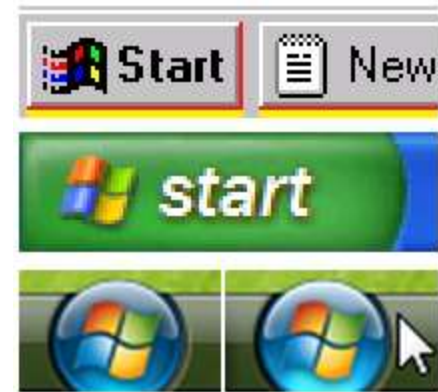


הלל: מה ששנא עליך על
תעשה לחברך



הנדסת תוכנה

12. חווית משתמש – UX כלים ||

"I have always wished for my computer to be as easy to use as my telephone; my wish has come true because I can no longer figure out how to use my telephone"

[Bjarne Stroustrup](#) (C++ Creator)



http://groups.google.com/group/microsoft.public.powerpoint/browse_thread/thread/27c8f0b03f69fd4d?hl=en&ie=UTF-8&q=powerpoint+undo+deleted+slides#6608dc06dad9ca8a



microsoft.public.powerpoint

can i get a deleted slide back?

★ 3 messages - [Collapse all](#)

Crowe [View profile](#)

I **deleted** a slide from my **powerpoint** slide and hit "save". I need that slide back! Any way to get it???

sigh

Thanks!

[Reply](#) [Reply to author](#) [Forward](#)

Rate this post: ★★★★★

Sandy Johnson [View profile](#)

Dear Crowe,

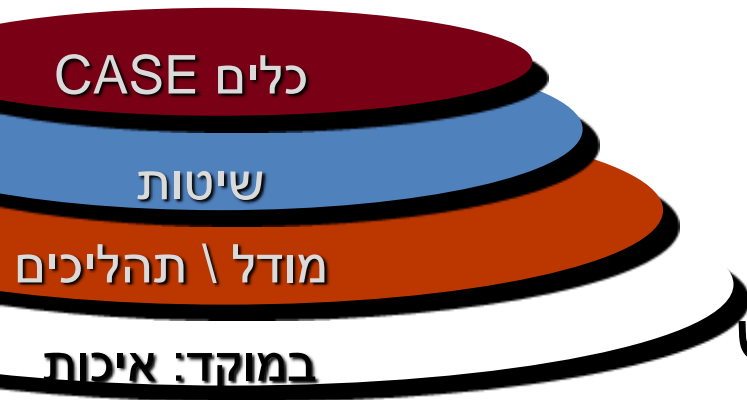
Sorry can't get it back -- unless you saved it earlier under a different name or perhaps emailed the file to someone previous to saving it...

Iran Air Flight 655

- “Iran Air Flight 655 was shot down by the USS *Vincennes*' Aegis system in 1988, killing 290 people. The error was initially attributed to operator error, but later some experts attributed the incident to the poor design of the Aegis user interface.”



היום?



- חווית משתמש
 - שמישות \ עיצוב ממשק משתמש
 - GUI מסורתי, web והתקנים ניידים
- כלים מתקדמים
- תרגיל/פרויקט – משימת סבב אחרון
 - Code Review with pull requests
 - בנוס סביבת CI / CD

Brooks (MMM): Question:
How does a large software
project get to be one year
late?
Answer: One day at a time!

Usability in the Front

- [Windows 8 UX Manager](#)
- [The Science and Art of User Experience at Google](#) (gtalk example)

[Web](#) [Images](#) [Videos](#) [Maps](#) [News](#) [Shopping](#) [Gmail](#) [more ▾](#)

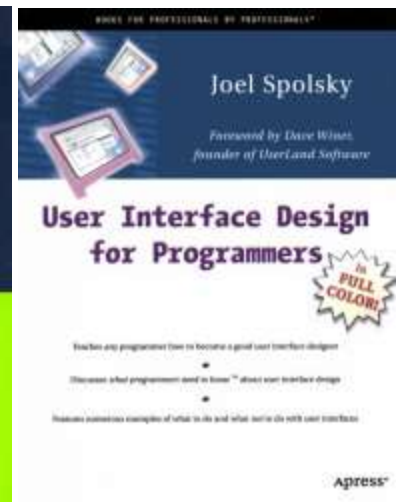
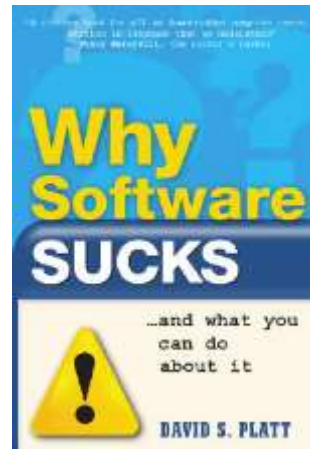
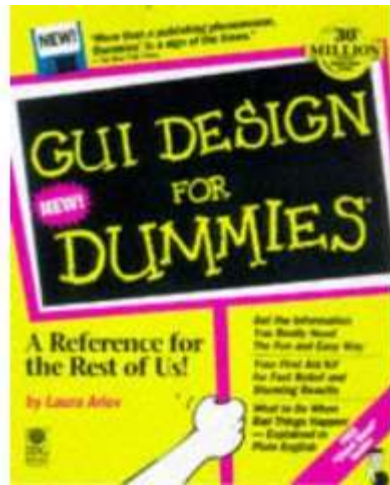
Reuven Y.



[Advanced search](#)
[Language tools](#)

מקורות

- Pressman Chap. 12
- ברק דנין, מדריכי שמישות (ול-סמארטפון)
- Amber, Agile Usability
- Spolsky, User Interface Design For Programmers
<http://www.joelonsoftware.com/printerFriendly/uibook/fog0000000249.html>
- Usability In Practice, MSDN series



קישורים

- <http://uxi.org.il/pages/10316> חווית משתמש מבוא
- http://www.developer.nokia.com/Design/User_experience/ נוקיה
- Platt, [Using WPF for Good and Not Evil](#)
- [mit course](#)
- [The Encyclopedia of Human-Computer Interaction](#)
[Ency. chapter: User Experience and Experience Design](#)
(+ interview)
- Bloch, [How to design a good API and why it matters](#), 2006 Google paper
- Hong, [Why is Great Design so Hard](#) ("So what is Apple doing right?"), blog post 2010

קישורים

Jakob Nielsen, Why You Only Need •
to Test With 5 Users

<http://www.useit.com/alertbox/20000319.html>

- Constantine, “*What do users want?* Engineering Usability into Software”
<http://www.foruse.com/articles/whatusers.pdf>
- Nielsen, [URL as UI](#)
- Patterns: <http://quince.infragistics.com>
- <http://en.wikipedia.org/wiki/Usability> (עברית)
- http://en.wikipedia.org/wiki/Web_usability
- http://en.wikipedia.org/wiki/User_experience_design
- [Web Style Guide](#) an online book
- <http://www.worldusabilityday.org/>

ועוד קישורים

- <http://www.usabilityfirst.com/index.txt>
 - [Usability First: Usability Glossary: usability](#)
 - [Usability First: Usability Glossary: affordance](#)
- http://www.snyderconsulting.net/article_7tricks.htm
 - http://www.snyderconsulting.net/article_paperprototyping.htm
- <http://www.uie.com/>
- <http://jerz.setonhill.edu/design/usability/intro.htm>
- **Windows User Experience Interaction Guidelines**
 - <http://msdn.microsoft.com/en-us/library/aa511258.aspx> (video)
- IE9 Usability <http://windows.microsoft.com/en-US/internet-explorer/products/ie-9/features/focused-on-your-websites>
- <http://www.useit.com/>
- <http://www.usability.gov/>
- פשוט שימושי <http://www.usable.co.il/>, <http://www.uex.co.il/>

מאפייני מערכת לדוגמא

- פונקציונליות
- גודל
- ביצועים
- מחיר
- אמינות
- בטיחות
- סטנדרטים
- חווית\ממשק משתמש\שמישות
- בתיכון מערכת צריך לשקלל מאפיינים שונים
- היום דגש מסוים

סוגי ממשקים

- ממשק בין רכיבים
- ממשק למערכות חיצוניות
- ממשק משתמש

שלוש חוקי הזהב [Mandel 97]

- Place the user in control
- Reduce the user's memory load
- Make the interface consistent

Place the User in Control

- Define **interaction** modes in a way that does not force a user into unnecessary or undesired actions.
- Provide for flexible interaction.
- Allow user interaction to be interruptible and undoable.
- Streamline interaction as skill levels advance and allow the interaction to be customized.
- Hide technical internals from the casual user.
- Design for direct interaction with objects that appear on the screen.

Reduce the User's Memory Load

- Reduce demand on short-term memory.
- Establish meaningful defaults.
- Define shortcuts that are intuitive.
- The visual layout of the interface should be based on a real world metaphor.
- Disclose information in a progressive fashion.

Make the Interface Consistent

- Allow the user to put the current task into a meaningful context.
- Maintain consistency across a family of applications.
- If past interactive models have created user expectations, do not make changes unless there is a compelling reason to do so.

ממשק משתמש – לא רק ציור

- עיצוב אינטראקטיביות משתמש
- ארגון המידע
- מחקר משתמשים
- עיצוב ויזואלי

חווית משתמש (נילסן\נורמן)

"חווית משתמש כוללת את כל האספקטים של האינטראקציה בין משתמש הקצה לבין החברה, השירותים שלה והמוצרים שלה. הדרישה הבסיסית לחווית משתמש מעולה היא לענות על הצרכים המדויקים של המשתמש, ללא סיבוכים מיותרים. על המוצר או השירות להיות פשוט ואלגנטי, **כך שיהיה כיף להחזיק בו, וכיף להשתמש בו.** חווית משתמש אמיתית נותנת ללקוחות הרבה יותר מאשר בדיוק את מה שהם אומרים שהם רוצים, או ממלאת אחרי רשימת יכולות ודרישות. כדי להגיע לחווית משתמש באיכות גבוהה על-פני כל החברה, חייב להיות **רצף אחד על-פני כל השירותים** שהיא מספקת בתחומים שונים, כולל הנדסה, שיווק, עיצוב גראפי ותעשייתי, ועיצוב ממשק".

Usability - שמישות

- מידת האפקטיביות שבה משתמש יכול להשיג את מטרותיו עם התוכנה
- חלק מ- Human Computer Interaction, חיתוך עם מדעים קוגניטיביים \ ארגונומיה-פיזיולוגיה \ הנדסת אנוש
- המשתמש במרכז!
- ISO Definition
- designed for use (video)

DILBERT by Scott Adams





מאפייני שמישות (Nielsen)

- **לימודיות** - המידה שבה הממשק ניתן ללמידה על ידי המשתמש
- **יעילות** - המידה שבה השימוש בממשק הוא יעיל
- **זכירות** - מידת הקלות שבה המשתמש זוכר את פעולות הממשק.
- **שגיאות** - כמות הטעויות הפוטנציאלית הקיימת בממשק
- **שביעות רצון** - מידת שביעות הרצון של המשתמש מהממשק

Affordance - מושג קרוב: נגישות |

- מצב שבו המאפיינים החושיים של אובייקט מרמזים על אופן השימוש בו

– כפתור בולט, מציע שאמורים בד"כ ללחוץ עליו
– פינה עבה של חלון מרמזת על אפשרות להגדלה



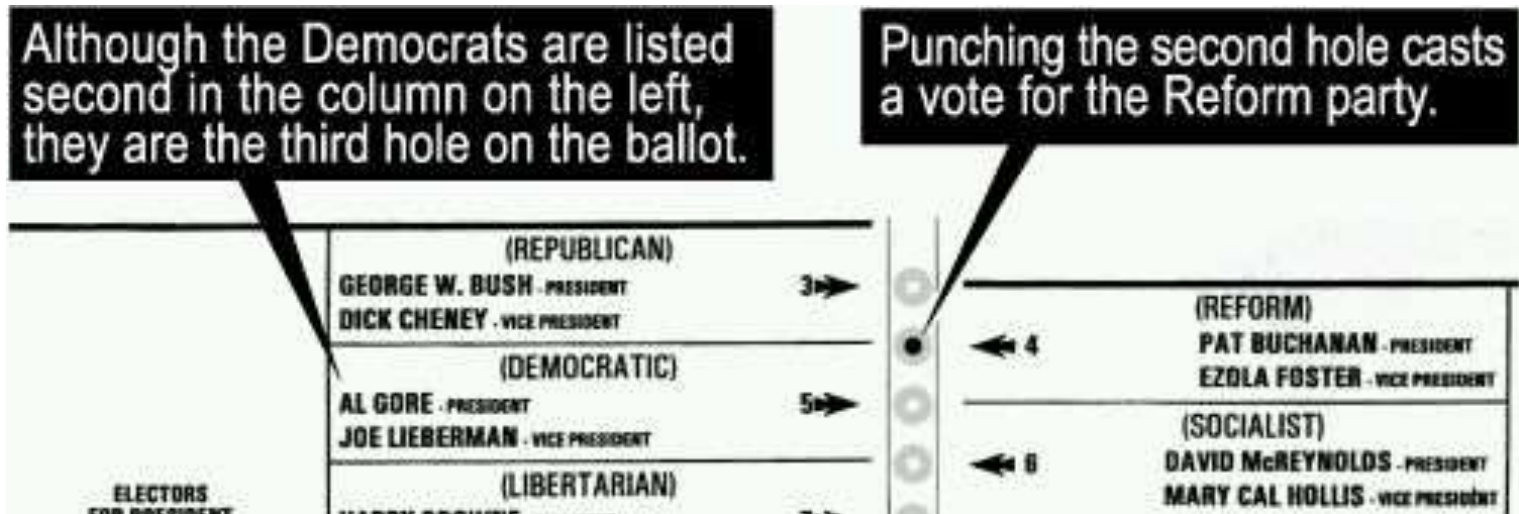
– תכונה רצויה לממשק משתמש, עוד בהמשך...

נגישות II - Accessibility

- נגישות למרות מוגבלויות שונות
 - ראייה
 - מוטוריקה
- נגישות למניעת מחלות וכדו'
- תקנים:
- Web Content Accessibility Guidelines (WCAG)
 - <http://www.w3.org/TR/WCAG20/>
- <http://www.access-board.gov/508.htm>
- Tools: Herra, Booby

שמישות וממשק משתמש

- קשר הדוק בין השניים
- לממשק גרוע יכולה להיות השפעה רבה



איך משיגים שמישות?

- מבדקי שמישות (Usability Testing)
 - Eye Tracking
- ראיונות\סקרים עם מומחים\ קבוצות מיקוד
- אב טיפוס
 - נייר
 - קוד (VB, Rails)
- פיתוח איטרטיבי
 - במיוחד ברשת
- תהליכי עיצוב



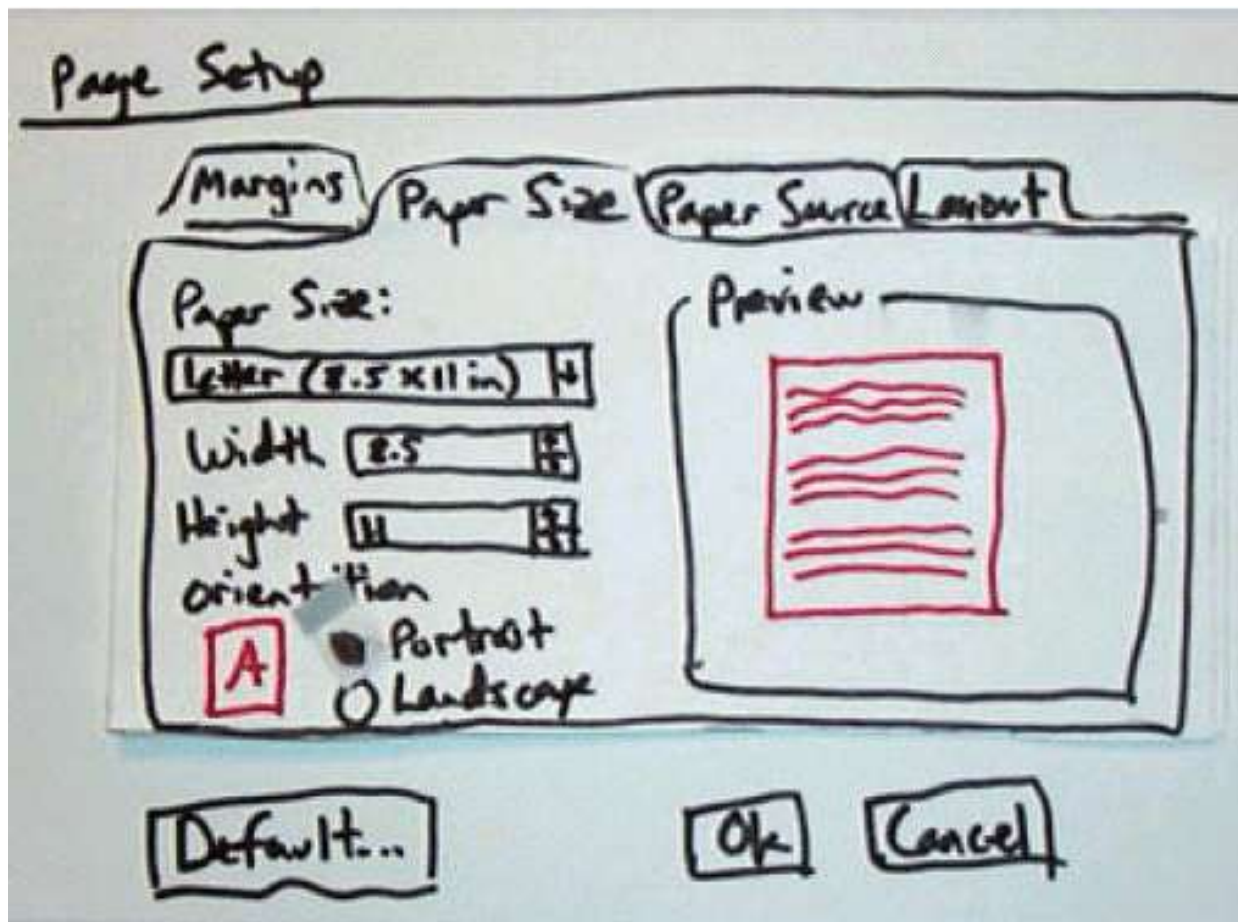
DILBERT reprinted by permission of United Feature Syndicate.

אב טיפוס מנייר

- משחק תפקידים, ללא מחשב



אב טיפוס מנייר



יתרונות לנייר

- בדיקת הרעיונות לפני המימוש
- ביצוע שינויים מהיר
- אפשרות לגלות מה באמת צריך
- נטרול משתנים טכנולוגיים ומימושיים
 - עוד צעד לקראת הלקוח
 - שינויים נראים קלים יותר
 - התמקדות בדברים העקרוניים

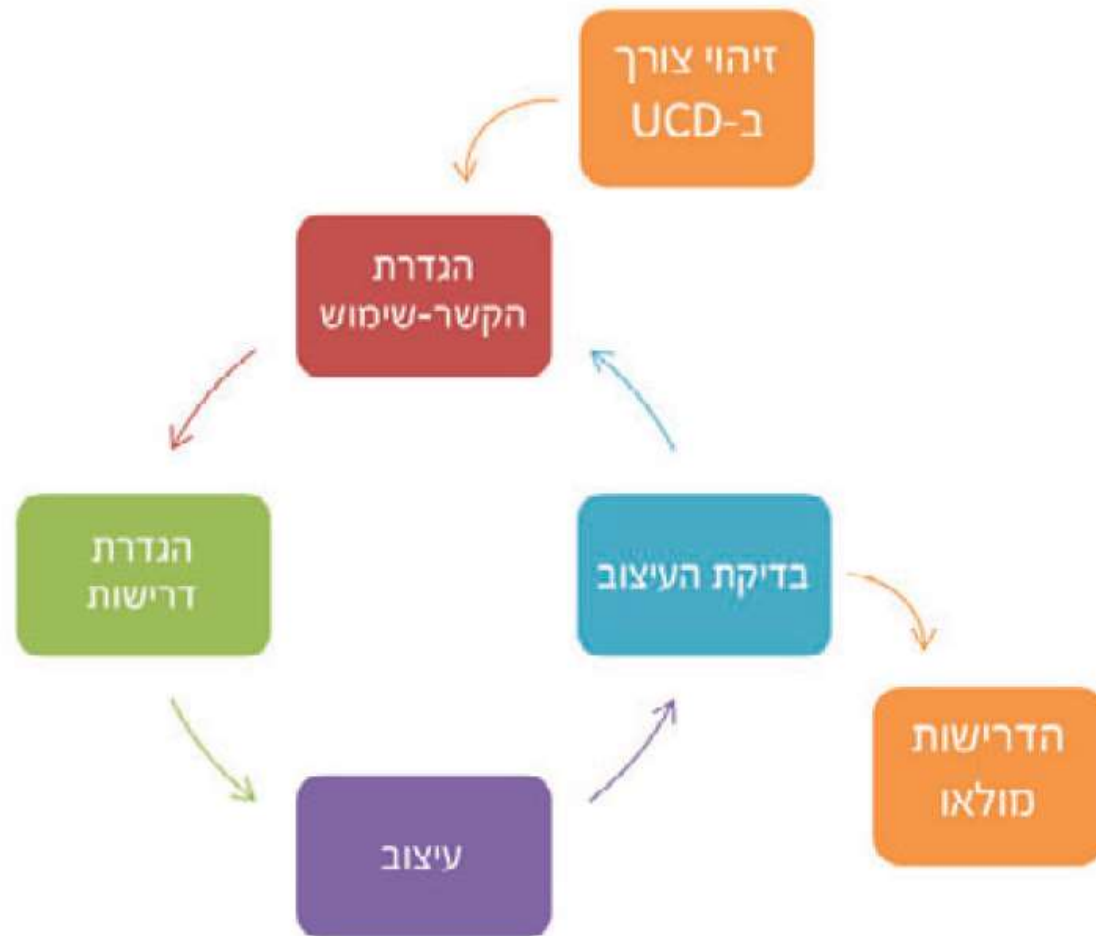
בתוכנה

- סביבות פיתוח מהירות, למשל:
 - [Ruby on Rails](#)
 - Nancy, [ASP.NET MVC](#)
- סביבות אבות טיפוס:
 - לאתרים: <https://gomockingbird.com/>
 - [SketchFlow](#), MS Expression Blend [Video](#)
 - [BalsamiQ Mockups](#), [וידאו](#)
 - <http://www.pretotyping.org/androgen>
 - <http://astrails.com/> איך עושים זאת? למשל:
 - VS2012 Story Boarding
- בדיקות אוטומטיות, למשל [Selenium](#)
 - Julian Harty, [Finding Usability Bugs with Automated Tests](#)

איזה סוג בדיקה כללית שלמדנו הכי משמעותי לחוויית משתמש

1. בדיקות קבלה (Acceptance)
2. בדיקות שמישות (מעבדת שמישות)
3. סקר דרישות
4. בדיקות יחידה

תהליך עיצוב מוכוון משתמש (UCD)



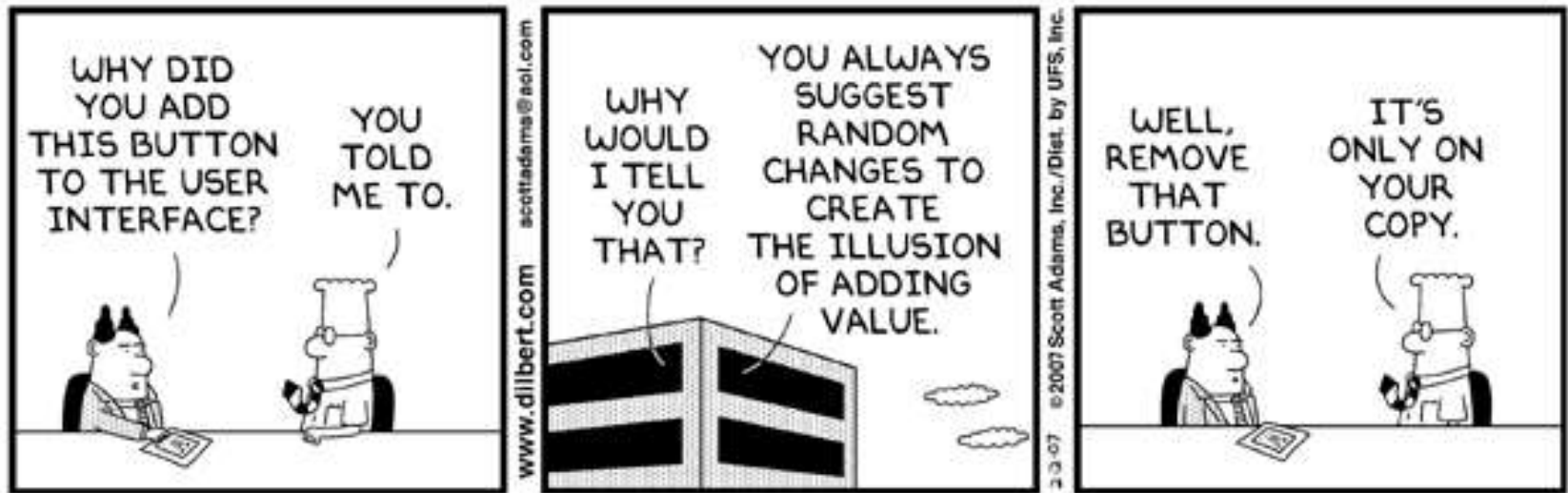
מתי בודקים?

- “The rule of thumb...is that the cost-benefit ratio for **usability** is **\$1:\$10-\$100**. Once a system is in **development**, correcting a problem costs 10 times as much as fixing the same problem in **design**. If the system has been **released**, it costs 100 times as much relative to fixing in design.” (Gilb, 1988, [here](#))

User Centered Design •

עוד לא בדקתם? •

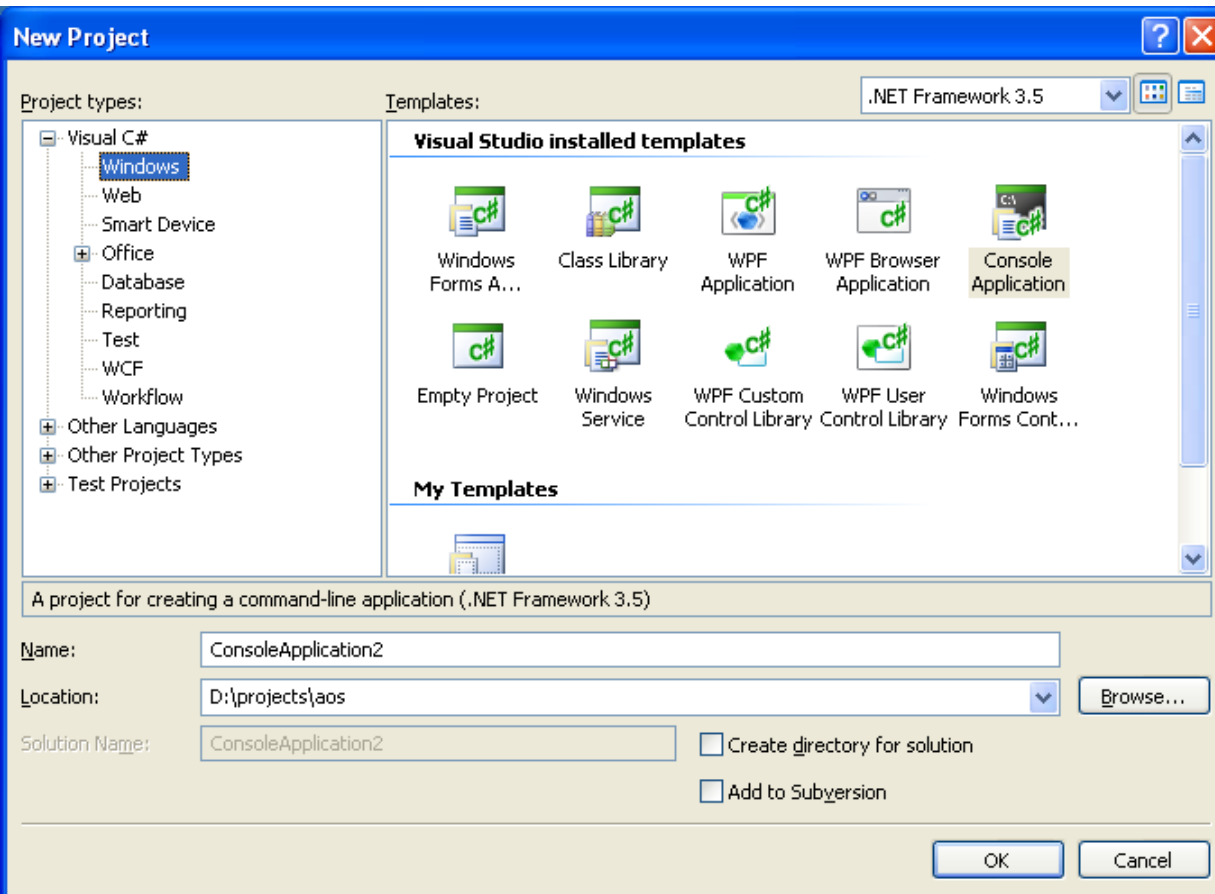
כמה עקרונות בסיסיים לתוכניות חלונאיות



© Scott Adams, Inc./Dist. by UFS, Inc.

מתי כדאי להשתמש ב:

- כפתור?
- תיבת סימון?
- כפתור רדיו?
- שדה טקסט?
- רשימה/נגללת?
- עץ?
- תפריט?
- תיבת שיחה?
- אחרים....?

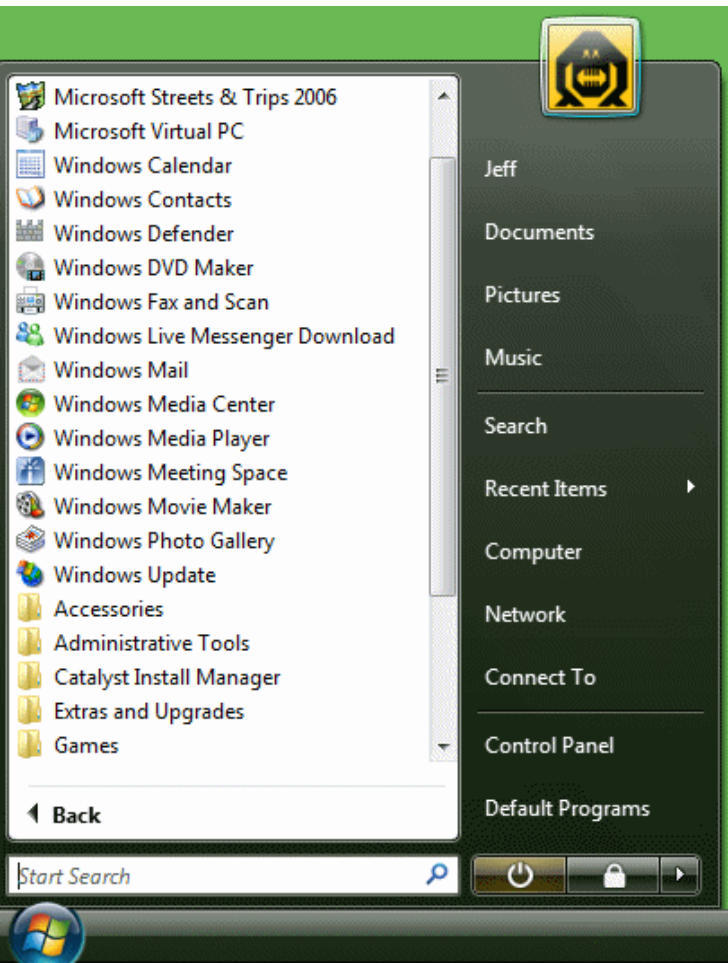


כפתורים, תפריטים

- כפתור – עבור פעולות עצמאיות, רלוונטיות למסך הנוכחי

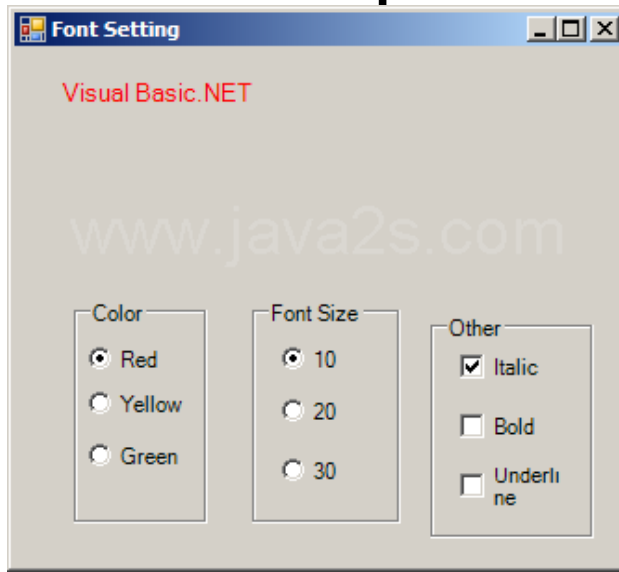
– מתי שמים ... (שלש נקודות)?

- סרגל כלים – פעולות נפוצות
- תפריט – פעולות פחות נפוצות
ישימות למספר מסכים
(או כולם)



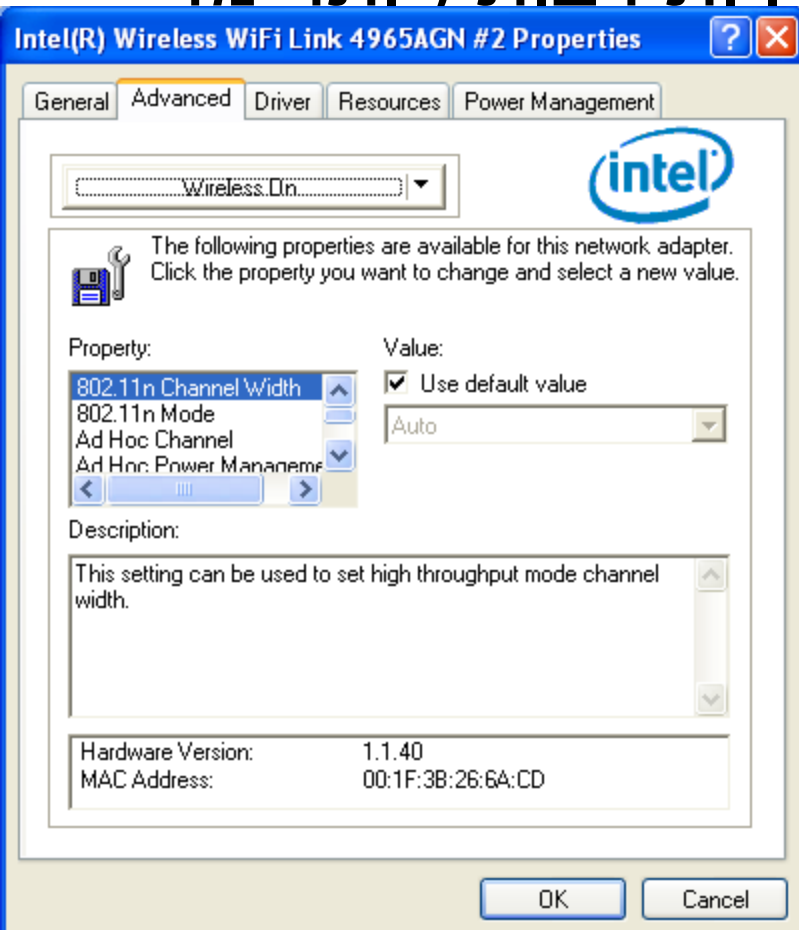
תיבות סימון

- Check-box – מפסקי הפעלה וכיבוי, כל אחד בלתי תלוי באחרים (לרוב תואם לערכים בוליאניים)
- Radio button – אפשרויות קשורות שרק אחת צריכה להיבחר (לרוב תואם לקבועים\enum)



רשימות

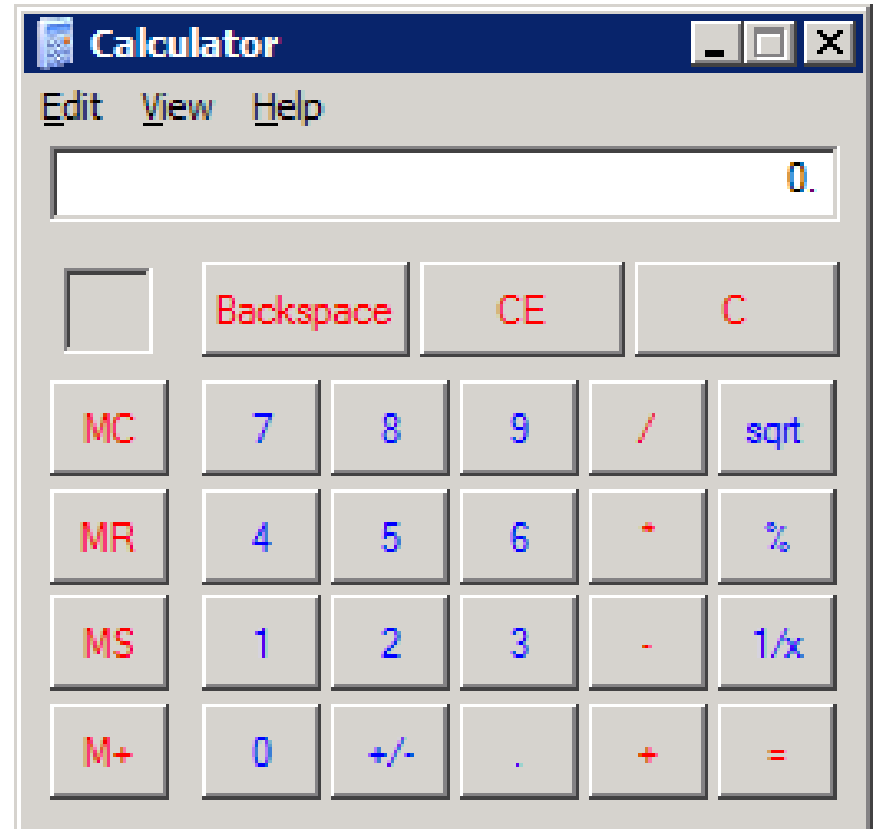
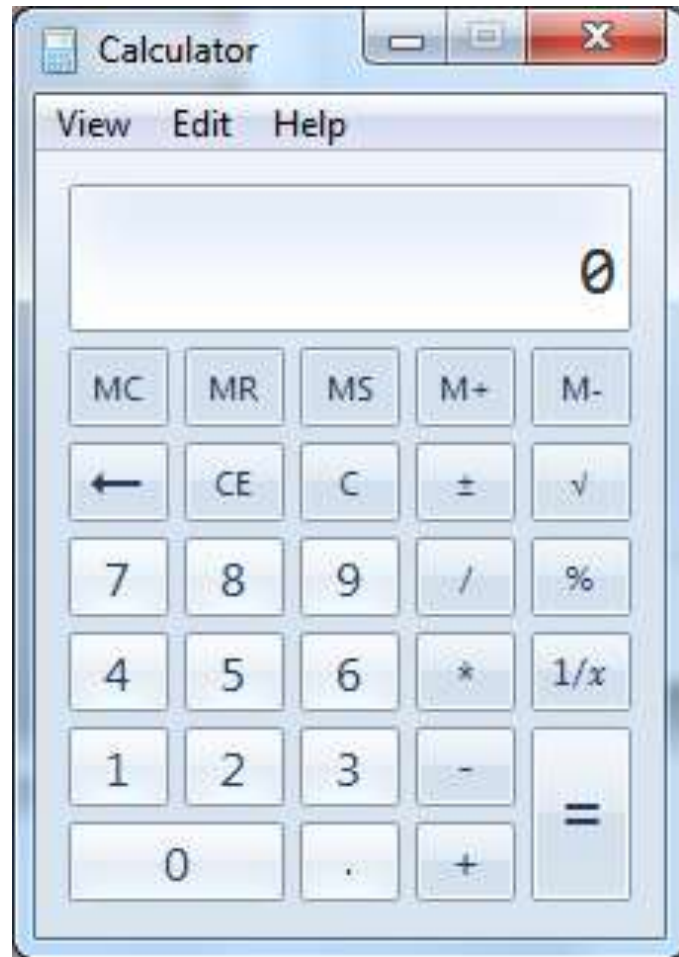
- טקסט (בד"כ בצרוף תגית) – קלט חופשי
- רשימה – בחירה מתוך אפשרויות רבות (יותר מדי בשביל רדיו) ורוצים שיראו את כולן
- Combo-box – כנ"ל, אך רוצים לחסוך במקום על המסך



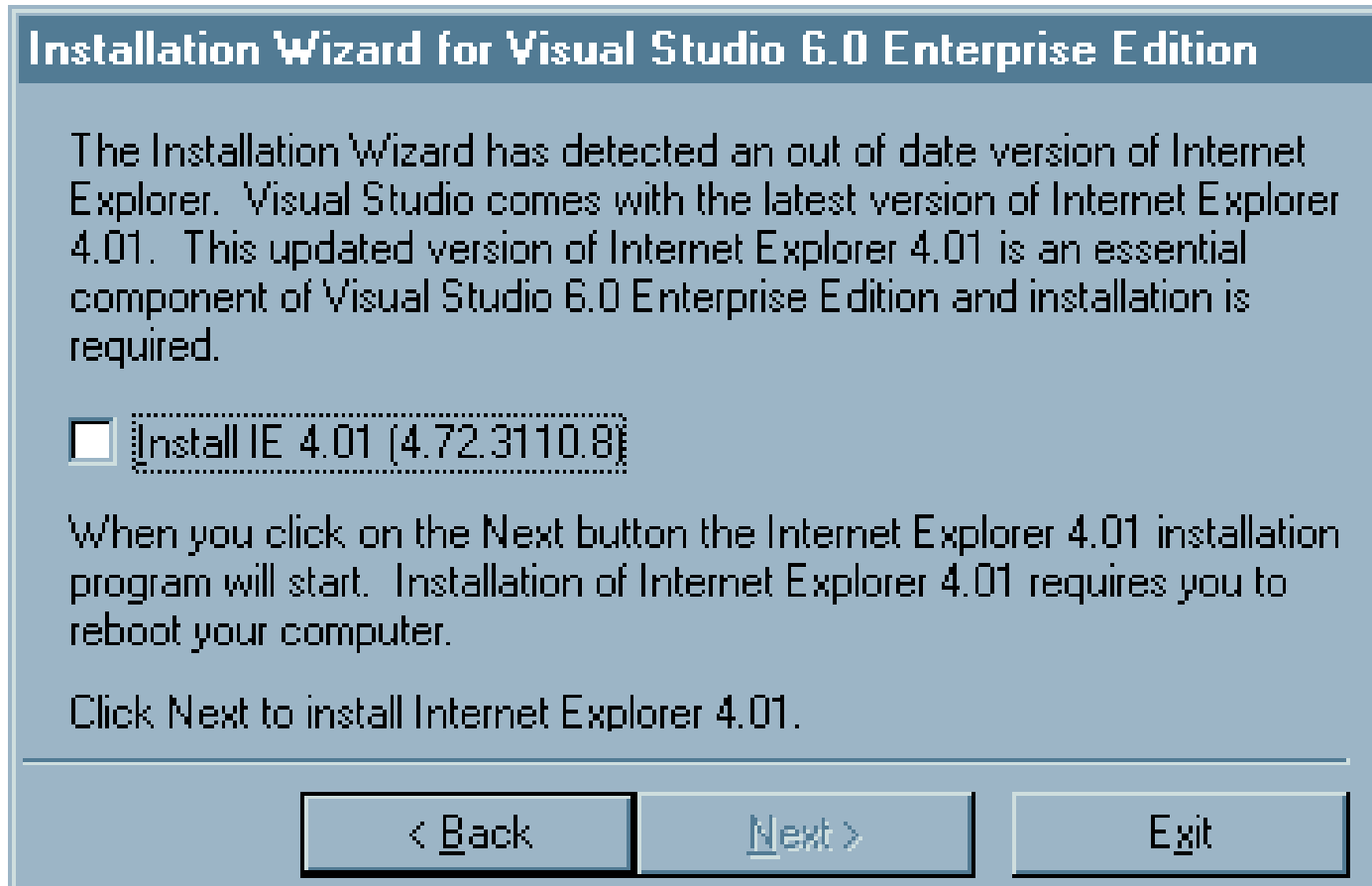
מסכים מרובים

- Tabbed Pane – המשתמש יכול לעבור בכל רגע בין המסכים השונים
- אשף (wizard) – הנחיית המשתמש לאורך תהליך
- תיבת שיחה – הצגת מסכים זמניים או תיבת מאפיינים

על ספסל הנאשמים



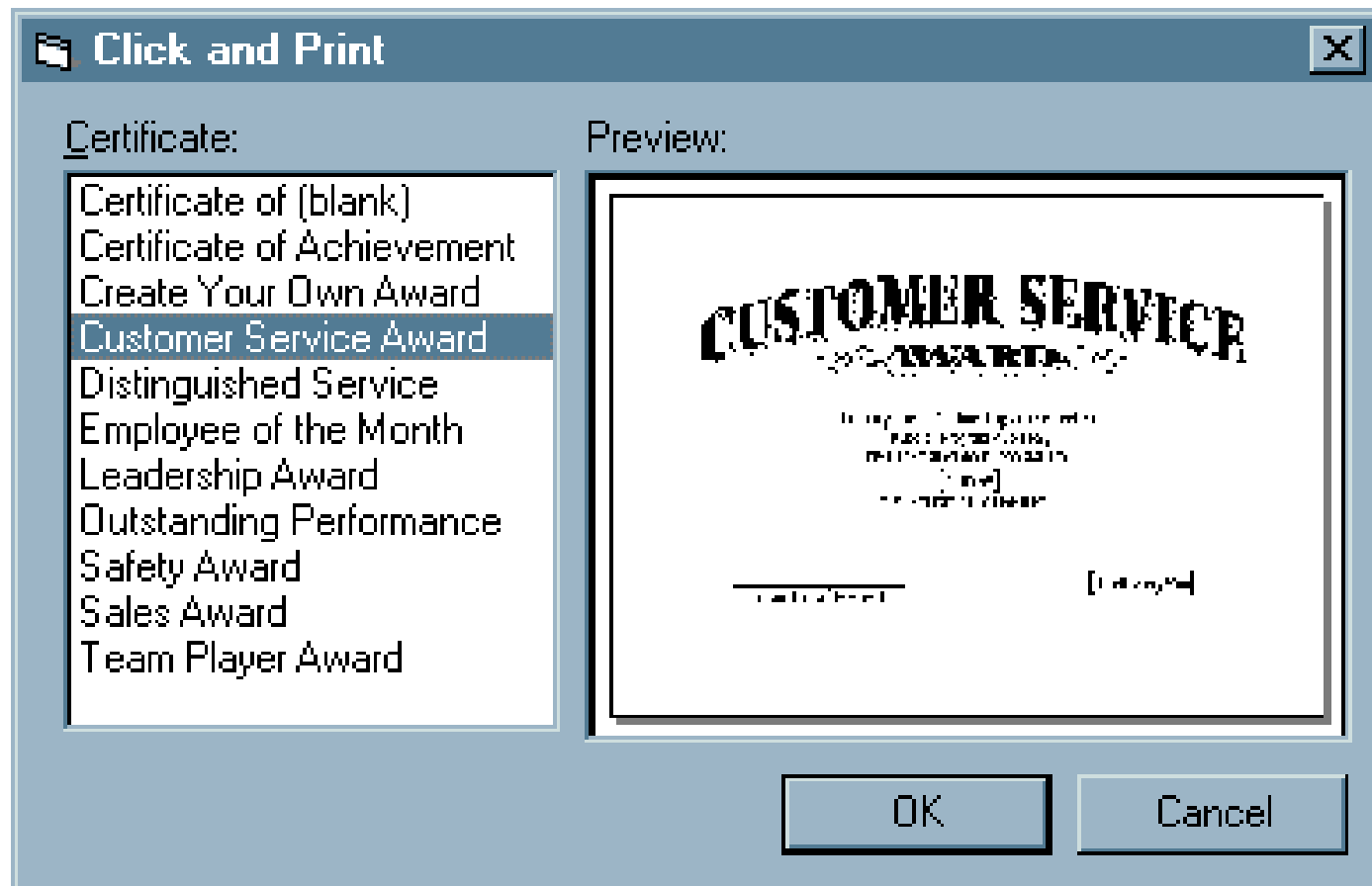
Interface Hall of Shame - דוגמאות



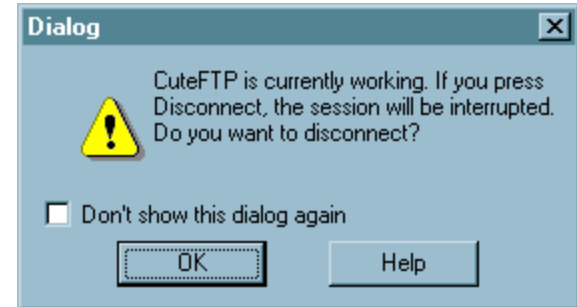
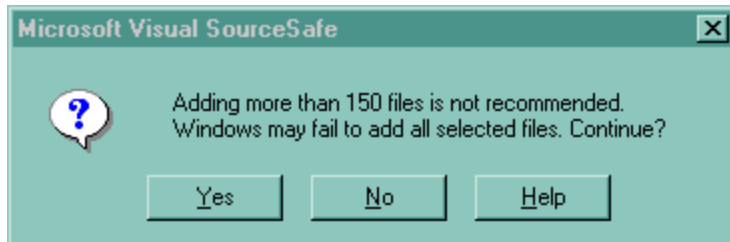
Interface Hall of Shame - דיון



קצת יותר טוב



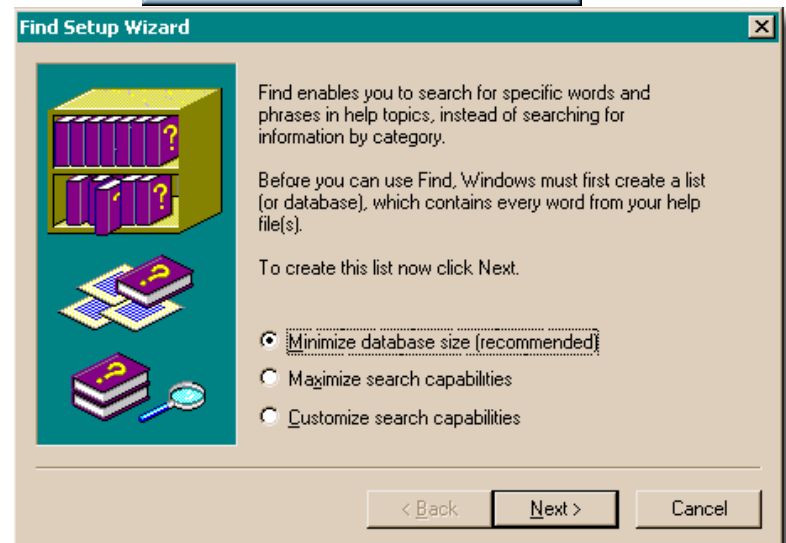
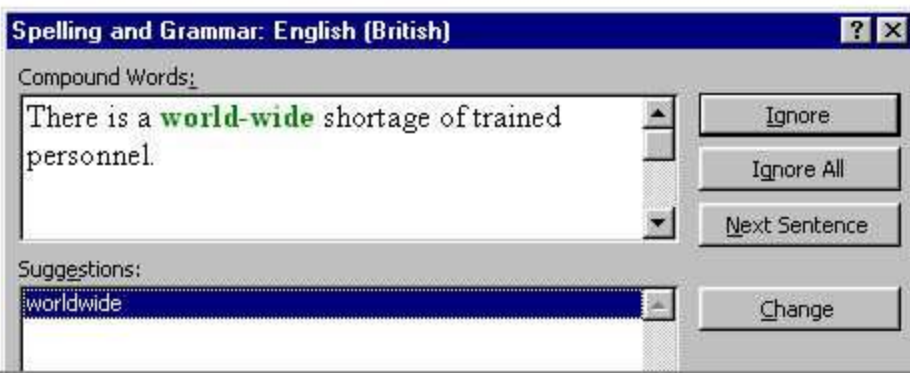
עוד דוגמאות והודעות שגיאה



There is a worldwide shortage of trained personnel.



There is a world-wide shortage of trained personnel.



שמישות ברשת (+300M\$)

Login

You must be a registered user to proceed. Please login or register below.

Already registered? Login

Username:

Password:

Login

☐ Remember my username on this machine.

[Forgot your login information? Click here.](#)

New to the Site?

Sign up in a few easy steps!

Register Now

Login

You must be a registered user to proceed. Please login or register below.

Already registered? Login

Username:

Password:

Login

☐ Remember my username on this machine.

[Forgot your login information? Click here.](#)

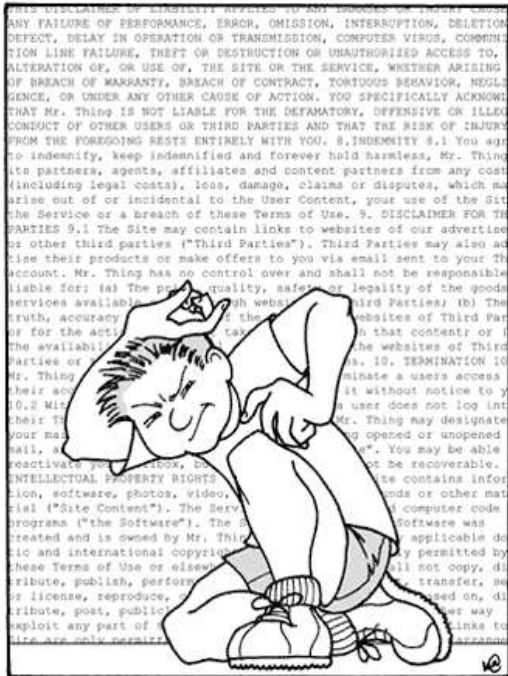
New to the Site?

Continue

You do not need to create an account to make purchases on our site. Simply click **Continue** to proceed to checkout.

To make your future purchases even faster, you can create an account during checkout.

טעויות נפוצות בעיצוב רשת



- עמודים עמוסים
- התאמה לרזולוציות שונות
- התאמה לתקנים/דפדפנים שונים
- התאמה לאיזורים שונים
- "קירות" של טקסט
- עיצוב סטנדרטי

Jakob's Law of the Web User Experience states that "users spend most of their time on *other* websites."

דברים שלא כל המשתמשים יודעים

- הלוגו מקשר לעמוד הבית



- אבטחה מול פרטיות



Billing Address

This is where you enter or review the billing information for your order. **The billing address needs to match the one on your credit card account.** Fields marked with an asterisk (*) must be completed for the order to be processed.

(Questions about your privacy? [Your Privacy & Security](#))

Title: *First name: *Last name:
*Address:

NetBank® Banking how you live.
Member FDIC

Apply Now

Get Free Bill Pay & Pay No Fees
NetValue Interest Checking

Only \$50 deposit to open, free unlimited transactions. [Apply Now](#)

Banking ▶
Investments ▶
Loans
About NetBank
Demos
Contact Us
Home
Security & Privacy
Site Map

Overview
Home Equity
Mortgages
Business Leasing
Loan Glossary
Loan Disclosures

| Services | Our Rates |
|--|---------------------------------|
| Money Market Account | 4.75% APY* open |
| 1 Year CDs (other terms available) | 5.26% APY* open |
| Home Equity Loans (intro rate fixed for 3 months) | 6.99% APR open |

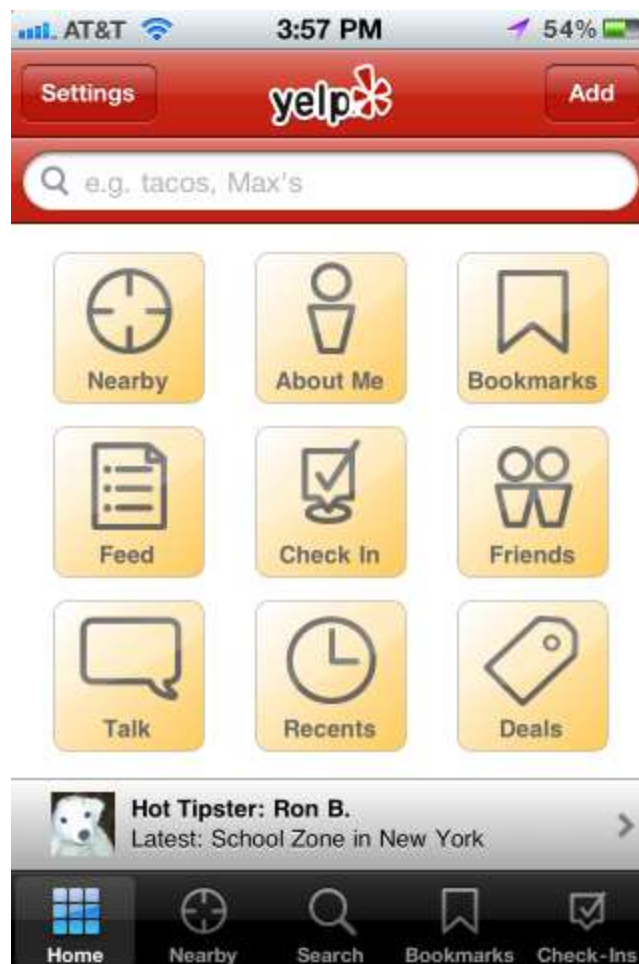
- שימוש בתפריטים נגללים
- כפתורי חיצים לרשימות
- ניווט באמצעות URL-ים
- הכרות עם כפתורי הדפדפן
- פתיחת חלון שני
- Web 2.0

דוגמת שמישות ברשת

<http://graphs.gapminder.org/world/> •

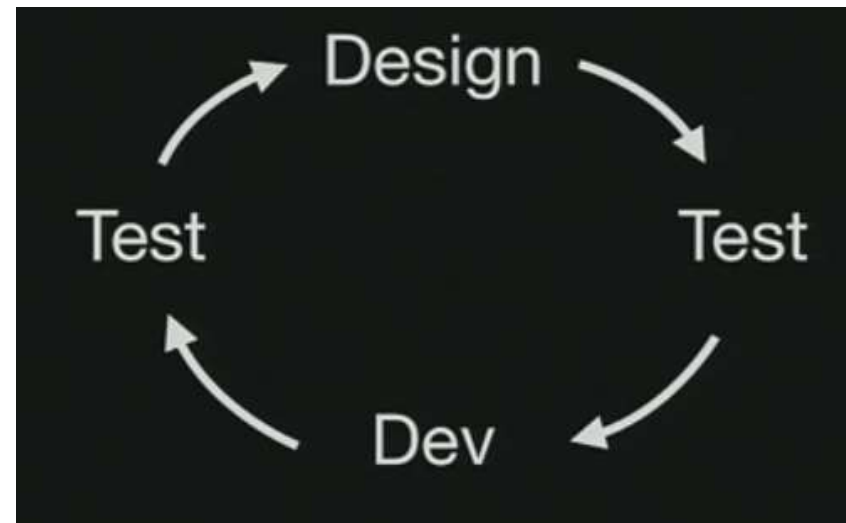


שמישות במחשוב נייד



עיצוב חווית משתמש

- Aral Balkan, **“A happy grain of sand”**
 - [Flush](#)
 - [Elevator](#)
 - [Washing Machine](#)
 - [DVD Burner @ Apple](#)
 - Ticket Machine
 - [Phone network](#)



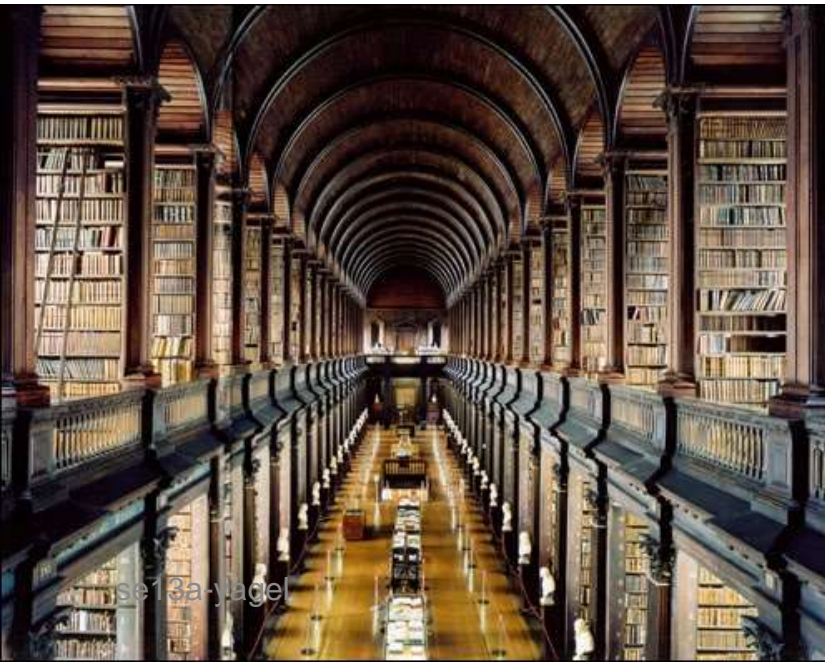
תרגיל עיצוב

- שרטטו ממשק משתמש לחיפוש בספרייה, הכולל את המאפיינים הבאים:

- אפשרות לחיפוש מחרוזת כמחבר, כותר או נושא

- אפשרות לחפש ספר או כתב-עת

- אפשרות לצפות במספר תוצאות מסודרות ע"פ זמינות או תאריך הוצאה
(אך לא שתי האפשרויות)



חווית משתמש ואנחנו

• ~~סקר שמישות עם נציג הלקוח במהלך הסבב~~

• ~~שיפורים בסבב הבא~~

• משוב (מקוון) מהמשתמשים, למשל

– UserVoice

Google moderator, idea informer, ideaTorrent, –
etc...



סיכום - UX

- חווית משתמש
- מי מאפיין? מתי? כיצד בודקים?
- דרישות משתמשים
- עיצוב חווית משתמש כמקצוע
- – ארגונומיה, עיצוב גראפי, קוגניציה, ניסוח טקסט, Gamification
- – מתי באחריות מהנדס התוכנה?
- – לקחים ל: API Usability, או
- Cambridge: Usability of Programming Languages
- MIT: User Interface Design & Implementation, קורס: בחירה,

כלים ||

- ראינו: תהליכים, שיטות וכלים (למשל: בדיקות יחידה)

- היום: כלים משלימים:

- סקרי קוד

- תיעוד

- בדיקות קבלה

- בדיקות כיסוי

- שילוב מתמשך והפצה מתמשכת



קישורים

- אלעד סופר: כלים אג'ילים – יש דבר כזה?
- משה קפלן: סקרי קוד, כלי הפיתוח שאתם פשוט חייבים
- Continuous Delivery: Anatomy of the Deployment Pipeline, Book chapter, 2010
- CI webcast
- Roy Osherov: Code Review detailed post

1. מהו סקר קוד (Code Review)?

- בחינה עקבית של הקוד
 - במטרה למצוא באגים ולתקנם $2 < 1 + 1$
 - שיפור הקוד ומיומנות המפתח

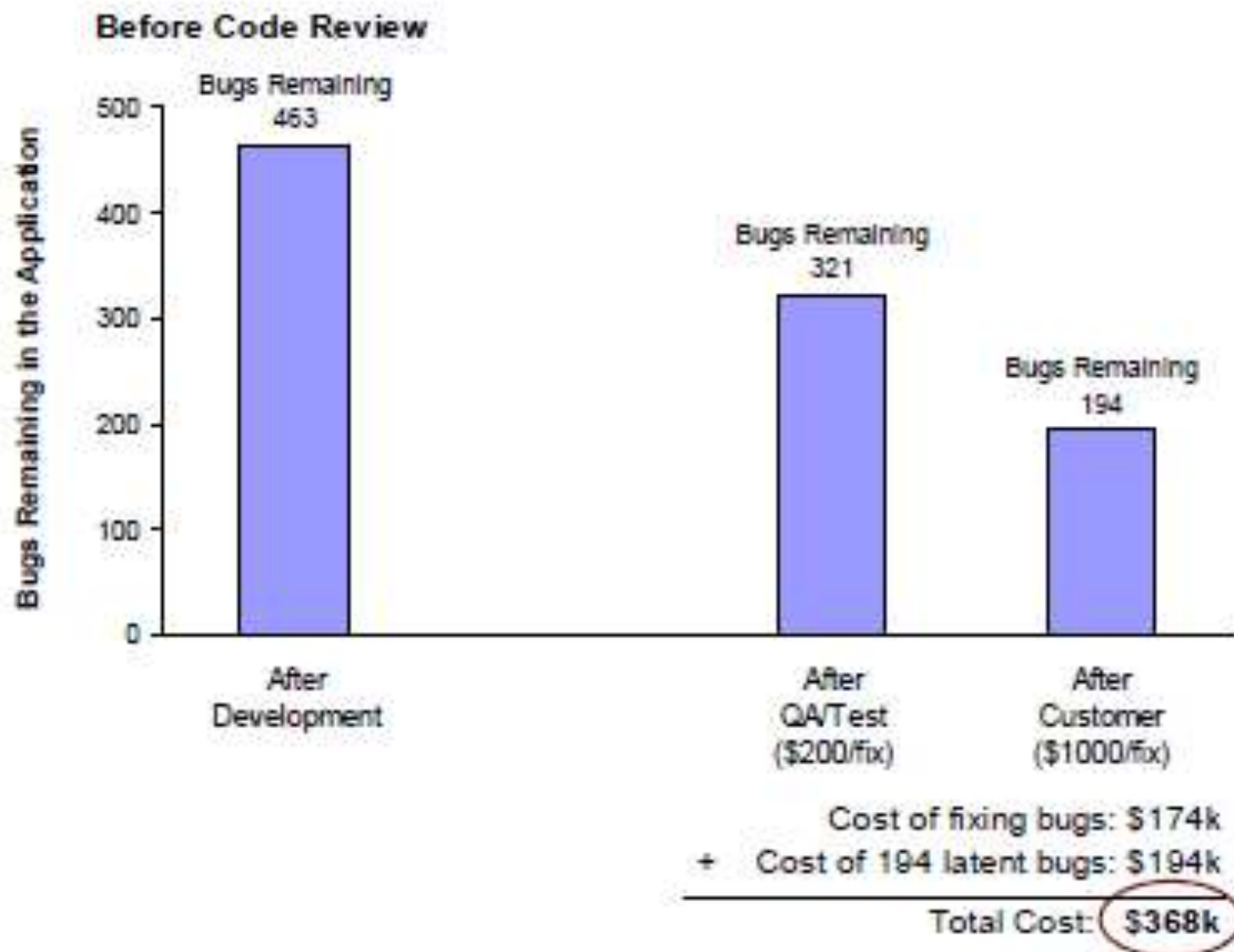


- האם אתם עושים זאת?
- האם אפשר לא לעשות זאת?

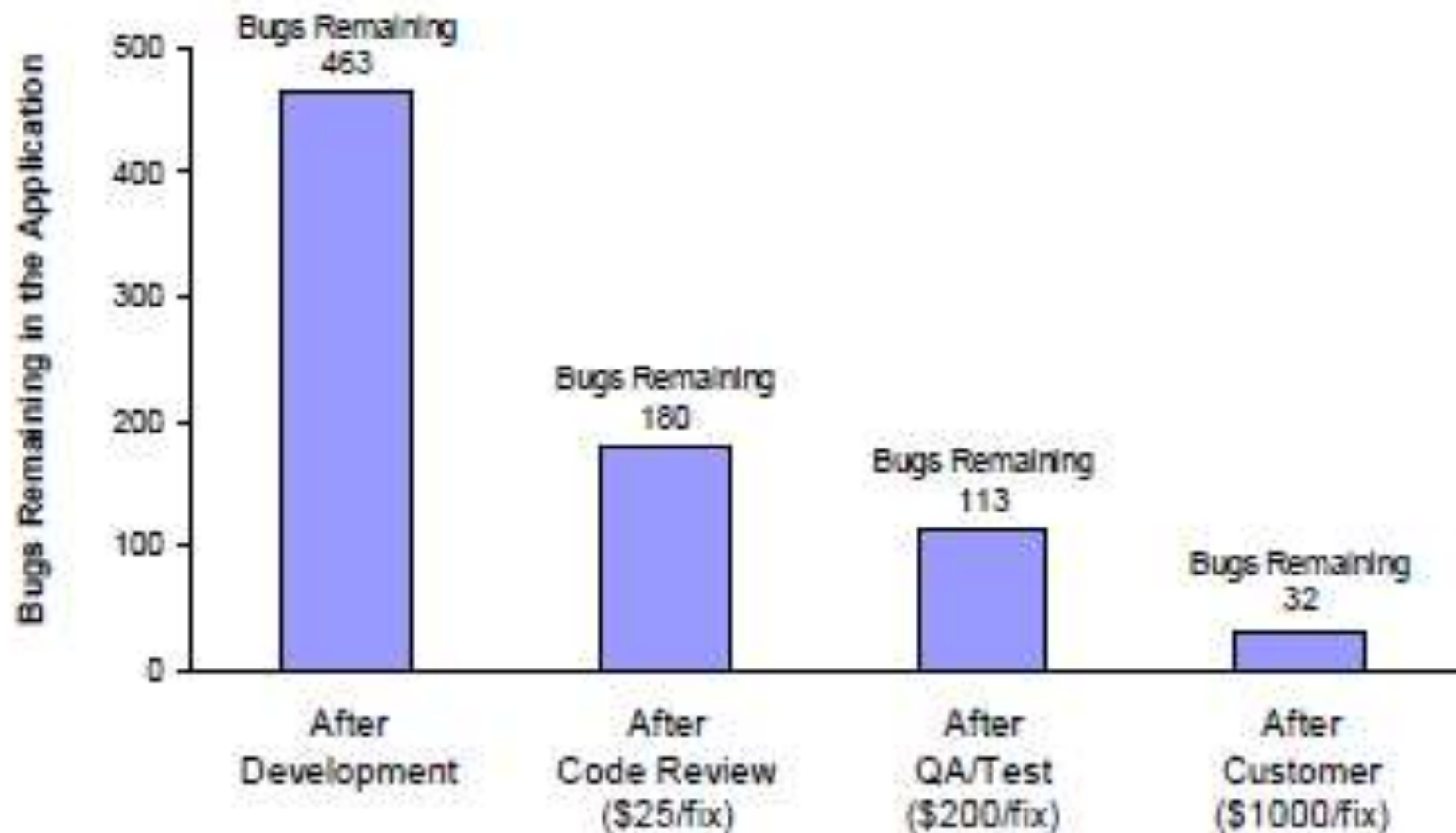
5 Reasons for Software Developers to Do Code Reviews (Even If You Think They're a Waste of Time)

- “testing cost between times finding and fixing a software problem after delivery is often 100 times more expensive than finding and fixing it during the requirements and design phase”.

Saving \$150k: A real-world case study



After Code Review



Cost of fixing bugs: \$120k
+ Cost of 32 latent bugs: \$ 32k

Total Cost: **\$152k**

חסרונות?

- זמן מפתחים – (אבל מצד שני יותר יקר לגלות בהמשך)

שיטות לסקר

- מאחורי הכתף
- מייל, IM
- Pair-programming
- בשילוב עם TDD ו-Refactoring
- כלים ייעודיים
 - [Kiln](#)
 - [Rietveld](#) ([דוגמא](#))
 - [Mondrian: Code Review on the Web](#)
 - <http://www.review-board.org/>
 - JCE Course: <http://pair2program.appspot.com/>

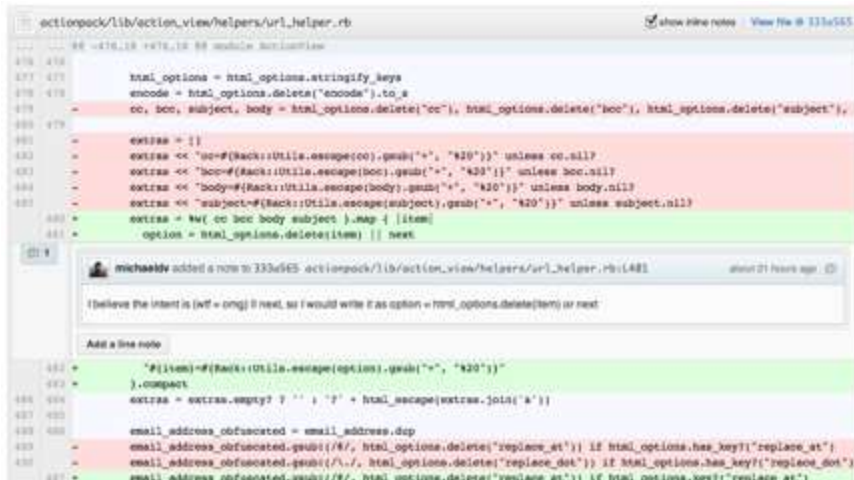
GitHub Code Review

GitHub Pull Request = Code + Issue + •
Code Comments

Pull Requests •

Comments •

Process! •



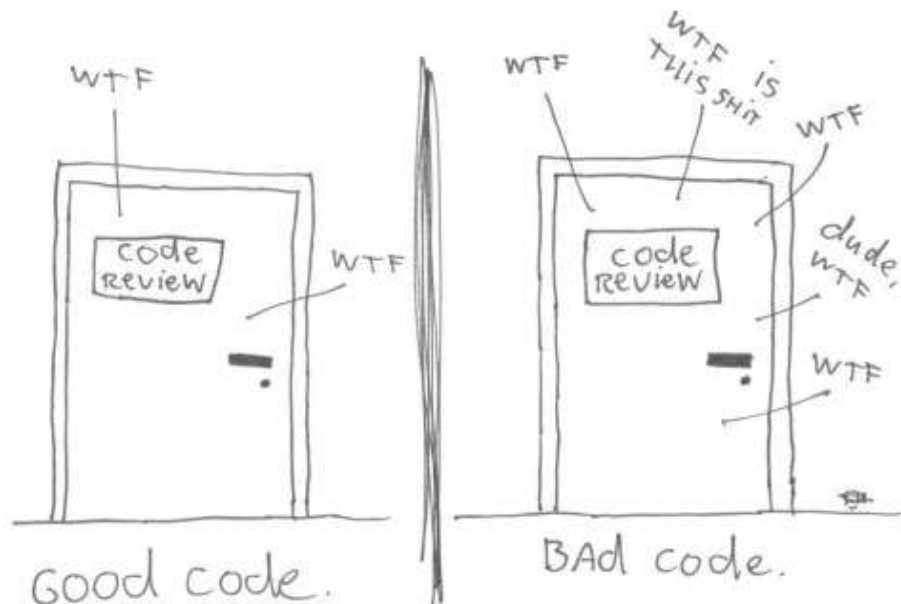
```
476 476
477 477      html_options = html_options.stringify_keys
478 478      encode = html_options.delete("encode").to_s
479 479      cc, bcc, subject, body = html_options.delete("cc"), html_options.delete("bcc"), html_options.delete("subject"),
480 480
481 481      extras = []
482 482      extras << "cc=#{Rack::Utils.escape(cc).gsub(" ", "%20")}" unless cc.nil?
483 483      extras << "bcc=#{Rack::Utils.escape(bcc).gsub(" ", "%20")}" unless bcc.nil?
484 484      extras << "body=#{Rack::Utils.escape(body).gsub(" ", "%20")}" unless body.nil?
485 485      extras << "subject=#{Rack::Utils.escape(subject).gsub(" ", "%20")}" unless subject.nil?
486 486
487 487      extras = %w[cc bcc body subject].map { |item|
488 488          option = html_options.delete(item) || next
489 489
490 490      }
491 491
492 492      # If item is not nil, escape it and add to extras
493 493      extras = extras.empty? ? "" : " " + html_escape(extras.join("&"))
494 494
495 495      email_addresses_obfuscated = email_addresses.dup
496 496      email_addresses_obfuscated.gsub!(/@/, html_options.delete("replace_at")) if html_options.has_key?("replace_at")
497 497      email_addresses_obfuscated.gsub!(/./, html_options.delete("replace_dot")) if html_options.has_key?("replace_dot")
498 498
499 499      email_addresses_obfuscated.gsub!(/%/, html_options.delete("replace_percent")) if html_options.has_key?("replace_percent")
```

הנשק הסודי

- של Extreme Programming
- של חברות שמצליחות להחזיק מוצרים לאורך זמן
- בפרויקט שלכם: תיעוד שליחת diff והערות שהתקבלו



The ONLY VALID MEASUREMENT
OF CODE QUALITY: WTFs/minute



(c) 2008 Focus Shift/OSNews/Thom Holwerda - <http://www.osnews.com/comics>

2. תיעוד ומסמכים

```
int five = 7; //11
```

- למה\האם לכתוב תיעוד? מי? מתי? סוגים?
- הערות: שנוי במחלוקת, למשל
<http://apdevblog.com/comments-in-code/>
- כלים ליצירת תיעוד מהערות
[Javadoc](#)/Ndoc, [Doxygen](#), Sandcastle, [docco](#),
[ועוד](#)
- – למשל [String in Java](#)
- מתי כלים אלו נצרכים?
- – איזה סוג תיעוד נותנות בדיקות יחידה? ([דוגמא](#)
CSVReader)
- דיון: [Agile Documentation](#), Amber,

כלים תומכי בדיקות

- Unit Testing Frameworks

 - ✓ xUnit

 - Parameterized tests, White-box, fuzzing (security) ,GUI

- Acceptance Tests Frameworks

- Code Coverage

- Continuous Integration

- A/B Testing ([paper](#))

```
בכמה מקרים תתרחש שגיאה?:  
int foo(int x) { // x is an input  
    int y = x + 3;  
    if (y == 13) abort(); // error  
    return 0;  
}
```

3. בדיקות קבלה

- שמות שונים: בדיקות משתמש, בדיקות קצה, פונקציונליות, מפרטים מורצים, BDD, ATDD
- סביבות מתפתחות לאחרונה, למשל
Fit, [Cucumber](#), SpecFlow, [BDDify](#), [TextTest](#) –
Web Automation: Selenium, Watir –
– <http://swarm.jquery.org/> סטטוס מעודכן לדוג'

4. כיסוי קוד

- כמה בדיקות צריך לכתוב?
- ב-TDD לא כותבים אף שורה בלי בדיקה
 ⇐ 100% כיסוי
 ⇐ אם יש 90% כיסוי, היכן מסתבר שרוב הבאגים?
- כלים לדוגמא:
 - .Net : Ncover (משולב ב- TestDriven.Net)
 - Eclipse : [EclEmma](#)
 - ~~נבדוק עוד דוגמא [CsvReader](#)~~



CsvReader_src.zip

5. שילוב מתמשך – CI/CD

- מתי לבנות הכל ולהריץ את הבדיקות?
- באופן אידיאלי: לפני ואחרי כל שינוי
- לפחות כל פעם שמכנסים משהו חדש לבקרת הגרסאות?
- למה זה חשוב
- מי מריץ?
- שרתים: למשל Cruise Control, TeamCity, Hudson/Jenkins
- כלי הידור לדוגמא: Make, MsBuild, Ant, Kayak
- <http://martinfowler.com/articles/continuousIntegration.html>

Continuous integration - Recommended practices

http://en.wikipedia.org/wiki/Continuous_integration#Recommended_practices

- Maintain a code repository
- Automate the build
- Make the build self-testing
- Everyone commits every day
- Every commit (to mainline) should be built
- Keep the build fast
- Test in a clone of the production environment
- Make it easy to get the latest deliverables
- Everyone can see the results of the latest build
- Automate deployment

יתרונות וחסרונות

- איתור באגים קל ומהיר
- לא נופלים על בעיות שילובים בסוף
- עותק זמין ועובד של המוצר כל הזמן
- משוב תמידי על איכות הקוד
- מעודד כתיבת קוד מודולרי ופחות תלותי
- מצריך הקמת תשתיות בהתחלה
- כדי להיות אפקטיבי מצריך פיתוח סט בדיקות

בפרויקט - בונים

• Jenkins

– הקמת שרת (צוות הקורסלסיסטם)

– הוספת פרויקט והגדרות (צוות)

- בקרת גרסאות

- הידור

- בדיקות

- הודעות

– (Tray Application)

הפצה מתמשכת - CD

- פרויקט תוכנה כפס ייצור
- לאילו פרויקטים מתאים? האם עדיין יש צורך באיטרציות?
- תשתית CI כשלב מקדים
- חמשת השלבים למימוש Continuous Deployment
- CI/CD/cloud Demo - "Bootstrapping an agile project with continuous deployment using cloudbees"

לסיכום - כלים

- צידה לדרך
 - סקרי קוד, תיעוד
 - בדיקות מעבר ליחידה, כיסוי
 - שילוב מתמשך
- כלים רבים נוספים, למשל
 - אכיפת סגנון FxCop
- מתי וכמה תשמשו בכלים !



בפעם הבאה

- סיכום הקורס
- מצגות למעוניינים (רישום חובה למגישי הבונוס)
- לקריאה: התקדמות אישית