Online Experimentation at Microsoft

Ron Kohavi
ronnyk@microsoft.com

Thomas Crook
tcrook@microsoft.com

Roger Longbotham
rogerlon@microsoft.com

Microsoft, Experimentation Platform, One Microsoft Way, Redmond, WA 98052

ABSTRACT

Knowledge Discovery and Data Mining techniques are now commonly used to find novel, potentially useful, patterns in data (Fayyad, et al., 1996; Chapman, et al., 2000). Most KDD applications involve post-hoc analysis of data and are therefore mostly limited to the identification of correlations. Recent seminal work on Quasi-Experimental Designs (Jensen, et al., 2008) attempts to identify causal relationships. Controlled experiments are a standard technique used in multiple fields. Through randomization and proper design, experiments allow establishing causality scientifically, which is why they are the gold standard in drug tests. In software development, multiple techniques are used to define product requirements; controlled experiments provide a way to assess the impact of new features on customer behavior. The Data Mining Case Studies workshop calls for describing completed implementations related to data mining. Over the last three years, we built an experimentation platform system (ExP) at Microsoft, capable of running and analyzing controlled experiments on web sites and services. The goal is to accelerate innovation through trustworthy experimentation and to enable a more scientific approach to planning and prioritization of features and designs (Foley, 2008). Along the way, we ran many experiments on over a dozen Microsoft properties and had to tackle both technical and cultural challenges. We previously surveyed the literature on controlled experiments and shared technical challenges (Kohavi, et al., 2009). This paper focuses on problems not commonly addressed in technical papers: cultural challenges, lessons, and the ROI of running controlled experiments.

1. INTRODUCTION

> *We're here to put a dent in the universe.*
> *Otherwise why else even be here?*
> *-- Steve Jobs*

On Oct 28, 2005, Ray Ozzie, Microsoft's Chief Technical Officer at the time, wrote *The Internet Services Disruption* memo (Ray Ozzie, 2005). The memo emphasized three key tenets that were driving a fundamental shift in the landscape: (i) The power of the advertising-supported economic model; (ii) the effectiveness of a new delivery and adoption model (discover, learn, try, buy, recommend); and (iii) the demand for compelling, integrated user experiences that "just work." Ray wrote that the "web is fundamentally a self-service environment, and it is critical to design websites and product 'landing pages' with sophisticated closed-loop measurement and feedback systems… This ensures that the most effective website designs will be selected…" Several months after the memo, the first author of this paper, Ron Kohavi, proposed building an Experimentation Platform at Microsoft. The platform would enable product teams to run controlled experiments.

The Workshop on Data Mining Case Studies calls for papers that "describe a completed implementation" that are "guided by the need to solve practical problems." In this paper, we intentionally avoid covering the technical aspects of controlled experiments, as these were covered elsewhere (Kohavi, et al., 2009)—rather the paper focuses on the cultural challenges, lessons, and the ROI of running controlled experiments through real examples. Over the last three years, we built an experimentation platform system (ExP) at Microsoft, capable of running and analyzing controlled experiments on web sites and services. Experiments ran on 18 Microsoft properties, including MSN® home pages in several countries (e.g., US, UK, Brazil), MSN Money, MSN Real Estate, www.microsoft.com, store.microsoft.com, support.microsoft.com, Office Online, www.xbox.com, several marketing sites, and Windows Genuine Advantage. In terms of scale, the larger experiments run with tens of millions of users, one ran with over 100M users when we need a lot of statistical power to detect a very small but critical effect. For ROI, we share relative improvements (percentages); several of our experiments had projected annual improvements of over $1M each.

The "story" we tell should **inspire** others to use controlled experiments, whether by implementing their own system or using a 3rd party system. The statistics we share about the percentage of ideas that pass all the internal evaluations, get implemented, and fail to improve the metrics they were designed to improve are **humbling**. The cultural challenges we faced in deploying a methodology that is foreign to many classical Microsoft teams should help others foster similar cultural changes in their organizations. We share stories on **education** campaigns we ran and how we raised awareness in a large company with close to 100,000 employees. We ran numerous controlled experiments on a wide variety of sites and analyzed the data using statistical and machine learning techniques. Real-world examples of experiments open people's eyes as to the potential and the return-on-investment. In this paper we share several interesting examples that show the power of controlled experiments to improve sites, establish best practices, and resolve debates with data rather than deferring to the HIghest-Paid-Person's Opinion (HiPPO) or to the loudest voice.

Our mission at the Experimentation Platform team is to accelerate software innovation through trustworthy experimentation. We have made a small dent in Microsoft's universe and would like to share the learnings so you can do the same in yours.

In Section 2, we briefly review the concept of controlled experiments. In Section 3, we describe the progress of experimentation at Microsoft over the last three years. In Section 4, we look at successful applications of experiments that help motivate the rest of the paper. In Section 5, we share some humbling statistics about the success and failure of ideas. In Section 6, we review the Application Implementation Continuum and discuss the sweet-spot for experimentation. Section 7 reviews the cultural challenges we faced and how we dealt with them. We conclude with a summary. Lessons and challenges are shared throughout the paper.

2. Controlled Experiments

> *It’s hard to argue that Tiger Woods is pretty darn good at what he does. But even he is not perfect. Imagine if he were allowed to hit four balls each time and then choose the shot that worked the best. Scary good.*
> -- *Michael Egan, Sr. Director, Content Solutions, Yahoo*

In the simplest controlled experiment, often referred to as an A/B test, users are randomly exposed to one of two variants: Control (A), or Treatment (B) as shown in Figure 1 (Kohavi, et al., 2009; Box, et al., 2005; Holland, et al., 2005; Eisenberg, et al., 2008). The key here is “random.” Users cannot be distributed “any old which way” (Weiss, 1997); no factor can influence the decision.

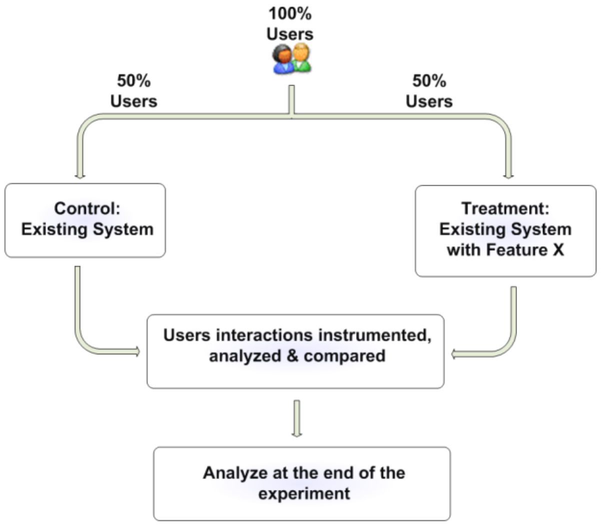

Figure 1: High-level flow for an A/B test

Based on observations collected, an Overall Evaluation Criterion (OEC) is derived for each variant (Roy, 2001). The OEC is sometimes referred to as a Key Performance Indicator (KPI) or a metric. In statistics this is often called the Response or Dependent Variable.

If the experiment was designed and executed properly, the only thing consistently different between the two variants is the change between the Control and Treatment, so any statistically significant differences in the OEC are the result of the specific change, establishing causality (Weiss, 1997 p. 215). This is where statistical tests (e.g., t-test) are used.

Common extensions to the simple A/B tests include multiple variants along a single axis (e.g., A/B/C/D) and multivariable tests where the users are exposed to changes along several axes, such as font color, font size, and choice of font.

For the purpose of this paper, the statistical aspects of controlled experiments, such as design of experiments, statistical tests, and implementation details are not important. We refer the reader to the paper *Controlled experiments on the web: survey and practical guide* (Kohavi, et al., 2009) for more details.

3. Experimentation at Microsoft

> *The most important and visible outcropping of the action bias in the excellent companies is their willingness to try things out, to experiment.*
> *There is absolutely no magic in the experiment…*
> *But our experience has been that most big institutions have forgotten how to test and learn. They seem to prefer analysis and debate to trying something out, and they are paralyzed by fear of failure, however small.*
> -- *Tom Peters and Robert Waterman,*
> *In Search of Excellence*

In 2005, when Ron Kohavi joined Microsoft, there was little use of controlled experiments at Microsoft outside Search and the MSN US home page. Only a few experiments ran as one-off ”split tests” in Office Online and on microsoft.com. The internet Search organization had basic infrastructure called “parallel flights” to expose users to different variants. There was appreciation for the idea of exposing users to different variant, and running content experiments was even patented (Cohen, et al., 2000). However, most people did not test results for statistical significance. There was little understanding of the statistics required to assess whether differences could be due to chance. We heard that there is no need to do statistical tests because “even election surveys are done with a few thousand people” and Microsoft’s online samples were in the millions. Others claimed that there was no need to use sample statistics because all the traffic was included, and hence the entire population was being tested.

In March 2006, the Experimentation Platform team (ExP) was formed as a small incubation project. By end of summer we were seven people: three developers, two program managers, a tester, and a general manager. The team’s mission was dual-pronged:

1. Build a platform that is easy to integrate
2. Change the culture towards more data-driven decisions

In the first year, a proof-of-concept was done by running two simple experiments. In the second year, we focused on advocacy and education. More integrations started, yet it was a “chasm” year and only eight experiments ultimately ran successfully. In the third year, adoption of ExP, the Experimentation Platform, grew significantly, with a new experiment starting about once a week. The search organization has evolved their parallel flight infrastructure to use statistical techniques and is executing a large number of experiments independent of the Experimentation Platform, but using the same statistical evaluations. Over 15 web properties at Microsoft ran at least one experiment with ExP, and several more properties are adopting the platform.

Testimonials from ExP adopters show that groups are seeing the value. The purpose of sharing the following testimonials isn’t self-promotion, but rather to share actual responses showing that cultural changes are happening and ExP partners are finding it highly beneficial to run controlled experiments. Getting to this point required a lot of work and many lessons that we will share in the following sections. Below are some testimonials.

- I’m thankful everyday for the work we’ve done together. The results of the experiment were in some respect counter intuitive. They completely changed our feature prioritization. It dispelled long held assumptions about <area>. Very, very useful.
- The Experimentation Platform is essential for the future success of all Microsoft online properties… Using ExP has been a tremendous boon for <team name>, and we’ve only just begun to scratch the surface of what that team has to offer.
- For too long in <team name>, we have been implementing changes on <online site> based on opinion, gut feeling or perceived belief. It was clear that

this was no way to run a successful business…Now we can release modifications to the page based purely on statistical data

- We are partnering with the ExP...and are planning to make their system a core element of our mission

The next section reviews several successful applications of controlled experiments.

4. Applications of Controlled Experiments at Microsoft

One of the best ways to convince others to adopt an idea is to show examples that provided value to others, and carry over to their domain. In the early days, publicly available examples were hard to find. In this section we share recent Microsoft examples.

4.1 Which Widget?

The MSN Real Estate site (http://realestate.msn.com) wanted to test different designs for their "Find a home" widget. Visitors to this widget were sent to Microsoft partner sites from which MSN Real estate earns a referral fee. Six different designs, including the incumbent, were tested.

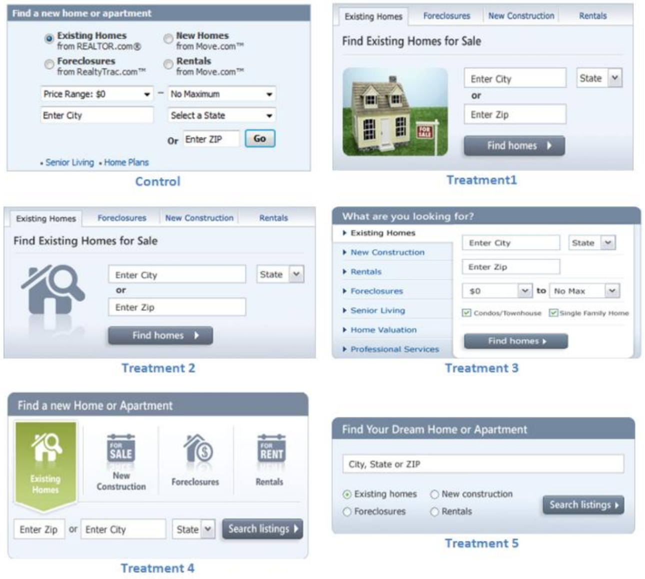

Figure 2 Widgets tested for MSN Real Estate

A "contest" was run by Zaaz, the company that built the creative designs, prior to running an experiment with each person guessing which variant will win. Only three out of 21 people guessed the winner, and the three were from the ExP team (prior experience in experiments seems to help). All three said, among other things, that they picked Treatment 5 because it was simpler. One person said it looked like a search experience.

The winner, Treatment 5, increased revenues from referrals by almost 10% (due to increased clickthrough). The Return-On-Investment (ROI) was phenomenal.

4.2 MSN Home Page Ads

A critical question that many site owners face is how many ads to place. In the short-term, increasing the real-estate given to ads can increase revenue, but what will it do to the user experience, especially if these are non-targeted ads? The tradeoff between increased revenue and the degradation of the end-user experience is a tough one to assess, and that's exactly the question that the MSN home page team at Microsoft faced.

The MSN home page is built out of modules. The Shopping module is shown on the right side of the page above the fold. The proposal was to add three offers right below it, as shown in Figure 3, which meant that these offers would show up below the fold for most users. The Display Ads marketing team estimated they could generate tens of thousands of dollars per day from these additional offers.

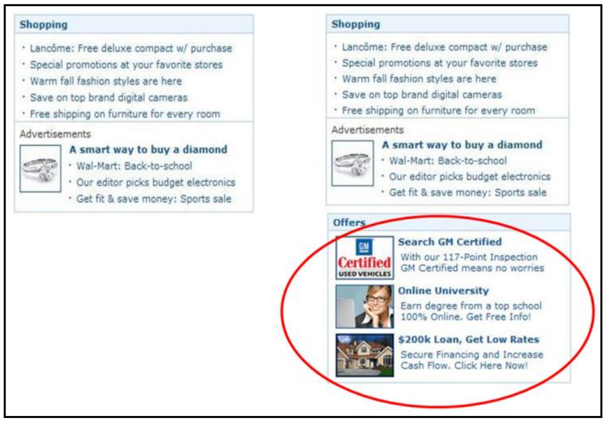
Shopping
Lancôme: Free deluxe compact w/ purchase
Special promotions at your favorite stores
Warm fall fashion styles are here
Save on top brand digital cameras
Free shipping on furniture for every room
Advertisements
A smart way to buy a diamond
Wal-Mart: Back-to-school
Our editor picks budget electronics
Get fit & save money: Sports sale
Shopping
Lancôme: Free deluxe compact w/ purchase
Special promotions at your favorite stores
Warm fall fashion styles are here
Save on top brand digital cameras
Free shipping on furniture for every room
Advertisements
A smart way to buy a diamond
Wal-Mart: Back-to-school
Our editor picks budget electronics
Get fit & save money: Sports sale
Offers
Search GM Certified
With our 117-Point Inspection GM Certified means no worries
Online University
Earn degree from a top school 100% Online. Get Free Info!
$200k Loan, Get Low Rates
Secure Financing and Increase Cash Flow. Click Here Now!

Figure 3 MSN Home Page Proposal.
Left: Control, Right: proposed Treatment

The interesting challenge here is how to compare the ad revenue with the "user experience." We refer to this problem as the OEC, or the Overall Evaluation Criterion. In this case, we decided to see if page views and clicks decreased, and assign a monetary value to each. (No statistically significant change was seen in visit frequency for this experiment.) Page views of the MSN home page have an assigned value based on ads; clicks to destinations from the MSN home page were estimated in two ways:

1. Monetary value that the destination property assigned to a click from the MSN home page. These destination properties are other sites in the MSN network. Such a click generates a visit to an MSN property (e.g., MSN Autos or MSN Money), which results in multiple page views.
2. The cost paid to search engines for a click that brings a user to an MSN property but not via the MSN home page (Search Engine Marketing). If the home page is driving less traffic to the properties, what is the cost of regenerating the "lost" traffic?

As expected, the number from #2 (SEM) was higher, as additional value beyond direct monetization is assigned to a click that may represent a new user, but the numbers were close enough to get agreement on the monetization value to use.

A controlled experiment was run on 5% of the MSN US home page users for 12 days. Clickthrough rate decreased by 0.35% (relative change), and the result was statistically significant. Page views per user-day decreased 0.35%, again a result that was highly statistically significant.

Translating the lost clicks to their monetary value, it was higher than the expected ad revenue. The estimated loss, had this feature been deployed, was millions of dollars per year.

4.3 Open in Place or in a Tab?

When a visitor comes to the MSN home page and they are recognized as having a Hotmail account, a small Hotmail convenience module is displayed. Prior to the experiment, if they clicked on any link in the module, Hotmail would open in the same tab/window as the MSN home page, replacing it. The MSN team wanted to test if having Hotmail open in a new tab/window would increase visitor engagement on the MSN because visitors will reengage with the MSN home page if it was still present when they finished reading e-mail.

The experiment included one million visitors who visited the MSN UK home page, shown in Figure 4 and clicked on the Hotmail module over a 16 day period. For those visitors the number of clicks per user on the MSN homepage increased 8.9% and the percentage of visitors who clicked on the homepage after opening Hotmail increased 6.6%. This change resulted in significant increase in user engagement and was implemented in the UK and in the US shortly after the experiment was completed.

One European site manager wrote: “This report came along at a really good time and was VERY useful. I argued this point to my team and they all turned me down. Funny, now they have all changed their minds.”

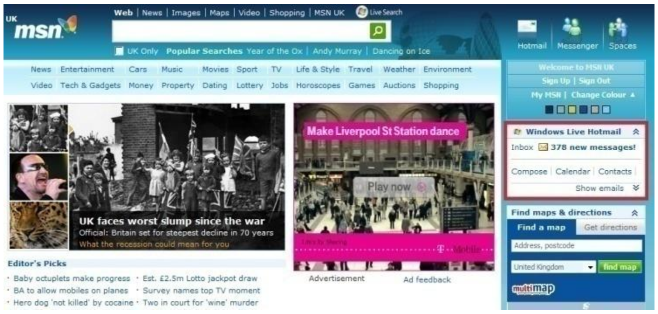
UK msn
Web | News | Images | Maps | Video | Shopping | MSN UK
Hotmail Messenger Spaces
News Entertainment Cars Music Movies Sport TV Life & Style Travel Weather Environment
Video Tech & Gadgets Money Property Dating Lottery Jobs Horoscopes Games Auctions Shopping
Welcome to MSN UK
Sign Up | Sign Out
Windows Live Hotmail
Inbox 378 new messages!
Compose | Calendar | Contacts
Show emails
Find maps & directions
Find a map Get directions
UK faces worst slump since the war
Make Liverpool St Station dance
Play now
Editor's Picks
Advertisement
Ad feedback
multimap

Figure 4 Hotmail Module highlighted in red box

4.4 Personalize Support?

The support site for Microsoft (http://support.microsoft.com) has a section near the top of the page that has answers to the most common issues. The support team wanted to test whether making those answers more specific to the user would be beneficial. In the Control variant, users saw the top issues across all segments. In the Treatment, users saw answers specific to their particular browser and operating system. The OEC was the click-through rate (CTR) on the links to the section being tested. The CTR for the treatment was over 50% higher the Control, proving the value of simple personalization.

This experiment ran as a proof of concept with manually generated issue lists. The support team now plans to add this functionality to the core system.

4.5 Pre-Roll or Post-Roll Ads?

Most of us have an aversion to ads, especially if they require us to take action to remove them or if they cause us to wait for our content to load. We ran a test with MSN Entertainment and Video Services (http://video.msn.com) where the Control had an ad that ran prior to the first video and the Treatment post-rolled the ad, after the content. The primary business question the site owners had was “Would the loyalty of users increase enough in the Treatment to make up for the loss of revenue from not showing the ad up front?” We used the first two weeks to identify a cohort of users that was then tracked over the next six weeks. The OEC was the return rate of users during this six week period. We found that the return rate increased just over 2% in the Treatment, not enough to make up for the loss of ad impressions, which dropped more than 50%.

4.6 Page Redesign?

The Microsoft Office Online store wanted to redesign their homepage to make it more appealing with the objective of increasing the percent of users who click on the Buy buttons. Any clicks within the red rectangles in Figure 5 (Control) or Figure 6 (Treatment) would count as a click on a Buy button.

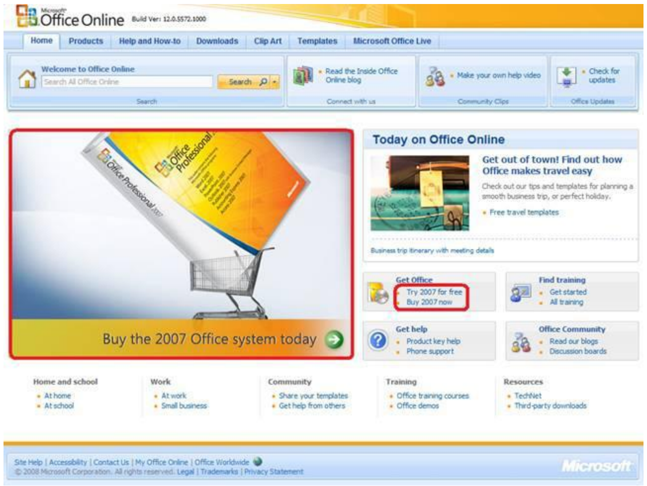

Figure 5 Control for Office Online experiment

Before you read any further, check your understanding of online customers by guessing which of these had more users click on the Buy button.

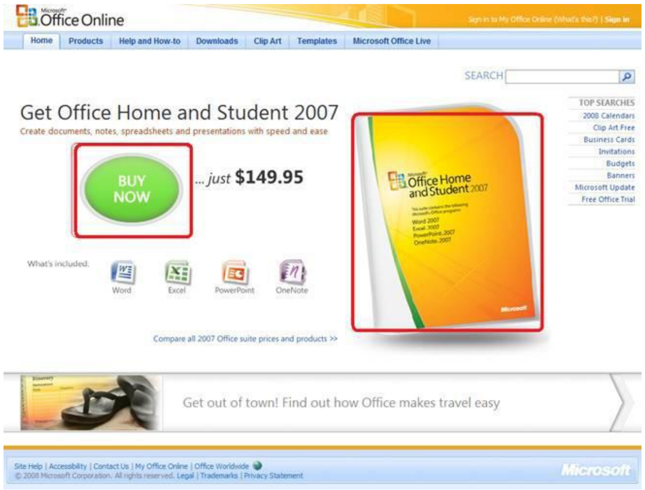
Office Online
Home Products Help and How-to Downloads Clip Art Templates Microsoft Office Live
SEARCH
Get Office Home and Student 2007
Create documents, notes, spreadsheets and presentations with speed and ease
BUY NOW
... just $149.95
What's included:
Word
Excel
PowerPoint
OneNote
TOP SEARCHES
2008 Calendars
Clip Art Free
Business Cards
Invitations
Budgets
Banners
Microsoft Update
Free Office Trial
Compare all 2007 Office suite prices and products >>
Get out of town! Find out how Office makes travel easy
Microsoft

Figure 6 Treatment for Office Online experiment

Even though most people find the Treatment more appealing, the Control had more users click on the Buy button. In fact, the Treatment had 64% fewer users click on the Buy button. Why did the Treatment perform so poorly? Unfortunately, even though a designed experiment can determine cause and effect, it cannot tell us why we got the results we did. In some cases we may see some clues in the data that could help explain the results, but the experiment can't tell us what the users were thinking. In this case there were quite a few changes between the Control and Treatment, but from our previous experience with e-commerce sites we believe the large drop is due to the fact that the Treatment shows the price of the product and the Control does not. We know from prior work that when the price is not shown there are more clicks because some users are looking for this crucial bit of information. Even though the Control had more than twice as many users click on the Buy button it is entirely possible that no more users purchased the product. This is an excellent example of a case where the experimenters should have done the additional work to measure what should have been the OEC, actual purchases.

4.7 Which Search Box?

MSN Homepage conducted a test of a change to the search box at the top of their page. A search conducted in this box uses the Live Search engine.

There were several changes between the Control (Figure 7) and the Treatment (Figure 8).

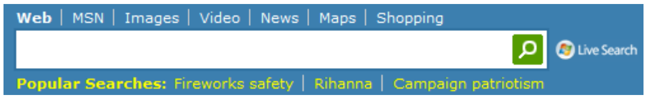
Web | MSN | Images | Video | News | Maps | Shopping
Live Search
Popular Searches: Fireworks safety | Rihanna | Campaign patriotism

Figure 7 Control for the MSN Search Box experiment

The Control has a larger area for the Search box and a magnifying glass icon while the Treatment has a larger button with the word "Search" in it. Additionally, the Control has the term Popular Searches below the search box and the Treatment simply lists the popular searches. There were a couple of other minor changes.

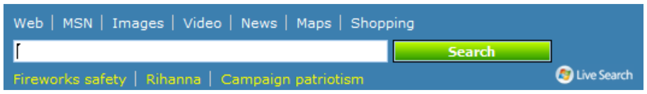
Web | MSN | Images | Video | News | Maps | Shopping
Search
Fireworks safety | Rihanna | Campaign patriotism
Live Search

Figure 8 Treatment for the MSN Search Box experiment

With so many competing changes to the search box no one knew which would draw more searches. In fact, it was a draw. Neither variant was statistically significantly better than the other.

5. Most Ideas Fail to Show Value

> *The fascinating thing about intuition is that a fair percentage of the time it's fabulously, gloriously, achingly wrong*
>
> -- *John Quarto-vonTivadar, FutureNow*

It is humbling to see how bad experts are at estimating the value of features (us included). Every feature built by a software team is built because *someone* believes it will have value, yet many of the benefits fail to materialize. Avinash Kaushik, author of *Web Analytics: An Hour a Day*, wrote in his Experimentation and Testing primer (Kaushik, 2006) that "80% of the time you/we are wrong about what a customer wants." In Do It Wrong Quickly (Moran, 2007 p. 240), the author writes that Netflix considers 90% of what they try to be wrong. Regis Hadiaris from Quicken Loans wrote that "in the five years I've been running tests, I'm only about as correct in guessing the results as a major league baseball player is in hitting the ball. That's right - I've been doing this for 5 years, and I can only "guess" the outcome of a test about 33% of the time!" (Moran, 2008).

We in the software business are not unique. QualPro, a consulting company specializing in offline multi-variable controlled experiments, tested 150,000 business improvement ideas over 22 years and reported that 75 percent of important business decisions and business improvement ideas either have no impact on performance or actually hurt performance (Holland, et al., 2005). In the 1950s, medical researchers started to run controlled experiments: "a randomized controlled trial called for physicians to acknowledge how little they really knew, not only about the treatment but about disease" (Marks, 2000 p. 156). In *Bad Medicine: Doctors Doing Harm Since Hippocrates*, David Wootton wrote that "For 2,400 years patients have believed that doctors were doing them good; for 2,300 years they were wrong." (Wooton, 2007). Doctors did bloodletting for hundreds of years, thinking it had a positive effect, not realizing that the calming effect was a side effect that was unrelated to the disease itself. When President George Washington was sick, doctors extracted about 35%-50% of his blood over a short period, which inevitably led to preterminal anemia, hypovolemia, and hypotension. The fact that he stopped struggling and appeared physically calm shortly before his death was probably due to profound hypotension and shock (Kohavi, 2008). In an old classic, Scientific Advertising (Hopkins, 1923 p. 23), the author writes that "[In selling goods by mail] false theories melt away like snowflakes in the sun... One quickly loses his conceit by learning how often his judgment errs--often nine times in ten."

When we first shared some of the above statistics at Microsoft, many people dismissed them. Now that we have run many experiments, we can report that Microsoft is no different. Evaluating well-designed and executed experiments that were designed to improve a key metric, **only about one-third were successful at improving the key metric!**

There are several important lessons here

1. Avoid the temptation to try and build optimal features through extensive planning without early testing of ideas. As Steve Kurg write: "The key is to start testing early (it's really never too early) and test often, at each phase of Web development" (Krug, 2005).
2. Experiment often. Because under objective measures most ideas fail to improve the key metrics they were designed to improve, it is important to increase the rate of experimentation and lower the cost to run experiments. Mike Moran phrased this lesson as follows: "You have to kiss a lot of frogs to find one prince. So how can you find your prince faster? By finding more frogs and kissing them faster and faster" (Moran, 2007).
3. A failure of an experiment is not a mistake: learn from it. Badly executed experiments are mistakes (Thomke, 2003), but knowing that an idea fails provides value can save a lot of time. It is well known that finding an error in requirements is 10 to 100 times cheaper than changing features in a finished product (McConnell, 2004). Use experimentation with software prototypes to verify requirements in the least costly phase of the software development lifecycle. Think of how much effort can be saved by building an inexpensive prototype and discovering that you do not want to build the production feature at all! Such insights are surprisingly common in organizations that experiment. The ability to fail fast and

try multiple ideas is the main benefit of a customer-driven organization that experiments frequently. We suggest that development teams launch prototype features regularly, and extend them, making them more robust, and then fully deploy them only if they prove themselves useful. This is a challenging proposition for organizations whose development culture has been to "do it right the first time".

4. Try radical ideas and controversial ideas. In (Kohavi, et al., 2009), we described the development of Behavior-Based Search at Amazon, a highly controversial idea. Early experiments by an intern showed the surprisingly strong value of the feature, which ultimately helped improve Amazon's revenue by 3%, translating into hundreds of millions of dollars in incremental sales. Greg Linden at Amazon created a prototype to show personalized recommendations based on items in the shopping cart (Linden, 2006). Linden notes that "a marketing senior vice-president was dead set against it," claiming it will distract people from checking out. Greg was "forbidden to work on this any further." Nonetheless, Greg ran a controlled experiment and the rest is history: the feature was highly beneficial. Multiple sites have copied cart recommendations. Sir Ken Robinson made this point eloquently when he said: "If you're not prepared to be wrong, you will not come up with anything original" (Robinson, 2006).

6. The Sweet-Spot for Experiments

> *When optimizing for conversion, we often find clients trying to improve engine torque while ignoring a flat tire*
> *-- Bryan Eisenberg and John Quarto-vonTivadar in Always Be Testing* (2008)

We now describe software development environments that present the sweet spot for controlled experiments along a continuum of product types: the Application Implementation Continuum, shown in Figure 9.

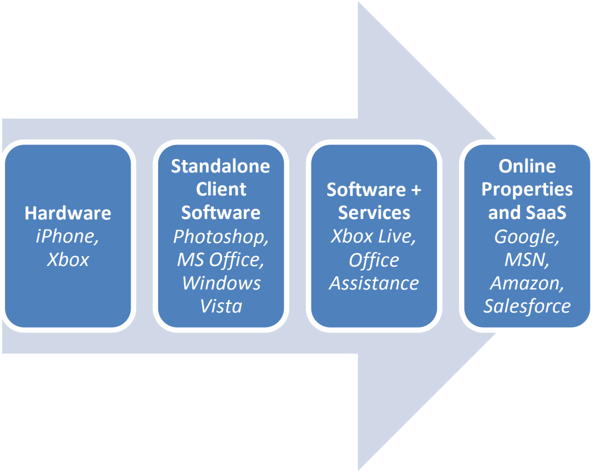
Hardware
iPhone, Xbox
Standalone Client Software
Photoshop, MS Office, Windows Vista
Software + Services
Xbox Live, Office Assistance
Online Properties and SaaS
Google, MSN, Amazon, Salesforce

Figure 9: The Application Implementation Continuum ranges from hardware devices, which are hard to change and experiment on, to online properties and SaaS, which are easy to change and experiment on.

Products on the right side of the continuum are more amenable to experimentation and agile development methodologies than those on the left. Microsoft, which had historically developed complex software products delivered on physical media, evolved development methodologies appropriate for the left side of the continuum. Younger companies, such as Amazon and Google, developed methodologies appropriate to the right side of the continuum. Below we review the ways to identify customer preferences and the ingredients necessary for running effective experiments.

6.1 Identifying Customer Preferences

Forty to sixty percent of software defects are due to errors in requirements (Wiegers, 2003). Historically, this motivated developers to increase the amount and depth of customer research carried out before design and coding commenced in earnest. Unfortunately, there are natural limits to what can be learned from this type of research:

- We cannot completely know customers' needs and usage environments before a feature is deployed. Some information about customer requirements is "sticky." i.e., it can only be discovered at the time and place where the customer uses the product (von Hippel, 1994). Beta testing can identify missing and erroneous requirements at a later stage, but such late discoveries are costly, and often arrive too late for developers to make the necessary changes.
- Customer research findings are not necessarily predictive of actual customer behavior. In the real world, customers must make tradeoffs that are not adequately captured in focus group or laboratory settings. Furthermore, what customers say in a focus group setting or a survey may not truly indicate what they prefer. A well-known example of this phenomenon occurred when Philips Electronics ran a focus group to gain insights into teenagers' preferences for boom box features. The focus group attendees expressed a strong preference for yellow boom boxes during the focus group, characterizing black boom boxes as "conservative." Yet when the attendees exited the room and were given the chance to take home a boom box as a reward for their participation, most chose black (Cross, 2005).
- Traditional customer research techniques are expensive. Teams must design usability studies and surveys, recruit participants, conduct focus groups, run user experience sessions, conduct in-depth interviews, and finally, compile, analyze and report on the results. Most studies are done with a dozen or so participants, leading to anecdotal evidence and little statistical significance.

Thomke wrote that organizations will recognize maximal benefits from experimentation when it is used in conjunction with an "innovation system" (Thomke, 2003). Agile software development is such an innovation system.

In contrast to development on the left side of the Application Implementation Continuum, development on the right side can leverage controlled experiments and fast iterations to converge on designs that please customer relatively quickly and inexpensively. Agile development teams can deploy prototypes to web sites and services as experiments, enabling the organization to learn from the customer behavior.

Ninety percent of the time to code features is typically spent in handling the edge cases of small populations. In online environments, these can be excluded from the experiments. For example, when implementing JavaScript, the browser support matrix is enormous and compatibility testing is very time consuming. For implementing a prototype, it may be enough to support a few most common browser versions. If 80% of the users do not behave as desired, it's time to go back to the drawing board. Conversely, if there is a significant boost in metrics of interest, above what was expected, the feature should be prioritized higher and possibly expanded. The development team can iterate quickly to converge on an optimized customer experience if they only have to develop the software for a few browsers. Then, once a good solution is found, they can spend the time to roll out the new experience to the long tail.

6.2 Necessary Ingredients

Controlled experiments are not applicable everywhere. In order to use agile development with controlled experiments, several ingredients have to exist.

1. A clear objective that can be practically evaluated. Controlled experiments require an Overall Evaluation Criterion, or OEC. Organizations that have not agreed what to optimize should first get agreement on that (which is sometimes hard), but as Lewis Carroll said, "If you don't know where you are going, any road will take you there." It is important to note that for many sites the OEC must represent a long-term customer value, not a short-term gain. For example, time on site and frequency of visit are better criteria than ads clicked, which is a short term metric that will lead to short-term gains and long-term doom as the site plasters itself with ads. As in the CRISP-DM model (Chapman, et al., 2000), Business Understanding is the first necessary step.
2. Easy to collect data about the user behavior. With client software, user behavior is hard to track and usually requires users to opt-in. As more of the user experience moves online, it becomes easier to track user behavior since server-side logging and client-side JavaScript logging are commonly used in industry and accepted as reasonable practices.
3. Easy to change and experiment with real users. As you move along the Application Implementation Continuum, experimentation becomes easier. At the left, we have hardware devices, which are hard to change and therefore make it harder to experiment with real users outside of focus groups and prototypes (although see several great examples in *Experimentation Matters* (Thomke, 2003)). At the other extreme are online properties, such as MSN, Amazon.com, Google and eBay, and Software as Services (Saas) implementations such as Salesforce.com, which are very easy to change. Iterative experiments with real users are easy to carry out at this end of the continuum. In between the extremes, we have standalone client software and Software+Services. In the former, experiments have to be planned and the opportunity to experiment may be limited to beta cycles. For the latter, experimentation capability has to be properly baked into the client so that server-side changes can be made to impact the client. For example, assistance/help for Microsoft Office products sends user queries to a service and receives articles to display. Software for hardware devices with connectivity can also use experimentation
4. Sufficient users exist. Very small sites or products with no customer base cannot use experimentation, but such sites typically have a key idea to implement and they need quick feedback after going live. Because new sites are aiming for big improvements, the number of users needed to detect the desired effects can be relatively small (e.g., thousands of users). Large sites, which are typically better optimized, benefit even from small improvements and therefore need many customers to increase their experiment's sensitivity level.

Most non-trivial online properties meet, or could meet, the necessary ingredients for running an agile development process based on controlled experiments. Many implementations of software+services could also meet the requirements relatively easily. For example, we are currently working with the Microsoft Zune team to use experiments to find the best music recommendation algorithms.

7. Cultural Challenges

> *There were three ways to get fired at Harrah's:*
> *steal, harass women, or institute a program or policy without first running an experiment*
>
> -- *Gary Loveman, quoted in*
> *Hard Facts* (Pfeffer, et al., 2006 p. 15)

Microsoft clearly knows how to build and ship classical "shrink-wrapped" or "client" software. There have been over 120 million Office licenses sold since the launch of Office 2007 to July 2008 (Elop, 2008). Office releases are well planned and executed over three to four years. But in the evolving world of the web and services, there is a different way of "shipping" software. Mark Lucovsky described it well (Lucovsky, 2005):

> *When an Amazon engineer fixes a minor defect, makes something faster or better, makes an API more functional and complete, how do they "ship" that software to me? What is the lag time between the engineer completing the work, and the software reaching its intended customers? A good friend of mine investigated a performance problem one morning, he saw an obvious defect and fixed it. His code was trivial, it was tested during the day, and rolled out that evening. By the next morning millions of users had benefited from his work*

Websites and services can iterate faster because shipping is much easier. In addition, getting implicit feedback from users through online controlled experiments is something that could not be done easily with shrink-wrapped products, but can easily be done in online settings. It is the combination of the two that can make a big difference in the development culture. Instead of doing careful planning and execution, one can try many things and evaluate their value with real customers in near-real-time.

Linsky and Heifetz in *Leadership on the Line* (Linsky, et al., 2002) describe *Adaptive Challenges* as those that are not amenable to standard operating procedures and where the technical know-how and procedures are not sufficient to address the challenge. We faced several non-technical challenges that are mostly cultural. It is said that the only population that likes change consists of wet babies. We share the things we did that were useful to nudge the culture toward an experimentation culture.

7.1 Education and Awareness

People have different notions of what “experiment” means, and the word “controlled” in front just doesn’t help to ground it. In 2005, no Microsoft groups that we are aware of ran proper controlled experiments with statistical tests.

In the few groups that ran “flights,” as they were called, traffic was split into two or more variants, observations were collected and aggregated, but no tests were done for statistical significance, nor were any power calculations done to determine how large a sample was needed and how long experiments should run. This led to overfitting the noise in some cases.

One of our first challenges was education: getting people to realize that what they have been doing was insufficient. Upton Sinclair wrote that “It is difficult to get a man to understand something when his salary depends upon his not understanding it.” People have found it hard to accept that many of their analyses, based on raw counts but no statistics, have been very “noisy,” to put it mildly.

We started teaching a monthly one-day class on statistics and design of experiments. Initially, we couldn’t fill the class (of about 20), but after a few rounds interest grew. To date more than 500 people at Microsoft have attended our class, which now commonly has a waiting list.

The challenge is ongoing, of course; we still find people who test ideas by comparing counts from analytical reporting tools without controlling for many factors and without running statistical tests.

We wrote the KDD paper *Practical Guide to Controlled Experiments on the Web* (Kohavi, et al., 2007) in our first year to help give the team credibility as “experts” in the field. The paper is now part of the class reading for several classes at Stanford University (CS147, CS376), USCD (CSE 291), and at the University of Washington (CSEP 510). It is getting referenced by dozens of articles and some recent book, such as King (2008).

We put posters across the Microsoft campus with examples of A/B tests or with quotations. One of the more controversial and successful ones was “Experiment or Die!,” shown in Figure 10, with a fossil and a quotation from Hal Varian at Google.

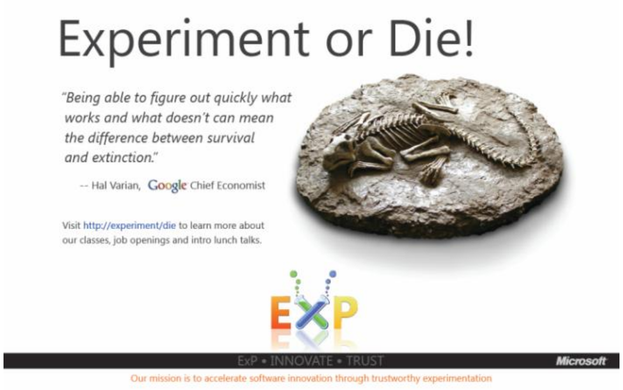
Experiment or Die!
“Being able to figure out quickly what works and what doesn’t can mean the difference between survival and extinction.”
-- Hal Varian, Google Chief Economist
Visit http://experiment/die to learn more about our classes, job openings and intro lunch talks.
EXP
ExP • INNOVATE • TRUST
Microsoft
Our mission is to accelerate software innovation through trustworthy experimentation

Figure 10: Example of a Poster: Experiment or Die!

We ate our own dog food and A/B tested our posters by creating two designs for each promotion. Each design was tagged with a unique URL offering more information about our platform, services and training classes. We compared page views for each URL to determine the effectiveness of each design.

One of our most successful awareness campaigns featured a HiPPO stress toy imprinted with our URL. HiPPO stands for the Highest Paid Person’s Opinion (Kohavi, et al., 2007). We gave away thousands of HiPPOs at the annual Microsoft employee company meetings, in our training classes, introductory talks, and through a HiPPO FAQ web site.[1] The campaign went viral, spawning word of mouth awareness and even a small fan club in Microsoft India.

We created an internal Microsoft e-mail distribution list for those interested in experimentation. There are now over 700 people on the alias.

In late 2008, enough experiments started to execute across groups that we decided to share interesting results and best practices. An internal Microsoft e-mail distribution list was created for sharing results, similar to the experiments we shared earlier in this paper.

7.2 Perceived Loss of Power

Linsky and Heifetz wrote that “People do not resist change, per se. People resist loss” (Linsky, et al., 2002). Some people certainly viewed experimentation as a risk to their power and/or prestige. Some believed it threatened their job as decision makers. After all, program managers at Microsoft select the next set of features to develop. Proposing several alternatives and admitting you don’t know which is best is hard. Likewise, editors and designers get paid to create a great design. In some cases an objective evaluation of ideas may fail and hurt their image and professional standing.

It is easier to declare success when the feature launches and not *if* it is liked by customers. We have heard statements such as “we know what to do. It’s in our DNA,” and “why don’t we just do the right thing?”

This was, and still is, a significant challenge, despite the humbling statistics about the poor success rate of ideas when evaluated objectively (see Section 5).

One actual example worth sharing revolves around the idea of automatically optimizing headlines for a major content site at Microsoft. Amazon, for example, optimized the home page slots automatically with great results (Kohavi, et al., 2004). At Microsoft, editors traditionally create headlines and order them manually, which led to several proposals to use multi-armed bandit algorithms (Wikepedia, 2008) for optimizing the headlines under some constraints, with the goal of increasing the overall click-through. An initial experiment showed some promising results, but the US editorial team pushed back strongly, claiming that this is part of their job and that they are optimizing for other criteria that are hard to quantify. Development stalled, as the development team looked for international countries willing to try this. The experiment launched outside the US and was successful even under less than ideal conditions (e.g., lower traffic). Yahoo’s paper on content optimization (Agarwal, et al., 2008) solidified the claim that this idea has merit. The issue escalated to higher management and the project got the go-ahead.

[1] See http://exp-platform.com/whatsahippo.aspx

What we found was that a great way to convince people that we are not good at predicting the outcomes of experiment is to challenge them. We created a survey with eight A/B tests, and offered a nice polo shirt for anyone who could correctly guess 6 out of 8 (the options were: A is statistically significantly better, B is statistically significantly better, or there's no statistically significant difference between them). With over 200 responses, we didn't have to hand out a single shirt! 6 out of 200 had 5 answers correct; the average was 2.3 correct answers. Humbling! At the 2008 CIKM conference (Pasca, et al., 2008), Kohavi gave an invited talk on controlled experiments and challenged the audience to predict the outcome of three actual A/B tests that ran. Out of about 150 people in the audience who stood up to the challenge, only 1 correctly guessed the outcome of two challenge questions[2]. Note that with three options to each question, this is much worse than random (150/9 = 16 people).

7.3 Reward Systems

Lee et al. (2004) write about the mixed effects of inconsistency on experimentation in organizations. They note that management can support experimentation and highlight it as a value (normative influence), but inconsistent reward systems that punish failure lead to aversion, especially in organizations that are under constant evaluation for perfect execution.

At Microsoft, as in many other large companies, employees are evaluated based on yearly goals and commitments. Conventional wisdom is that the best goals and commitments need to be SMART: specific, measurable, attainable, realistic and timely[3]. Most goals in software development organizations at Microsoft are around "shipping" products, not about their impact on customers or key metrics. In most projects, the classical triangular tradeoff exits between features, time, and quality. Some teams, such as Microsoft Office, traditionally focused on time and quality and cut features; others focused on features and quality and delayed the release schedule. Either way, features are commonly defined by their perceived value and are prioritized by program managers. Controlled experiments and the humbling results we shared bring to question whether a-priori prioritization is as good as most people believe it is. One possible change to goal definitions is to avoid tying them to products and features, but rather tie them to key metrics, and empower the development organizations to regularly test their ideas using controlled experiments. The feature development pace will undoubtedly slow down, but more corrections will be made on the way, ultimately leading to a better customer experience in shorter time.

It is hard for us to judge whether we are making any change in people's goals; cultural changes take time and it is unlikely that we have made a dent in many people's yearly performance goals. This is an ongoing challenge worth highlighting.

7.4 Incorrect Reasons Not to Experiment

Controlled experiments are a tool that has its limitations, which we discussed in *Controlled experiments on the web: survey and practical guide* (Kohavi, et al., 2009). A recent article by Davenport (2009) points out that controlled experiments are best suited for strategy execution, not strategy formulation; they are not suited for assessing a major change in business models, a large merger or acquisition (e.g., you can't run a randomized experiments on whether Microsoft should acquire Yahoo!). We agree, of course. However, we have also heard many incorrect reasons why not to experiment and would like to address them.

1. Claim: Experimentation leads to incremental innovations. While it is true that one can limit experiments to trivial UI changes like choosing colors, there is no reason experiments can't be used for radical changes and non-UI changes. Amazon makes heavy use of experimentation and its page design has evolved significantly—its first home page did not even have a search box. Multiple industry-leading innovations came from experimenting with prototypes that showed significant value and were reprioritized quickly once their value was apparent. Two such examples were described in Section 5 (item 4). One lesson that we learned is that many of our initial examples were indeed highlighting a big difference achieved through a small UI change, something that may have solidified the thinking that experimentation is best used for small incremental changes. Now we emphasize more sophisticated examples, such as whether to show ads (Section 4.5) and backend changes (Section 4.4).
2. Claim: Team X is optimizing for something that is not measurable. Here we need to differentiate between not measurable and non-economical to measure. We believe that the former is a bad way to run a business. If you can't articulate what you're optimizing for, how can the organization determine if you are doing a good job? If you are not improving a measurable metric, perhaps the other direction is also true: no measurable change will be observable without you in the organization!
   The other interpretation is more reasonable: it may be non-economical to measure the change. While this is valid at times, we would like to point to Amazon as an example of a company that did decide to measure something hard: the value of TV ads. After a 15-month-long test of TV advertising in two markets, it determined that TV ads were not a good investment and stopped them (Bezos, 2005). Is your organization avoiding experiments whose answer they would rather not know?
3. Claim: It's expensive to run experiments.
   Holland (2005) wrote that based on 150,000 business improvement ideas over 22 years, "there is no correlation between what people in the organization think will work and what actually does work… The lack of correlation between what people think will work and what does work has nothing to do with the level of the people in the organization who make these judgments. The experts are no better than the front-line workers or senior executives in determining which ideas will improve results." While we think Holland's sample is biased because his consulting company, QualPro, is brought in to help evaluate more controversial ideas, we do believe that people and organizations are overly confident of their ideas, and the poor success rate described in Section 5 strongly supports that. While it is expensive to experiment, it is more expensive to continue developing and supporting features that are not improving the metrics they were supposed to improve, or hurting them,

[2] A video recording of the presentation with the live quiz is available at http://videolectures.net/cikm08_kohavi_pgtce/

[3] Guy Kawasaki in Reality Check (2008 p. 94) suggests that goals be "rathole resistant," to avoid short-term target that dead-ends. We agree, and we have emphasized the importance of setting OECs for long-term customer lifetime-value.

and at Microsoft, these two cases account for 66% of experiments.

The flip side is to reduce costs and develop infrastructure to lower the cost of experimentation, and that's why we embarked on building the Experimentation Platform.

8. SUMMARY

It doesn't matter how beautiful your theory is, it doesn't matter how smart you are. If it doesn't agree with experiment[s], it's wrong.

-- Richard Feynman

Experimentation lies at the heart of every company's ability to innovate (Thomke, 2001; Thomke, 2003). Running physical experiments is relatively expensive, so companies have had to be parsimonious with the number of experiments. The electric light bulb required more than 1,000 complex experiments. In modern times, with the magic of software, experimentation is much cheaper, and the ability to test innovative ideas unprecedented.

Even with online experiments where experimentation is relatively cheap, the experimenter should consider the cost of experimentation. It is debatable whether a small site should experiment since the cost of installing the necessary software code, testing and maintaining it may not be less than the benefit from testing. The small site owner should also take into account that their experiments may need to run for a very long time to see any but the largest effects. Even larger sites may question whether the experimentation gain is worth the cost. Let us take the case of a site that has the necessary infrastructure for running experiments. If it is a large site the cost of developing and maintaining the code will be insignificant compared to the potential gains from experimentation. We should consider two experimental situations. The first is where you don't know whether the Treatment is better or not. In this case you should definitely test since it has been demonstrated many times that our ability to determine correctly whether the Treatment will be positive, negative or neutral is very poor. For example, in one case we ran an experiment for a site where the management was reluctant to run the test because they considered it a "no-brainer" that the Treatment would win. We agreed the value proposition looked quite promising but proceeded with the experiment. The Treatment had some unexpected and subtle negative aspects that would not have been detected had we not run the experiment. If the Treatment had been launched we estimate the annual loss to the site would have been in the millions of dollars. It only cost them $6,000 to run the experiment. The second situation to consider is when you know the Treatment is positive and you want to estimate the magnitude of the effect. For example, you want to estimate the effect of personalization for your site or the value of improving the page load time. You can prove the value of personalization without incurring large development costs by experimenting with a large or influential subset of your population and proving the value for that subset before making the decision to personalize the site for all users. You can also prove the value of improving page load time without heavy development investment by testing the effect of *slowing* the page by 100-500 msec. One situation where we would recommend you not run an experiment is when the site wants to run a continual test as a type of "scorecard" on the value of the Treatment. If a Treatment is a winner you should implement it for all applicable users. A continual test with 10% of users in the Treatment, say, will deprive 10% of your users from the better experience and your site from the benefit of the Treatment. In addition, maintaining the code for multiple versions of your site is expensive as well.

Changing the culture at a large company like Microsoft, with over 95,000 employees, is not easy. As more software is written in the form of services and web sites, the value of running controlled experiments and getting direct feedback in near-real-time rises. In the last three years, experimentation at Microsoft grew significantly in usage, but we are only at the early stages. We presented successful applications of experimentation, the many challenges we faced and how we dealt with them, and many lessons. The humbling results we shared in Section 5 bring to question whether a-priori prioritization is as good as most people believe it is. We hope this will help readers initiate similar changes in their respective organizations so that data-driven decision making will be the norm, especially in software development for online web sites and services.

ACKNOWLEDGMENTS

We would like to thank members of the Experimentation Platform team at Microsoft, especially Randy Henne and Lisa Ambler. Special thanks to David Treadwell and Ray Ozzie; without their support the experimentation platform would not have existed.

REFERENCES

Agarwal, Deepak, et al. 2008. Online Models for Content Optimization. *Neural Information Processing Systems (NIPS).* 2008. http://research.yahoo.com/node/2428.

Bezos, Jeff. 2005. The Zen of Jeff Bezos. [ed.] Chris Anderson. *Wired Magazine.* January 2005, 13.01. http://www.wired.com/wired/archive/13.01/bezos.html.

Box, George E.P., Hunter, J Stuart and Hunter, William G. 2005. *Statistics for Experimenters: Design, Innovation, and Discovery.* 2nd. s.l. : John Wiley & Sons, Inc, 2005. 0471718130.

Chapman, Peter, et al. 2000. *CRISP-DM: Step-by-step data mining guide.* 2000. http://www.crisp-dm.org/CRISPWP-0800.pdf.

Cohen, Jules S, Kromann, Paul K and Reeve, Thomas S. 2000. *Systems and methods for conducting internet content usage experiments. Patent 7,343,390* December 20, 2000. http://www.google.com/patents?vid=USPAT7343390.

Cross, Robert G. and Dixit, Ashutosh. 2005. Customer-centric pricing: The surprising secret for profitability. *Business Horizons.* 2005, Vol. 48, p. 488.

Davenport, Thomas H. 2009. How to Design Smart Business Experiments. *Harvard Business Review.* 2009, February.

Eisenberg, Bryan and Quarto-vonTivadar, John. 2008. *Always Be Testing: The Complete Guide to Google Website Optimizer.* s.l. : Sybex , 2008. 978-0470290637 .

Elop, Stephen. 2008. Financial Analyst Meeting 2008. *Microsoft Investor Relations.* [Online] July 24, 2008. http://www.microsoft.com/msft/speech/FY08/ElopFAM2008.mspx.

Fayyad, Usama, Piatetsky-Shapiro, Gregory and Smyth, Padhraic. 1996. From Data Mining to Knowledge Discovery: An Overview. [book auth.] Usama Fayyad, et al. *Advances in Knowledge Discovery and Data Mining.* Menlo Park, CA : The MIT Press, 1996, pp. 1-34.

Foley, Mary-Jo. 2008. Microsoft looks to make product planning more science than art. *ZDNet: All About Microsoft.* [Online] April 16, 2008. http://blogs.zdnet.com/microsoft/?p=1342.

Holland, Charles W and Cochran, David. 2005. *Breakthrough Business Results With MVT: A Fast, Cost-Free, "Secret Weapon" for Boosting Sales, Cutting Expenses, and Improving Any Business Process* . s.l. : Wiley, 2005. 978-0471697718 .

Hopkins, Claude. 1923. *Scientific Advertising.* New York City : Crown Publishers Inc., 1923.

Jensen, David D, et al. 2008. Automatic Identification of Quasi-Experimental Designs for Discovering Causal Knowledge. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 2008, pp. 372-380.

Kaushik, Avinash. 2006. Experimentation and Testing: A Primer. *Occam's Razor.* [Online] May 22, 2006. http://www.kaushik.net/avinash/2006/05/experimentation-and-testing-a-primer.html.

Kawasaki, Guy. 2008. *Reality Check: The Irreverent Guide to Outsmarting, Outmanaging, and Outmarketing Your Competition.* s.l. : Portfolio Hardcover , 2008. 978-1591842231 .

King, Andrew. 2008. *Website Optimization: Speed, Search Engine & Conversion Rate Secrets* . s.l. : O'Reilly Media, Inc, 2008. 978-0596515089 .

Kohavi, Ron and Round, Matt. 2004. *Front Line Internet Analytics at Amazon.com.* [ed.] Jim Sterne. Santa Barbara, CA : s.n., 2004. http://ai.stanford.edu/~ronnyk/emetricsAmazon.pdf.

Kohavi, Ron. 2008. Bloodletting: Why Controlled Experiments are Important . [Online] May 19, 2008. http://exp-platform.com/bloodletting.aspx.

Kohavi, Ron, et al. 2009. Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery.* February 2009, Vol. 18, 1, pp. 140-181. http://exp-platform.com/hippo_long.aspx.

Kohavi, Ron, Henne, Randal M and Sommerfield, Dan. 2007. Practical Guide to Controlled Experiments on the Web: Listen to Your Customers not to the HiPPO. [ed.] Rich Caruana, and Xindong Wu Pavel Berkhin. August 2007, pp. 959-967. http://exp-platform.com/hippo.aspx.

Krug, Steve. 2005. *Don't Make Me Think: A Common Sense Approach to Web Usability.* 2nd. s.l. : New Riders Press, 2005. 978-0321344755 .

Lee, Fiona, et al. 2004. The Mixed Effects of Inconsistency on Experimentation in Organizations. *Organization Science.* 2004, Vol. 15, 3, pp. 310-326.

Linden, Greg. 2006. Early Amazon: Shopping cart recommendations. *Geeking with Greg.* [Online] April 25, 2006. http://glinden.blogspot.com/2006/04/early-amazon-shopping-cart.html.

Linsky, Martin and Heifetz, Ronald. 2002. *Leadership on the Line: Staying Alive Through the Dangers of Leading.* s.l. : Harvard Business School Press, 2002. 978-1578514373 .

Lucovsky, Mark. 2005. Shipping Software . *Markl's Thoughts* . [Online] February 12, 2005. http://mark-lucovsky.blogspot.com/2005/02/shipping-software.html.

Marks, Harry M. 2000. *The Progress of Experiment: Science and Therapeutic Reform in the United States, 1900-1990.* s.l. : Cambridge University Press, 2000. 978-0521785617 .

McConnell, Steven C. 2004. *Code Complete.* 2nd Edition. s.l. : Microsoft Press, 2004.

Moran, Mike. 2007. *Do It Wrong Quickly: How the Web Changes the Old Marketing Rules* . s.l. : IBM Press, 2007. 0132255960.

—. 2008. Multivariate Testing in Action. *Biznology Blog by Mike Moran.* [Online] December 23, 2008. http://www.mikemoran.com/biznology/archives/2008/12/multivariate_testing_in_action.html.

Pasca, Marius and Shanahan, James G. 2008. Industry Event. *ACM 17th Conference on Information and Knowledge Management* . [Online] Oct 29, 2008. http://cikm2008.org/industry_event.php#Kohavi.

Pfeffer, Jeffrey and Sutton, Robert I. 2006. *Hard Facts, Dangerous Half-Truths, and Total Nonsense: Profiting from Evidence-Based Management.* s.l. : Harvard Business School Press, 2006. 978-1591398622 .

Ray Ozzie. 2005. Ozzie memo: 'Internet services disruption'. [Online] 10 28, 2005. http://news.zdnet.com/2100-3513_22-145534.html.

Robinson, Ken. 2006. Do schools kill creativity? *TED: Ideas Worth Spreading.* Feb 2006. http://www.ted.com/index.php/talks/ken_robinson_says_schools_kill_creativity.html.

Roy, Ranjit K. 2001. *Design of Experiments using the Taguchi Approach : 16 Steps to Product and Process Improvement.* s.l. : John Wiley & Sons, Inc, 2001. 0-471-36101-1.

Thomke, Stefan. 2001. Enlightened Experimentation: The New Imperative for Innovation. Feb 2001.

Thomke, Stefan H. 2003. Experimentation Matters: Unlocking the Potential of New Technologies for Innovation. 2003.

von Hippel, Eric. 1994. "Sticky information" and the locus of problem solving: Implications for innovation. *Management Science.* 1994, Vol. 40, 4, pp. 429-439.

Weiss, Carol H. 1997. *Evaluation: Methods for Studying Programs and Policies.* 2nd. s.l. : Prentice Hall, 1997. 0-13-309725-0.

Wiegers, Karl E. 2003. *Software Requirements.* 2nd Edition. s.l. : Microsoft Press, 2003.

Wikepedia. 2008. Multi-armed Bandit. *Wikipedia.* [Online] 2008. http://en.wikipedia.org/wiki/Multi-armed_bandit.

Wooton, David. 2007. *Bad Medicine: Doctors Doing Harm Since Hippocrates* . s.l. : Oxford University Press, 2007. 978-0199212798 .