceived in the sixties, the idea of laying out numbers in a grid was far back as the Babylonian times. The oldest known example of a grid of numbers is a tablet from the 18th century BC, lists the Pythagorean triplets in Plimpton 322, a Babylonian tablet.

The basic quick start is just to upload your slides with notes.

But if you want to go a bit further, think about adding a storyline to your presentation. Always a good idea, but especially for slides you put online, it is useful to think about engaging your audience.

The slides are not a summary of my thesis.

Introduction

1.1 Spreadsheets

Spreadsheets can be considered the most successful programming paradigm in the history of computers. End-user programmers outnumber software programmers [Sca05] and it has been estimated that 90% of computers have Excel installed [Bra09].

Their use is diverse, ranging from inventory administration to educational applications and from scientific modeling to financial systems, a domain in which their use is particularly prevailing. Panko [Pan06] estimates that 95% of U.S. firms, and 80% in Europe, use spreadsheets in some form for financial reporting.

One of the success factors of spreadsheets is their easy-to-use interface and great flexibility. Users do not have to think about a design of their spreadsheet programs: they can just start entering data and formulas. However, there are some scenarios in which a spreadsheet user wants to have information about the underlying design of the spreadsheet. For instance, when debugging, when reading a spreadsheet created by someone else or when making big changes to the spreadsheets. Given the fact that the design is hidden 'behind' the spreadsheet and inaccessible for the user, difficulties can arise in those scenarios where a user does need to understand the spreadsheet's design.

This research aims at providing spreadsheet users with information about their spreadsheet's design, whenever they need and want that information.

1.2 A brief history of spreadsheets

Although the electronic spreadsheet was first conceived in the sixties, the idea of laying out numbers in a grid goes as far back as the Babylonian times. The Plimpton 322, a Babylonian tablet, lists the Pythagorean triplets in

Look at the difference between
the first page of my dissertation



The slides are not a summary of my
thesis.

Look at the difference between
the first page of my dissertation

And the first slide of my
presentation →

Introduction

1.1 The history of spreadsheets

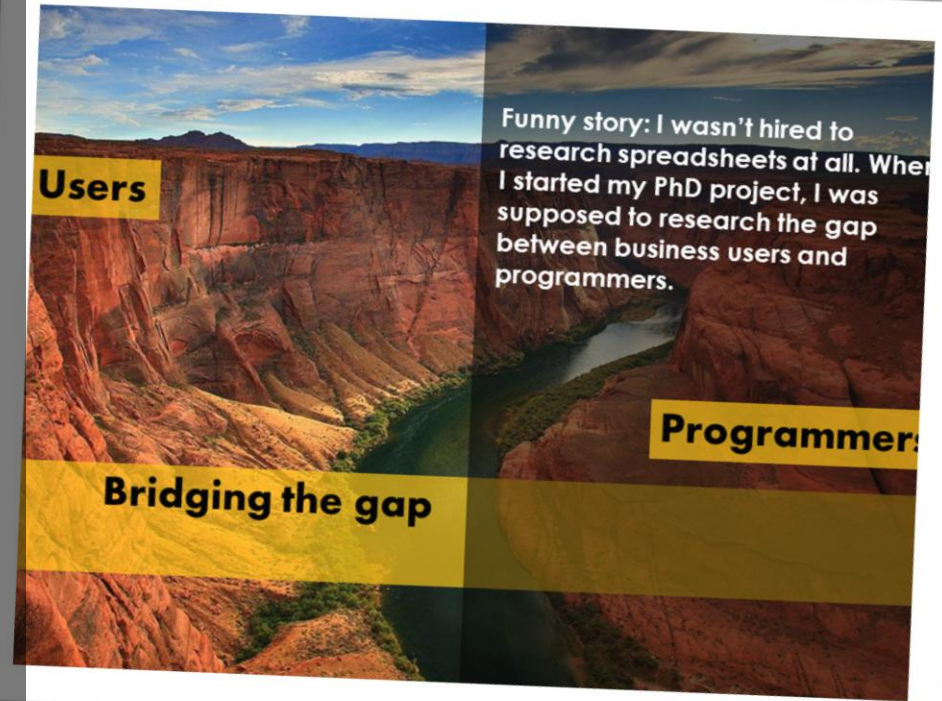
Spreadsheets can be considered the most successful programming paradigm in the history of computers. End-user programmers outnumber software programmers [Sca05] and it has been estimated that 90% of computers have Excel installed [Bra09].

Their use is diverse, ranging from inventory administration to educational applications and from scientific modeling to financial systems, a domain in which their use is particularly prevailing. Panko [Pan06] estimates that 95% of U.S. firms, and 80% in Europe, use spreadsheets in some form for financial reporting. One of the success factors of spreadsheets is their easy-to-use interface and great flexibility. Users do not have to think about a design of their spreadsheet programs: they can just start entering data and formulas. However, there are some scenarios in which a spreadsheet user wants to have information about the underlying design of the spreadsheet. For instance, when debugging, when reading a spreadsheet created by someone else or when making big changes to the spreadsheets. Given the fact that the design is hidden 'behind' the spreadsheet and inaccessible for the user, difficulties can arise in those scenarios where a user does need to understand the spreadsheet's design.

This research aims at providing spreadsheet users with information about their spreadsheet's design, whenever they need and want that information.

1.2 A brief history of spreadsheets

Although the electronic spreadsheet was first conceived in the sixties, the idea of laying out numbers in a grid as far back as the Babylonian times. The Plimpton 322, a Babylonian tablet, lists the Pythagorean triplets in



The slides are not a summary of my
thesis.

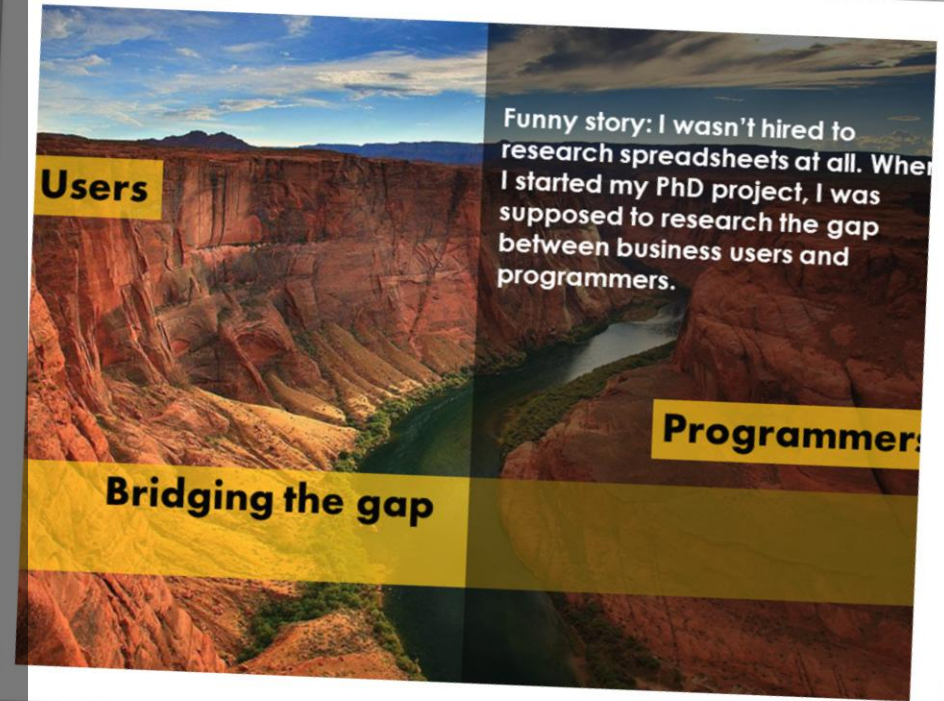
Look at the difference between
the first page of my dissertation

And the first slide of my
presentation.

The first slide tells a story: 'I wasn't
hired to work on spreadsheets'

It makes you wonder how on
earth I ended up writing a
dissertation on them.

The slides are not a summary of my
thesis.






Tip 1: story, story, story



Quickstart 2) Progress reports



Quickstart 2 concerns progress reports. If you are involved in any projects, you'll probably have to write progress reports (that no one reads...)

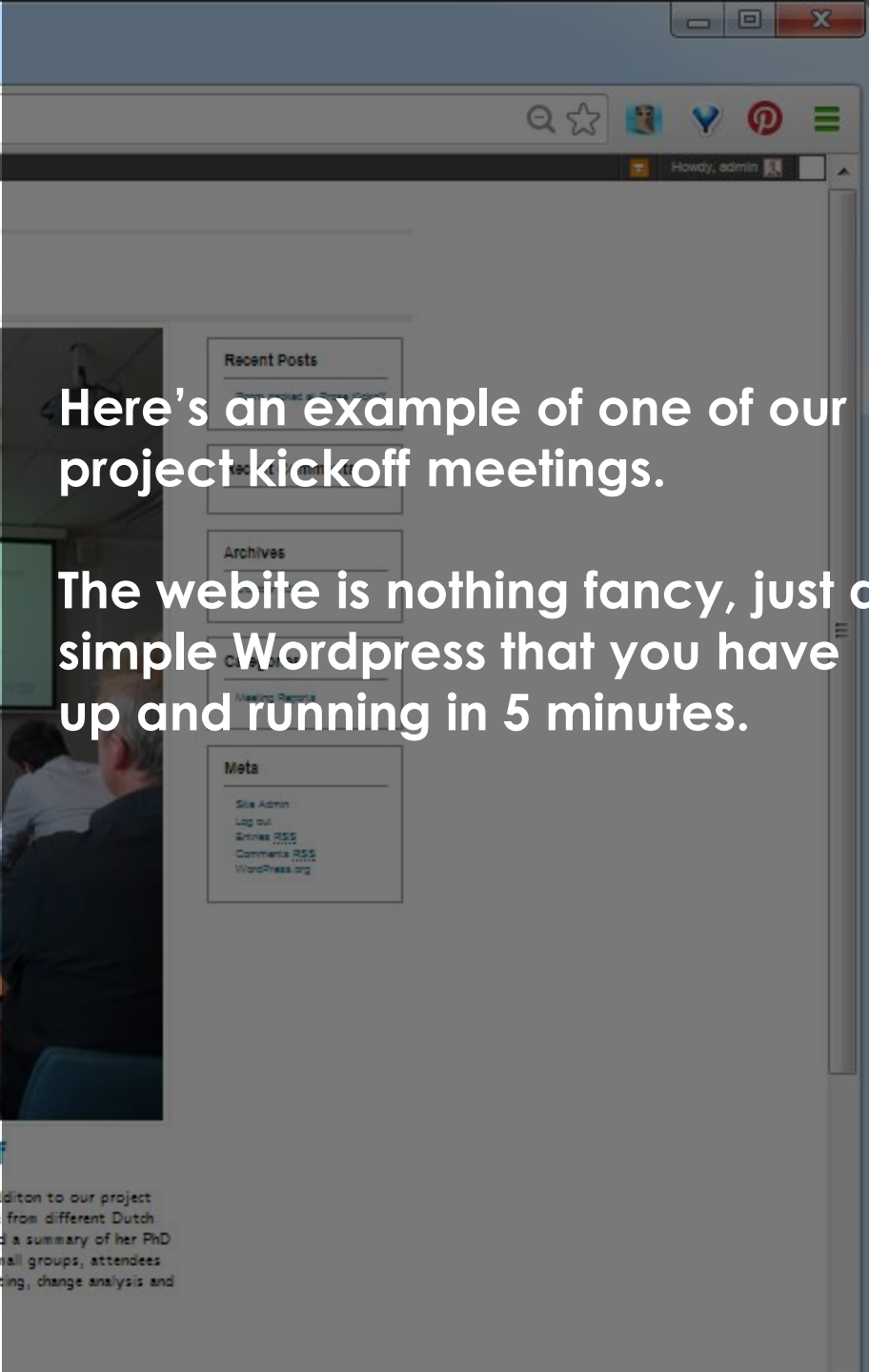
Put them online to make writing more fun, this makes awesome social content to keep your fans updated.

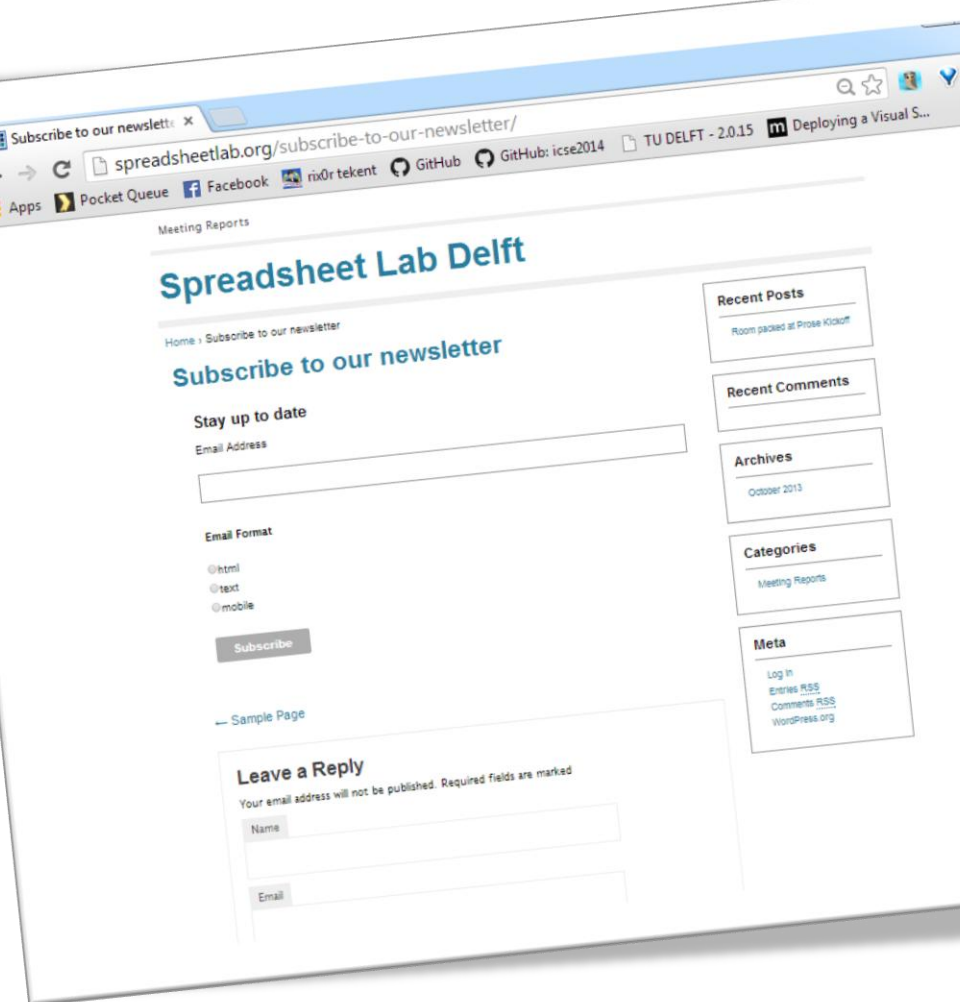
Quickstart 2) Progress reports



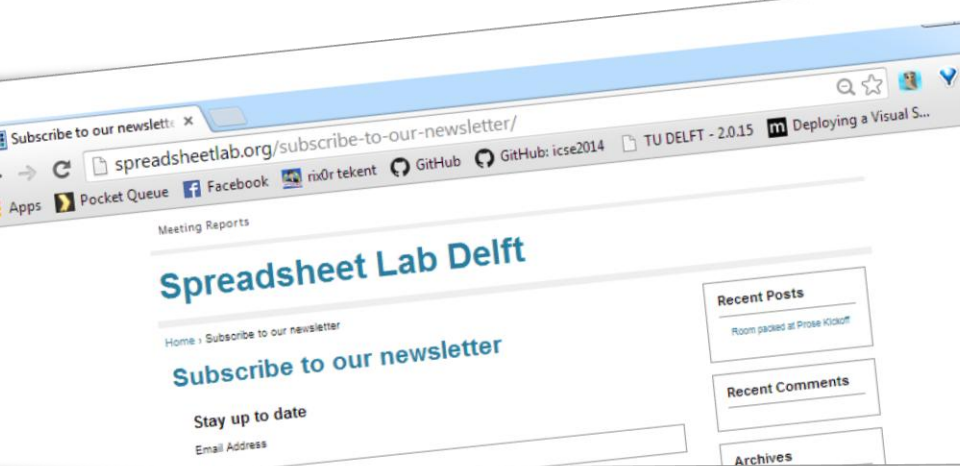
Here's an example of one of our project kickoff meetings.

The website is nothing fancy, just a simple Wordpress that you have up and running in 5 minutes.

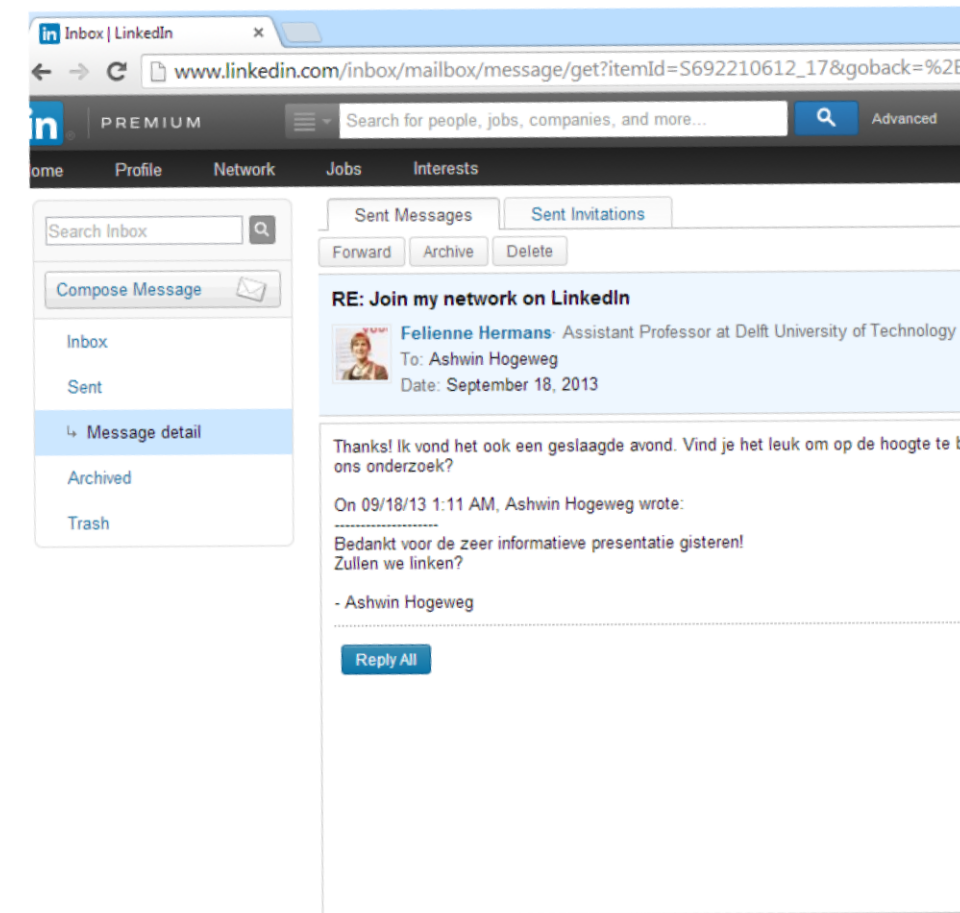




The next step is to build an audience for these reports. You can create a newsletter (mailchimp is nice for this) and have people subscribe on your website.



The next step is to build an audience for these reports. You can create a newsletter (mailchimp is nice for this) and have people subscribe on your website.



LinkedIn is also great for building an audience. If people connect with you, ask them whether they want to receive updates on your research and place them on your mailing list too.



Tip 2) Build an audience

Detecting Code Smells in Spreadsheet Formulas

Felienne Hermans, Martin Pinzger and Arie van Deursen
Delft University of Technology
Delft, the Netherlands
{f.f.j.hermans, m.pinzger, arie.vandeursen}@tudelft.nl

Abstract—Spreadsheets are used extensively in business processes around the world and just like software, spreadsheets are changed throughout their lifetime causing maintainability issues. This paper adapts known *code smells* to spreadsheet formulas. To that end we present a list of metrics by which we can detect *smelly* formulas and a visualization technique to highlight these formulas in spreadsheets. We implemented the metrics and visualization technique in a prototype tool to evaluate our approach in two ways. Firstly, we analyze the Euses spreadsheet corpus, to study the occurrence of the formula smells. Secondly, we analyze ten real life spreadsheets, and interview the spreadsheet owners about the identified smells. The results of these evaluations indicate that formula smells are common and that they can reveal real errors and weaknesses in spreadsheet formulas.

Keywords—spreadsheets; code smells; refactoring;

I. INTRODUCTION

The use of spreadsheets is very common in industry. Winston [1] estimates that 90% of all analysts in industry perform calculations in spreadsheets. Spreadsheet developers are in fact end-user programmers that are usually not formally trained software engineers. There are many of those end-user programmers, more than there are traditional programmers, and the artifacts they create can be just as important to an organization as regular software. Technically speaking, spreadsheets and source code are similar. Both consist of formulas, which are pieces of source code, since both consist of constants, variables, conditional statements and references to other parts of the software. It therefore seems logical to research what principles from software engineering are also applicable to spreadsheets.

In previous work [2] we have defined code smells between worksheets, such as high coupling between worksheets and *middle men* worksheets. The evaluation of those smells showed that they can indeed reveal weak spots in a spreadsheet's design. In this paper we follow that line of thought, but focus our attention on smells within spreadsheet formulas. To that end we present a set of *formula smells*, based on Fowler's code smells. We subsequently define metrics for each of the formula smells, to enable the automatic detection of the smells. We then describe a method to detect these formula smells. Our detection approach uses thresholds to divide the severeness of each formula smell into low, moderate, and high. The thresholds are based on

Thereon we address the issue of communicating identified smells to spreadsheet users. We choose to do this within the spreadsheet itself, with a spreadsheet *risk map*, a colored overlay on the spreadsheet, indicating risk in the spreadsheet formulas. Finally we evaluate the catalog of smells in two ways, with a quantitative and qualitative evaluation. We perform a quantitative evaluation on the Euses spreadsheet corpus. The qualitative analysis was performed with ten real life spreadsheets and their developers from industry. With both studies we aim to answer the three research questions: R_1 What formula smells are most common, and why? R_2 To what extent do formula smells expose threats to spreadsheet quality? R_3 To what extent are risk maps an appropriate way to visualize formula smells?

The results of these evaluations show that formula smells can indeed reveal weaknesses, and even find real mistakes in a spreadsheet. The risk maps, although not yet perfect, are a good aid in helping to locate and understand formula smells.

II. FORMULA SMELLS

We define a number of *formula smells*, based on the existing work in the field of source code smells, initiated by Fowler [4]. Smells in source code indicate suspicious code, where the developer might want to refactor to improve the code. We define formula smells, which aim to minimize the chance of future errors. Formula smells are inspired by source code smells: they indicate formulas that are suspicious, not easy to read or error-prone. In the following we present our set of formula smells plus ways to refactor them.

A. Multiple Operations

One of the most well-known code smells is the Long Method. Inspired by this code smell, we define the formula smell *Multiple Operations*. Analogous to a long method, a formula with many different operations will likely be harder to understand than a shorter one. Especially since in most spreadsheet programs, there is limited space to view a formula, causing long formulas to be cut off.

A corresponding refactoring is the division of the Multiple Operations over multiple cells in a spreadsheet. For instance, instead of putting $SUM(A1:A6)*(B1+8)/100$ in one cell, this could be split into two cells, one for the SUM, and one for

Quickstart 3) Papers

Detecting Code Smells in Spreadsheet Formulas

Felienne Hermans, Martin Pinzger and Arie van Deursen
Delft University of Technology
Delft, the Netherlands
{f.f.j.hermans, m.pinzger, arie.vandeursen}@tudelft.nl

Abstract—Spreadsheets are used extensively in business processes around the world and just like software, spreadsheets are changed throughout their lifetime causing maintainability issues. This paper adapts known *code smells* to spreadsheet formulas. To that end we present a list of metrics by which we can detect *smelly* formulas and a visualization technique to highlight these formulas in spreadsheets. We implemented the metrics and visualization technique in a prototype tool to evaluate our approach in two ways. Firstly, we analyze the Euses spreadsheet corpus, to study the occurrence of the formula smells. Secondly, we analyze ten real life spreadsheets, and interview the spreadsheet owners about the identified smells. The results of these evaluations indicate that formula smells are common and that they can reveal real errors and weaknesses in spreadsheet formulas.

Keywords—spreadsheets; code smells; refactoring;

I. INTRODUCTION

The use of spreadsheets is very common in industry. Winston [1] estimates that 90% of all analysts in industry perform calculations in spreadsheets. Spreadsheet developers are in fact end-user programmers that are usually not formally trained software engineers. There are many of those end-user programmers, more than there are traditional programmers, and the artifacts they create can be just as important to an organization as regular software. Technically speaking, spreadsheets and software source code are very similar. Both consist of formulas, while pieces of source code, since both consist of constants, variables, conditional statements and references to other parts of the software. It therefore seems logical to research what principles from software engineering are also applicable to spreadsheets.

In previous work [2] we have defined code smells between worksheets, such as high coupling between worksheets and *middle men* worksheets. The evaluation of those smells showed that they can indeed reveal weak spots in a spreadsheet's design. In this paper we follow that line of thought, but focus our attention on smells within spreadsheet formulas. To that end we present a set of *formula smells*, based on Fowler's code smells. We subsequently define metrics for each of the formula smells, to enable the automatic detection of the smells. We then describe a method to detect these formula smells. Our detection approach uses thresholds to divide the severeness of each formula smell into low, moderate, and high. The thresholds are based on

Thereon we present a set of *code smells* to provide code smells to spreadsheet users. We choose to do this within the spreadsheet itself, with a spreadsheet *risk map*, a colored overlay indicating the severity of the identified spreadsheet formulas. Finally we evaluate the catalog of smells in two ways, with a quantitative and qualitative evaluation. We perform a quantitative evaluation on the Euses spreadsheet corpus. The qualitative analysis was performed with ten real life spreadsheets and their developers from industry. With both studies we aim to answer the three research questions: R_1 What formula smells are most common, and why? R_2 To what extent do formula smells expose threats to spreadsheet quality? R_3 To what extent are risk maps an appropriate way to visualize formula smells?

The results of these evaluations show that formula smells can indeed reveal weaknesses, and even find real mistakes in a spreadsheet. The risk maps, although not yet perfect, are a good aid in helping to locate and understand formula smells.

II. FORMULA SMELLS

We define a number of *formula smells*, based on the existing work in the field of source code smells, initiated by Fowler [4]. Smells in source code indicate suspicious code that the developer might want to refactor to improve code quality and minimize the chance of future errors. Formula smells are inspired by source code smells: they indicate formulas that are suspicious, not easy to read or error-prone. In the following we present our set of formula smells plus ways to refactor them.

A. Multiple Operations

One of the most well-known code smells is the Long Method. Inspired by this code smell, we define the formula smell *Multiple Operations*. Analogous to a long method, a formula with many different operations will likely be harder to understand than a shorter one. Especially since in most spreadsheet programs, there is limited space to view a formula, causing long formulas to be cut off.

A corresponding refactoring is the division of the Multiple Operations over multiple cells in a spreadsheet. For instance, instead of putting $SUM(A1:A6)*(B1+8)/100$ in one cell, this could be split into two cells, one for the SUM, and one for

The third easy quickstart are papers. You are writing them anyway.

Quickstart 3) Papers

Felienne's Blog

Meta

Site Admin

Log out

ABOUT

COMMITTEES

PUBLICATIONS

LIVE BLOG

PRESS

BLOG

Detecting Code Smells in Spreadsheet Formulas

Felienne · June 27, 2012 · Publications · 550

We just learned this paper is accepted at ICSM 2012.

The idea

The basic idea of this paper is to investigate whether we could apply the known code smells invented by Martin Fowler in his book [Refactoring](#) to spreadsheet formulas. You could view this as a piece of code, so it makes sense to assume this can 'smell' just as code can.

The smells

But, not all smells on his catalog are applicable to spreadsheets. Some, like Refuse to inherit, do with inheritance, and that concept does not really exist in spreadsheets. And in a previous paper, we have already investigated smells between worksheets. This leaves us with the following smells that could occur in formulas:

1. Multiple Operations Analogous to a long method, a formula with many different operations will likely be harder to understand than a shorter one. Especially since in most spreadsheet programs, there is limited space to view a formula, causing long formulas to be cut off.
2. Multiple References Another code smell we use as a basis is the Many parameters code smell. A method that uses many input values might be split into multiple methods to improve readability. The formula equivalent of this smell occurs when a formula references many different other cells, like `SUM(A1:A5; B7:C18; C19:F19)`
3. Conditional Complexity Fowler states that the nesting of many conditional operations should be considered a threat to code readability, since conditionals are hard to read. Since spreadsheet systems also allow for the use of conditional, spreadsheet formulas are at risk of this trait too.
4. Long Calculation Chain Spreadsheet formulas can refer to each other, hence creating chains of calculation. To understand the meaning of such a formula, a spreadsheet user has to trace

The third easy quickstart are papers. You are writing them anyway.

Just put a summary on your webpage and they are way more accessible than they are now. Even if you have a pdf on your website, a post is better, as it is easier to share and easier to read on mobile devices.

Detecting Code Smells in Spreadsheet Formulas

Felienne June 21, 2012 Publications Edit

We just learned this paper is accepted at ICSM 2012.

The idea

The basic idea of this paper is to investigate whether we could apply the known code smells invented by Martin Fowler in his book [Refactoring](#) to spreadsheet formulas. You could view a formula as a small piece of code, so it makes sense to assume this can 'smell' just as code can.

The smells

But, not all smells on his catalog are applicable to spreadsheets. Some, like Refused Bequest, has to do with inheritance, and that concept does not really exist in spreadsheets. And in a [previous paper](#), we have already investigated smells between worksheets. This leaves us with the following smells that could occur in formulas:

1. **Multiple Operations** Analogous to a long method, a formula with many different operations will likely be harder to understand than a shorter one. Especially since in most spreadsheet programs, there is limited space to view a formula, causing long formulas to be cut off.
2. **Multiple References** Another code smell we use as a basis is the Many parameters code smell. A method that uses many input values might be split into multiple methods to improve readability. The formula equivalent of this smell occurs when a formula references many different other cells, like `SUM(A1:A5; B7;C18;C19;F19)`
3. **Conditional Complexity** Fowler states that the nesting of many conditional operations should be considered a threat to code readability, since conditionals are hard to read. Since spreadsheet systems also allow for the use of conditional, spreadsheet formulas are at risk of this treat too.
4. **Long Calculation Chain** Spreadsheet formulas can refer to each other, hence creating chains of calculation. To understand the meaning of such a formula, a spreadsheet user has to trace along multiple steps to find the origin of the data. This is a task that spreadsheet users find tedious.
5. **Duplicated Formula** This smell indicates that similar snippets of code are used throughout a class. This is a concept common in spreadsheets too, where some formulas are partly the same as others.

Here is the first paper I put on my website, in the summer of 2012.

Sidenote: As you can see, I am also new to this, we all are.

Detecting Code Smells in Spreadsheet

Formulas

Felienne June 21, 2012

We just learned this paper is acc

The idea

The basic idea of this paper is to Martin Fowler in his book *Refac* code, so it makes sense to assure

The smells

But, not all smells on his catalog with inheritance, and that conce already investigated smells betw formulas:

1. Multiple Operations Analog be harder to understand th limited space to view a form
2. Multiple References Another that uses many input value equivalent of this smell occ B7;C18;C19;F19)
3. Conditional Complexity For considered a threat to code also allow for the use of co
4. Long Calculation Chain Spr calculation. To understand along multiple steps to find
5. Duplicated Formula This sm is a concept common in spr

Measuring the smells

Now we have the smells, we need to come up with a way to measure them, in order to detect what cells are smelly. We did that by analyzing a large body of spreadsheets (the EUSES corpus) and setting smelliness thresholds based on the values in that set. We used 70, 80 and 90% as thresholds, since these are values that are common in source code smells. In the evaluation we validated whether those thresholds made sense.

Evaluation

For the evaluation, first we analyzed the EUSES corpus a second time, to check the thresholds. We set them based on the formulas, and we wanted to know how many spreadsheets were smelly, given the proposed thresholds. This is important, since we would not want to be in a situation where all, or no spreadsheets were smelly, that would mean we had to adjust the thresholds. Luckily, the results were promising, not too many smelly spreadsheet, but also not too few:

Smell	> 70%	> 80%	> 90%
Multiple References	23.8%	18.4%	6.3%
Multiple Operations	21.6%	17.1%	6.3%
Duplication	10.8%	7.1%	3.7%
Long Calculation Chain	9.0%	7.9%	3.3%
Conditional Complexity	4.4%		
Any of the above smells	42.7%	31.4%	19.7%

Table II
PERCENTAGE OF SPREADSHEETS IN THE EUSES CORPUS THAT WERE SMELLY, BASED ON THE PROPOSED THRESHOLDS, FROM AT LEAST ONE OF THE FIVE SPREADSHEET SMELLS IN THE EUSES CORPUS, FOR THE THREE THRESHOLDS

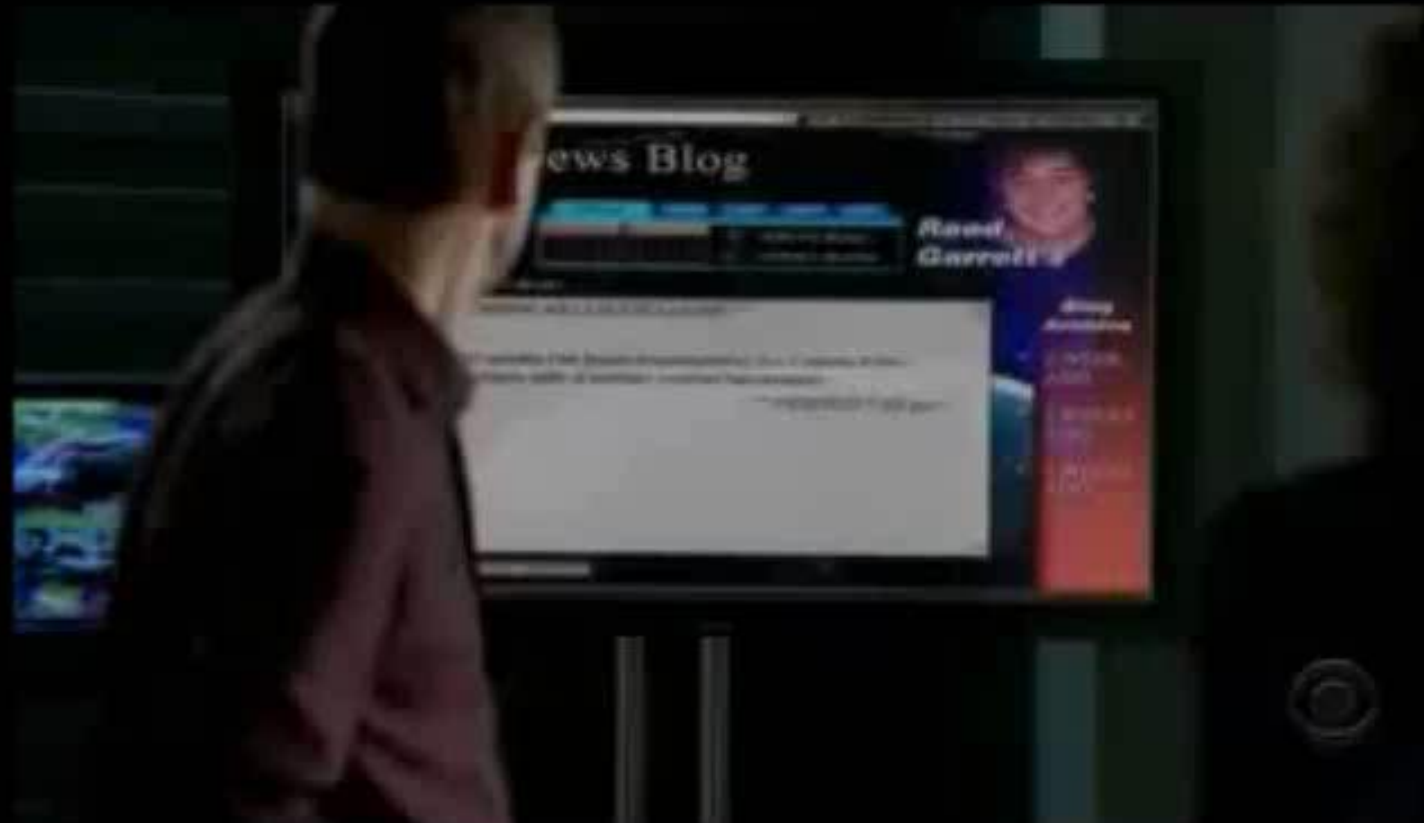
Then, we went onto the second evaluation, in which we analyzed the cell dependencies and the complexity of their owners. In this evaluation we learned that there is little awareness of the risks of complicated formulas (like Multiple Operations). Spreadsheet users seem not to be concerned too much about maintainability of formulas. They keep extending formulas with more operations and more references, causing formulas to become long and complicated.

Furthermore, we found two actual faults in a spreadsheet by looking at the Duplication Smell. With respect to the other smells, the concern caught is lack of understandability. Spreadsheet users found that our current smell detection strategies reveal the formulas that are the least maintainable. These formulas will be time consuming to change, and changes made will be more error prone.

Here is the first paper I put on my website, in the summer of 2012.

Sidenote: As you can see, I am also new to this, we all are.

As you can also see, this post really looks like a paper still, is even has a table. Not really for the general public yet.



Let me remind you what the general public knows about our field



**If you missed it for a lack of video,
this is what they say.**

This is in real-time



So far, so good

I'll create a
GUI interface using
visual basic to track
the killer's IP address

This is in real-time



OMG are you serious???

I'll create a
GUI interface using
visual basic to track
the killer's IP address

This is in real-time



Tip 3) Simple is better

Fellienne's Blog

Meta

Site Admin

Log out

ABOUT

COMMITTEES

PUBLICATIONS

LIVE BLOG

PRESS

BLOG

New paper: BumbleBee, a tool for spreadsheet formula transformations

Fellienne October 1, 2013 Publications 536

Some spreadsheets can be improved

While looking at spreadsheet and how they are used, over the past years I have noticed that many users don't make their spreadsheets as easy as they could be. For instance, they use $A1+A2+A3+A4+A5$ instead of the simpler $SUM(A1:A5)$. Sometimes because they are unaware of a simpler construct, or because the spreadsheet evolved over time. For instance, in used to be $A1+A2$, then $A3$ was added and so forth. Such complex formulas were exactly the aim of our previous work on smell detection.

If you say smell, you say.... refactorings!

So in order to improve spreadsheets, we and other researchers have developed a number of refactorings to improve spreadsheet formulas. Over the last few months, I have been working on BumbleBee, a tool to perform not only refactorings, but more general transformations on spreadsheet formulas.



About Fellienne

Daytime: researching end-user programming
Weekends: running, gaming, PGL volunteering

On this blog I share my papers and thoughts and occasionally I live blog events that I attend.

[Read more here](#)

Recent Comments

Martin Rogala on On bad arguments and stereotypes

Birthday Madness! - A weird link compilation | Instagram on Excel Turing Machine

Infosecurity US - Turing's Excel on Excel Turing Machine

What is the type of an anonymous type? | My .NET Corner on Dynamic typing in C#

Dr. Evgenii Kolosov on The missing link between industry and academia

Tags

academia Andre Alexandrescu

Berovo 2012 Best practices

Blog Bugs C#

clone detection code clones

Google Analytics

New paper: BumbleBee, a tool for spreadsheet formula transformations

Felienne October 3, 2013 Publications Edit

Some spreadsheets can be improved

While looking at spreadsheet and how they are used, over the past years I have noticed that many users don't make their spreadsheets as easy as they could be. For instance, they use $A1+A2+A3+A4+A5$ instead of the simpler $SUM(A1:A5)$. Sometimes because they are unaware of a simpler construct, or because the spreadsheet evolved over time. For instance, it used to be $A1+A2$, then $A3$ was added and so forth. Such complex formulas were exactly the aim of our [previous work](#) on smell detection.

If you say smell, you say... refactorings!

So in order to improve spreadsheets, we and other researchers have developed a number of refactorings to improve spreadsheet formulas. Over the last few months, I have been working on BumbleBee, a tool to perform not only refactorings, but more general transformations on spreadsheet formulas.

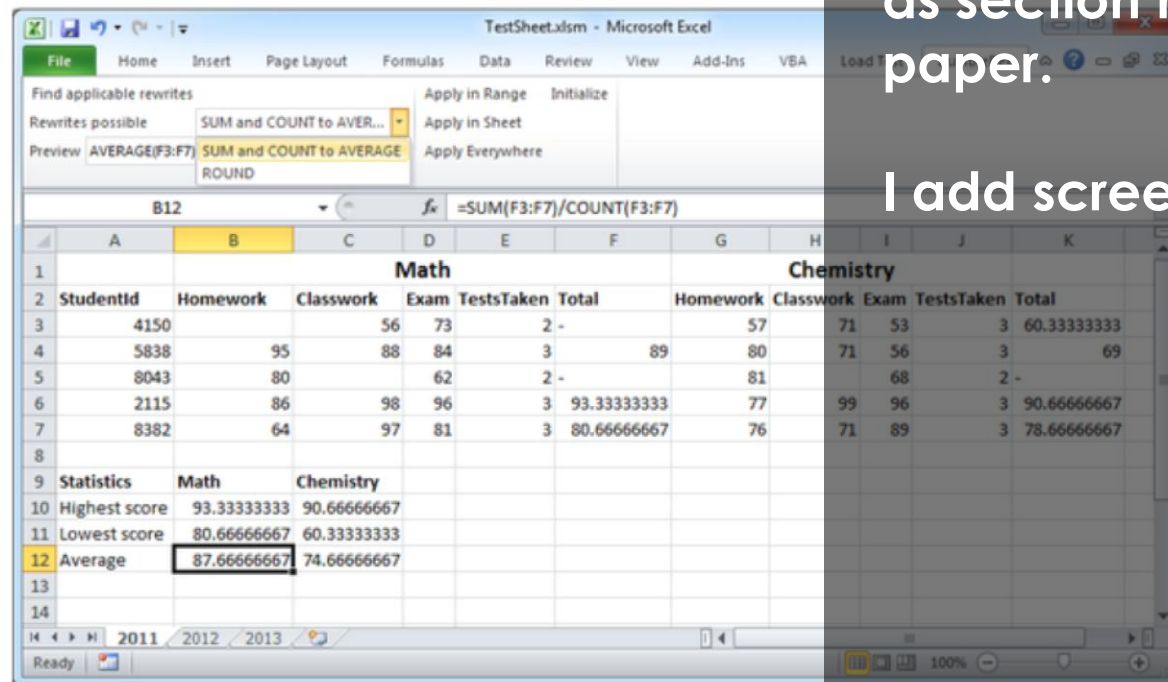
This is how my newest paper blog post looks like. Images, tag lines as section heads. Nothing like a paper.



above, you can perform transformations on spreadsheet formulas.

Spreadsheet refactoring with BumbleBee

In the add-in interface, shown below, a user can find transformations that can be applied on the current formula with the 'Find applicable rewrites' button. Once they are shown in the dropdown list, a user can select one. The result of this transformation is then shown in the preview box. In the example below, the SUM and COUNT to AVERAGE transformation is shown and B12, with `=SUM(F3:F7)/COUNT(F3:F7)` is selected, so the preview box shows `AVERAGE(F3:F7)`. The user can now apply this transformation in any selected range, sheet or in the entire workbook.



The screenshot shows the BumbleBee add-in interface in Microsoft Excel. The interface has a menu bar with options: File, Home, Insert, Page Layout, Formulas, Data, Review, View, Add-Ins, and VBA. Below the menu bar, there is a section for 'Find applicable rewrites'. It includes a dropdown menu for 'Rewrites possible' showing 'SUM and COUNT to AVERAGE...', a preview box showing 'AVERAGE(F3:F7)', and buttons for 'Apply in Range', 'Initialize', 'Apply in Sheet', and 'Apply Everywhere'. The main spreadsheet area shows a table with student scores for Math and Chemistry. The table has columns for StudentId, Homework, Classwork, Exam, TestsTaken, and Total. The summary row (row 12) shows the average score for each subject: 87.66666667 for Math and 74.66666667 for Chemistry.

	Math	Chemistry
StudentId	Homework	Classwork
4150	56	71
5838	88	71
8043	62	68
2115	96	99
8382	81	71
Average	87.66666667	74.66666667

This is how my newest paper blog post looks like. Images, tag lines as section heads. Nothing like a paper.

I add screenshots

A few refactorings come pre-loaded with BumbleBee, but you as a user can use our BumbleBee language to express your own transformation of choice.

above, you can perform transformations on spreadsheet formulas.

Spreadsheet

In the

applied

are some

transformations

COUNT

=SUM

user

work

In our example spreadsheet for instance, students now need to take all three tests in order to get a grade. Suppose we would loosen that into at least 2 tests, then the formula above needs to change into $\text{IF}(\text{E3} \leq 2, \text{SUM}(\text{B3:D3})/\text{E3}, "-")$

For this, we cannot simply use search and replace, as we have to replace 3 with a different cell each time (on row 4, it will be E4 etc) This is where BumbleBee comes in. With the transformation

$\text{IF}(\{i+3,j\}=3, \text{SUM}(\{i,j\}:\{i+2,j\})/3, "-") \leftrightarrow \text{IF}(\{i+3,j\} \leq 2, \text{SUM}(\{i,j\}:\{i+2,j\})/\{i+3,j\}, "-")$ perform this change everywhere.

That last transformation does not look like something a user would do.

You are right, in the case of changing business rules, the transformation can become complex quickly. This is why, in future work, we plan to generate these transformations based on edits by the user. We envision a BumbleBee 2.0 that can ask a user that just updated a formula, whether he wants to perform the same edit throughout the spreadsheet.

That looks awesome, can I have it?

Absolutely. [Here](#) is an installer for Excel 2010. By adding your own transformations to the worksheet 'Transformations' and hitting 'Initialize' you can create your own rules and play around with them. [This](#) spreadsheet contains all transformations needed to migrate to Excel 2010 and [this one](#) contains common (behavior preserving) refactorings.

If you have questions, drop me a line or post below. If you want to know more, here is a [preprint](#) of our paper about BumbleBee.

A few refactorings come pre-loaded with BumbleBee, but you as a user can use our BumbleBee language to express your own transformation of choice.

This is how my newest paper blog post looks like. Images, tag lines as section heads. Nothing like a paper.

I add screenshots

And communicate with the reader directly.

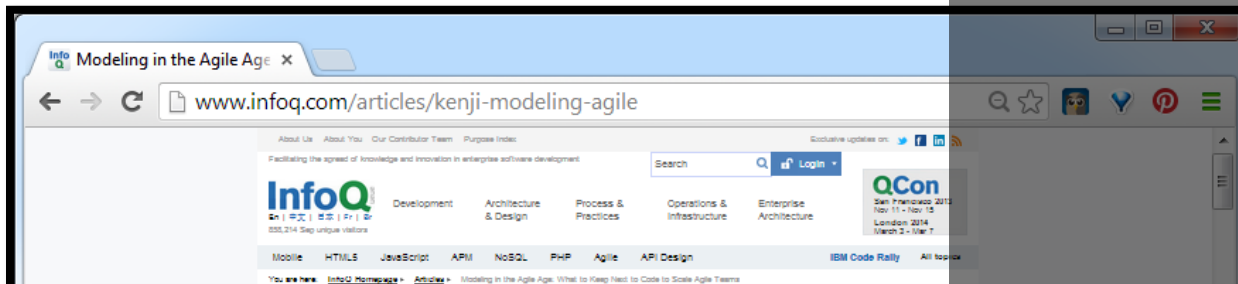


This is how my newest paper blog post looks like. Images, tag lines as section heads. Nothing like a paper.

I add screenshots

And communicate with the reader directly.

For inspiration on how to write, I look at website that my audience (developers) frequent for information.

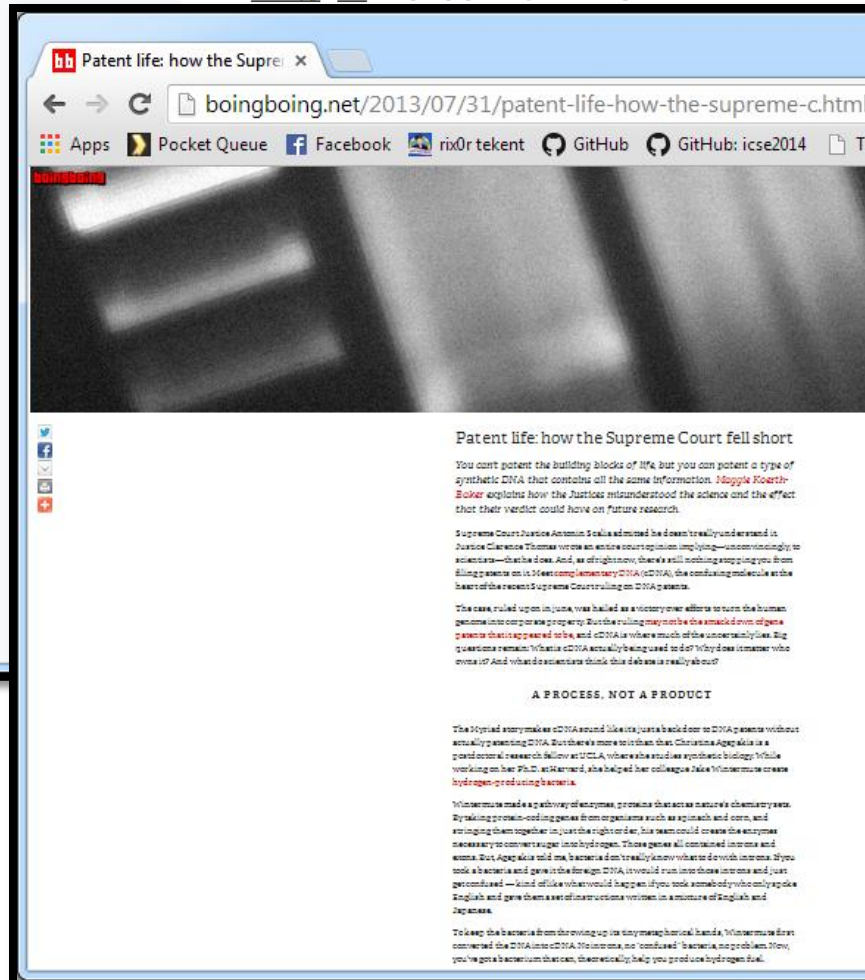


This is how my newest paper blog post looks like. Images, tag lines as section heads. Nothing like a paper.

I add screenshots

And communicate with the reader directly.

For inspiration on how to write, I look at website that my audience (developers) frequent for information.



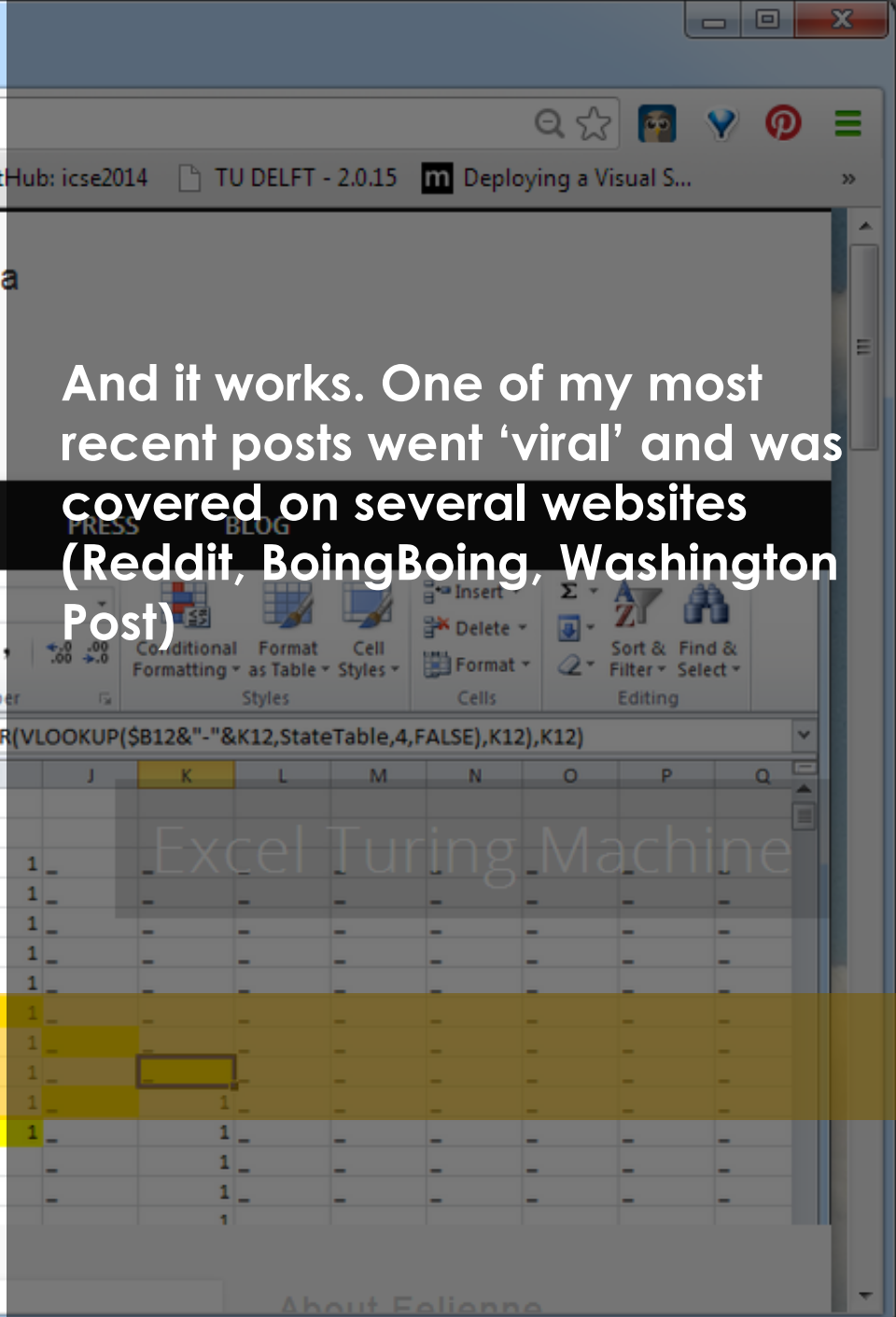


ABOUT COMMITTEES PUBLICATIONS LIVE BLOG

Clipboard Font Alignment Number

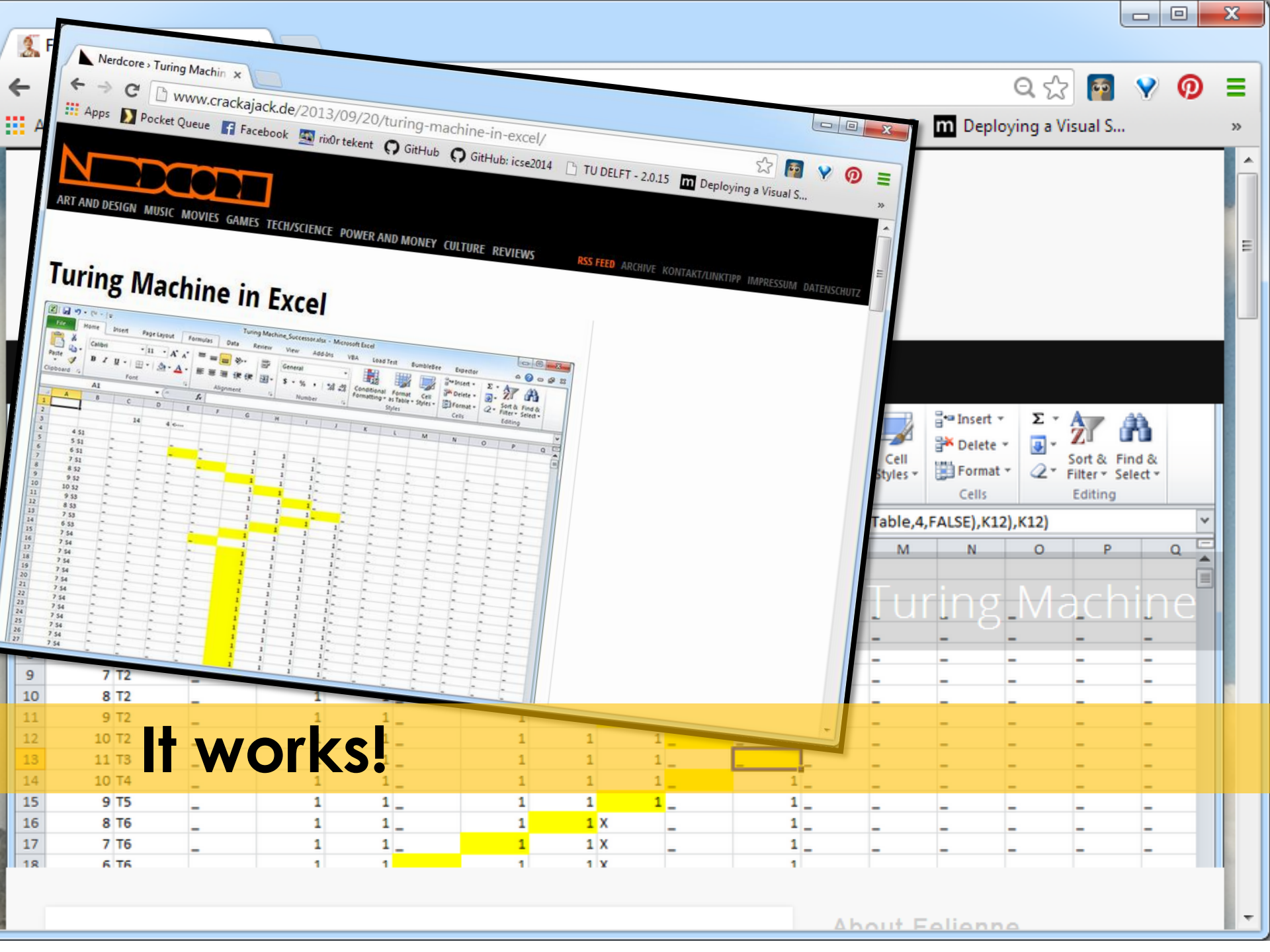
K13 $=IF(COLUMN(K13)=$A12,IFERROR(VLOOKUP($B12&"-"&K12,StateTable,4,FALSE),K12),K12)$

	A	B	C	D	E	F	G	H	I
4									
5									
6	4 T1	-		1	1	-	1	1	
7	5 T1	-		1	1	-	1	1	
8	6 T1	-		1	1	-	1	1	
9	7 T2	-		1	1	-	1	1	
10	8 T2	-		1	1	-	1	1	
11	9 T2	-		1	1	-	1	1	
12	10 T2	-		1	1	-	1	1	
13	11 T3	-		1	1	-	1	1	
14	10 T4	-		1	1	-	1	1	
15	9 T5	-		1	1	-	1	1	
16	8 T6	-		1	1	-	1	1	X
17	7 T6	-		1	1	-	1	1	X
18	6 T6	-		1	1	-	1	1	X



And it works. One of my most recent posts went 'viral' and was covered on several websites (Reddit, BoingBoing, Washington Post)

It works!



Turing Machine in Excel

It works!

Is Richmond's mortgage seizure scheme even legal? "The arguments will now proceed to the two parts of eminent domain law: demonstrating public purpose for the takings and offering fair-value. Since this is the furthest an eminent domain case has made it, it might be useful to step back and walk through the arguments. If the case succeeds, it is likely other cities, which have been hesitant, will consider going forward." *Mike Konczal* in *The Washington Post*.

CompSci is cool interlude: [Building a Turing machine in Excel](#).

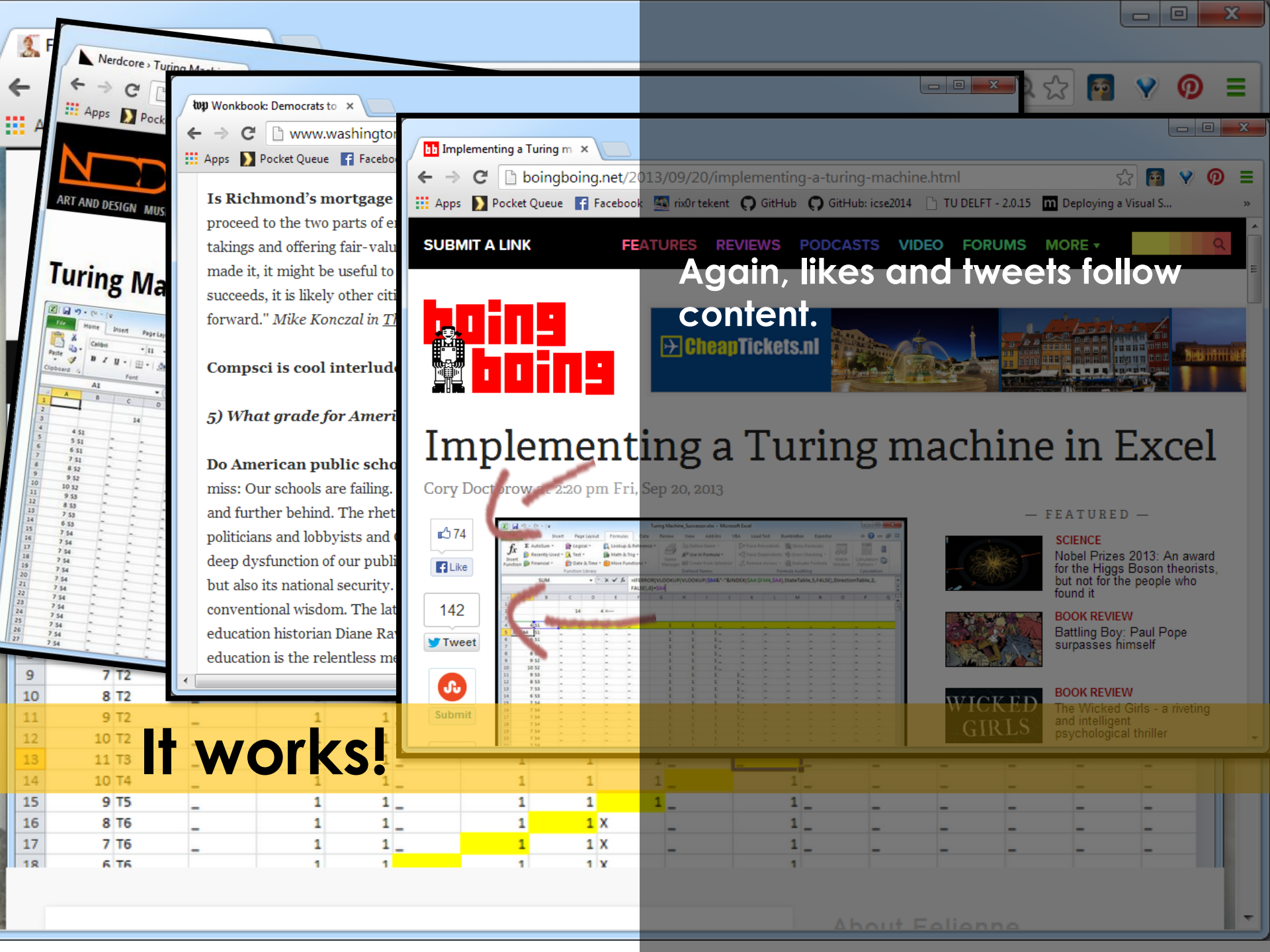
5) What grade for America's schools?

Do American public schools really stink? Maybe not. "The drumbeat is hard to miss: Our schools are failing. Public education is in crisis. Our students are falling further and further behind. The rhetoric comes from the left and right, from educators and politicians and lobbyists and CEOs and even Education Secretary Arne Duncan. The deep dysfunction of our public schools is said to threaten not only America's economy but also its national security. But a vocal group of contrarians is challenging that conventional wisdom. The latest weapon in their arsenal: A new book out this week by education historian Diane Ravitch, who argues that the biggest crisis facing public education is the relentless message that public education is in crisis." *Stephanie Simon*

It works!

The collage features several overlapping elements:

- Browser Window (Top Right):** Displays the article "Implementing a Turing machine in Excel" by Cory Doctorow on boingboing.net. The article title is "Implementing a Turing machine in Excel" and the author is "Cory Doctorow at 2:20 pm Fri, Sep 20, 2013". The page includes a navigation bar with links like "SUBMIT A LINK", "FEATURES", "REVIEWS", "PODCASTS", "VIDEO", "FORUMS", and "MORE". There are also social media links for Facebook, GitHub, and others.
- Spreadsheet (Bottom Left):** A Microsoft Excel spreadsheet titled "Turing Machine". It contains a grid of numbers and letters, likely representing the state transitions of the Turing machine. The grid is partially highlighted in yellow.
- Book Cover (Bottom Right):** A book cover for "The Wicked Girls" by Paul Pope. The cover features a dark, atmospheric illustration of a city street at night.
- Text Overlay (Center):** Large, bold, black text that reads "It works!".
- Other Elements:** A small "Nerdcore" logo is visible in the top left corner. A "CheapTickets.nl" banner is also present in the browser window.



wp Workbooks: Democrats to x

www.washington

Apps Pocket Queue Facebook

Is Richmond's mortgage proceed to the two parts of e takings and offering fair-val made it, it might be useful to succeeds, it is likely other citi forward." Mike Konczal in T

CompSci is cool interlude

5) What grade for Ameri

Do American public scho miss: Our schools are failing. and further behind. The rhet politicians and lobbyists and deep dysfunction of our publi but also its national security. conventional wisdom. The lat education historian Diane Ra education is the relentless me

Implementing a Turing m x

boingboing.net/2013/09/20/implementing-a-turing-machine.html

Apps Pocket Queue Facebook rix0r tekent GitHub GitHub: icse2014 TU DELFT - 2.0.15 Deploying a Visual S...

SUBMIT A LINK FEATURES REVIEWS PODCASTS VIDEO FORUMS MORE

boingboing

74

Like

142

Tweet

Submit

Again, likes and tweets follow content.

CheapTickets.nl

Implementing a Turing machine in Excel

Cory Doctorow at 2:20 pm Fri, Sep 20, 2013

— FEATURED —

SCIENCE
Nobel Prizes 2013: An award for the Higgs Boson theorists, but not for the people who found it

BOOK REVIEW
Battling Boy: Paul Pope surpasses himself

BOOK REVIEW
The Wicked Girls - a riveting and intelligent psychological thriller

It works!



I promised you 3 easy ways to start social media



**Slides: you make them anyway.
For level 1, just add notes and
upload.**

**I promised you 3 easy ways to start
social media**



Slides: you make them anyway.
For level 1, just add notes and
upload.

If you want to go the extra mile,
think about what 'story' your
presentation told and add that.

**I promised you 3 easy ways to start
social media**



Quick start 1) Sl



Tip 1: story, story, story



Quickstart 2) Progress reports

Progress reports: you write them anyway. For the basic coverage, just put them on a simple website.

I promised you 3 easy ways to start social media



Quick start 1) Sl



Tip 1: story, story, story



Quickstart 2) Progre



Tip 2) Built an audience

Progress reports: you write them anyway. For the basic coverage, just put them on a simple website.

Want more? Think about who you want to share those notes with and build an audience.

I promised you 3 easy ways to start social media



Quick start 1) Sl



Tip 1: story, story, story



Quickstart 2) Progr



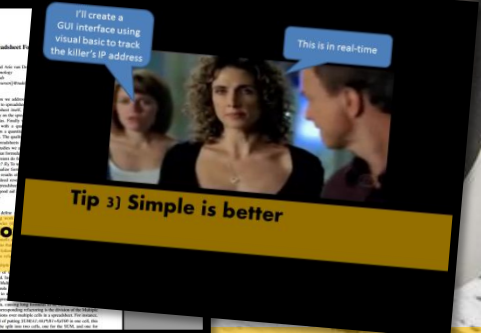
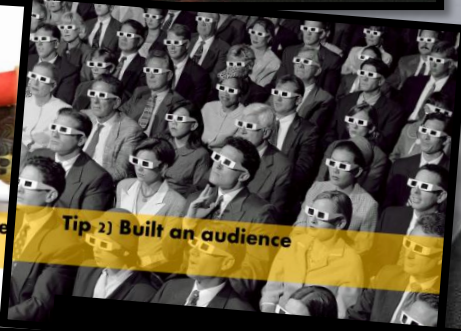
Tip 2) Built an audience



Quick start 3) A blog post per paper

I promised you 3 easy ways to start social media


Papers: you write them anyway.
Put a simple summary online.



Papers: you write them anyway.
Put a simple summary online.

Level 2: Make it understandable
for the general public, by
simplifying and looking like other
magazine/blogs your audience
reads.

I promised you 3 easy ways to start
social media



Want to know more? Feel free to
have a look at my website and
imitate what you like (or comment
on what you don't)

Or send me a tweet, I'm at
@feliienne and also still learning.

Content is king

www.feliienne.com @feliienne