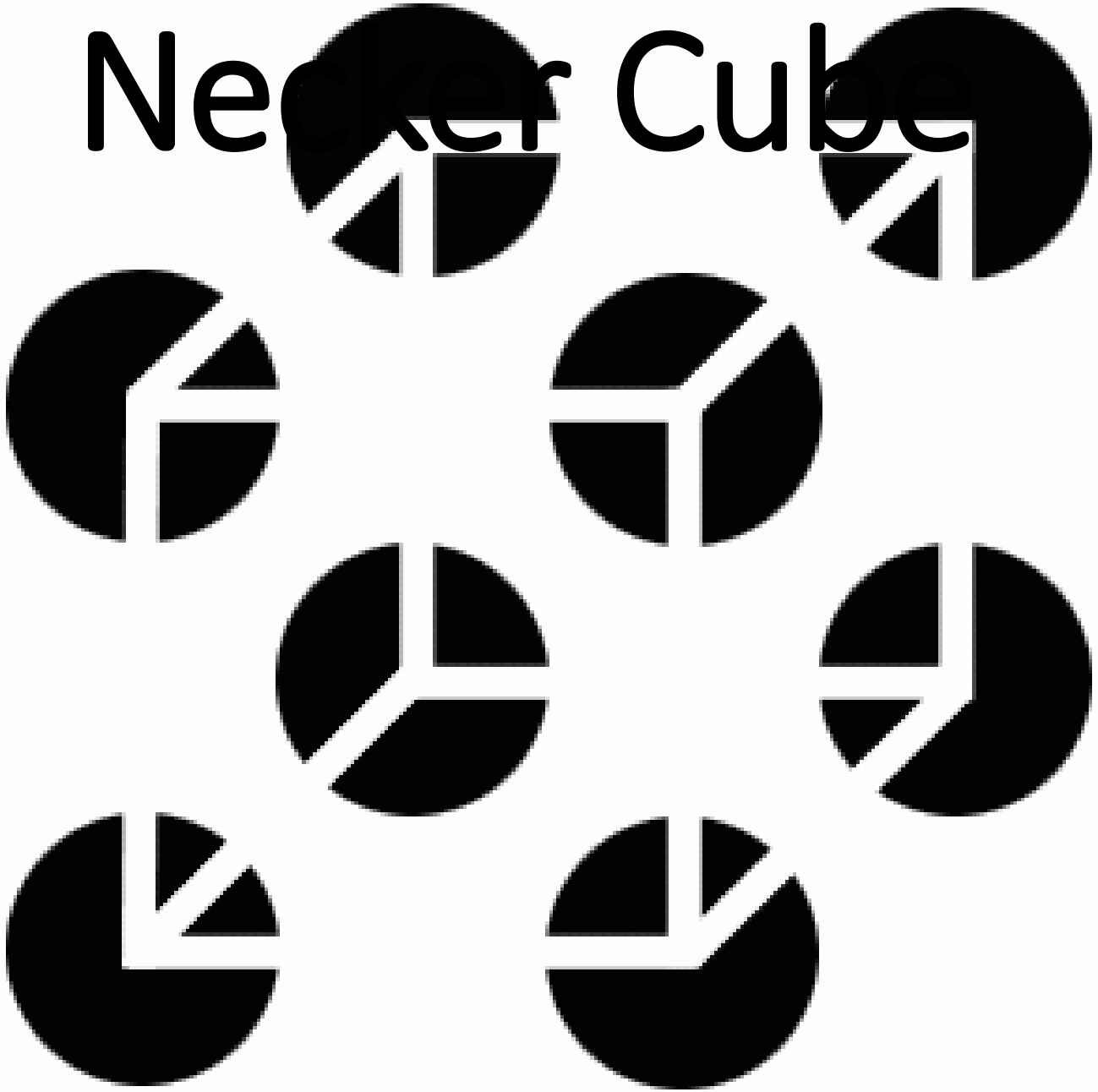
UI, UX, Users, and You

A Developer's Guide to Designing User Interfaces and User Experiences for your Users

Necker Cube



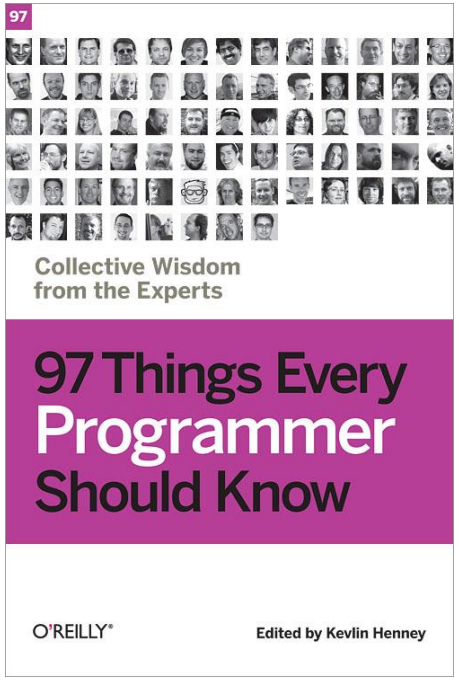
UI, UX, Users, and You

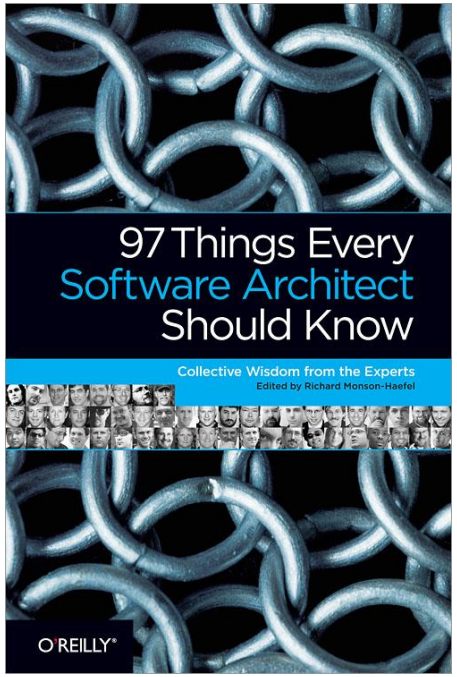
A Developer's Guide to Designing User Interfaces and User Experiences for your Users

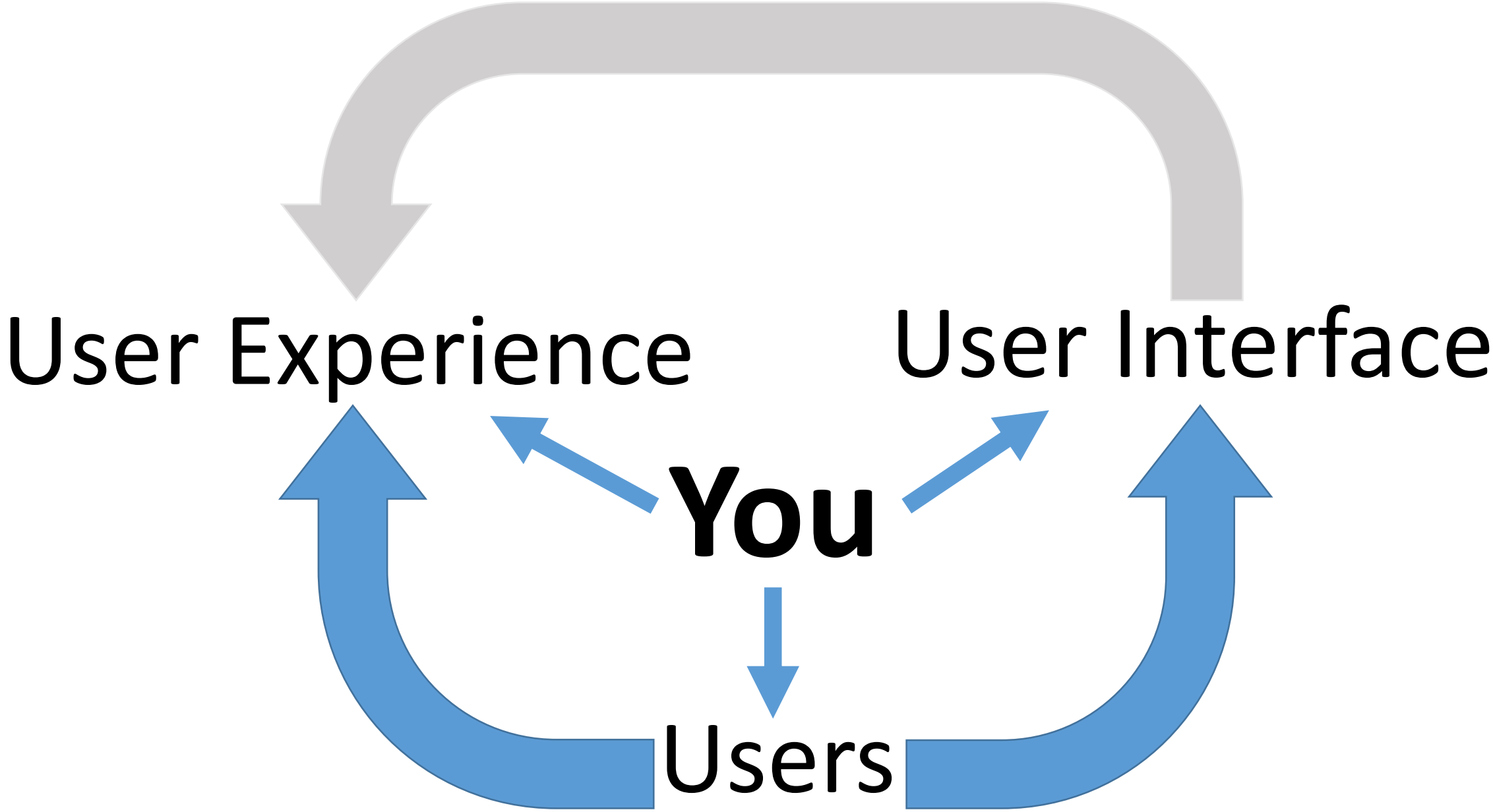
Burk Hufnagel

Programmer and Solution Architect with Daugherty Business Solutions

Speaker at JavaOne twice; voted RockStar in 2010






User Experience
User Interface
You
Users

Quick Definitions

Users

The people who use what we create.

User Interface

The part of your creation they interact with.

User Experience

How they feel about the interaction.

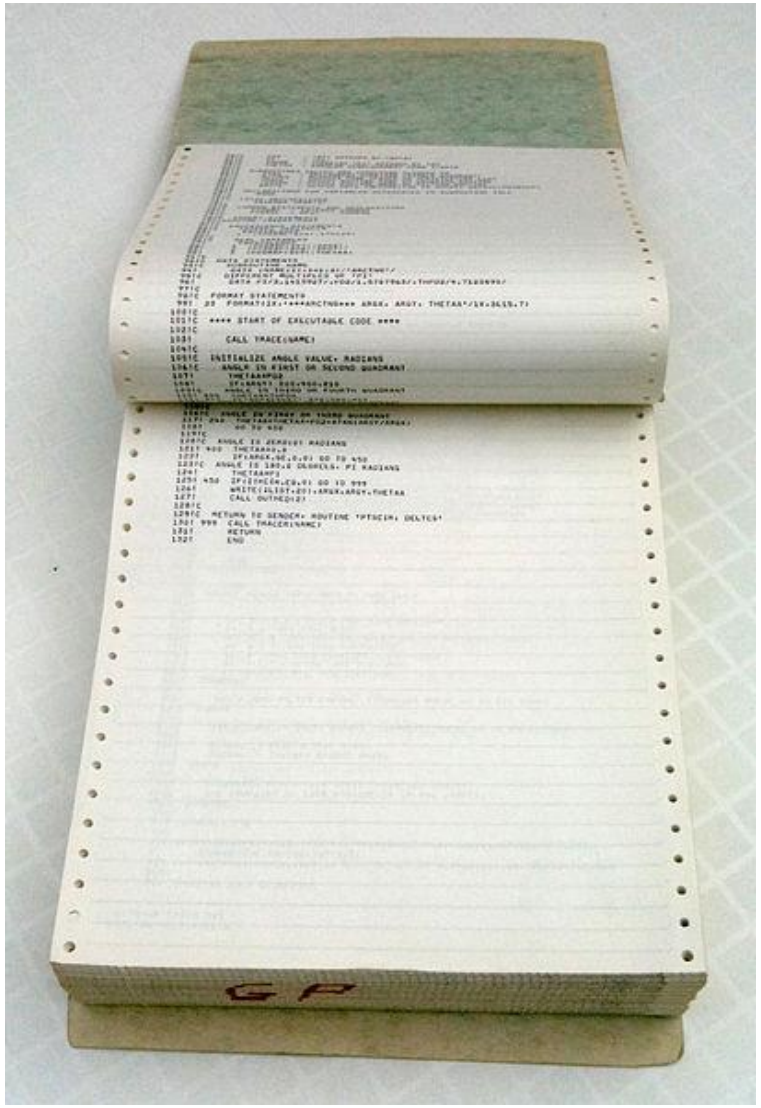
Public Service Announcement

Learning about UX reminds me of Master Yoda's quote about the Dark Side of the Force:

"Once you start down the dark path, forever will it dominate your destiny. Consume you it will, as it did Obi-Wan's apprentice."

Users

- Historically, a User was a trained technician who knew how to run computer jobs, allocating memory and other resources as they were needed. This is why the programs in TRON believed in the power of the Users.
- End-Users were people who didn't necessarily know how to use the computer and just worked with the results it returned; often as printed reports on continuous form paper.



Controversy about “Users"

Don Norman and others have made the point that the term user is derogatory, and dehumanizes the people we’re supposed to be serving.

Still others, like Jimmy Guterman at O’Reilly, pointed out that “the only industries that refer to their customers as users are high tech and illicit drugs.”

Software = Drug?

No. But, you probably know people who seem hooked on games, social media, or even email and chat.

Great UX can be compelling.

Searching Google for the phrase “x addiction” where x is:

Games: Angry Birds -> 1,630,000

World of Warcraft -> 20,400,000

Twitter -> 85,500,000

Facebook -> 91,400,000

Email -> 51,500,000

Google -> 54,300,000

User Experience

User Experience is how someone feels about interacting with your software – before, after, and especially while, they are using it.

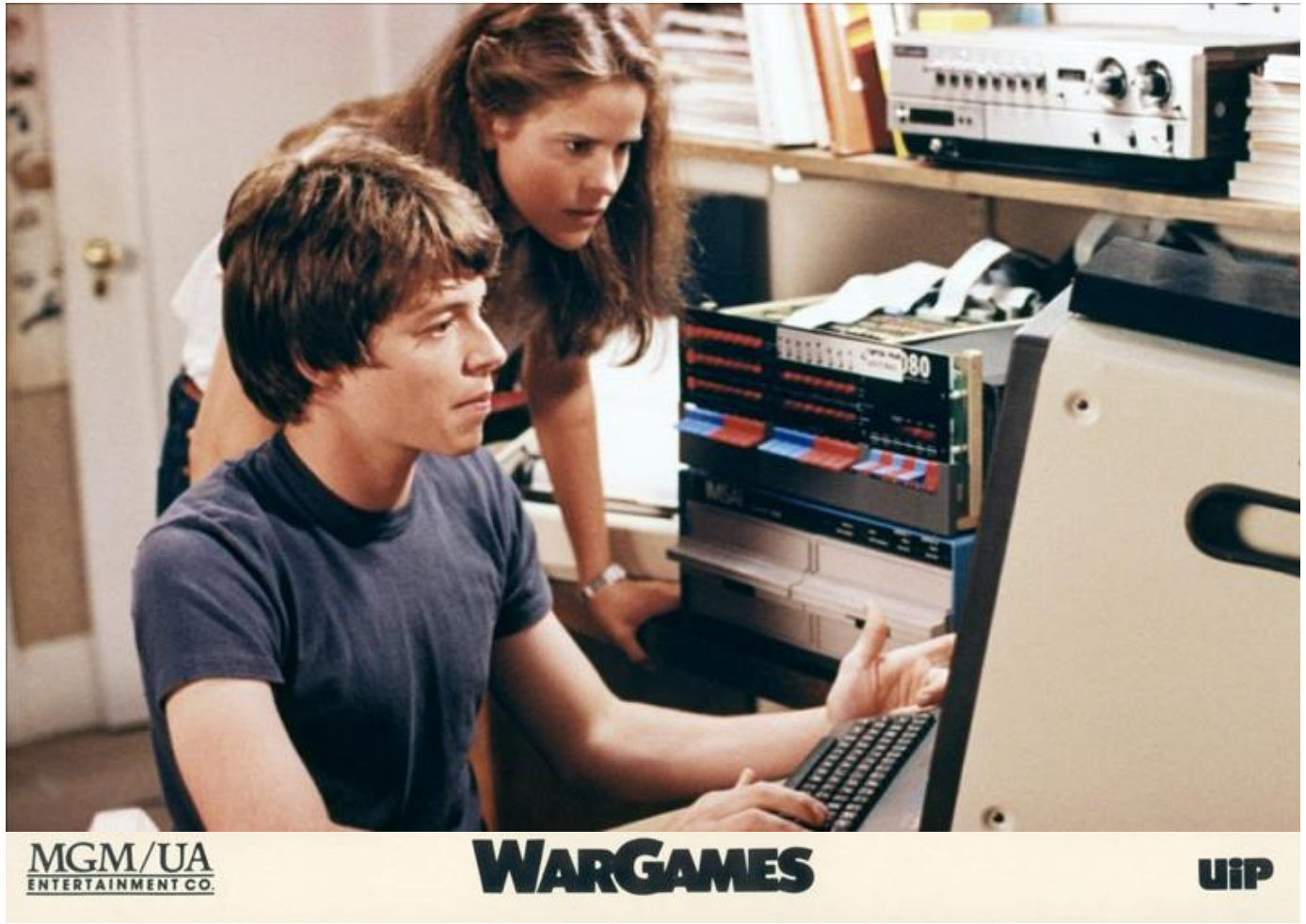
User Experience

Rule 1: There is always a user experience.

Corollary: If you don't intentionally design the UX then you have no control over what the user experience will be.

This is probably a Bad Thing.

UX Can Change Over Time



So what?

Yodlee and Mint

Yodlee spent 10 years and $116 million creating services connecting to about 11,000 financial data sources.

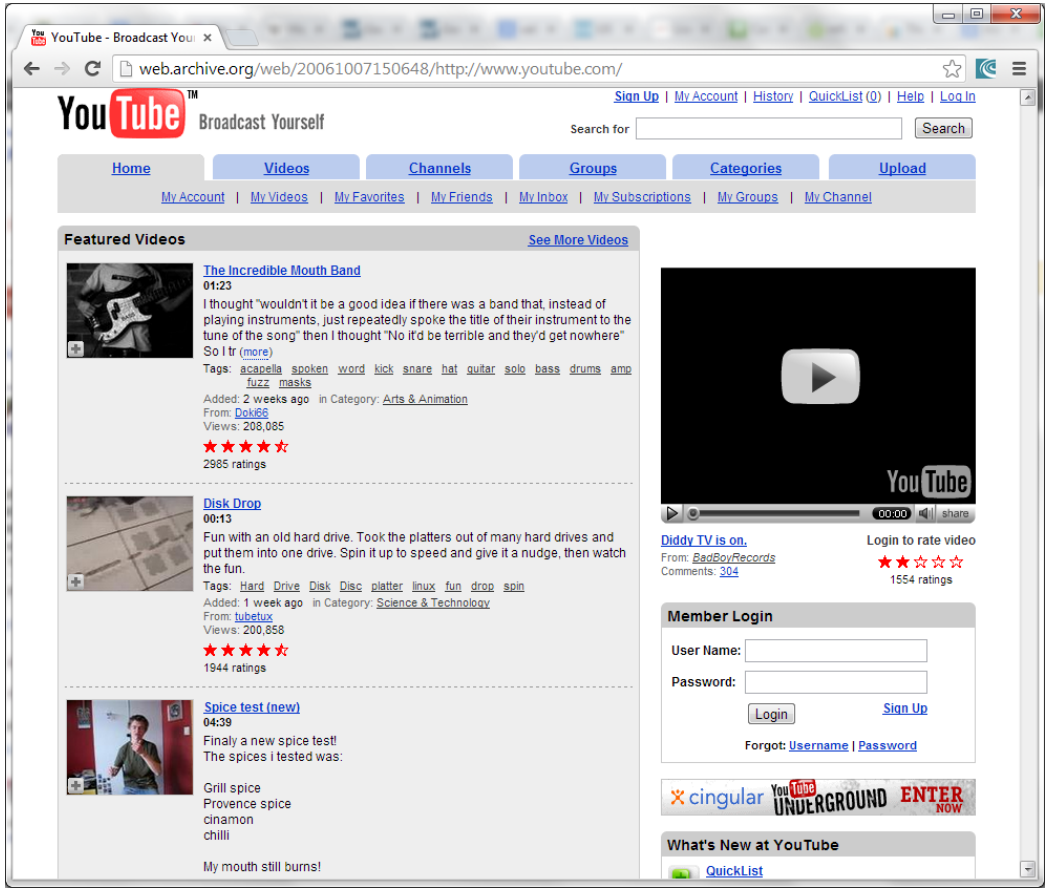
Mint spent two years and $32 million building a great UI/UX layer on top of Yodlee and got bought for $170 million.

Yodlee did the “hard work: and only made about $4 million from Mint.

Adobe and YouTube

Adobe create Flash and figured out how to play video over the web.

YouTube built a great UI/UX that let people upload and share their videos, and got bought 19 months later for $1.65 billion.



Teehan + Lax Fund (UX Fund)

Geoff Teehan and Jon Lax (founders of Canadian UX design agency) invested $50,000 in 10 companies, including Apple, Google, Netflix, and RIM that they felt provided great UX. In four and a half years, it increased over 100%.

TEEHAN+LAX UX FUND

Change Since November 1, 2006

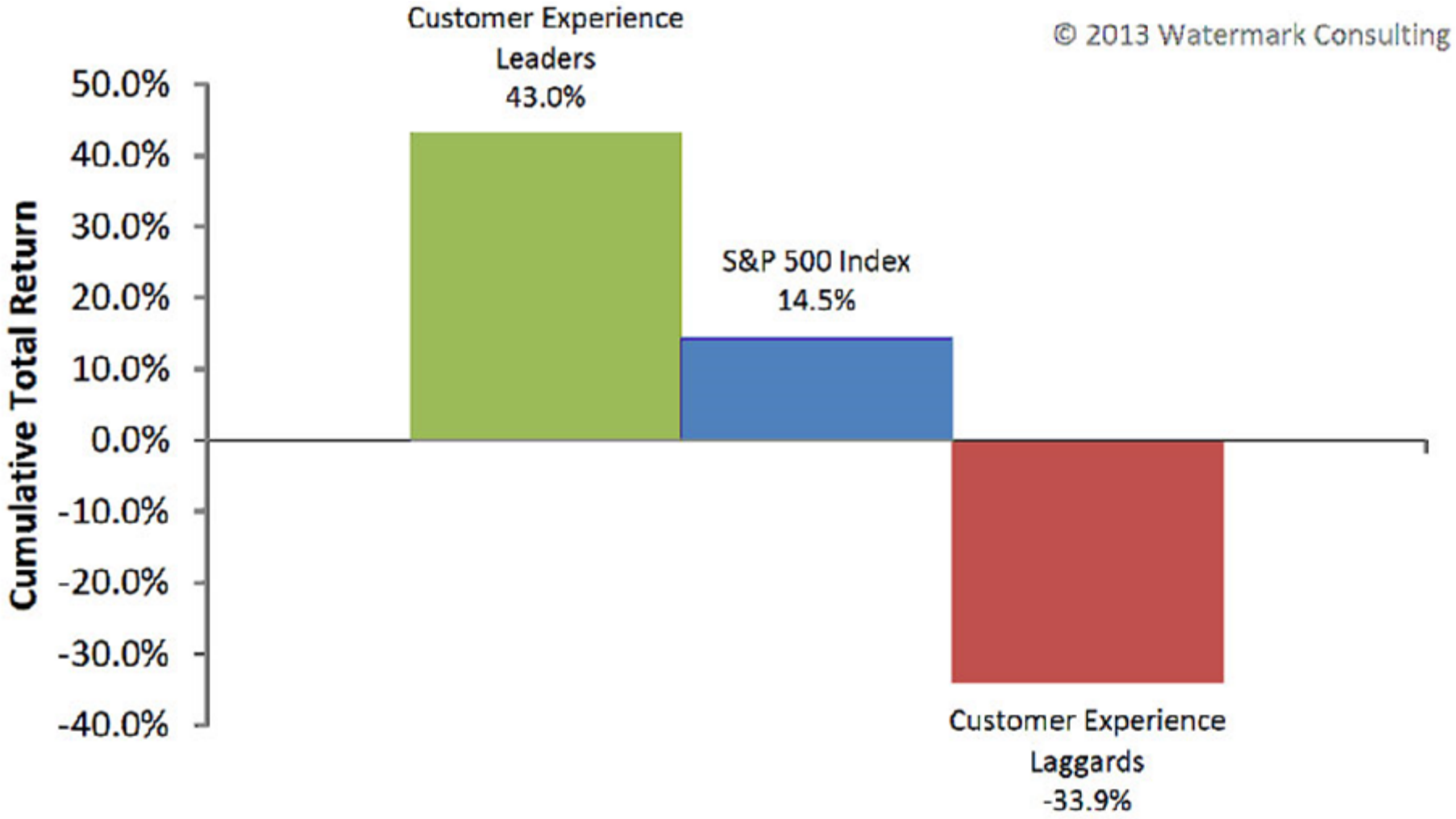
TEEHAN+LAX UX FUND

+101.8%

NASDAQ	+21.76%	NASDAQ 100	+39.15%	Total Initial Investment:	$49,672.90
S&P 500	-0.92%	NYSE	-0.79%	Inception Date:	Nov. 1, 2006
DOW	-100%			Days Until Maturity:	NA

Symbol	Last Trade	Today's Change		Shares	Buy Price	Current Value	Gain/Loss	
AAPL	350.13	$0.00	0%	62	$80.86	$21,708.06	+$16,694.74	+333.01%
ERTS	20.18	$0.00	0%	94	$53.10	$1,896.92	-$3,094.48	-62.00%
GOOG	544.10	$0.00	0%	10	$478.37	$5,441.00	+$657.30	+13.74%
JBLU	5.66	$0.00	0%	400	$12.57	$2,264.00	-$2,764.00	-54.97%
NFLX	232.67	$0.00	0%	180	$27.62	$41,880.60	+$36,909.00	+742.40%
NKE	82.32	$0.00	0%	108	$46.03	$8,890.56	+$3,919.32	+78.84%
PGR	21.94	$0.00	0%	207	$24.19	$4,541.58	-$465.75	-9.30%
RIMM	48.65	$0.00	0%	126	$39.34	$6,129.90	+$1,173.06	+23.67%
TGT	49.10	$0.00	0%	84	$59.31	$4,124.40	-$857.64	-17.21%
YHOO	17.70	$0.00	0%	190	$26.14	$3,363.00	-$1,603.60	-32.29%
Total:	--	--	--	--	--	**$100,240.02**	**+$50,567.95**	**+101.8%**

Watermark Consulting, 2013
6-Yr Cust Experience ROI Study



Designing a Great User Experience

Understand who is in your target audience, and who is not.

Figure out the user model and match it as closely as possible.

What are they trying to accomplish?

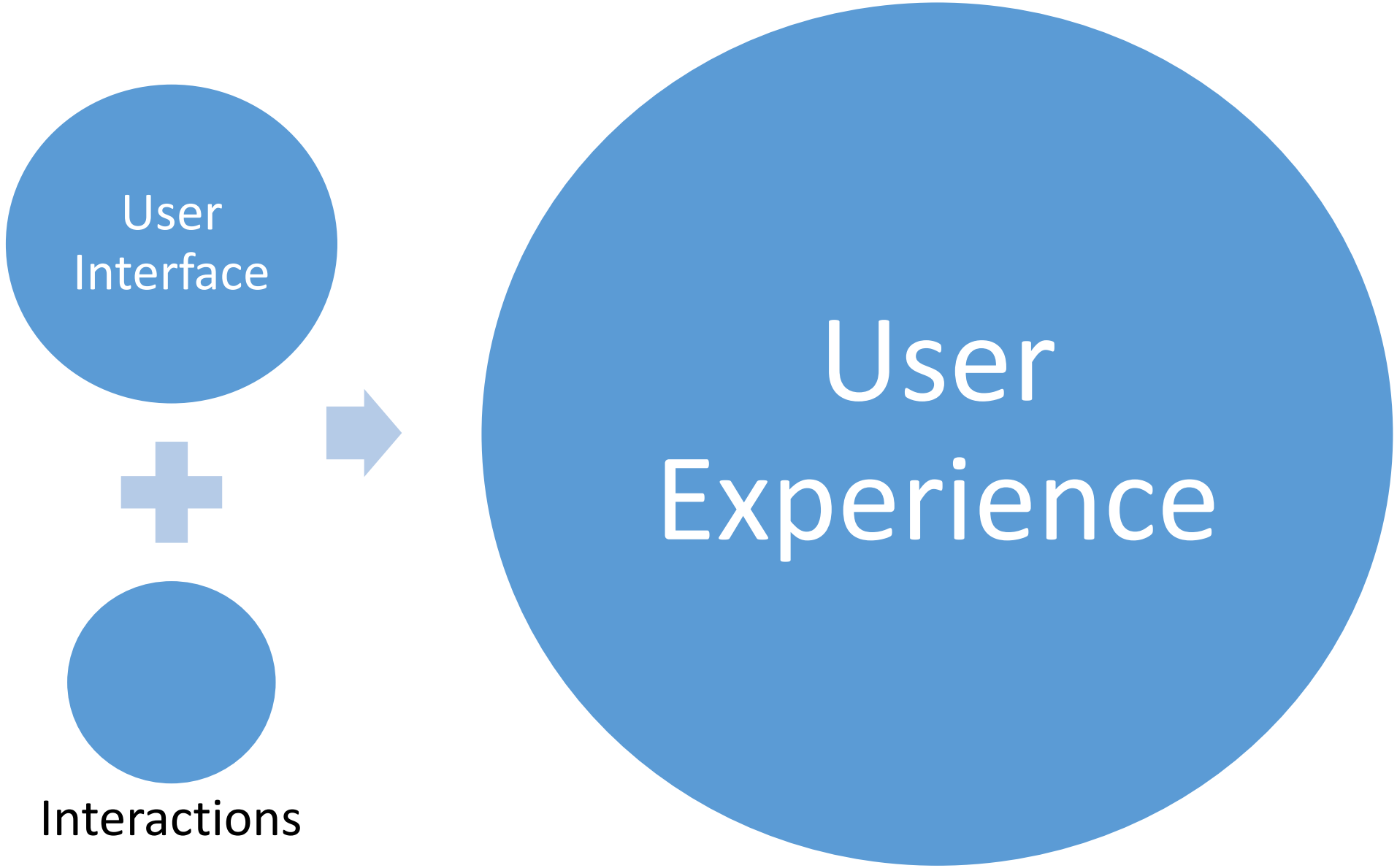
What's their skill level?

What's the context?

- Desktop, mobile, outdoors, etc. Constraints help determine what you need to supply.

Attractive things work better

User Interface

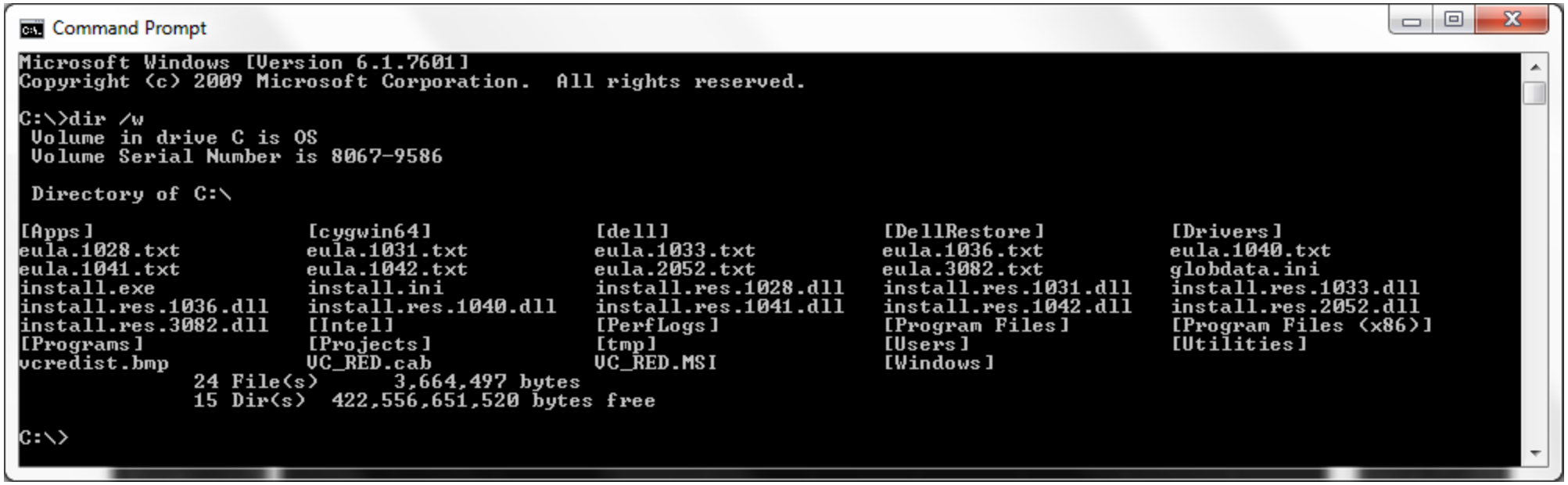

User Interface
Interactions
User Experience

Common Types of User Interfaces

Text-based: Command line interface

Graphical User Interface (GUI)

Command line user interface

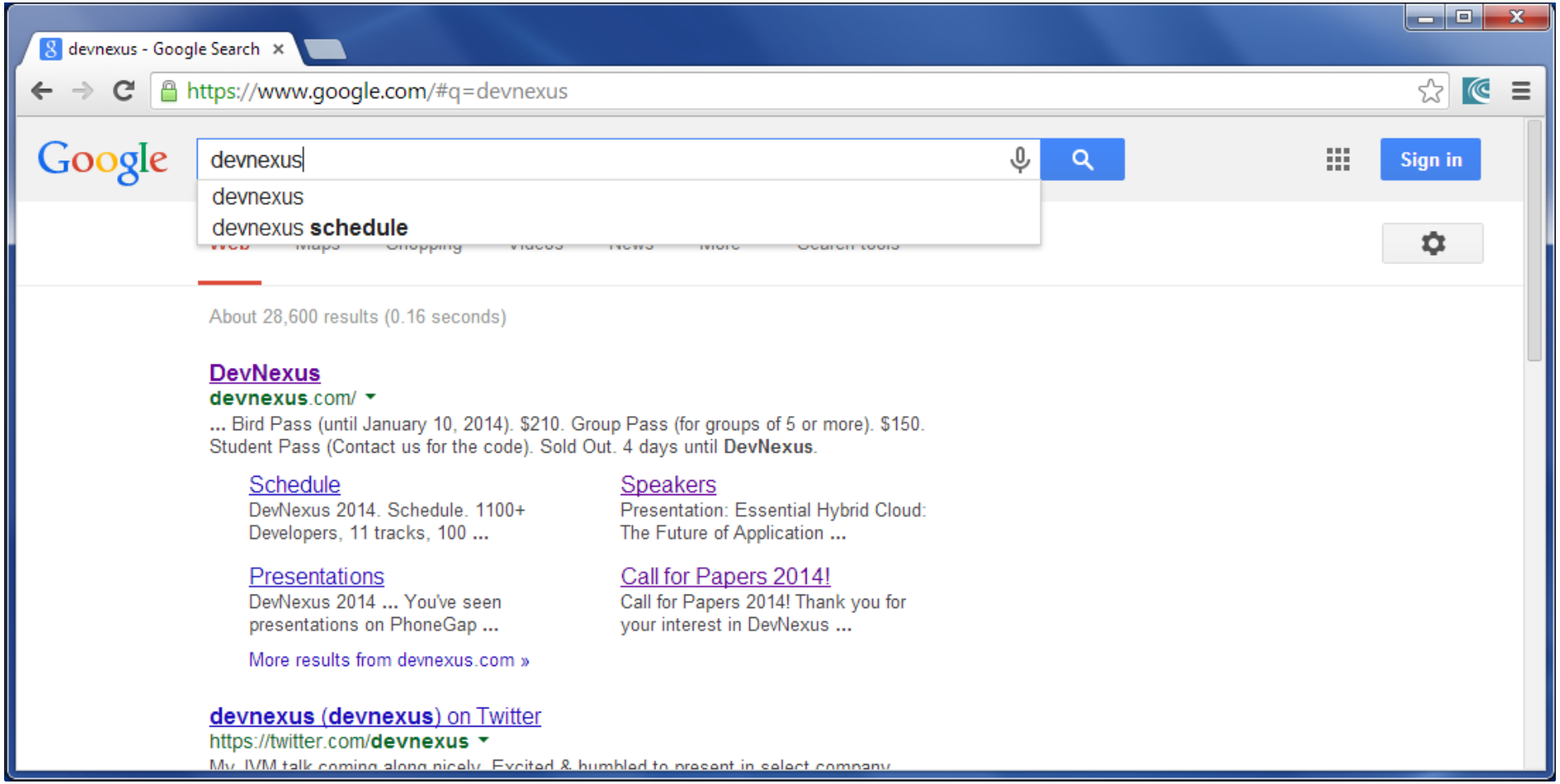


Command line user interface

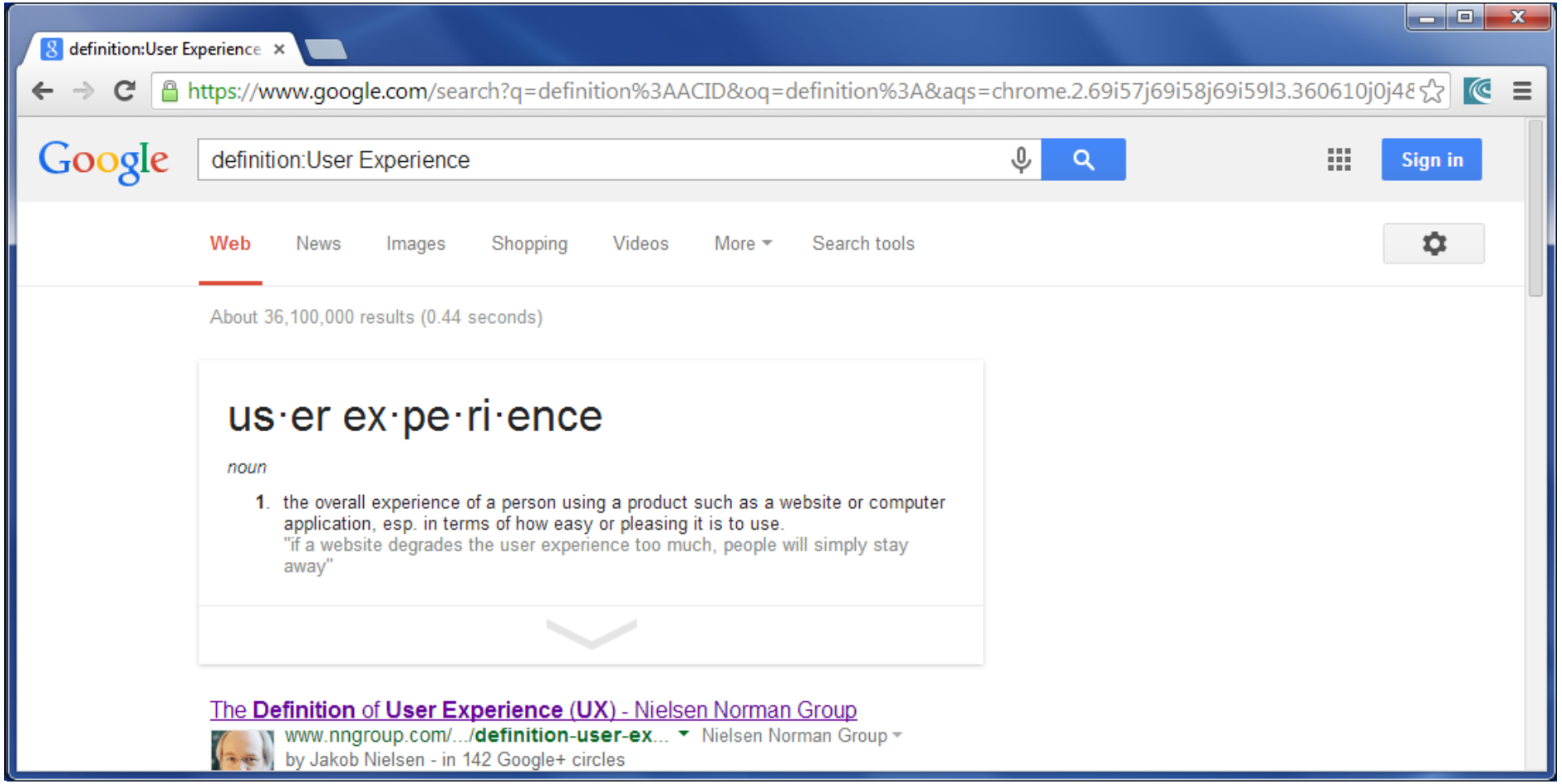
Grep command to find all the properties specified in files,
excluding the target directory and it's subdirectories:

```
grep -r --include=*.xml --exclude-dir=target
        -o $\{com.efx.channel.batch.[[:alpha:].]*\}
```

Command line user interface



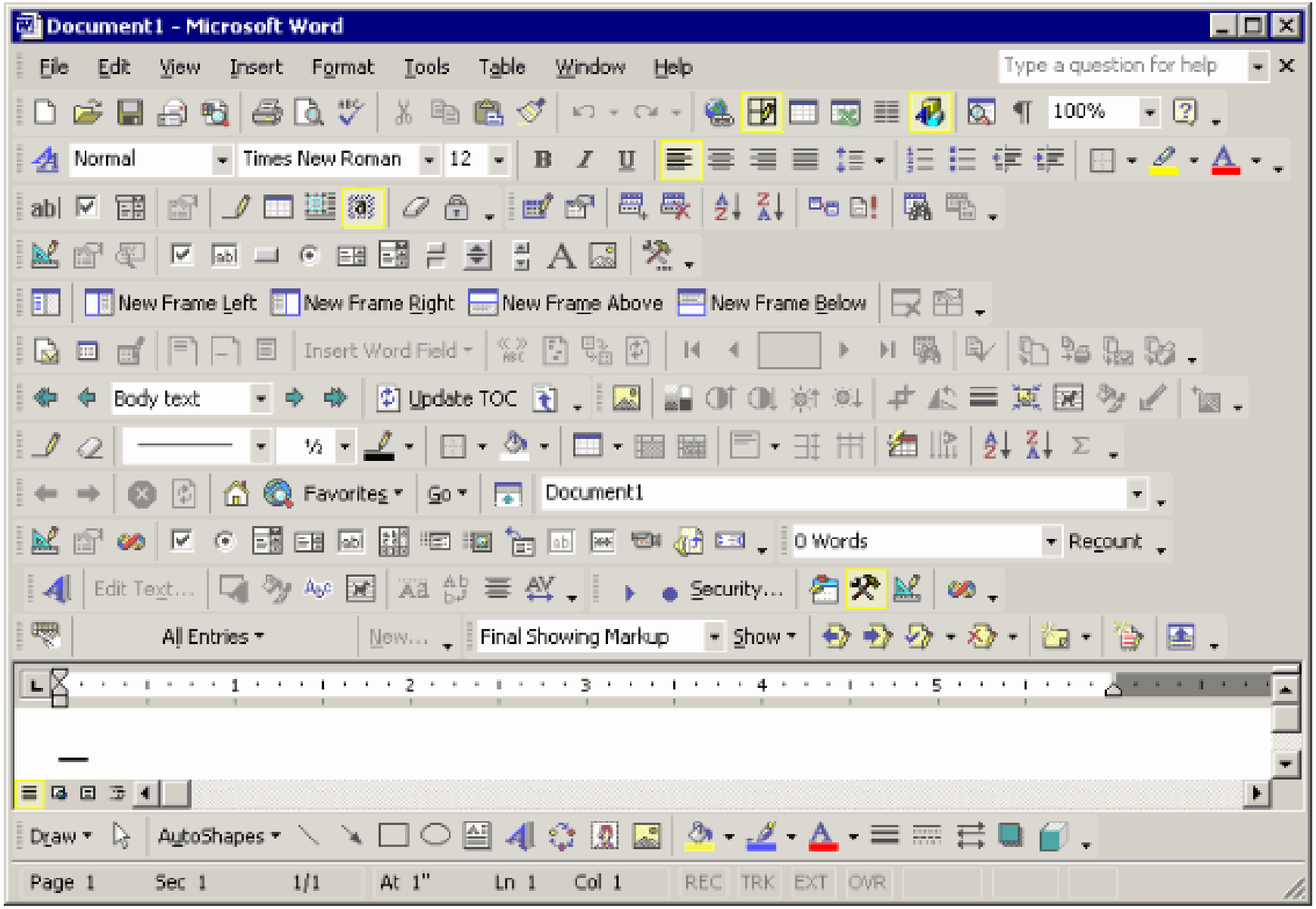
Command line user interface

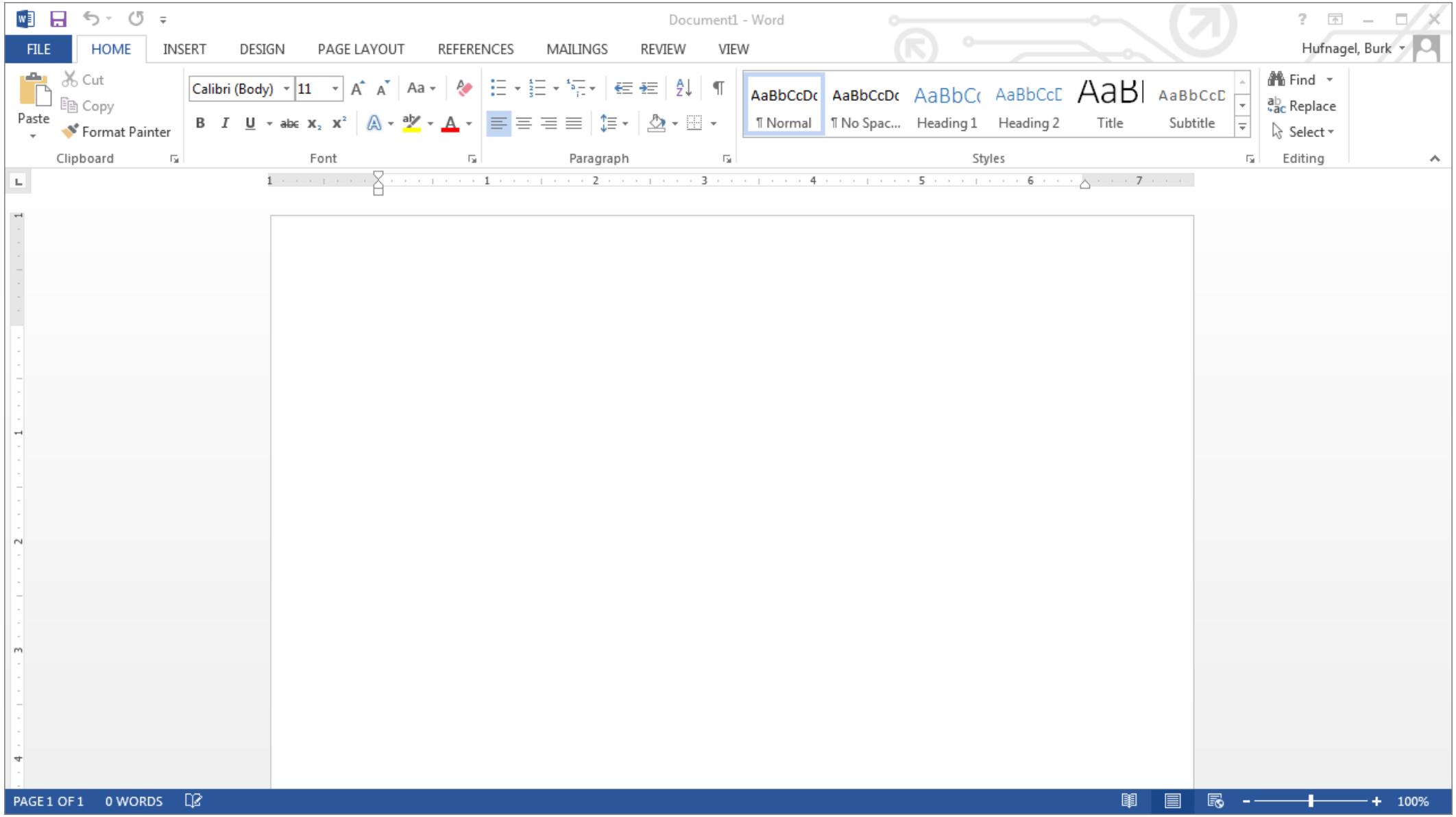


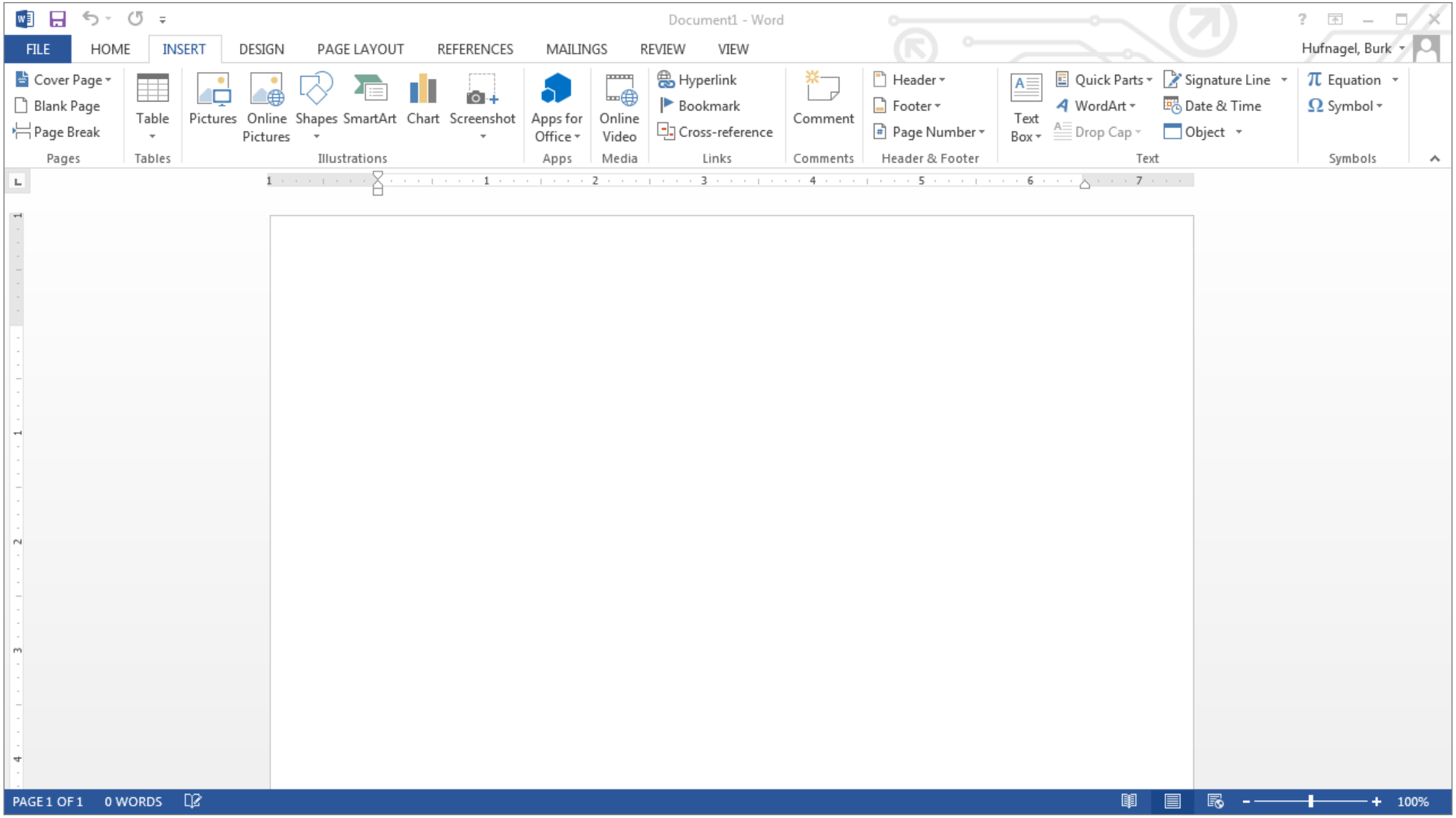
Designing simple use experiences often turns out not to be about “How can I make this simpler?” but rather “Where should I move the complexity.”

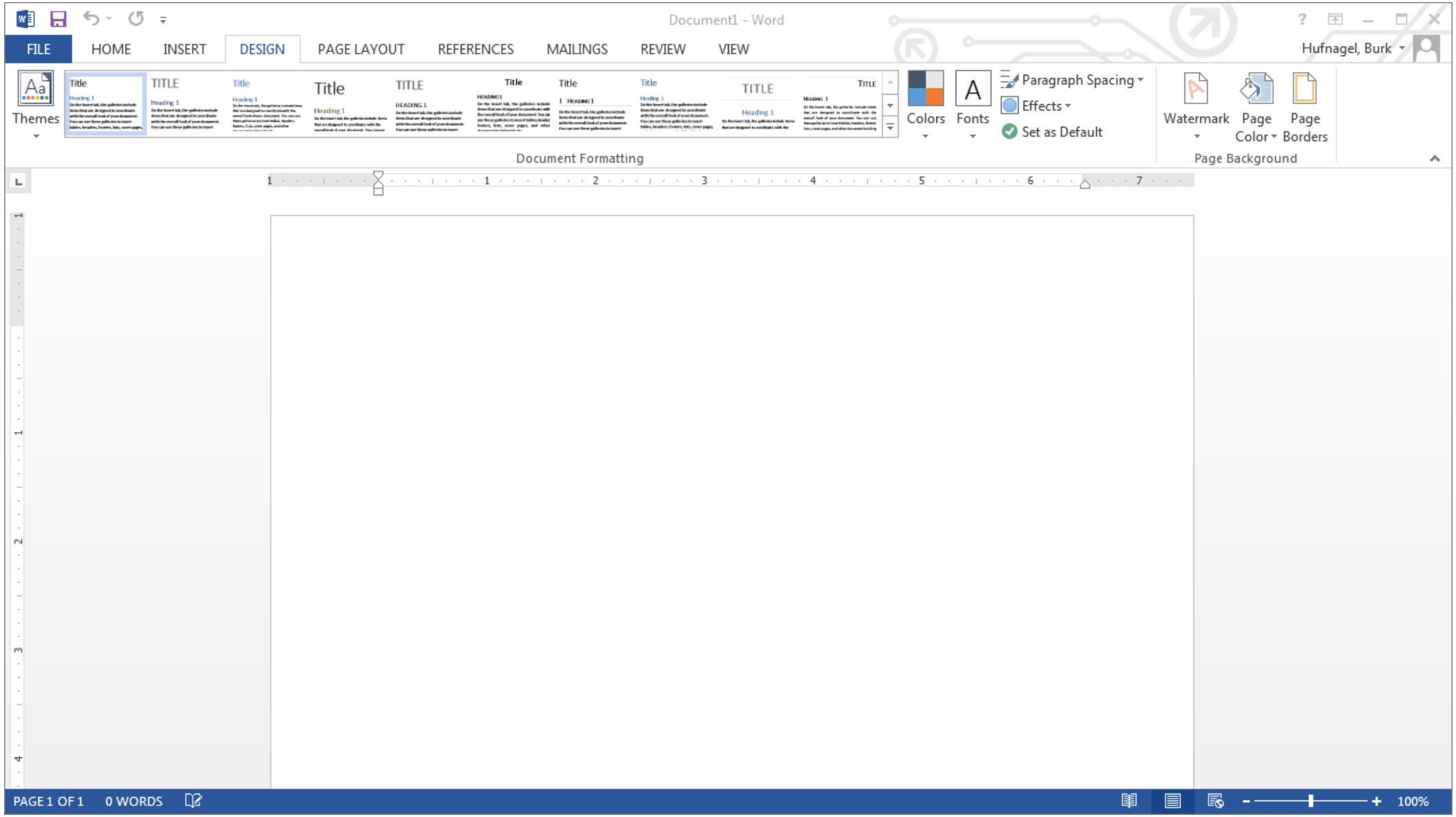
Giles Colborne – Simple and Usable

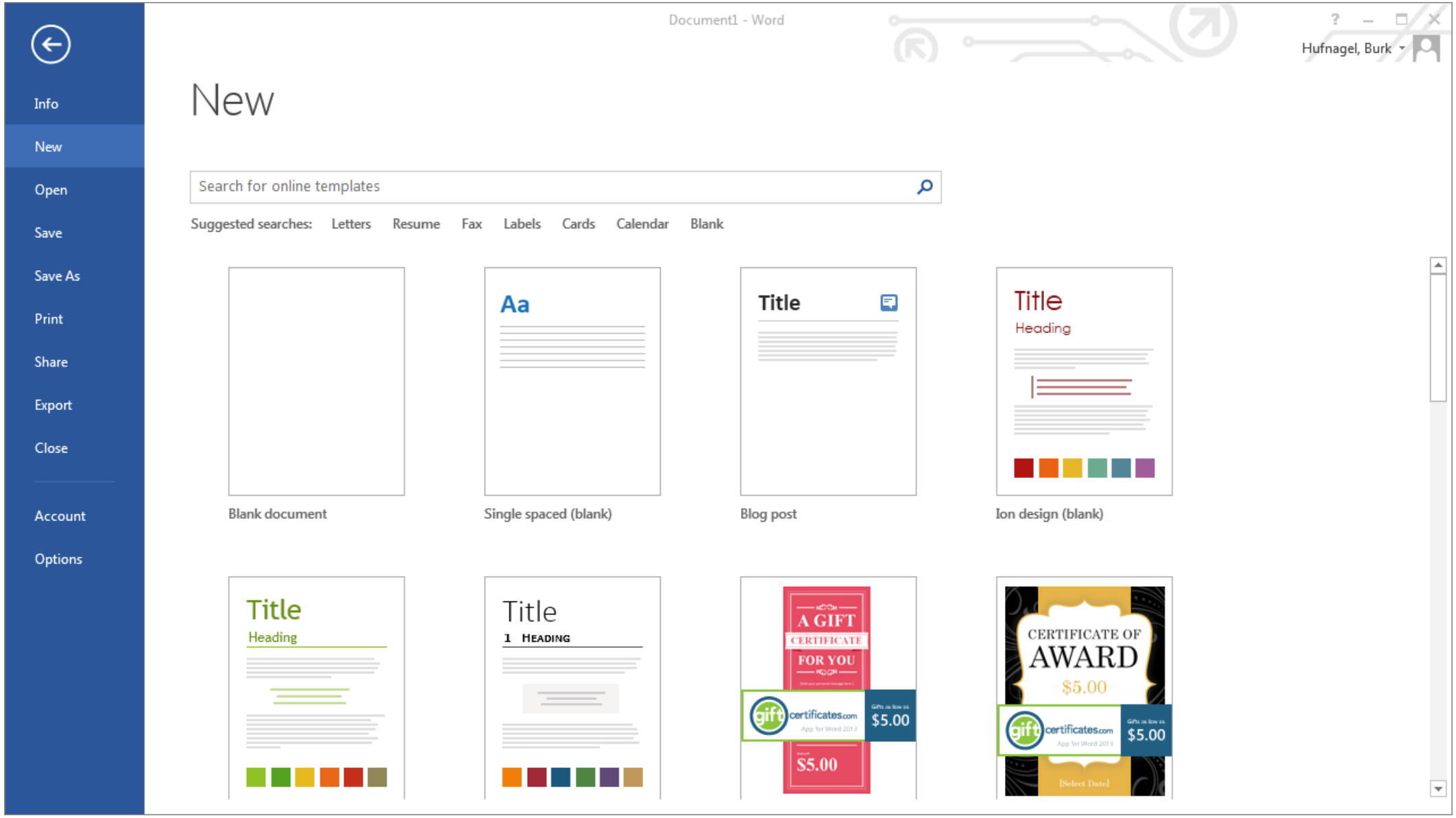
Conserve and move complexity

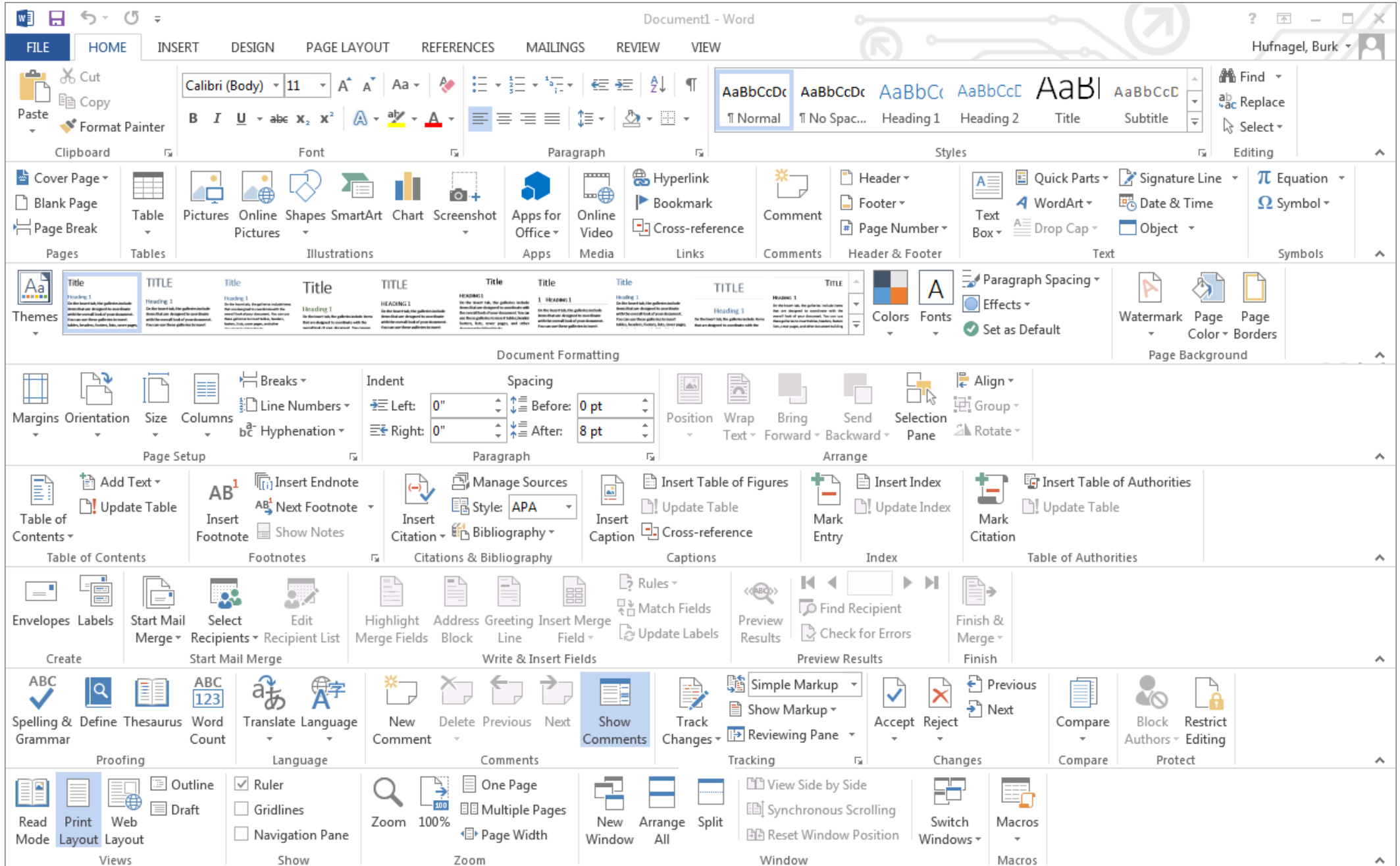

Document1 - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Type a question for help
100%
Normal
Times New Roman
12
New Frame Left
New Frame Right
New Frame Above
New Frame Below
Insert Word Field
Body text
Update TOC
Favorites
Go
Document1
0 Words
Recount
Edit Text...
Security...
All Entries
New...
Final Showing Markup
Show
Draw
AutoShapes
Page 1
Sec 1
1/1
At 1"
Ln 1
Col 1
REC
TRK
EXT
OVR


Document1 - Word
FILE
HOME
INSERT
DESIGN
PAGE LAYOUT
REFERENCES
MAILINGS
REVIEW
VIEW
Hufnagel, Burk
Cut
Copy
Format Painter
Paste
Clipboard
Calibri (Body)
11
Font
Paragraph
AaBbCcDc ¶ Normal
AaBbCcDc ¶ No Spac...
AaBbC Heading 1
AaBbCcE Heading 2
AaB Title
AaBbCcD Subtitle
Styles
Find
Replace
Select
Editing
PAGE 1 OF 1
0 WORDS
100%


Document1 - Word
FILE
HOME
INSERT
DESIGN
PAGE LAYOUT
REFERENCES
MAILINGS
REVIEW
VIEW
Hufnagel, Burk
Cover Page
Blank Page
Page Break
Pages
Table
Tables
Pictures
Online Pictures
Shapes
SmartArt
Chart
Screenshot
Illustrations
Apps for Office
Apps
Online Video
Media
Hyperlink
Bookmark
Cross-reference
Links
Comment
Comments
Header
Footer
Page Number
Header & Footer
Text Box
Quick Parts
WordArt
Drop Cap
Signature Line
Date & Time
Object
Text
Equation
Symbol
Symbols
PAGE 1 OF 1
0 WORDS
100%


Document1 - Word
FILE
HOME
INSERT
DESIGN
PAGE LAYOUT
REFERENCES
MAILINGS
REVIEW
VIEW
Hufnagel, Burk
Themes
Colors
Fonts
Paragraph Spacing
Effects
Set as Default
Watermark
Page Color
Page Borders
Document Formatting
Page Background
PAGE 1 OF 1
0 WORDS
100%


Document1 - Word
Hufnagel, Burk
Info
New
Open
Save
Save As
Print
Share
Export
Close
Account
Options
New
Search for online templates
Suggested searches: Letters Resume Fax Labels Cards Calendar Blank
Blank document
Aa
Single spaced (blank)
Title
Blog post
Title
Heading
Ion design (blank)
Title
Heading
Title
1 HEADING
A GIFT
CERTIFICATE
FOR YOU
$5.00
gift certificates.com
$5.00
CERTIFICATE OF
AWARD
$5.00
gift certificates.com
$5.00
[Select Date]


Document1 - Word
FILE
HOME
INSERT
DESIGN
PAGE LAYOUT
REFERENCES
MAILINGS
REVIEW
VIEW
Hufnagel, Burk
Paste
Cut
Copy
Format Painter
Clipboard
Calibri (Body)
11
Font
Paragraph
AaBbCcDc ¶ Normal
AaBbCcDc ¶ No Spac...
AaBbC Heading 1
AaBbCcE Heading 2
AaBI Title
AaBbCcD Subtitle
Styles
Find
Replace
Select
Editing
Cover Page
Blank Page
Page Break
Pages
Table
Tables
Pictures
Online Pictures
Shapes
SmartArt
Chart
Screenshot
Illustrations
Apps for Office
Apps
Online Video
Media
Hyperlink
Bookmark
Cross-reference
Links
Comment
Comments
Header
Footer
Page Number
Header & Footer
Text Box
Quick Parts
WordArt
Drop Cap
Signature Line
Date & Time
Object
Text
Equation
Symbol
Symbols
Themes
Colors
Fonts
Paragraph Spacing
Effects
Set as Default
Document Formatting
Watermark
Page Color
Page Borders
Page Background
Margins
Orientation
Size
Columns
Breaks
Line Numbers
Hyphenation
Page Setup
Indent
Left: 0"
Right: 0"
Spacing
Before: 0 pt
After: 8 pt
Paragraph
Position
Wrap Text
Bring Forward
Send Backward
Selection Pane
Align
Group
Rotate
Arrange
Table of Contents
Add Text
Update Table
Table of Contents
Insert Footnote
Insert Endnote
Next Footnote
Show Notes
Footnotes
Insert Citation
Manage Sources
Style: APA
Bibliography
Citations & Bibliography
Insert Caption
Insert Table of Figures
Update Table
Cross-reference
Captions
Mark Entry
Insert Index
Update Index
Index
Mark Citation
Insert Table of Authorities
Update Table
Table of Authorities
Envelopes
Labels
Create
Start Mail Merge
Select Recipients
Edit Recipient List
Start Mail Merge
Highlight Merge Fields
Address Block
Greeting Line
Insert Merge Field
Rules
Match Fields
Update Labels
Write & Insert Fields
Preview Results
Find Recipient
Check for Errors
Preview Results
Finish & Merge
Finish
Spelling & Grammar
Define
Thesaurus
Word Count
Proofing
Translate
Language
Language
New Comment
Delete
Previous
Next
Show Comments
Comments
Track Changes
Simple Markup
Show Markup
Reviewing Pane
Tracking
Accept
Reject
Previous
Next
Changes
Compare
Compare
Block Authors
Restrict Editing
Protect
Read Mode
Print Layout
Web Layout
Outline
Draft
Views
Ruler
Gridlines
Navigation Pane
Show
Zoom
100%
One Page
Multiple Pages
Page Width
Zoom
New Window
Arrange All
Split
View Side by Side
Synchronous Scrolling
Reset Window Position
Switch Windows
Window
Macros
Macros

Biological Design Principles

We are visual thinkers and great at seeing patterns

DOG HOUSE CAT

RED ORANGE YELLOW

GREEN BLUE PURPLE

BLACK RED ORANGE

BLUE YELLOW GREEN

PURPLE BLACK RED

Biological Design Principles

We are visual thinkers and great at seeing patterns

Chunking

Things near each other are related somehow

Grouping

Things farther apart are not related

Importance

Size determines importance. Big threat vs small threat.

Billing Information

*First Name:	Jane	*Address:	123 Sesame Street
*Last Name:	Doe		
Company:		*City:	Good Towne
Phone:	678.555.1212	*State:	GA
Email:		*Zip Code:	30022
Re-type Email:		*Country:	United States

Billing Information

***First Name:**	Jane
***Last Name:**	Doe
Company:	
Phone:	678.555.1212
Email:	
Re-type Email:	

***Address:**	123 Sesame Street
***City:**	Good Towne
***State:**	GA
***Zip Code:**	30022
***Country:**	United States

Billing Information

*First Name: Jane

*Last Name: Doe

*Address: 123 Sesame Street

*City: Good Towne

*State: GA

*Zip Code: 30022

*Country: United States

Company:

Phone: 678.555.1212

Email:

Re-type Email:

Billing Information

Customer Name

*First: Jane

*Last: Doe

Billing Address

*Street: 123 Sesame Street

*City: Good Towne

*State: GA

*Zip Code: 30022

*Country: United States

Company Information

Name:

Phone: 678.555.1212

Email:

Re-type Email:

Production Support / Operations

Hidden Users

Production Support/Operations

Most common UI is: Log files

```
2014-02-25 23:15:02.517 Error:Ur
```

Production Support/Operations

Most common UI is: Log files

```
2014-02-25 23:07:01.020 Info:Pro
2014-02-25 23:15:02.517 Info:Sec
2014-02-25 23:15:04.213 Info:Dat
2014-02-25 23:15:04.562 Info: Pr
2014-02-25 23:15:04.666 Error:Ur
2014-02-25 23:15:04.823 Info: Pr
```

Production Support/Operations

- Ask them what would make their lives easier.
  - Monitoring and logging
- Make sure log files answer the important questions:
- When did it happen?
  - Use YYYY-MM-DD HH:MM:SS:mmm for sorting and duration calculations
- What happened? Standardize terminology on error levels
- What was impacted? Transaction IDs, file info, non-PI data involved.

Hidden Users

Production Support / Operations

QA / Testers

QA / Testers

Most common UI is the end users’.

May also use debug-level log file entries.

HTML with IDs

```
<form id="BillingInfo" action="billingInformationProcessor.html" method="post"
   <fieldset>
      <legend>Billing Information</legend>
      <p><label for="FirstName" id="FirstName">First:</label><input…
      <p><label for="LastName" id="LastName">Last:</label><input…
      <p><label for="Street:" id="StreetAddr">Street:</label><input…
      <p><label for="Apt" id="Apartment">Apt:</label><input…
```

Hidden Users

Production Support / Operations

QA / Testers

Other Developers

API users

API Users

Traditional UI: JavaDoc

Better UI: Unit Tests

Source Code Users

Traditional UI: Just the Source

Better UI: Executable Specifications and Unit Tests

```
public class Decider{
    public Decider(){
    }
    public boolean isIt(int number){
        if(number<=1) {
            return false;
        }
        List<Integer> values = new ArrayList<Integer>();
        values.add(1);
        values.add(number);
        for(int i=2;i<Math.sqrt(number);i++)
            if( number%i==0){
                values.add(i);
                if(number/i!= i)
                    values.add(number/i);
        }
        int value=0;
        for(Integer i:values)
            value+=i;
        return value-number==number;
    }
}
```

```
public class Decider {
    public Decider(){
    }

    public boolean isIt(int number){
        if(number<=1) {
            return false;
        }

        List<Integer> values = new ArrayList<Integer>();
        values.add(1);
        values.add(number);
        for(int i=2;i<Math.sqrt(number);i++)
            if( number%i==0){
                values.add(i);
                if(number/i!= i)
                    values.add(number/i);
        }

        int value=0;
        for(Integer i:values)
            value+=i;

        return value-number==number;
    }
```

```
public class Decider {
    public static boolean isIt( int number ) {
        if(number<=1) {
            return false;
        }

        List<Integer> values = new ArrayList<Integer>();
        values.add( 1 );
        values.add( number );
.        for( int i = 2; i<Math.sqrt(number); i++ )
            if(  number % i == 0) {
                values.add( i );
                if( number / i !=  i )
                    values.add( number / i );
        }

        int value = 0;
        for( Integer i :values )
            value +=  i;

        // return comparison
        return value-number == number;
    }
}
```

```
public class Decider {
    public static boolean isIt( int number ) {
        if(number<=1) {
            return false;
        }

        List<Integer> values = new ArrayList<Integer>();
        values.add( 1 );
        values.add( number );
.        for( int i = 2; i<Math.sqrt(number); i++ )
            if(  number % i == 0) {
                values.add( i );
                if( number / i !=  i )
                    values.add( number / i );
        }

        // calc total values
        int total = 0;
        for( Integer value : values ) {
            total += value;
        }

        // return comparison
        return value-number == number;
    }
```

```
public class Decider {
    public static boolean isIt( int number ) {
        if(number<=1) {
            return false;
        }

        List<Integer> values = new ArrayList<Integer>();
        values.add( 1 );
        values.add( number );

.        for( int i = 2; i<Math.sqrt(number); i++ )
            if(  number % i == 0) {
                values.add( i );
                if( number / i !=  i )
                    values.add( number / i );
        }

        // calc total values
        int total = 0;
        for( Integer value : values ) {
            total += value;
        }

        // return comparison
        return value-number == number;
```

```
public class Decider {
    public static boolean isIt( int number ) {
        if(number<=1) {
            return false;
        }

        // create and prepopulate list of values
        List<Integer> values = new ArrayList<Integer>();
        values.add( 1 );
        values.add( number );


        // add more to list of values
        for( int i = 2; i<Math.sqrt(number); i++ )
            if(  number % i == 0) {
                values.add( i );
                if( number / i !=  i )
                    values.add( number / i );
        }

        // calc total values
        int total = 0;
        for( Integer value : values ) {
            total += value;
        }
```

```
public class Decider {
    public static boolean isIt( int number ) {
        if(number<=1) {
            return false;
        }

        // create and prepopulate list of values
        List<Integer> values = new ArrayList<Integer>();
        values.add( 1 );
        values.add( number );

        // add more to list of value
        for( int i = 2; i < Math.sqrt(number); i++ ) {
            if(  number % i == 0) {
                values.add( i );

                if( number / i !=  i )
                    values.add( number / i );
            }
        }

        // calc total values
        int total = 0;
        for( Integer value : values ) {
            total += value;
```

```
public class Decider {
    public static boolean isIt( int number ) {
        if(number<=1) {
            return false;
        }

        // create and prepopulate list of factors
        Set<Integer> factors = new TreeSet<Integer>();
        factors.add( 1 );
        factors.add( number );

        // add more factors to the list
        for( int i = 2; i < Math.sqrt(number); i++ ) {
            if(  number % i == 0) {
                factors.add( i );
            }
        }

        // calc total values
        int total = 0;
        for( Integer value : values ) {
            total += value;
        }

        // return comparison
```

```
public class Decider {
    public static boolean isIt( int number ) {
        if(number<=1) {
            return false;
        }

        // create and prepopulate list of factors
        Set<Integer> factors = new TreeSet<Integer>();
        factors.add( 1 );
        factors.add( number );

        // add more factors to the list
        for( int i = 2; i < Math.sqrt(number); i++ ) {
            if(  number % i == 0) {
                factors.add( i );
                factors.add( number / i );
            }
        }

        // calc total values
        int total = 0;
        for( Integer value : values ) {
            total += value;
        }
```

```
public class Decider {
    public static boolean isIt( int number ) {
        if(number<=1) {
            return false;
        }

        // create and prepopulate list of factors
        Set<Integer> factors = new TreeSet<Integer>();
        factors.add( 1 );
        factors.add( number );

        // add more factors to the list
        for( int i = 2; i < Math.sqrt(number); i++ ) {
            if(  number % i == 0) {
                factors.add( i );
                factors.add( number / i );
            }
        }

        // calc total values
        int total = 0;
        for( Integer value : values ) {
            total += value;
        }
```

```
public class Decider {
    public static boolean isIt( int number ) {
        if(number<=1) {
            return false;
        }

        // create and prepopulate list of factors
        Set<Integer> factors = getFactors( number );

        // calc total values
        int total = 0;
        for( Integer value : values ) {
            total += value;
        }

        // return comparison
        return total - number == number;
    }
}
```

```
public class Decider {
    public static boolean isIt( int number ) {
        if(number<=1) {
            return false;
        }

        // get list of factors
        Set<Integer> factors = getFactors( number );

        // get the sum of the factors
        int total = sumFactors( factors );

        // return comparison
        return total-number == number;
    }
}
```

```
public class Decider {
    public static boolean isPerfectNumber( int number ) {
        if(number<=1) {
            return false;
        }

        Set<Integer> factors = getFactors( number );

        int sumOfFactors = sumFactors( factors );

        return sumOfFactors - number == number;
    }
}
```

Hidden Users

Production Support / Operations

QA / Testers

Other Developers

UX Designers, Business Analysts, etc.

Parting thoughts

There is ALWAYS a user experience

Write and test your code as if the person maintaining it is a homicidal maniac who knows where you live.

Questions?

Email: Burk.Hufnagel@Daugherty.com

Blog: MindLikeaSword.blogspot.com