Market Research + JavaScript Frameworks

CS 370 SE Practicum, Cengiz Günay

(Some slides courtesy of Eugene Agichtein and the Internets)
Agenda

Upcoming milestones:

4/10: Demo functional prototype for testing
4/17: Testing/surveying ends; fix bugs, improve concept
4/24: Final demo to tech transfer and business people
5/5: Final deliverables
Upcoming milestones:

4/10: Demo functional prototype for testing
4/17: Testing/surveying ends; fix bugs, improve concept
4/24: Final demo to tech transfer and business people
5/5: Final deliverables

Today:

- JavaScript frameworks: Backbone, Node.js, Yii
Agenda

Upcoming milestones:

4/10: Demo functional prototype for testing
4/17: Testing/surveying ends; fix bugs, improve concept
4/24: Final demo to tech transfer and business people
5/5: Final deliverables

Today:

- JavaScript frameworks: Backbone, Node.js, Yii
- First, let’s learn more about business: market research
Introduction to Market Research

Kaufmann Institute 2012

What's your story?
“It is a capital mistake to theorise before one has data”

Sir Arthur Conan Doyle
Your Target Market Must Have:

- Willingness to buy
- Purchasing power (budget/money)
- Authority to buy
Market research:

The systematic gathering, recording, and analyzing of

- Primary data
- Secondary data

with respect to a particular market, where ‘market’ refers to a specific group in a specific geographic area.
Do it

But don’t overdo it

U.S. Smartphone versus Market Purchase Consideration Factors
(1-10 scale, 10 = Most important)
Source: comScore MobiLens, Dec-2010

<table>
<thead>
<tr>
<th>Factor</th>
<th>Market</th>
<th>Smartphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phone OS</td>
<td>6.8</td>
<td>8.0</td>
</tr>
<tr>
<td>Selection of Apps</td>
<td>6.5</td>
<td>7.6</td>
</tr>
<tr>
<td>Music/Video Capabilities</td>
<td>6.4</td>
<td>7.2</td>
</tr>
<tr>
<td>Brand Name of Phone</td>
<td>6.4</td>
<td>7.0</td>
</tr>
<tr>
<td>Social Networking Features</td>
<td>6.0</td>
<td>6.9</td>
</tr>
</tbody>
</table>

©2011 Perception Partners® – Proprietary Information – Do Not Distribute – Contains No Legal Advice
Market Segmentation

**Descriptive**

**Demographic Segmentation**
- Gender
- Age
- Family Life Cycle
- Race/Ethnic Group
- Education
- Income
- Occupation
- Family Size
- Religion
- Home Ownership

**Geographic Segmentation**
- Country
- Region
- (Sub)Urban/Rural
- Population Density
- City Size
- Climate

**Behavioral**

**Psychographic Segmentation**
- Lifestyles
- Psychological variables
  - personality
  - self-image

**Benefit Segmentation**
- Expected benefits from product use

*From Richardson, Market Segmentation*
What Will Market Research Tell You?

Defining your target market

- Is the market clearly identifiable? What, exactly, is it? What could it be?
- How large is the currently served market?
- How fast is it growing?
- Who are the competitors, how large, growing?
- How will you be different from competitors?
- Current trends in the industry?
- Regulations?
- Why are competitors’ customers loyal (are they?)
- If you are successful, who benefits, who hurts?
- Comparable sales growth
- Comparable profit margins
- Likely funders in your industry
Relevance to Marketing

- **Product**
  - Helps define your product’s *standout attributes* in terms of varieties, quality, design, features, brand, packaging, sizes, service, and warranties.
  - *A technology is not a product.* Products are made and manufactured for sale, while technologies enable the product to be made.

- **Price**
  - Helps *differentiate* a pricing strategy as regards list price, discounts, allowances, payment periods, credit terms, etc.

- **Place**
  - Helps *distinguish strengths and weaknesses* regarding how your products get to your customers; which channels will you use (retail, wholesale, foodservice); best locations to reach your target market; aspects of transportation, inventory, and storage.

- **Promotion**
  - Helps *demonstrate the impact* of the following: sales promotion (coupons, allowances, discounts), advertising, salespeople, public relations?

**Positioning**

- Knowledge gained from market research helps create an image or identity in the minds of a target market.
- Permits the ability to distinguish the intangible attributes that adds value to customers.
Guess Who Else Loves Market Research?

Good ‘ole VCs (and other investors)

- They may be familiar with your market and think your product is attractive (you want to be at least as informed as they are)
- They may be familiar with your market and think your product is a bad one (you need to persuade them otherwise)
- They may be unfamiliar with your market and therefore less likely to be interested (you will need to show them the opportunity)

So is Market Research Essential?

Yes!

- Why waste your own time developing a product with little/no viable market?
- Knowing the market before you develop will drive revenues
- Knowing the market better than others will probably land you funding.
Potential Value of a Marketing Research Effort Should Exceed Its Estimated Costs

Value

- Decreased uncertainty
- Increased likelihood of correct decisions
- Improved marketing performance and resulting higher profits

Costs

- Research expenditures
- Delay of marketing decision and possible disclosure of information to rivals
- Possible erroneous research results

Value
1. Primary/Direct Research

- Basic Stages
  - Define the Question
  - Select a Research Method
  - Collect Data
  - Analyze Data
  - Draw Conclusions

2. Secondary/Indirect Research (Actually, do this first!)

- Method relevant for most research objectives
- industry research (size of market, consumption patterns)
- analysis of competitors' products
Online Scanning
- Government studies
- Industry magazines
- Business magazines
- Business directories
- Newspapers
- Demographics
- Competitor literature

Database Mining
- Statistical abstracts
- Patents/literature
- Trade organizations
- Financial reports
- Governmental authorities
- Non-governmental organizations

Sources
- Library (who has a library card?)
- Chamber of commerce
- Agricultural organizations
- Economic development offices
- Tourist offices

Conducting Secondary Research

SWOT
- Strengths
- Weaknesses
- Opportunities
- Threats

Competitors
- Identities
- Sizes
- Growth rates
- Resources
- Targets

Environment
- Regulation
- Economics
- Labor
- Uncertainty
Build off secondary research

- Review results of any secondary market research
- Decide what other decisions you will make based on the information collected and what additional information you need to make the decision

Design the primary research strategy

- Decide which method(s) is/are most appropriate given the objectives
- Decide on the appropriate sample size
- Decide who you will contact for your sample
- Develop, e.g., a survey and letter needed to gather the information
- Pre-test your survey and make changes if necessary
- Gather the information
- Organize and analyze the information
- Make a decision
- Evaluate the results of the decision
Research methods

- Primary research methods inside your business
  - Brainstorming
  - Analysis of customer/employee suggestions and complaints

- Primary research methods outside your business (claimed behaviors)
  - Consumer surveys (would you buy this product?)
  - Focus groups
  - Interviews

- Outside your business (actual behaviors)
  - Observation
  - Mystery shopping

Reaching potential customers

- How to get answers
  - Ask
  - Coax
  - Incentivize
  - Get referred

- How many to talk to
  - Sample size is critical
  - Must be large enough for the data obtained to make sense
  - Key here is to be sufficiently representative of a population
  - Determine acceptable margin of error in advance
Challenges

- No/poor data
  - Underinvestment
  - Market too early
  - Market not defined
- Weak assumptions
  - Reveals planning gaps
  - Risks discontinuous story
  - Open to challenge
- Extrapolating too much, too far
  - Overestimation
  - Small samples
  - Lack of conservatism

Outcomes with weak market research

- Low/no revenues
- Slow sales cycles with high cost of sales
- Reactive product development
- Brand dilution
- Overinvestment
- May be (way) too early
- May be (very) delayed
- Disappointed stakeholders

What Could Go Wrong?
1. Start early
2. Have a point of view in your research design
3. Involve users/customers/suppliers
4. Treat market research as ongoing
5. Tailor the research to further establish your positioning
6. Communicate findings to stakeholders, get feedback
7. Refine assumptions as you proceed
“It ain’t the things we don’t know that gets us in trouble. It’s the things we know that ain’t so.”

Artemus Ward
Understand, quantify and maximize the value from innovation. That's the IP Advantage™

For More Information | Barry Brager |
bbbrager@perceptionpartners.com
Marketing for software projects

- Targeting technology users
- Certain age groups
- Research target groups consumption devices
  - Mobile: OS version, screen resolution
  - Web: OS, browser, version
- Usage patterns:
  - What other competitor apps they use
  - What other programs they use to achieve same result?
Your market research?

- Tell me more!
Post-AJAXic Web Technologies
AJAX Web Design Pattern

Facebook like button?
AJAX Web Design Pattern

- Facebook like button?
- Each piece on the website is independent: e.g., “Edit” buttons
AJAX Web Design Pattern

- Facebook like button?
- Each piece on the website is independent: e.g., “Edit” buttons
- For complex websites, even simple AJAX of jQuery is insufficient
- What would be better? (remember Django)
AJAX Web Design Pattern

- Facebook like button?
- Each piece on the website is independent: e.g., “Edit” buttons
- For complex websites, even simple AJAX of jQuery is insufficient
- What would be better? (remember Django)
- The Top 10 Javascript MVC Frameworks Reviewed
  - Most popular: Backbone.js, Angular.js, Ember.js, Google Web Toolkit
Backbone.js: A JavaScript MVC framework

- Remember?
  - Model, View, Controller
Remember?

- Model, View, Controller

**The model:** contains data + logic: conversions, validations, computed properties, and access control. *(Backbone video tutorial (1 hr))*

```javascript
Person = Backbone.Model.extend({
    defaults: {
        name: 'Fetus',
        age: 0,
        child: '
    },
    initialize: function(){
        alert("Welcome to this world");
    }
});

var person = new Person({ name: "Thomas", age: 67, child: 'Ryan' });

var age = person.get("age"); // 67
var name = person.get("name"); // "Thomas"
var child = person.get("child"); // 'Ryan'
```
Interacting with the server

Models are used to represent data from your server and actions you perform on them will be translated to RESTful operations.

The `id` attribute of a model identifies how to find it on the database usually mapping to the surrogate key.

For the purpose of this tutorial imagine that we have a mysql table called `Users` with the columns `id`, `name`, `email`.

The server has implemented a RESTful URL `/user` which allows us to interact with it.

Our model definition shall thus look like;

```javascript
var UserModel = Backbone.Model.extend({
  urlRoot: '/user',
  defaults: {
    name: '',
    email: ''
  }
});
```
A RESTful Web Syndrome

REST: Representational state transfer: it’s just a standard CGI interface

- client-server: stateless, cacheable, layered
A RESTful Web Syndrome

REST: Representational state transfer: it’s just a standard CGI interface
- client-server: stateless, cacheable, layered

<table>
<thead>
<tr>
<th>Resource</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DELETE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection URI, such as <a href="http://example.com/resources">http://example.com/resources</a></td>
<td>List the URIs and perhaps other details of the collection’s members.</td>
<td>Replace the entire collection with another collection.</td>
<td>Create a new entry in the collection. The new entry’s URI is assigned automatically and is usually returned by the operation.</td>
<td>Delete the entire collection.</td>
</tr>
<tr>
<td>Element URI, such as <a href="http://example.com/resources/item17">http://example.com/resources/item17</a></td>
<td>Retrieve a representation of the addressed member of the collection, expressed in an appropriate Internet media type.</td>
<td>Replace the addressed member of the collection, or if it doesn’t exist, create it.</td>
<td>Not generally used. Treat the addressed member as a collection in its own right and create a new entry in it.</td>
<td>Delete the addressed member of the collection.</td>
</tr>
</tbody>
</table>
A RESTful Web Syndrome

REST: Representational state transfer: it’s just a standard CGI interface
- client-server: stateless, cacheable, layered

### RESTful API HTTP methods

<table>
<thead>
<tr>
<th>Resource</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DELETE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection URI, such as <a href="http://example.com/resources">http://example.com/resources</a></td>
<td>List the URIs and perhaps other details of the collection's members.</td>
<td>Replace the entire collection with another collection.</td>
<td>Create a new entry in the collection. The new entry's URI is assigned automatically and is usually returned by the operation.</td>
<td>Delete the entire collection.</td>
</tr>
<tr>
<td>Element URI, such as <a href="http://example.com/resources/item17">http://example.com/resources/item17</a></td>
<td>Retrieve a representation of the addressed member of the collection, expressed in an appropriate Internet media type.</td>
<td>Replace the addressed member of the collection, or if it doesn't exist, create it.</td>
<td>Not generally used. Treat the addressed member as a collection in its own right and create a new entry in it.</td>
<td>Delete the addressed member of the collection.</td>
</tr>
</tbody>
</table>

- Start scrubbing with AJAX and SOAP!
Creating a new model

If we wish to create a new user on the server then we will instantiate a new UserModel and call `save`. If the `id` attribute of the model is `null`, Backbone.js will send a POST request to the `urlRoot` of the server.

```javascript
var UserModel = Backbone.Model.extend({
    urlRoot: '/user',
    defaults: {
        name: '',
        email: ''
    }
});

var user = new UserModel();
// Notice that we haven't set an `id`

var userDetails = {
    name: 'Thomas',
    email: 'thomasalwyndavis@gmail.com'
};
// Because we have not set a `id` the server will call
// POST /user with a payload of `{name:'Thomas', email: 'thomasalwyndavis@gmail.com'}`
// The server should save the data and return a response containing the new `id`
user.save(userDetails, {
    success: function (user) {
        alert(user.toJSON());
    }
});
```

Our table should now have the values

1. 'Thomas', 'thomasalwyndavis@gmail.com'
I’m getting RESTless, how do I implement a RESTful backend?

Example: Django REST framework
Next: Backbone “View” uses jQuery

What is a view?

Backbone views are used to reflect what your applications’ data models look like. They are also used to listen to events and react accordingly. This tutorial will not be addressing how to bind models and collections to views but will focus on view functionality and how to use views with a JavaScript templating library, specifically Underscore.js’s template.

We will be using jQuery 1.8.2 as our DOM manipulator. It’s possible to use other libraries such as MooTools or Sizzle, but official Backbone.js documentation endorses jQuery. Backbone.View events may not work with other libraries other than jQuery.

For the purposes of this demonstration, we will be implementing a search box. A live example can be found on jsFiddle.

```javascript
searchView = backbone.View.extend({
    initialize: function(){
        alert("Alerts suck.");
    }
});

// The initialize function is always called when instantiating a Backbone View.
// Consider it the constructor of the class.
var searchView = new SearchView();
```

The "el" property

The "el" property references the DOM object created in the browser. Every Backbone.js view has an "el" property, and if it not defined, Backbone.js will construct its own, which is an empty div element.

Let us set our view’s "el" property to div#search_container, effectively making Backbone.View the owner of the DOM element.

```html
<div id="search_container"></div>

<script type="text/javascript">
    SearchView = Backbone.View.extend({
        initialize: function(){
            alert("Alerts suck.");
        }
    });

    var searchView = new SearchView({ el: $('#search_container') });
</script>
```

Note: Keep in mind that this binds the container element. Any events we trigger must be in this element.
Next: Backbone “View” uses jQuery

What is a view?
Backbone views are used to reflect what your applications’ data models look like. They are also used to listen to events and react accordingly. This tutorial will not be addressing how to bind models and collections to views but will focus on view functionality and how to use views with a JavaScript templating library, specifically Underscore.js’s _.template.

We will be using jQuery 1.8.2 as our DOM manipulator. It’s possible to use other libraries such as MooTools or Sizzle, but official Backbone.js documentation endorses jQuery. Backbone.View events may not work with other libraries other than jQuery.

For the purposes of this demonstration, we will be implementing a search box. A live example can be found on jsFiddle.

```javascript
searchView = backbone.View.extend({
    initialize: function(){
        alert("Alerts suck.");
    }
});

// The initialize function is always called when instantiating a Backbone View.
// Consider it the constructor of the class.
var search_view = new SearchView();
```

The "el" property
The "el" property references the DOM object created in the browser. Every Backbone.js view has an "el" property, and if it not defined, Backbone.js will construct its own, which is an empty div element.

Let us set our view’s "el" property to div#search_container, effectively making Backbone.View the owner of the DOM element.

```html
<div id="search_container"></div>

<script type="text/javascript">
    SearchView = Backbone.View.extend({
        initialize: function(){
            alert("Alerts suck.");
        }
    });

    var search_view = new SearchView({ el: $('#search_container') });
</script>
```

Note: Keep in mind that this binds the container element. Any events we trigger must be in this element.
Backbone “Controller” is called *router*

**What is a router?**

Backbone routers are used for routing your applications URL's when using hash tags (#). In the traditional MVC sense they don't necessarily have the semantics and if you have read "What is a view?" it will elaborate on this point. Though a Backbone "router" is still very useful for a router/feature that needs URL routing/history capabilities.

Defined routers should always contain at least one route and a function to map the particular route to. In the example below we are going to define a route that is always called.

Also note that routes interpret anything after "#" tag in the URL. All links in your application should target "#/action" or "#action". (Appends a forward slash after the hashtag looks a bit nicer e.g. http://example.com/#/user/help)

```javascript
<script>
    var AppRouter = Backbone.Router.extend({
        routes: {
            "*actions": "defaultRoute" // matches http://example.com/#anything-here
        }
    });

    // Initiate the router
    var app_router = new AppRouter;

    app_router.on('route:defaultRoute', function(actions) {
        alert(actions);
    })

    // Start Backbone history a necessary step for bookmarkable URL's
    Backbone.history.start();
</script>
```
Backbone “Controller” is called *router*

What is a router?

Backbone routers are used for routing your applications URL’s when using hash tags(#). In the traditional MVC sense they don’t necessarily have the semantics and if you have read "What is a view?" it will elaborate on this point. Though a Backbone "router" is still very useful for any application/feature that needs URL routing/history capabilities.

Defined routers should always contain at least one route and a function to map the particular route to. In the example below we are going to define a route that is always called.

Also note that routes interpret anything after "#" tag in the URL. All links in your application should target "#/action" or "#action". (Appends forward slash after the hashtag looks a bit nicer e.g. http://example.com/#/user/help)

```javascript
<script>
    var AppRouter = Backbone.Router.extend({
        routes: {
            "*actions": "defaultRoute" // matches http://example.com/#anything-here
        }
    });

    // Initiate the router
    var app_router = new AppRouter;

    app_router.on('route:defaultRoute', function(actions) {
        alert(actions);
    })

    // Start Backbone history a necessary step for bookmarkable URL's
    Backbone.history.start();
</script>
```
What about a framework with a backend?

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Current Version: v0.10.26
What about a framework with a backend?

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Current Version: v0.10.26

- Server written in Javascript: platform independent
- Async events: fast
- Internally uses Google V8 JS engine
AN EXAMPLE: WEBSERVER

This simple web server written in Node responds with "Hello World" for every request.

```javascript
var http = require('http);
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(1337, '127.0.0.1');
console.log('Server running at http://127.0.0.1:1337/');
```

To run the server, put the code into a file example.js and execute it with the node program from the command line:

```
% node example.js
Server running at http://127.0.0.1:1337/
```

Here is an example of a simple TCP server which listens on port 1337 and echoes whatever you send it:

```javascript
var net = require('net');

var server = net.createServer(function (socket) {
  socket.write('Echo server\n');
  socket.pipe(socket);
});

server.listen(1337, '127.0.0.1');
```
THIS IS AN AMAZING TIME, OWEN. AND WE'RE HERE TO WITNESS IT!

ABSOLUTELY! WHAT?

NODE.JS! IT'S NOT EVERY DAY A NEW WEB FRAMEWORK IS BORN.

OKAY, IT IS, BUT THIS IS THE BIG ONE! IT'S GOING TO CHANGE EVERYTHING!

HOW DOES ONE CELEBRATE A NEW WEB FRAMEWORK?

IT SAYS "IN LIEU OF GIFTS PLEASE CONTRIBUTE TO THE GITHUB REPOSITORY."

NotInventedHere™ © Bill Barnes & Paul Southworth

NotInventedHere.com
Emory Bubble uses node.js :)

A node.js tutorial book
The Fast, Secure and Professional PHP Framework

Yii is a high-performance PHP framework best for developing Web 2.0 applications.

Yii comes with rich features: MVC, DAO/ActiveRecord, I18N/L10N, caching, authentication and role-based access control, scaffolding, testing, etc. It can reduce your development time significantly.

Three steps to build your application rapidly:

1. You create the database;
2. Yii generates the base PHP code;
3. You customize the code to fit your exact needs.

Take the Tour

Or dive directly into Yii with screencasts or guide

DownloadYii

v1.1.14 released on Aug 11, 2013
Release notes • License (BSD)

Fast

Yii only loads the features that you need. It has powerful caching support. It is explicitly designed to work efficiently with AJAX.

Secure

Security comes as standard with Yii. It includes input validation, output filtering, SQL injection and Cross-site scripting prevention.

Professional

Yii helps you develop clean and reusable code. It follows the MVC pattern, ensuring a clear separation of logic and presentation.
Upcoming episode

- Thursday: your demos!
- Next week: Testing and more design patterns
- Scrum now!