

Memo to the CEO

Aral Balkan (<http://aralbalkan.com>)



Creative Commons Attribution 2.0 UK: England & Wales

Who am I?



Who am I?

- User experience designer



Who am I?

- User experience designer
- Software architect



Who am I?

- User experience designer
- Software architect
- Flash/Flex developer



Who am I?

- User experience designer
- Software architect
- Flash/Flex developer
- Coach



Who am I?

- User experience designer
- Software architect
- Flash/Flex developer
- Coach
- Trainer



Who am I?

- User experience designer
- Software architect
- Flash/Flex developer
- Coach
- Trainer
- Professional speaker



Who am I?

- User experience designer
- Software architect
- Flash/Flex developer
- Coach
- Trainer
- Professional speaker



INFORMATION
FREEDOM
RULES



Who am I?

- User experience designer
- Software architect
- Flash/Flex developer
- Coach
- Trainer
- Professional speaker
- Open source advocate



INFORMATION
FREEDOM
RULES



Who am I?

- User experience designer
- Software architect
- Flash/Flex developer
- Coach
- Trainer
- Professional speaker
- Open source advocate
- I can tie my own shoelaces



INFORMATION
FREEDOM
RULES



Who am I?

- User experience designer
- Software architect
- Flash/Flex developer
- Coach
- Trainer
- Professional speaker
- Open source advocate
- I can tie my own shoelaces
- Desperate Housewives addict!



I get excited by...

I get excited by...

- Agile development and XP

I get excited by...

- Agile development and XP
- User-centered development

I get excited by...

- Agile development and XP
- User-centered development
- Usability design, patterns, and testing

I get excited by...

- Agile development and XP
- User-centered development
- Usability design, patterns, and testing
- Software architecture

I get excited by...

- Agile development and XP
- User-centered development
- Usability design, patterns, and testing
- Software architecture
- Open source software

I get excited by...

- Agile development and XP
- User-centered development
- Usability design, patterns, and testing
- Software architecture
- Open source software
- Evangeline Lilly



but enough about me

let's talk about

software development

Houston, we have a
problem...

50-70% of all IT projects fail

Source: Standing Group Chaos Report and other sources.

and what about the
human cost?

of developers

who toil daily

under unrealistic
deadlines

implicit expectations

(e.g., usability and
accessibility)

that are impossible to
satisfy

(because they are
implicit)

(and thus not budgeted
for)

(and thus not planned in)

(or tracked)

Developers: are you
stressed out?

daily?

(Beyond the numbers
there is a very real
human welfare issue
here)

50-70% of all IT projects fail

Source: Standing Group Chaos Report and other sources.

What is failure?

Project failure

Project failure

- Cancellation

Project failure

- Cancellation
- Schedule delays

Project failure

- Cancellation
- Schedule delays
- Cost overruns

Project failure

- Cancellation
- Schedule delays
- Cost overruns
- User rejection

What is success?

Project success

Project success

- Meets requirements

Project success

- Meets requirements
- Delivered on schedule

Project success

- Meets requirements
- Delivered on schedule
- Delivered within budget

Project success

- Meets requirements
- Delivered on schedule
- Delivered within budget
- Accepted by users

If 50-70% of projects fail

The *norm* in our industry
is failure.

norm | nôrm |

noun

1 (**the norm**) something that is usual,
typical, or standard

Usual, typical, standard
= failure.

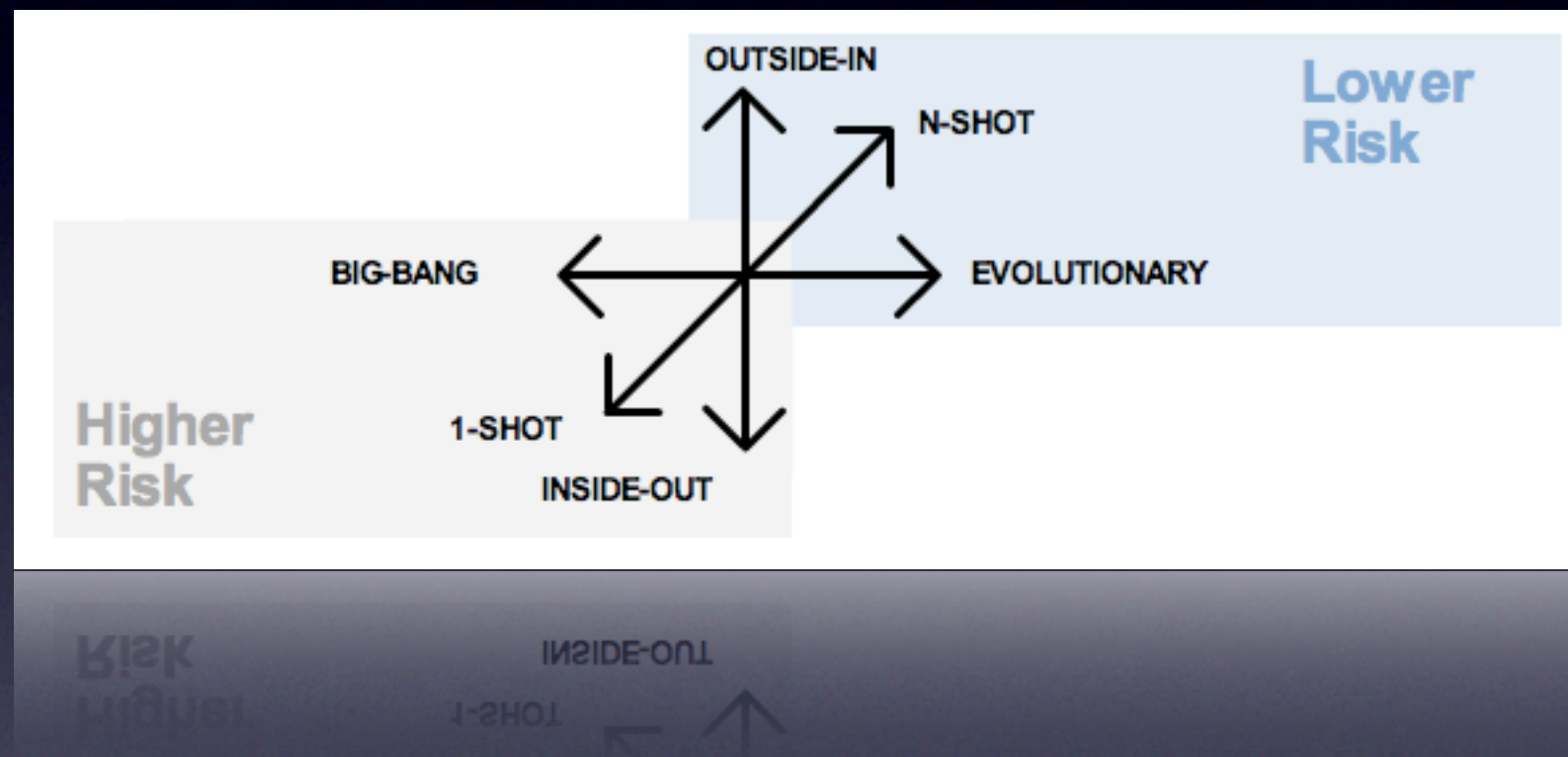
Why?

To understand why, we
have to understand a
core software
development concept...

risk

Yes, *risk*.

software development =
risk management



Development method and risk

Higher risk

Higher risk

- Big-bang

Higher risk

- Big-bang
- One-shot

Higher risk

- Big-bang
- One-shot
- Inside-out

Waterfall process

Waterfall process

I. Planning and requirements

Waterfall process

1. Planning and requirements
2. Design

Waterfall process

1. Planning and requirements
2. Design
3. Construction

Waterfall process

1. Planning and requirements
2. Design
3. Construction
4. Testing

Waterfall process

1. Planning and requirements
2. Design
3. Construction
4. Testing
5. Deployment

a waterfall process
assumes that there is an
end to development

this is a fundamentally
flawed assumption

a product that is not
being actively developed
is a *dead* product

Software development is
a *process*

of refinement.

Waterfall = norm

norm | nôrm |

noun

1 (**the norm**) something that is usual,
typical, or standard

The norm in our
industry is failure.

So: waterfall = failure.

OK, let's not be harsh...

waterfall \cong failure

(with the right people,
almost any process can
succeed)

(but will these right
people be happy working
against a process?)

Lower risk

Lower risk

- Evolutionary

Lower risk

- Evolutionary
- N-shot

Lower risk

- Evolutionary
- N-shot
- Outside-in

Agile Methodologies

Agile manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Extreme Programming (XP)

Extreme Programming (XP)

- Take processes that work to the extreme.

Extreme Programming (XP)

- Take processes that work to the extreme.
- Customer is part of the team.

Extreme Programming (XP)

- Take processes that work to the extreme.
- Customer is part of the team.
- Responsibilities are correctly allocated:
Customer makes business decisions,
development team makes technical decisions.

Extreme Programming (XP)

- Take processes that work to the extreme.
- Customer is part of the team.
- Responsibilities are correctly allocated:
Customer makes business decisions,
development team makes technical decisions.
- Change happens: how do we stay flexible and deal with it?

XP: Planning game

XP: Planning game

- Development team meets with the customer to plan

XP: Planning game

- Development team meets with the customer to plan
- Customer writes stories using plain English

XP: Planning game

- Development team meets with the customer to plan
- Customer writes stories using plain English
- i.e., no talk of buttons, windows, clicking

XP: Planning game

- Development team meets with the customer to plan
- Customer writes stories using plain English
- i.e., no talk of buttons, windows, clicking
- i.e., nothing technical

XP: Planning game

- Development team meets with the customer to plan
- Customer writes stories using plain English
- i.e., no talk of buttons, windows, clicking
- i.e., nothing technical
- (I mean it, that's the team's job!)

XP: Planning game

- Development team meets with the customer to plan
- Customer writes stories using plain English
- i.e., no talk of buttons, windows, clicking
- i.e., nothing technical
- (I mean it, that's the team's job!)
- Customer writes acceptance tests

XP: Planning game

XP: Planning game

- Team takes stories and breaks them down into tasks

XP: Planning game

- Team takes stories and breaks them down into tasks
- Team estimates tasks

XP: Planning game

- Team takes stories and breaks them down into tasks
- Team estimates tasks
- Team adds up the estimates for the tasks to arrive at estimates for the stories

XP: Planning game

- Team takes stories and breaks them down into tasks
- Team estimates tasks
- Team adds up the estimates for the tasks to arrive at estimates for the stories
- Team meets with the customer again.

XP: Planning game

XP: Planning game

- We work in iterations

XP: Planning game

- We work in iterations
- An iteration is 1 - 2 weeks

XP: Planning game

- We work in iterations
- An iteration is 1 - 2 weeks
- We ask the customer to choose the stories she wants for the first iteration

XP: Planning game

- We work in iterations
- An iteration is 1 - 2 weeks
- We ask the customer to choose the stories she wants for the first iteration
- (Yes, the customer chooses the stories)

XP: Planning game

- We work in iterations
- An iteration is 1 - 2 weeks
- We ask the customer to choose the stories she wants for the first iteration
- (Yes, the customer chooses the stories)
- If a story has dependencies, we bring it up at this point and choose those alongside.

XP

XP

- We start on an iteration

XP

- We start on an iteration
- We design sufficiently for that iteration

XP

- We start on an iteration
- We design sufficiently for that iteration
- (We don't plan too far ahead)

XP

- We start on an iteration
- We design sufficiently for that iteration
- (We don't plan too far ahead)
- (Because requirements change)

XP

- We start on an iteration
- We design sufficiently for that iteration
- (We don't plan too far ahead)
- (Because requirements change)
- So don't waste time today designing for things that may change tomorrow. Design for today!

XP

- We start on an iteration
- We design sufficiently for that iteration
- (We don't plan too far ahead)
- (Because requirements change)
- So don't waste time today designing for things that may change tomorrow. Design for today!
- Complexity is not your friend

Complexity happens;
simplicity, you have to
strive for.

XP

XP

- We start working on a story

XP

- We start working on a story
- We work in pairs

XP

- We start working on a story
- We work in pairs
- (Because writing code isn't what takes the most time, debugging and maintaining code are.)

XP

- We start working on a story
- We work in pairs
- (Because writing code isn't what takes the most time, debugging and maintaining code are.)
- Everyone has ownership of the code

XP

- We start working on a story
- We work in pairs
- (Because writing code isn't what takes the most time, debugging and maintaining code are.)
- Everyone has ownership of the code
- (To reduce the Truck Factor)

XP

- We start working on a story
- We work in pairs
- (Because writing code isn't what takes the most time, debugging and maintaining code are.)
- Everyone has ownership of the code
- (To reduce the Truck Factor)

XP

XP

- We finish a story

XP

- We finish a story
- We take it to the customer

XP

- We finish a story
- We take it to the customer
- Customer checks it against the acceptance test she wrote for the story

XP

- We finish a story
- We take it to the customer
- Customer checks it against the acceptance test she wrote for the story
- If it passes her acceptance test, customer accepts the story

XP

- We finish a story
- We take it to the customer
- Customer checks it against the acceptance test she wrote for the story
- If it passes her acceptance test, customer accepts the story
- (Great way of controlling feature creep.)

XP

XP

- We use source control

XP

- We use source control
- So we can be brave and not afraid of change

XP

- We use source control
- So we can be brave and not afraid of change
- We unit test

XP

- We use source control
- So we can be brave and not afraid of change
- We unit test
- To create a safety net for ourselves

XP

- We use source control
- So we can be brave and not afraid of change
- We unit test
- To create a safety net for ourselves
- So we can be brave and not afraid of change

XP

XP

- We update the customer if a story takes less time than we thought so she can add a story to the current iteration.

XP

- We update the customer if a story takes less time than we thought so she can add a story to the current iteration.
- If a story goes over, we tell the customer and she can take a story out.

Do not differentiate
between bugs and
feature requests

Remove that distinction
from your ticket tracker!

There are only stories.

Software development is
a process

of refinement.

But don't re-invent the
wheel

use software design
patterns

(common solutions to
common problems)

e.g., a Flash/Flex
framework that is based
on implementations of
design patterns

e.g., Arp
<http://osflash.org/arp>

XP: To summarize

XP: To summarize

- Agile methodology

XP: To summarize

- Agile methodology
- Customer is central to development

XP: To summarize

- Agile methodology
- Customer is central to development
- Iterative development

XP: To summarize

- Agile methodology
- Customer is central to development
- Iterative development
- Unit tests

XP: To summarize

- Agile methodology
- Customer is central to development
- Iterative development
- Unit tests
- Pair programming

XP: To summarize

- Agile methodology
- Customer is central to development
- Iterative development
- Unit tests
- Pair programming
- Sufficient design

XP has the customer at
the center of the
development process

but what about the *user*?



User-Centered Product Development (UCPD)

Why ask the user?

Once upon a time in
Ireland...

Eircom

Sign up for Voucher

The voucher has been ordered and paid for. You can now use the account information provided below to login to any of the *eircom* wireless broadband hotspots in Ireland.

Please save your login codes and receipt information using the download button below. Lost codes cannot be replaced.

Customer: Aral Balkan
Address line 1: 36 Frederick Street
Address line 2: Brighton
E-mail address: aral@aralbalkan.com

Order number: 89702
Payment reference number: 473439
Time: 2006-10-04 08:15

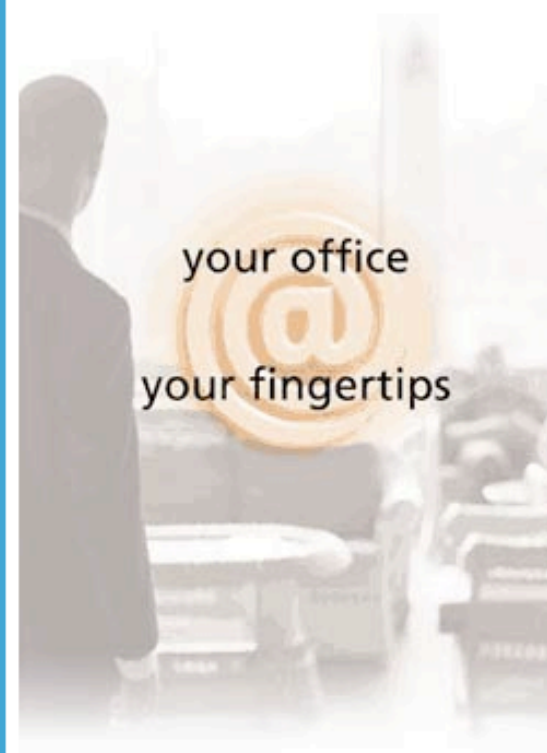
Price (incl. VAT): 3.00 EUR
VAT: 0.52 EUR
Price (excl. VAT): 2.48 EUR

Product: 30 Minute Voucher (lasts 30 minutes from first login)

Username: **u27500880**
Password: **ACP7BKKQ**

[Download receipt](#)

[Go to the login page](#)



Done

Done

signup.wlan.eircom.net

signup.wlan.eircom.net

6IRCOM

Welcome to *eircom* wireless broadband

Welcome to *eircom* wireless broadband at eircom Wireless broadband

You can't see it. You can't touch it. But right now, you're in the middle of an eircom wireless broadband access zone - a Wi-Fi hotspot.

For information on how to avail of eircom wireless broadband and to obtain a username and password, please ask at reception. Alternatively, you can signup for thirty (30) minutes, one (1) hour, twenty-four (24) hours or seven (7) days access online by clicking on the "buy online" button.

[Buy online](#)

eircom wireless broadband is also available in a number of other venues across the country. To see a full listing of the hotspots, please visit www.eircom.net/wirelessbroadband.

By using this service you are bound by Terms & Conditions of the [Acceptable Usage Policy](#). For help click [here](#)

Please type your username and password to achieve login.

Username:



Password:

(Do not insert username and password using the copy and paste function)



Communications working for you

Username:

Password:

Go


(Do not insert username and password using the copy and paste function)

Respect user effort!

User Interface Design Principles

<http://aralbalkan.com/687>

sign-up



What kind of customer are you?

STEP 1

What kind of Customer are You?

STEP 2

Choose your Internet Package

STEP 3

Customer Details

STEP 4

Set up your email address

STEP 5

Confirm your details

Welcome to eircom net.

eircom net is Ireland's leading Internet Service Provider bringing the Internet to over half a million people across Ireland.

To sign-up now follow the simple steps below.

First you need to tell us what kind of customer you are.

Note : Please note phone numbers in Portlaoise, Abbeylax, Tullamore and Birr now all begin with 057.

Birr now all begin with 027.


Note : Please note phone numbers in Portlaoise, Abbeylax, Tullamore and

First you need to tell us what kind of customer you are.

To sign-up now follow the simple steps below:

Whose life are you trying to make easier:
your's or the user's?

sign-up



What kind of customer are you?

STEP 1

What kind of Customer are You?

STEP 2

Choose your Internet Package

STEP 3

Customer Details

STEP 4

Set up your email address

STEP 5

Confirm your details

Note : Please note phone numbers in Portlaoise, Abbeylax, Tullamore and Birr now all begin with 057.

Birr now all begin with 027.

Note : Please note phone numbers in Portlaoise, Abbeylax, Tullamore and

First you need to tell us what kind of customer you are.

To sign-up now follow the simple steps below:

Whose life are you trying to make easier:
your's or the user's?

Abbey National Business



Abbey

about e-banking

security

technical
requirements

e-banking demo

logon

contact us

logon

23rd July 2006 [Changes to e-banking](#)
30th June 2006 [Latest updates on internet fraud](#)

Please enter your username, followed by your passcode and either your mother's maiden name, memorable question, or memorable answer (whichever is asked for) in the boxes below.

[Click here if you have forgotten your security details.](#)

If you are a first time user or unsure what to do please [click here for log on help](#)

User Name

Passcode

Memorable Answer

Submit

Clear

Done

ibank.anbusiness.com



Done

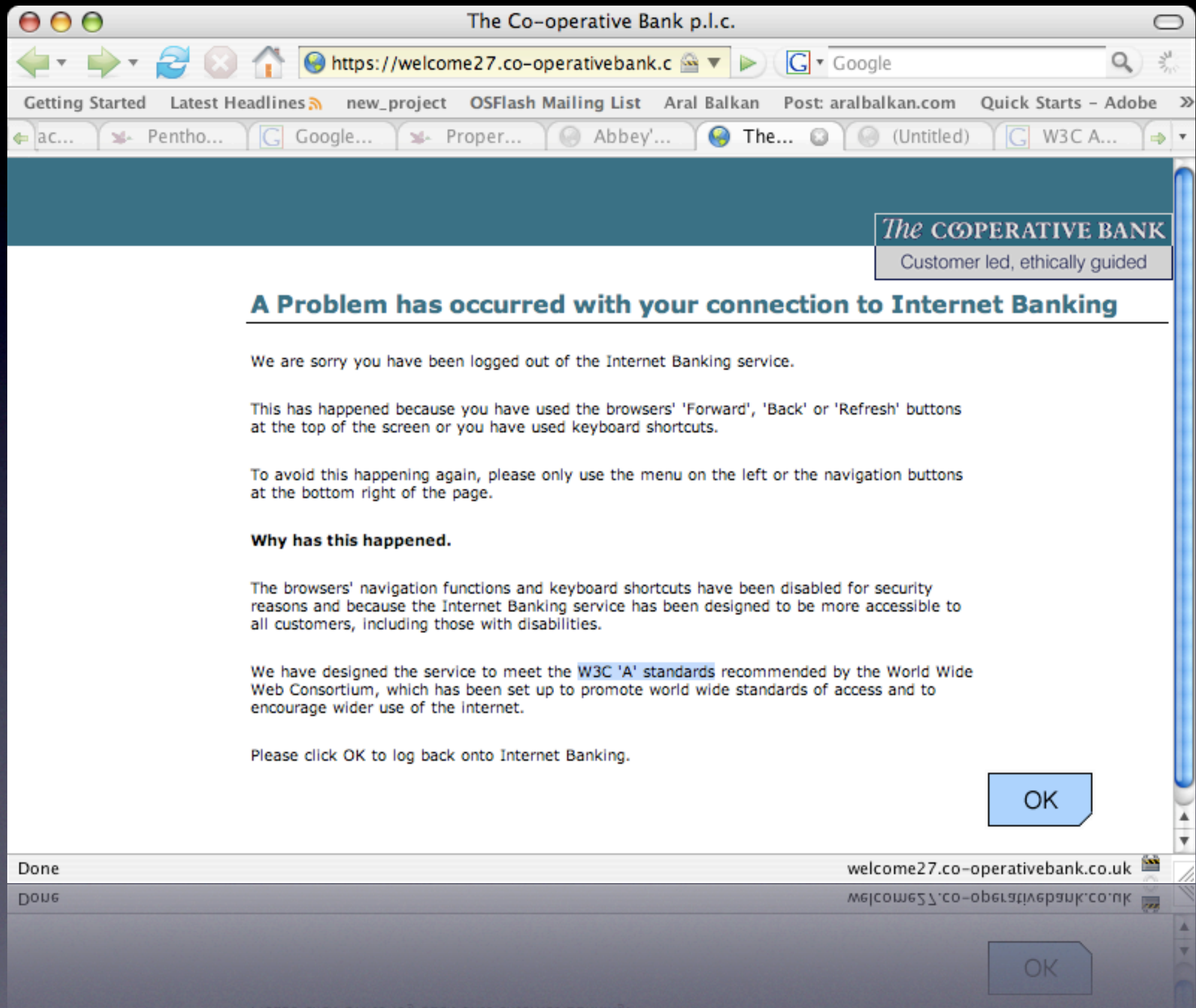
ibank.anbusiness.com



Submit

Clear

Co-operative Bank



Checkbox accessibility

is bad

Usability approach to accessibility

is good

User Interface Design Principles

<http://aralbalkan.com/687>

But, Aral, isn't all this
“common sense?”

“common sense”

Common Sense is a
dangerous myth.

Almost every bad design decision
can be traced back to someone
thinking it was “common sense.”

If you want usable
applications

that are accepted by
your users

(projects that succeed)

You need to move
beyond “common sense”

to embrace usability
design and testing

and make the user
central to your
development process.

User-Centered Product Development (UCPD)

But Aral, “anyone can
design a UI”

(and when camcorders
came out, everyone
became Spielberg)

High-level design of UI
appears simple

“anyone can design a UI”

(just look at MySpace
pages and some
shareware applications)

Special UI expertise is needed when constraints must be met for...

- Competitiveness
- Usability
- Consistency
- Cost
- Resources
- Schedule

UI is a complex creature

Why are UIs complex?

- Not uncommon for 50% of software code to be UI
- Large number of factors and unknowns
- Non-linear
- Nondeterministic
- Nonorthogonal

UI is competitive
advantage

UI as competitive advantage

- Product feature list is growing; complex
- UI should be a solution to complexity
- Simple, easy-to-use, right features, right price
= competitive advantage
- But how do you achieve this?

UCPD

UCPD

- Software development process

UCPD

- Software development process
- Iterative

UCPD

- Software development process
- Iterative
- Focus on achieving product goals for usability & other measurable factors in the product life cycle.

usability is measurable

if you gather usability
requirements alongside
functional requirements

UCPD: Requirements

UCPD: Requirements

- Easy: Business and functional requirements.

UCPD: Requirements

- Easy: Business and functional requirements.
- Overlooked: Usability, accessibility, consistency and integration requirements.

UCPD: Requirements

- Easy: Business and functional requirements.
- Overlooked: Usability, accessibility, consistency and integration requirements.
- Quantifiable usability requirements are the first step towards accountability.

UCPD: Requirements

- Easy: Business and functional requirements.
- Overlooked: Usability, accessibility, consistency and integration requirements.
- Quantifiable usability requirements are the first step towards accountability.
- Move from implicit, ambiguous expectations to explicit, measurable requirements.

UCPD

UCPD

- Involve the user throughout

UCPD

- Involve the user throughout
- (Note: This does not equate to “design by committee”)

UCPD

- Involve the user throughout
- (Note: This does not equate to “design by committee”)
- Evaluate/test throughout (test early, test often)

UCPD

- Involve the user throughout
- (Note: This does not equate to “design by committee”)
- Evaluate/test throughout (test early, test often)
- Pros/cons to domain expert in team

and in case I failed to
mention it...

test, test, test!

Testing

Testing

- Opinions of UI and usability experts important but

Testing

- Opinions of UI and usability experts important but
- Users & usability testing ultimately determine whether requirements are met

Testing

- Opinions of UI and usability experts important but
- Users & usability testing ultimately determine whether requirements are met
- No test = religious debates

Testing

- Opinions of UI and usability experts important but
- Users & usability testing ultimately determine whether requirements are met
- No test = religious debates
- Waste time, erode respect & prevent critical decisions

Myth of the average user

Myth of the average user

- Good design is not about what most people like

Myth of the average user

- Good design is not about what most people like
- There are no right answers

Myth of the average user

- Good design is not about what most people like
- There are no right answers
- Instead: Does this control work, with these contents, and this wording, in this context create a good experience for people who use this application?

User satisfaction = function of
features, UI, response time, reliability,
installability, information,
maintainability, etc.

UCPD cheat sheet

UCPD cheat sheet

- Know the user: goals and tasks

UCPD cheat sheet

- Know the user: goals and tasks
- Involve users: Test often and iterate

UCPD cheat sheet

- Know the user: goals and tasks
- Involve users: Test often and iterate
- Work to the 80/20 rule

UCPD cheat sheet

- Know the user: goals and tasks
- Involve users: Test often and iterate
- Work to the 80/20 rule
- Layer UIs according to user tasks

UCPD cheat sheet

- Know the user: goals and tasks
- Involve users: Test often and iterate
- Work to the 80/20 rule
- Layer UIs according to user tasks
- Remove needless words and features

UCPD cheat sheet

- Know the user: goals and tasks
- Involve users: Test often and iterate
- Work to the 80/20 rule
- Layer UIs according to user tasks
- Remove needless words and features
- Simplify! As Steve Krug says, “Don’t make me think!”

Complexity happens;
simplicity, you have to
strive for.

But how simple?

“...as simple as possible,
but not simpler”

(Albert Einstein)

“Cookies! Umm-num-
num-num-num!”

(Cookie Monster)

CEO Cheat Sheet

1. Implement an agile development methodology such as eXtreme Programming (XP)
2. Implement a user-centered development process.
3. Hire good people and trust them to their jobs (which will, no doubt include not reinventing the wheel and using software design patterns in their architectures.)

So, why “Memo to the
CEO?”

“Why me?”

because...

The only way these
processes will get
implemented is if they
have buy in at the
highest levels

because...

it's a matter of budget

and process

you cannot secretly
implement XP

or have a user-centered
process when usability
testing isn't budgeted
into every iteration

Grand summary

Grand summary

- Software projects are risky

Grand summary

- Software projects are risky
- In order to control the risk you need to break from tradition and cut the risk of big-bang, one-shot, inside-out development.

Grand summary

- Software projects are risky
- In order to control the risk you need to break from tradition and cut the risk of big-bang, one-shot, inside-out development.
- Embrace agility and implement a user-centered development process.

Grand summary

- Software projects are risky
- In order to control the risk you need to break from tradition and cut the risk of big-bang, one-shot, inside-out development.
- Embrace agility and implement a user-centered development process.
- Increase ROI via better understanding of user needs, tasks, goals and thus higher product adoption.

yes, I said ROI

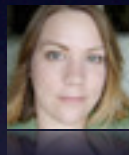
and that's the bottom line!

Thank you! :)

Credits



Aral Balkan: Dry humour and cheesy transitions



Kystal Pritchett: “The Texas Flasher”

Attribution-NonCommercial-NoDerivs 2.0



Mus’IFetish: “red3”

Attribution-NonCommercial-NoDerivs 2.0

<http://flickr.com/photos/getnlean/20926369/in/photostream>

Questions?



<http://aralbalkan.com>

<http://osflash.org>

<http://ariaaware.com>



Creative Commons Attribution 2.0 UK: England & Wales