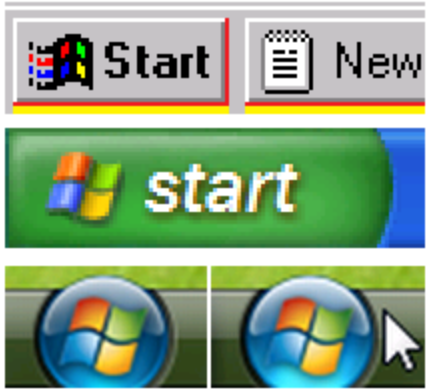
הלל: מה ששנוא עליך על תעשה לחברך

המכללה האקדמית להנדסה ירושלים

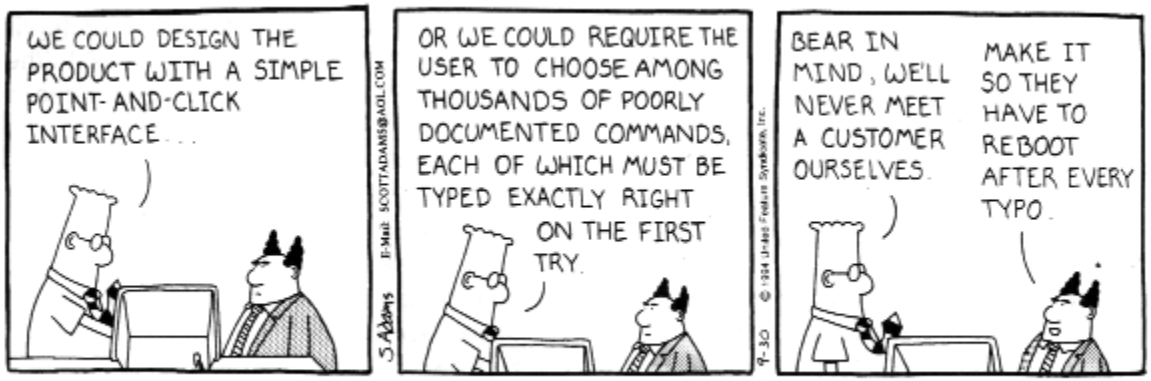


הנדסת תוכנה

12. חווית משתמש – UX

כלים II

"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

Google Groups

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!

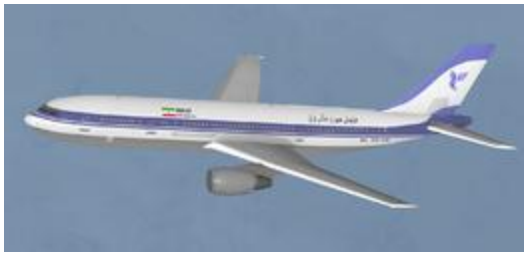
Reply Reply to author Forward Rate this post:

Sandy Johnson View profile

Dear Crowe,

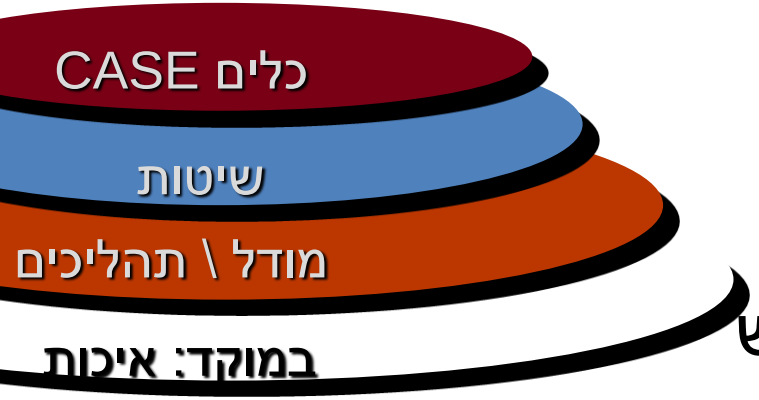
Sorry can't get it back -- unless you saved it earlier under a different name or perhaps emailed the file to someone previous to saving it...

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
  בונוס סביבת CI / CD

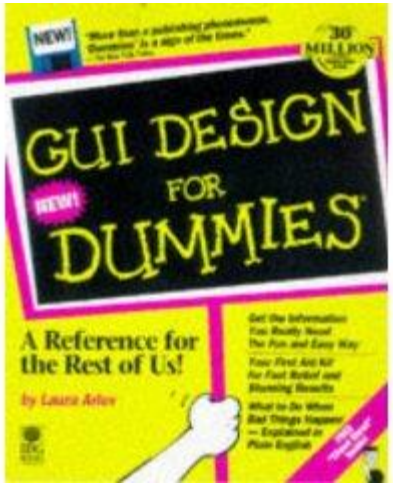
Brooks (MMM): Question: How does a large software project get to be one year late?
Answer: One day at a time!

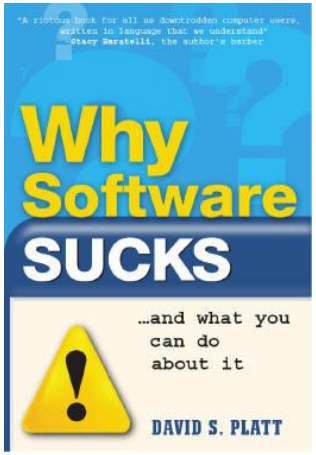
Usability in the Front

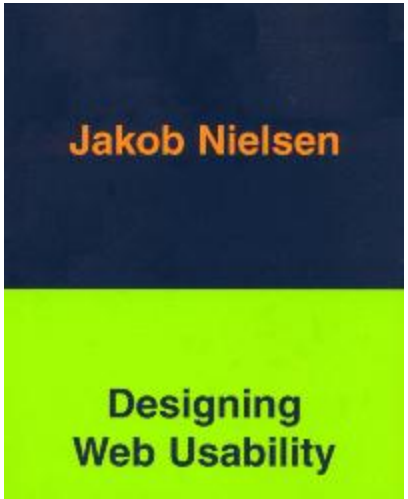
- Windows 8 UX Manager
- The Science and Art of User Experience at Google (gtalk example)

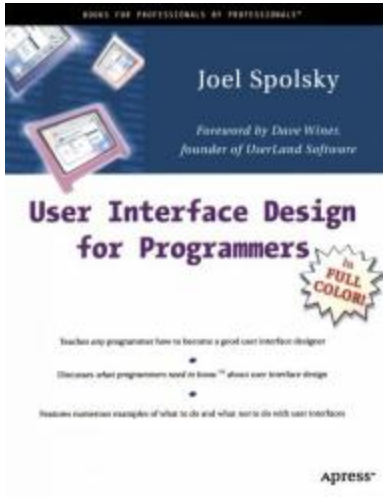
מקורות

- 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
- Usabiliy In Practice, MSDN series









קישורים

- חוויית משתמש מבוא http://uxi.org.il/pages/10316
- נוקיה http://www.developer.nokia.com/Design/User_experience/
- 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.txl
  - 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 - ביצועים
- גודל - 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

שלשת 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 אינטראקציית משתמש
- ארגון המידע
- מחקר משתמשים
- עיצוב ויזואלי

חווית משתמש (נילסן\נורמן)

"חוויית משתמש כוללת את כל האספקטים של האינטראקציה בין משתמש הקצה לבין החברה, השירותים שלה והמוצרים שלה. הדרישה הבסיסית לחוויית 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**. חוויית משתמש אמיתית נותנת ללקוחות הרבה יותר מאשר בדיוק את מה שהם אומרים שהם רוצים, או ממלאת אחרי רשימת יכולות ודרישות. כדי להגיע לחוויית משתמש באיכות גבוהה על-פני כל החברה, חייב להיות **רצף אחיד על-פני כל השירותים** שהיא מספקת בתחומים שונים, כולל הנדסה, שיווק, עיצוב גראפי ותעשייתי, ועיצוב ממשק".

שמישות - Usability

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

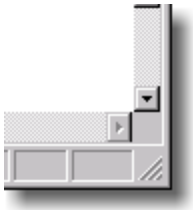

זה
מטורף
לא
השליכו בקבוקי פלסטיק
למיכל המיחזור

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

מושג קרוב: נגישות I - Affordance

- מצב שבו המאפיינים החושיים של אוביקט 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...

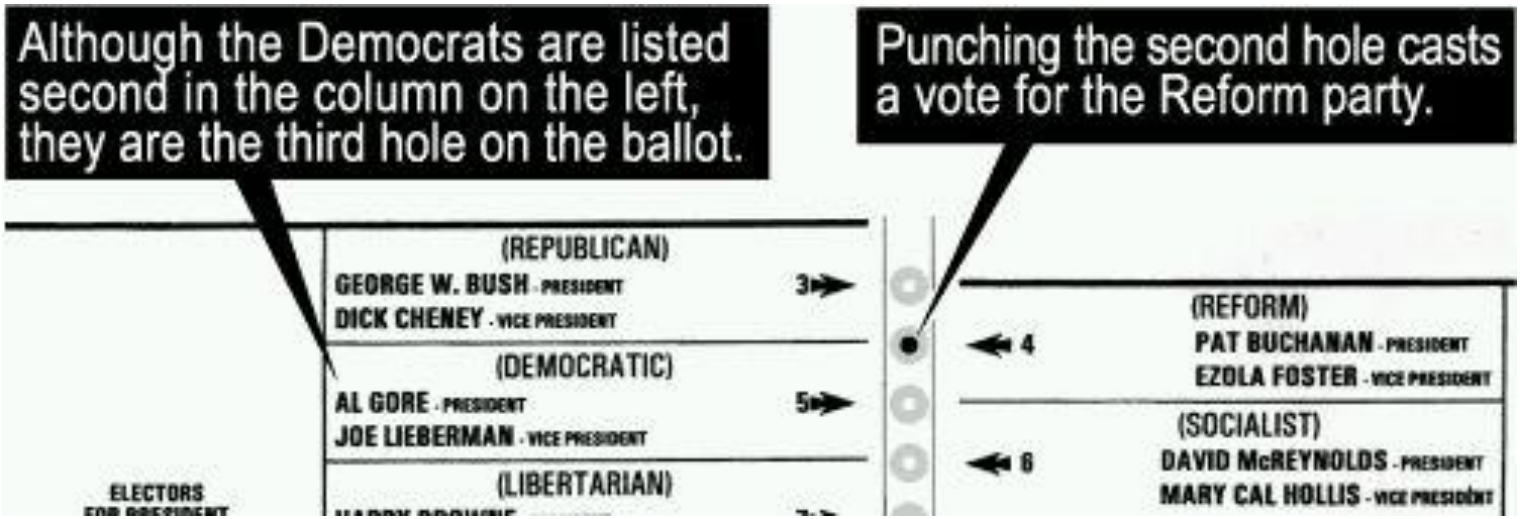
Accessibility - II נגישות

- נגישות למרות מוגבלויות שונות
  - ראיה
  - 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?

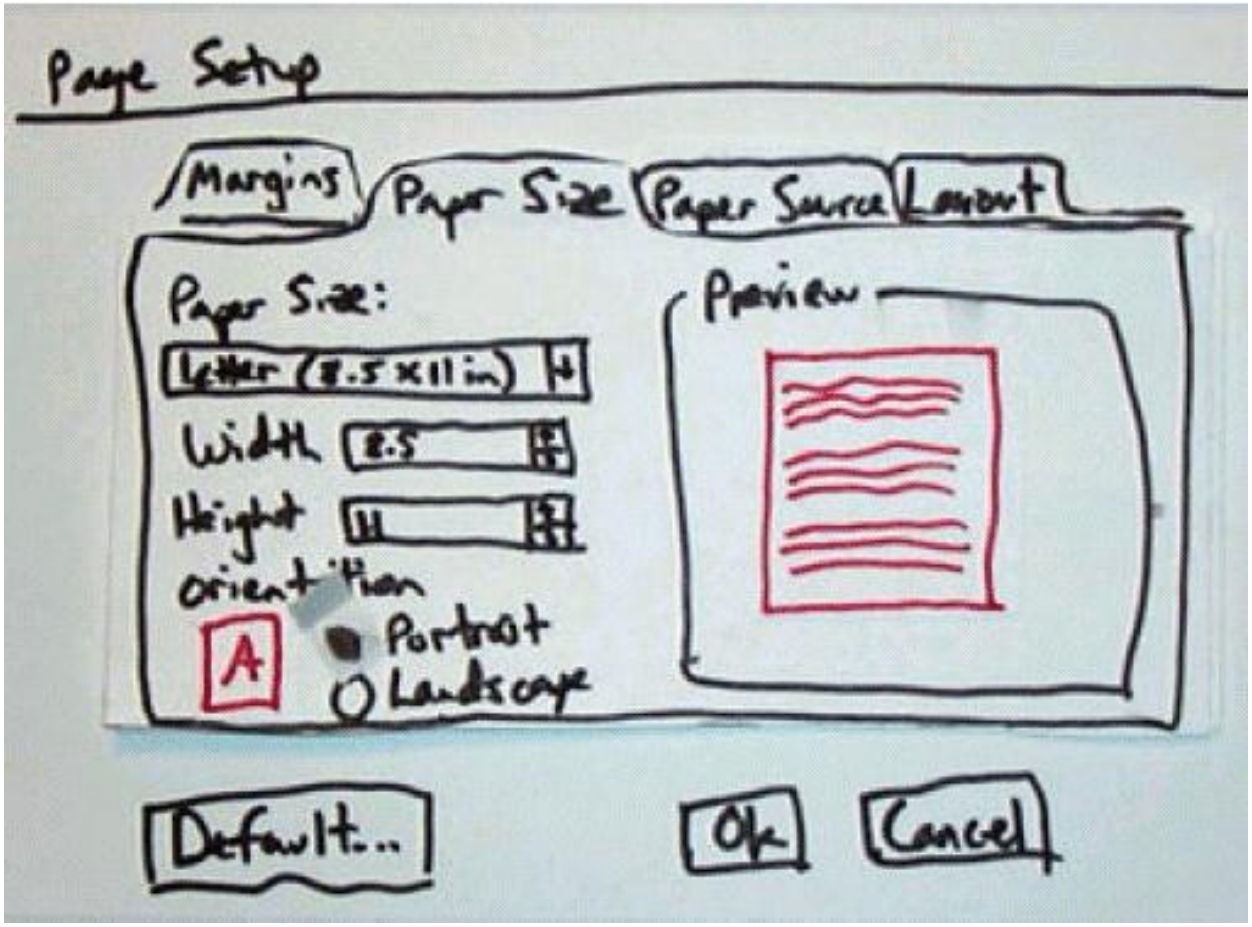
- מבדקי שמישות (Usability Testing)
  - Eye Tracking
- ראיונות\סקרים עם מומחים\ קבוצות מיקוד
- אב טיפוס
  - נייר
  - קוד (Rails ,VB)
- פיתוח איטרטיבי
  - במיוחד ברשת
- תהליכי עיצוב

DILBERT reprinted by permission of United Feature Syndicate.

אב טיפוס מנייר

- משחק תפקידים, ללא מחשב

אב טיפוס מנייר




pilot™
WED • NOV 5


counter
speaker
level indicators
Record
Record Stop
Play
play stop
Focus: Primary t
Think top down
Consistent UI
Gestures?
Orientation?
Check target siz
Reduce settings
Audio Levels
Speaker Grille
Record
View
Carrier
4:30 PM
PLAYBACK
RECORD
STOP

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

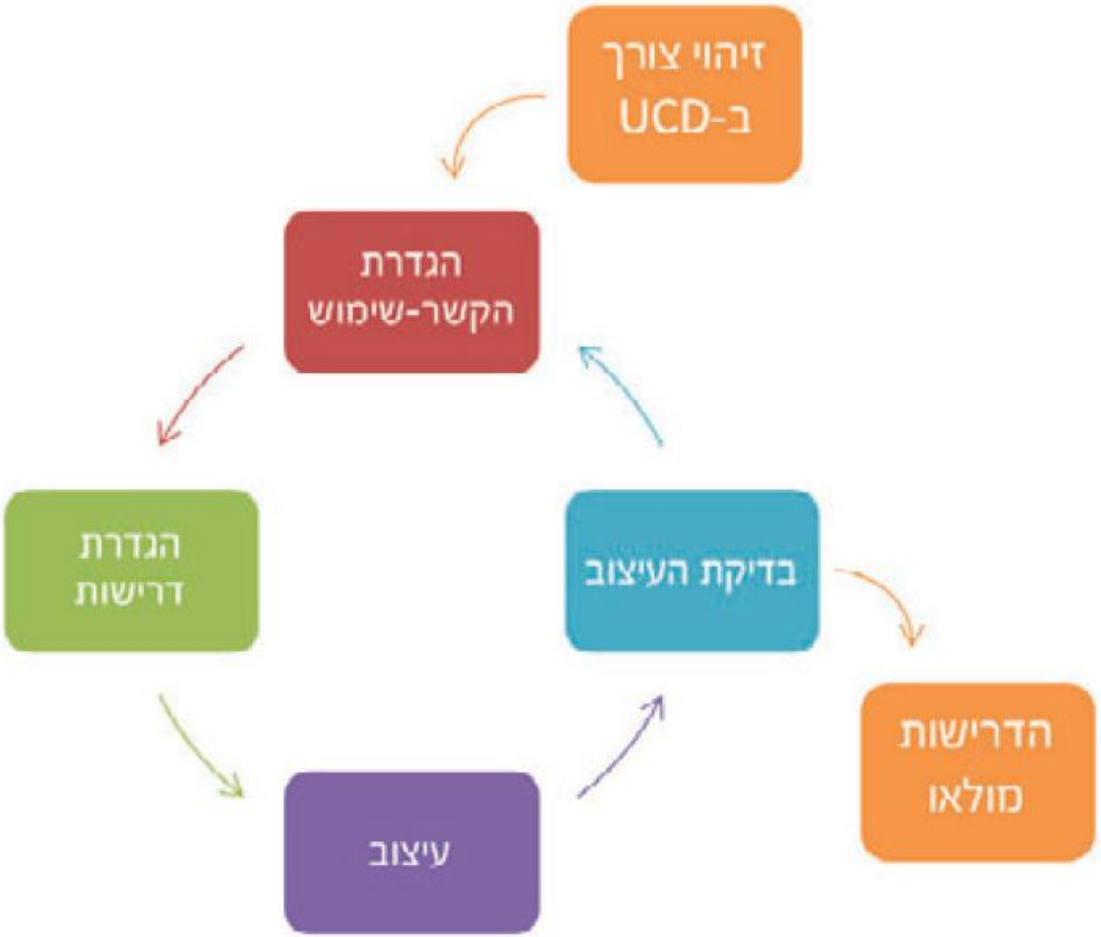
בתוכנה

- סביבות פיתוח מהירות, למשל:
  - Ruby on Rails
  - ASP.NET MVC, Nancy
- סביבות אבות טיפוס:
  - לאתרים: https://gomockingbird.com/
  - SketchFlow MS Expression Blend, Video
  - BalsamiQ Mockups, וידאו
  - http://www.pretotyping.org/androgen
    איך עושים זאת? למשל: http://astrails.com/
  - 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

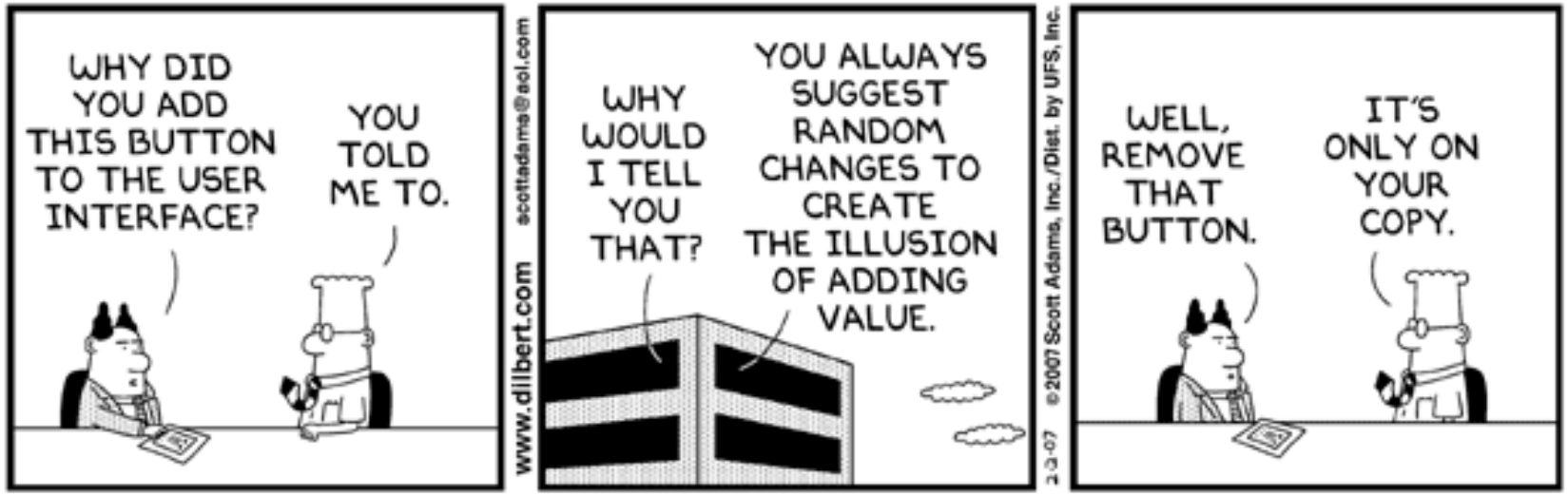
תהליך עיצוב מוכוון משתמש (UCD)



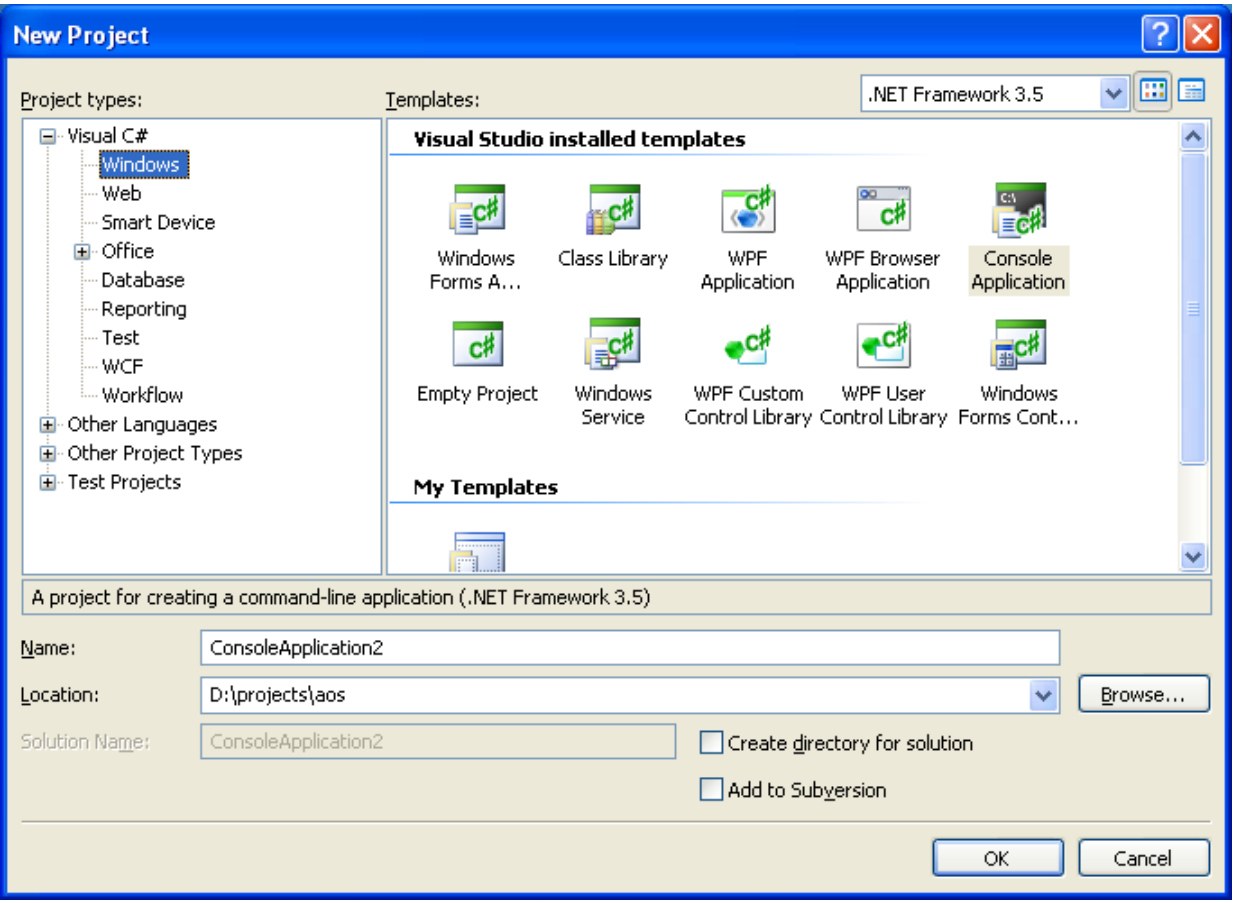
מתי בודקים?

- “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



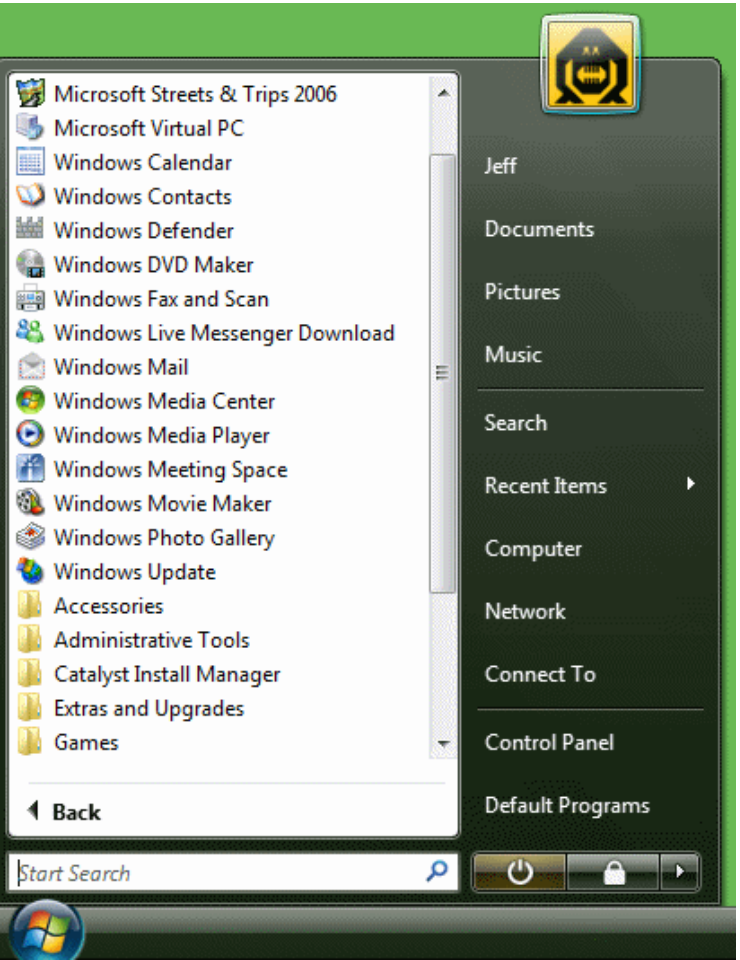
מתי כדאי להשתמש ב:



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

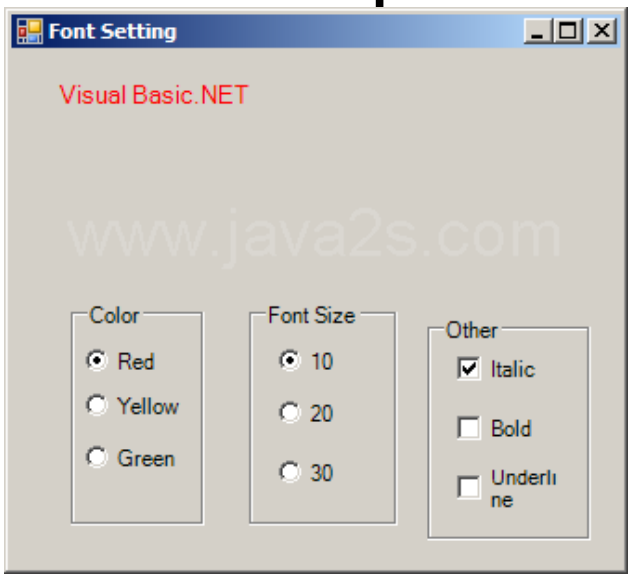
כפתורים, תפריטים

- כפתור – עבור פעולות עצמאיות, רלוונטיות למסך הנוכחי
  - מתי שִׂמים ...(שלש נקודות)?
- סרגל כלים – פעולות נפוצות
- תפריט – פעולות פחות נפוצות ישימות למספר מסכים (או כולם)



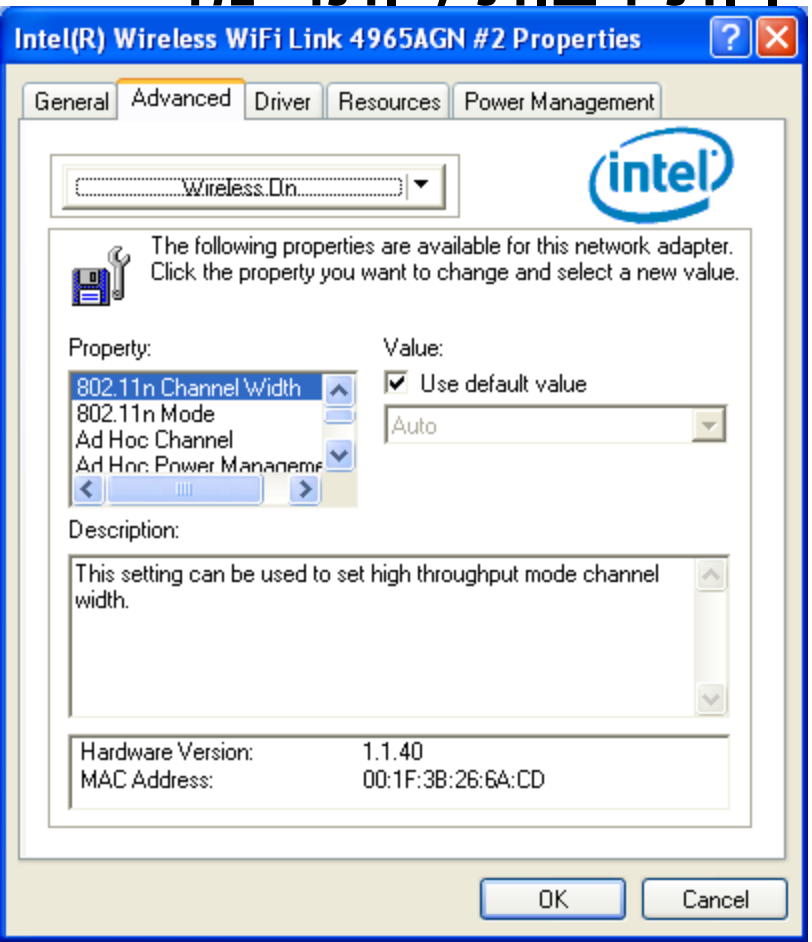
תיבות סימון

- 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

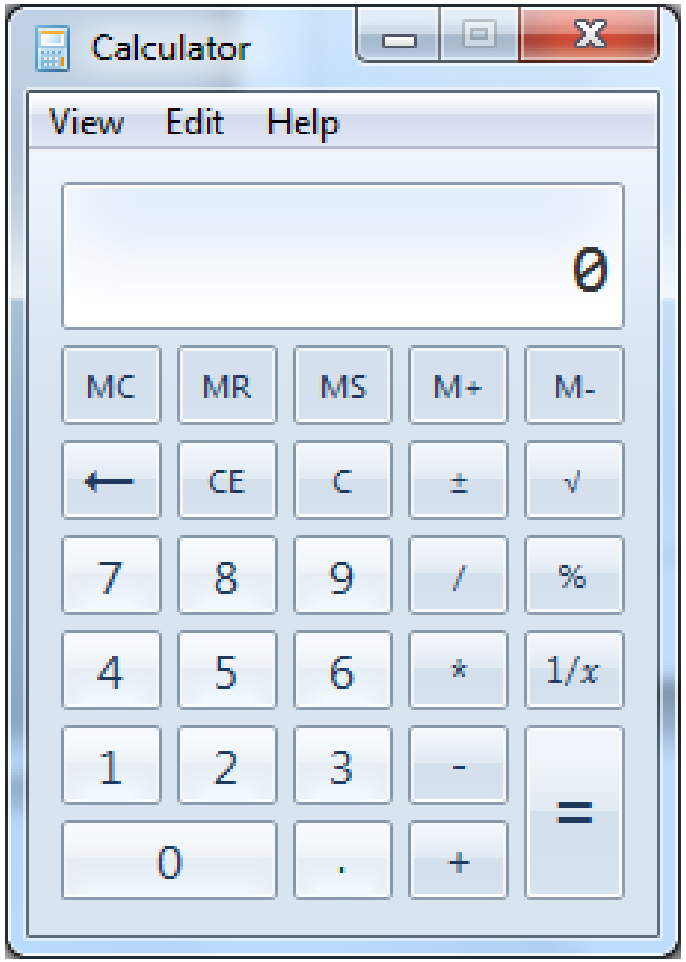
- טקסט (בד"כ בצרוף תגית) – קלט חופשי
- רשימה – בחירה מתוך אפשרויות רבות (יותר מדי בשביל רדיו) ורוצים שיראו את כולן
- Combo-box – כנ"ל, אך רוצים לחסוך במקום על המסך

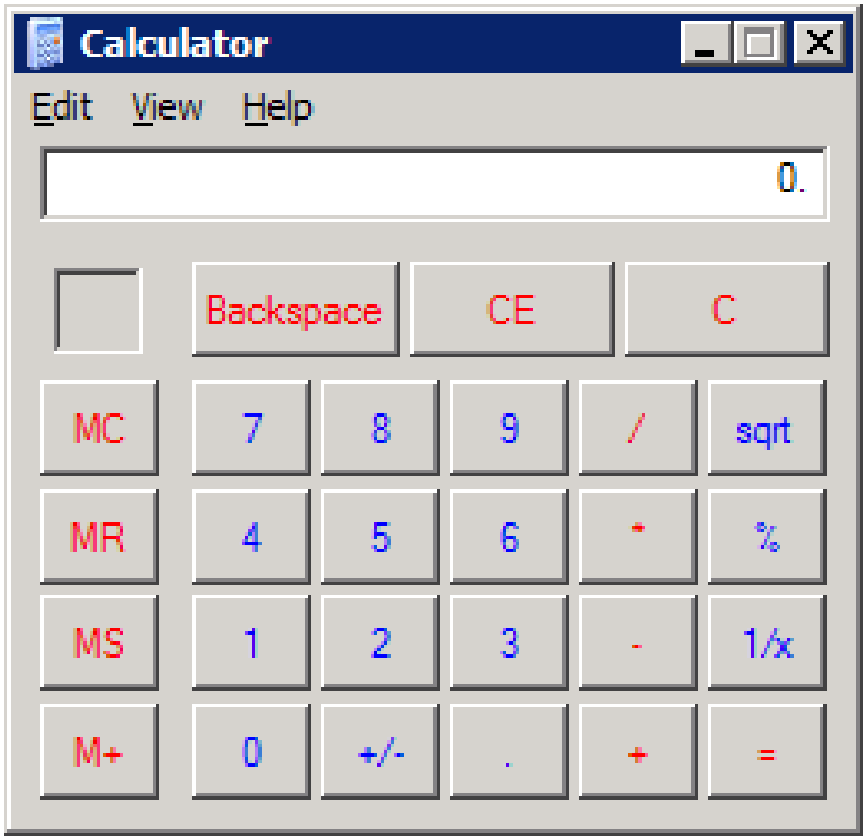


מסכים מרובים

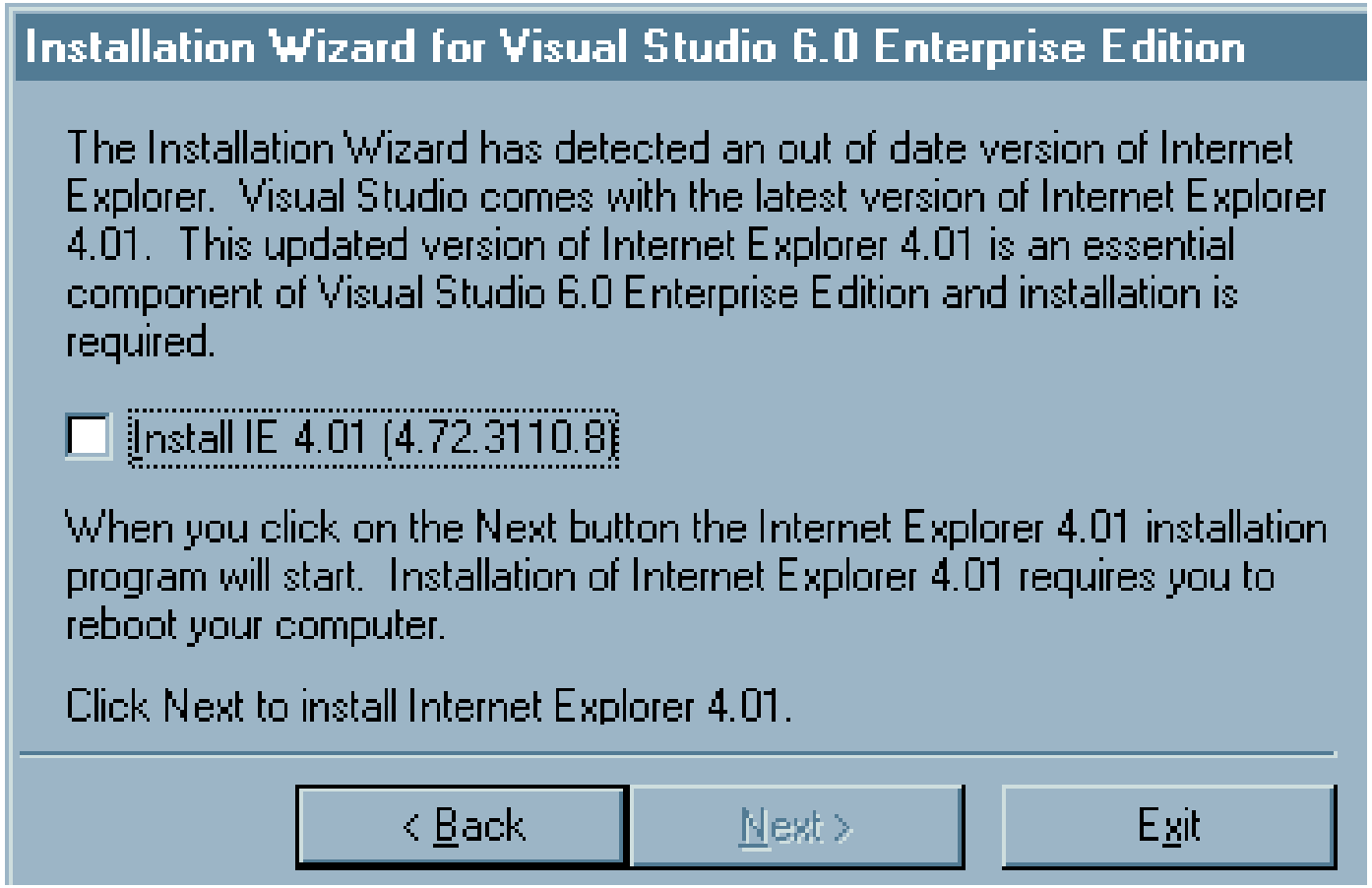
- Tabbed Pane – המשתמש יכול לעבור בכל רגע בין המסכים השונים
- אשף (wizard) – הנחיית המשתמש לאורך תהליך
- תיבת שיחה – הצגת מסכים זמניים או תיבת מאפיינים

על ספסל הנאשמים



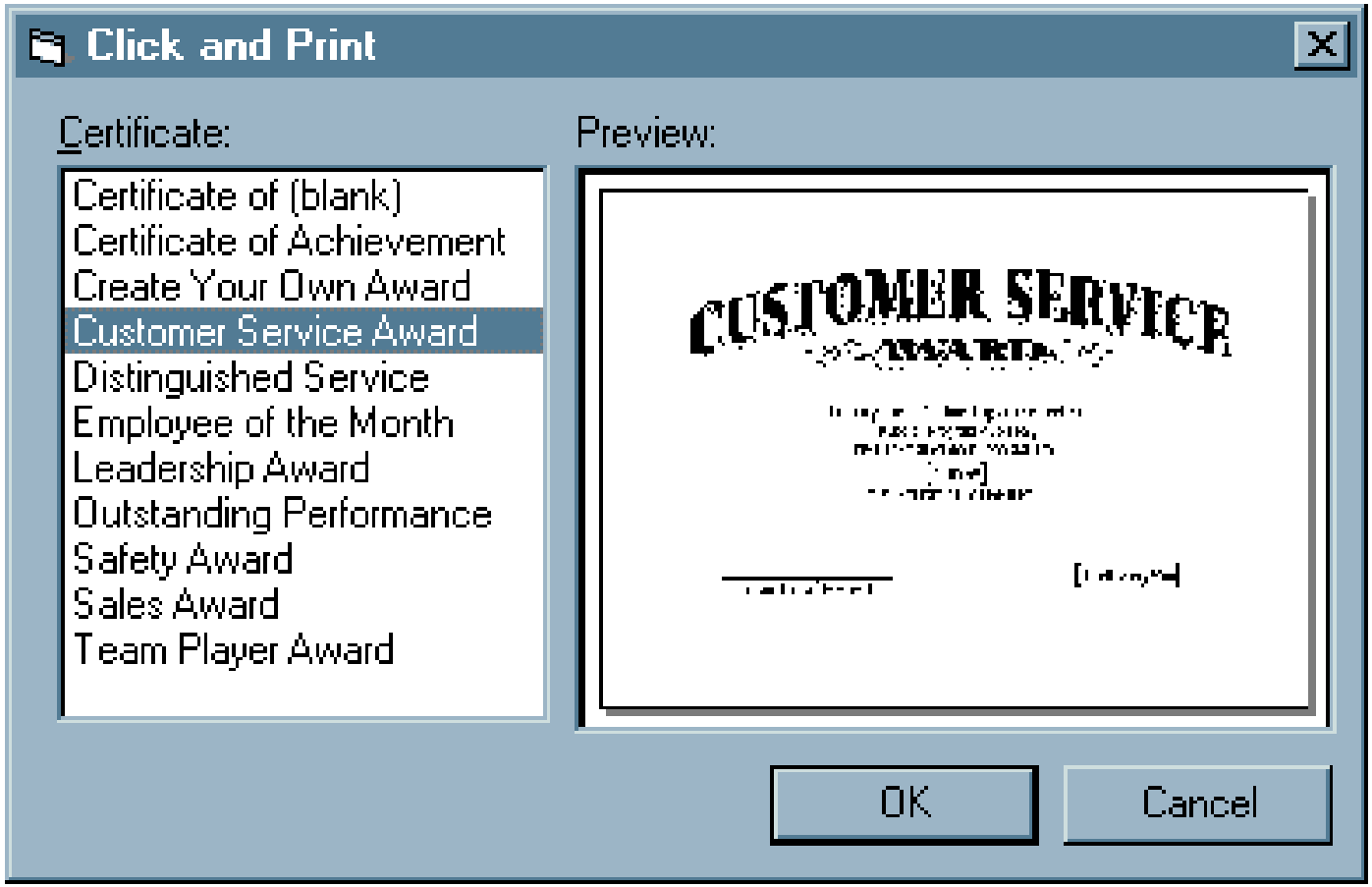


Interface Hall of Shame - דוגמאות

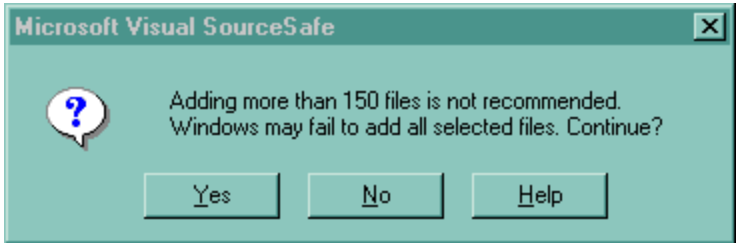


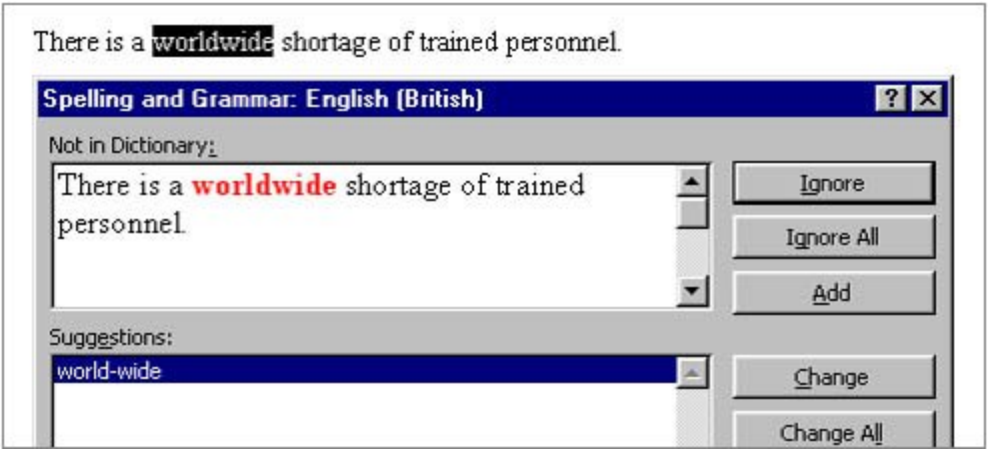
Interface Hall of Shame -דיון

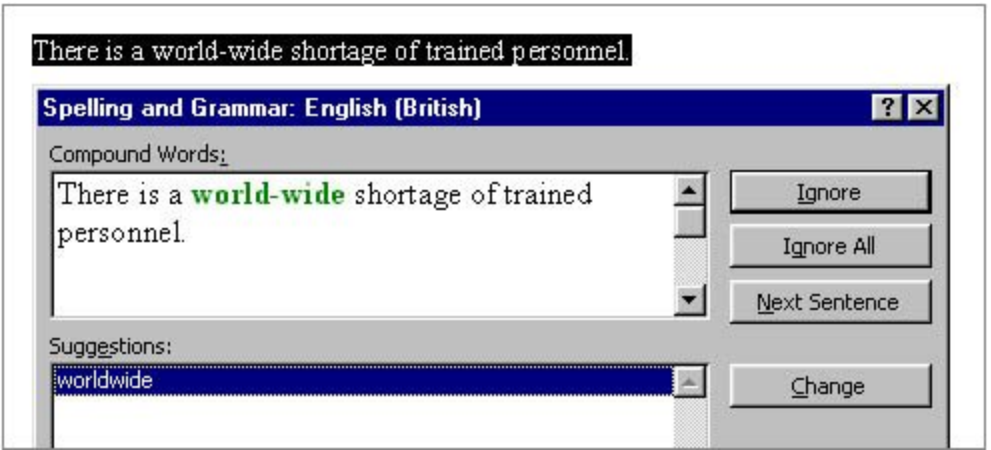
קצת יותר טוב

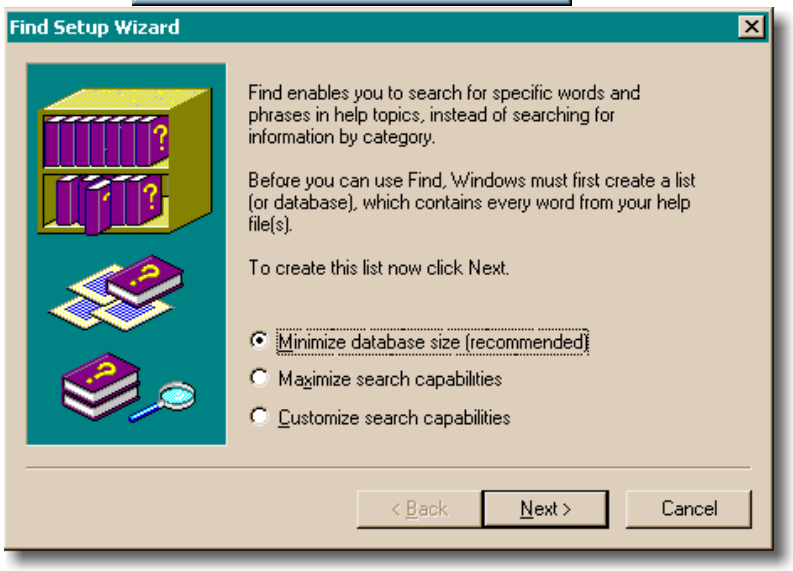

Click and Print
Certificate:
Preview:
Certificate of (blank)
Certificate of Achievement
Create Your Own Award
Customer Service Award
Distinguished Service
Employee of the Month
Leadership Award
Outstanding Performance
Safety Award
Sales Award
Team Player Award
CUSTOMER SERVICE
AWARD
OK
Cancel

עוד דוגמאות והודעות שגיאה

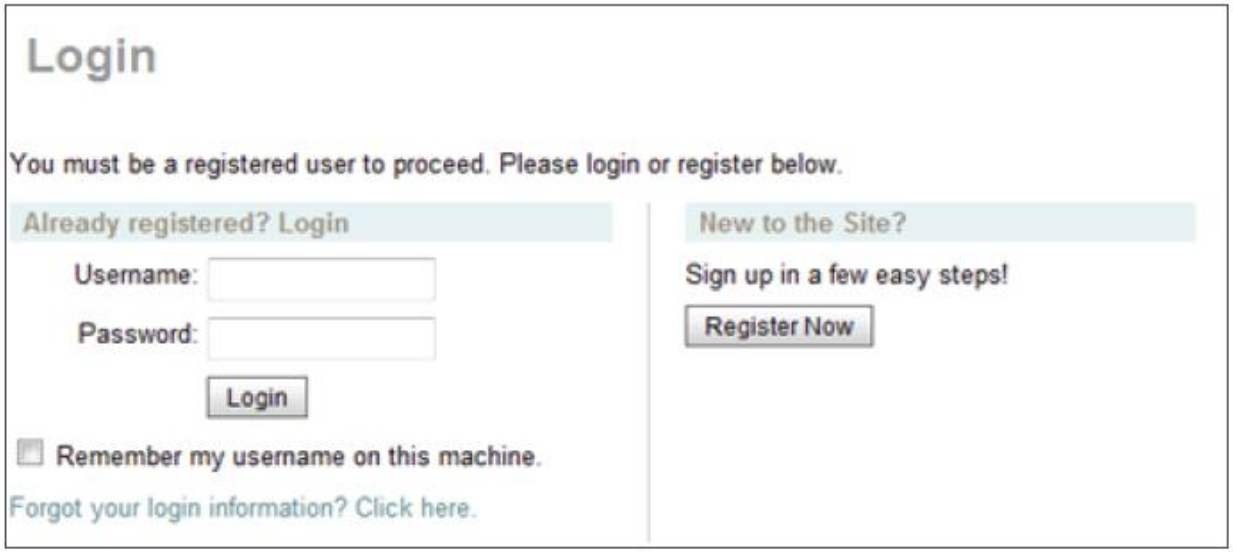




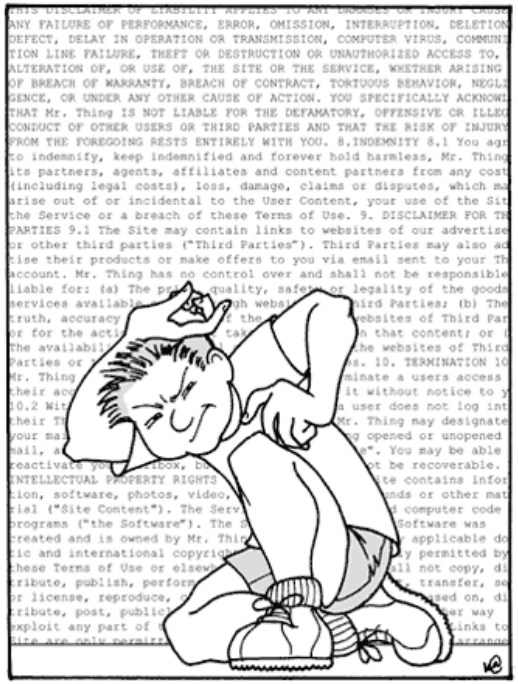




שמישות ברשת (+300M$)



טעויות נפוצות בעיצוב רשת

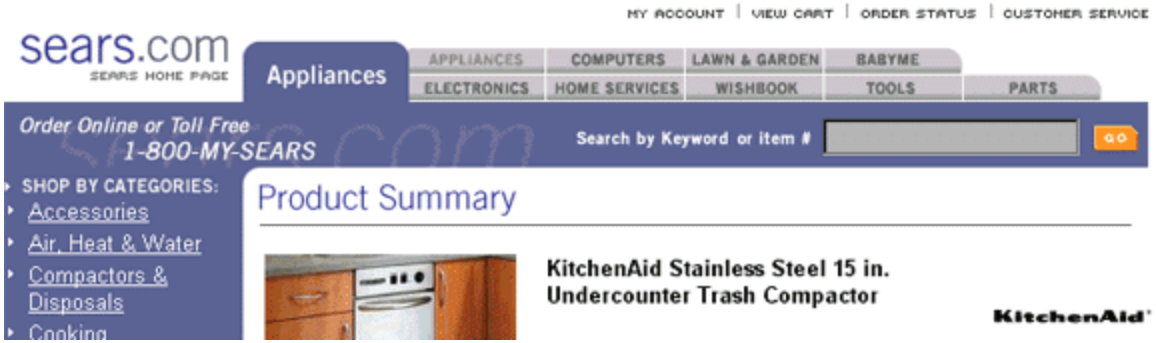


- עמודים עמוסים
- התאמה לרזולוציות שונות
- התאמה לתקנים/דפדפנים שונים
- התאמה לאיזורים שונים
- "קירות" של טקסט
- עיצוב סטנדרטי

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



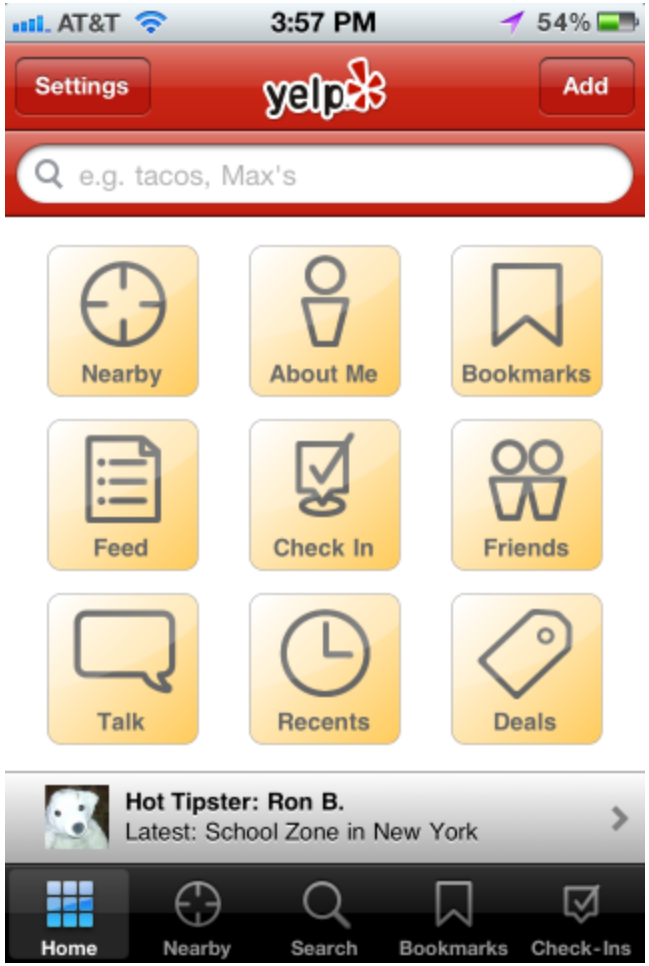
- אבטחה מול פרטיות

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

דוגמת שמישות ברשת

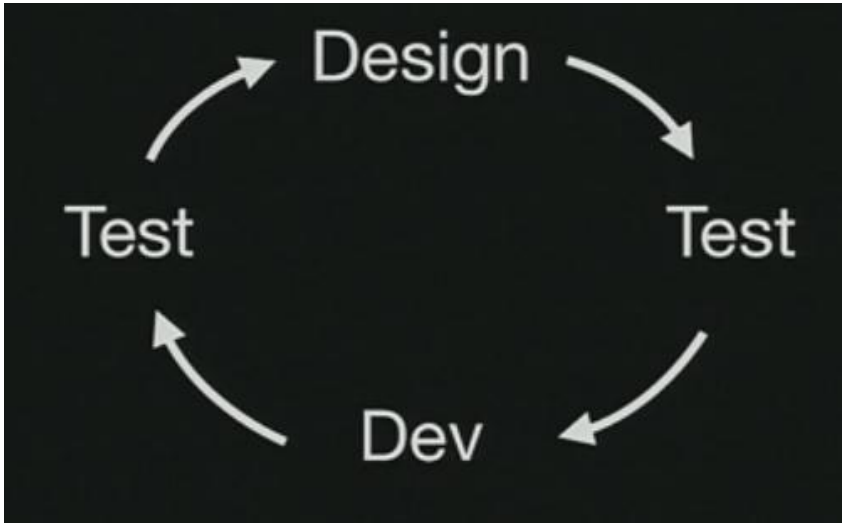
- http://graphs.gapminder.org/world/

שמישות במחשוב נייד



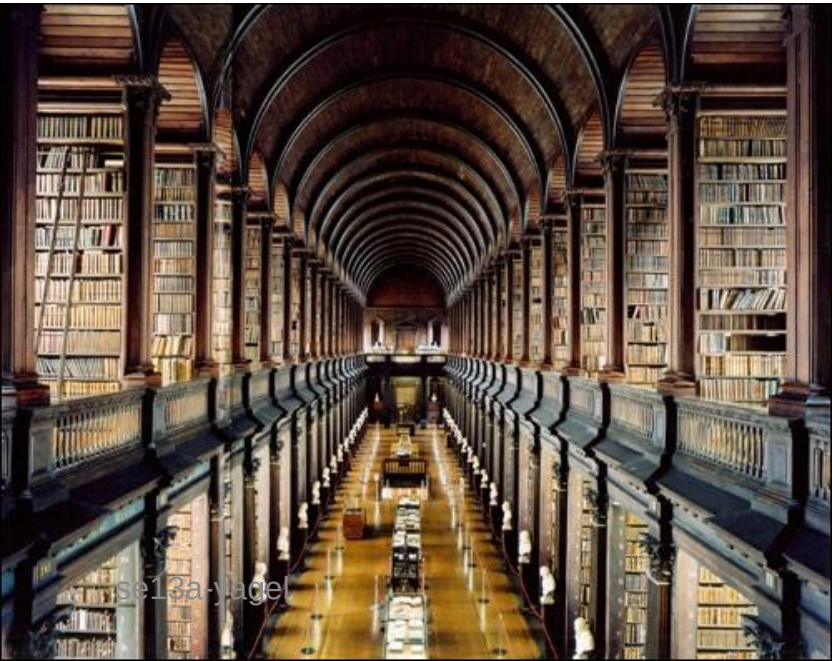
עיצוב חווית משתמש

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



תרגיל עיצוב

- שרטטו ממשק משתמש לחיפוש בספריה, 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 <u>משתמשים</u>
- 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
- קורס: בחירה, MIT: User Interface Design & Implementation

כלים II

- 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)?

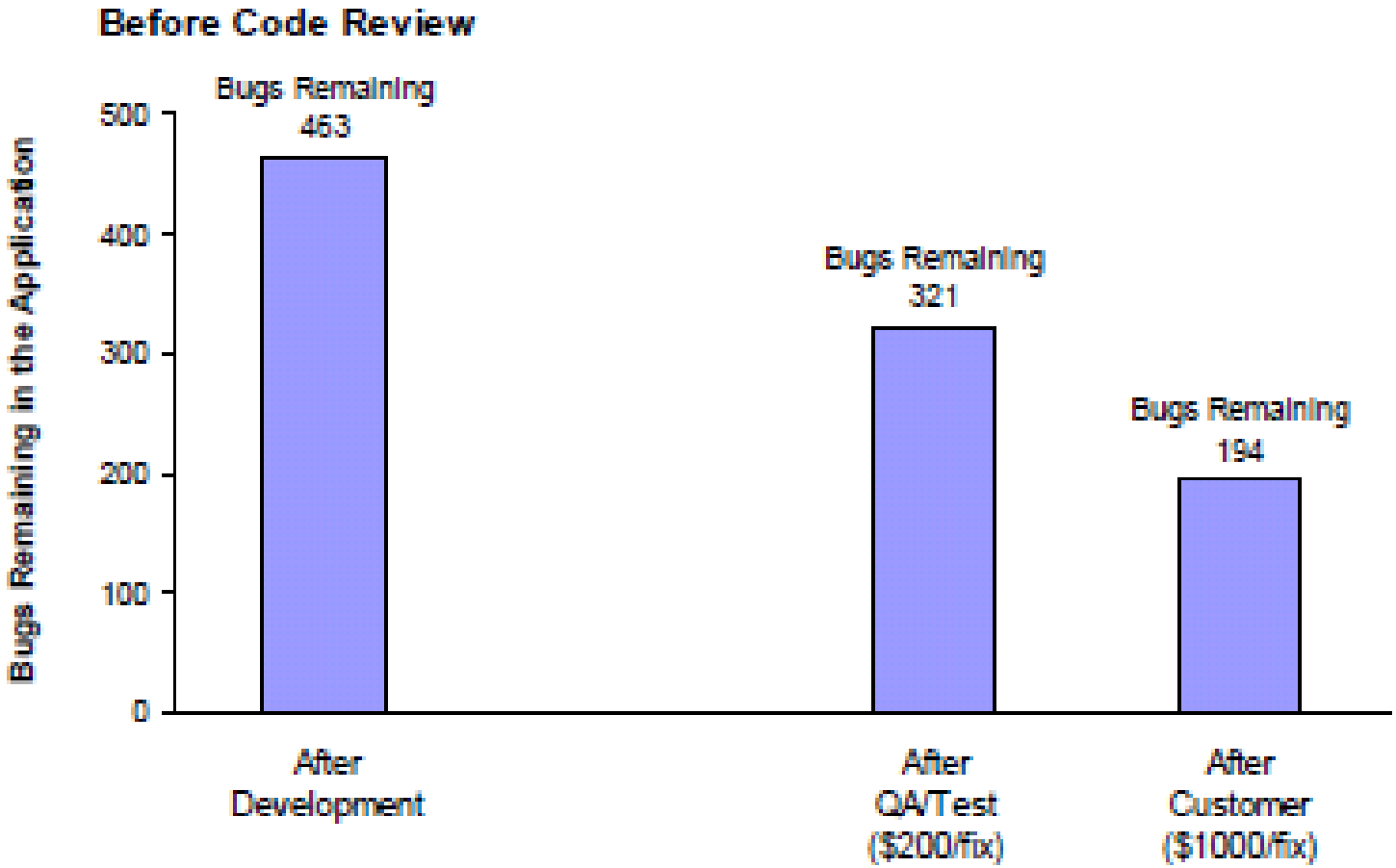
- בחינה עקבית של הקוד
  - במטרה למצוא באגים ולתקנם 2<1+1
  - שיפור הקוד ומיומנות המפתח

- 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

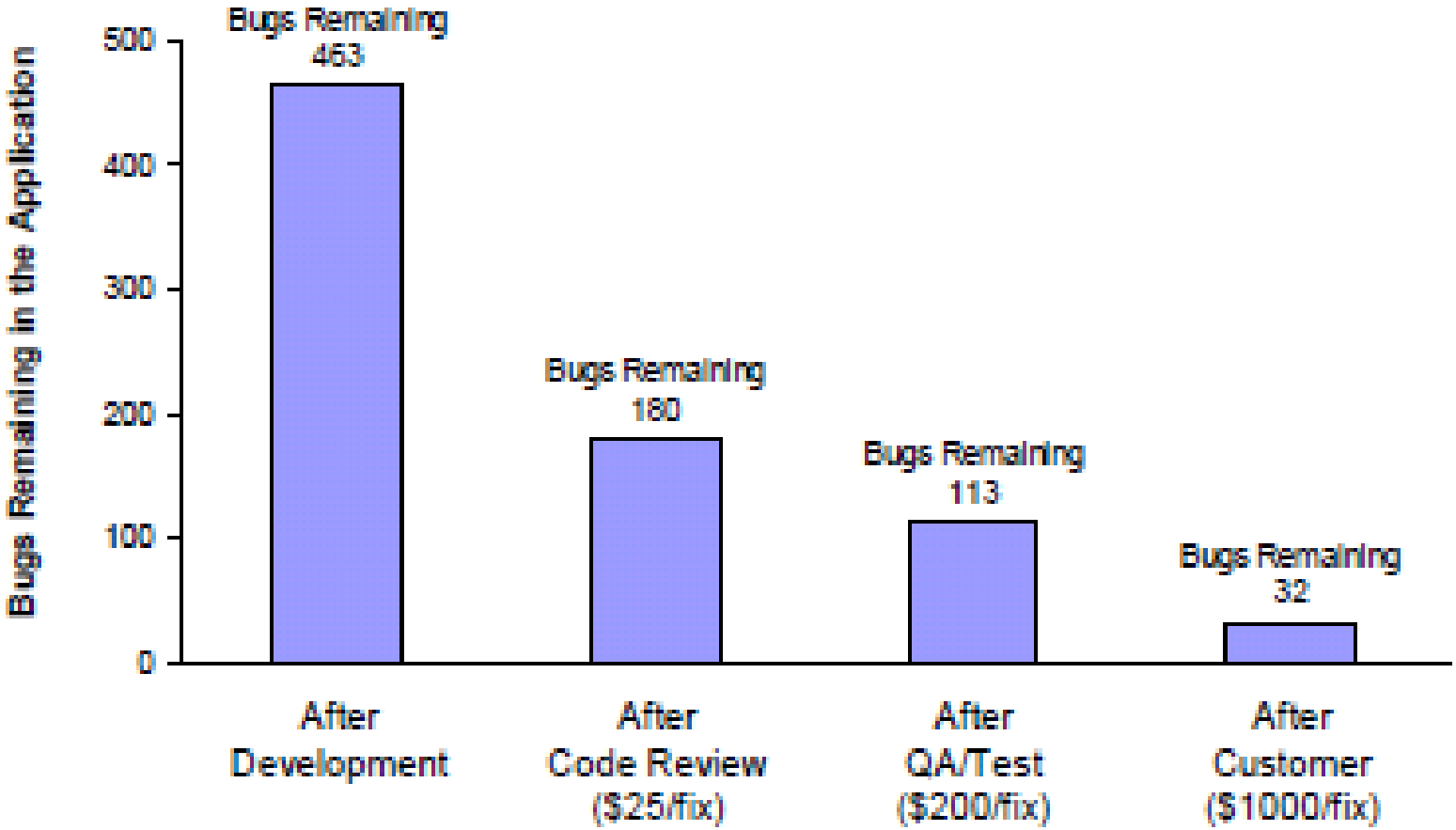


Cost of fixing bugs: $174k
\+ Cost of 194 latent bugs: $194k

Total Cost: **$368k**

http://smartbear.com/docs/book/code-review-works.pdf

After Code Review



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

Total Cost: **$152k**

חסרונות?

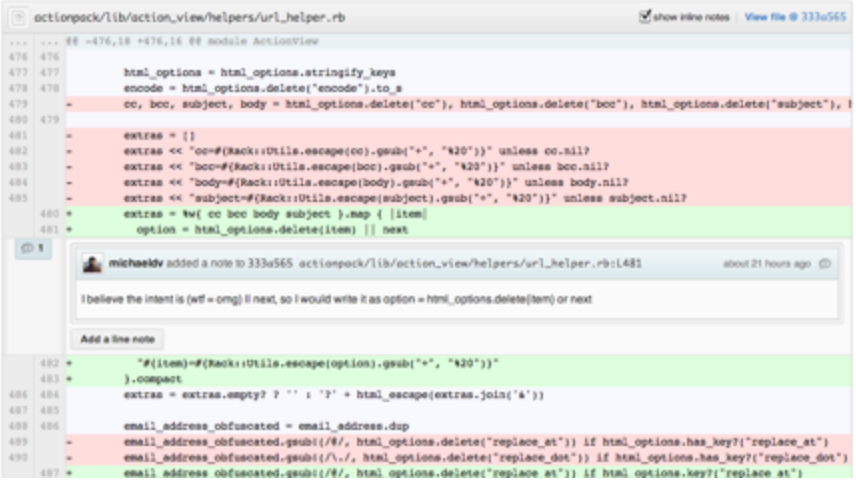
- זמן מפתחים
  - (אבל מצד שני יותר יקר לגלות בהמשך)

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/

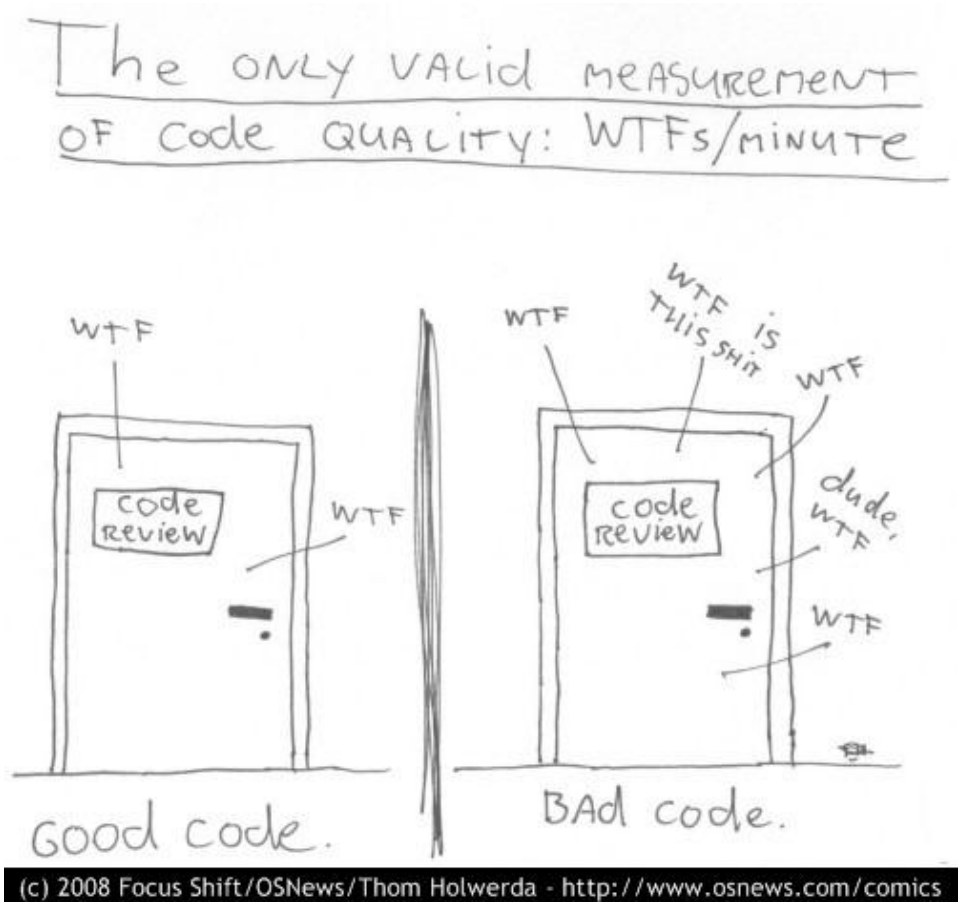
GitHub Code Review

- **GitHub Pull Request =** Code + Issue + Code Comments
- Pull Requests
- Comments
- Process!



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
WTF
code review
WTF
Good code.
WTF
WTF is this shit
WTF
code review
dude, WTF
WTF
BAd code.
(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: Amber, Agile Documentation

כלים תומכי בדיקות

- Unit Testing Frameworks
  - ✔ xUnit
  - – Parameterized tests, White-box, fuzzing (security) ,GUI
  - – (pexforfun)
- 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: Selinum, Watir
  - סטטוס מעודכן לדוג' http://swarm.jquery.org/

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
- ענן: Heroku, Cloudbee, MS Azure, jelastic
- 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 - בונוס

- Jenkins
  - הקמת שרת (צוות הקורס\סיסטם)
  - 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"

בפעם הבאה

- סיכום הקורס
- מצגות למעוניינים (רישום חובה למגישי הבונוס)
- לקריאה: התקדמות אישית
- Norvig, Teach Yourself Programming in Ten Years (עברית)
- שאלה: מה הם הקריטריונים של המחבר לבחירת שפת תכנות? מה דעתך?

לסיכום - כלים

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