How we make MOO
What, moo?
• Founded April 2006
• Launched September 2006
• shipped to 87 countries in the first month
• Offices in London
• 30 people
• Partners: Flickr, Facebook, Etsy, Bebo, iStockPhoto, etc, and you

A brief history
The MOO Philosophy

• Digital Print can be different, better (as well as) cheaper
• Make everyone a designer
• A consumer to[yo], not a geeky machine
• Simplicity of experience trumps 'features'
• A global business from day 1
• How do you sell a 2.01D object?
• MOO has few 'advanced' users
• Printing, compiling, distributing, returning, fixing
• Agile (-ish)
• weekly releases
• short projects (max 6-8 weeks)
• bugs can lead to physical costs
• Lots and lots of reporting and alerting, feedback.
• All the usals, re: uptime etc
• 5% feedback to order ratio, declining each year
• <1% reprint rate due to code errors, declining each year
• Growth slower than a pure online play
• Higher barrier to participation
• A data sink,
• Not a communication platform, either asynchronous, nor synchronous
• Hurrah, keeps costs low!
• Build for integrity, accuracy

Scaling Characteristics
• Store every image for reorders but must be accessible
• decrease in HD cost outstrips growth rate
• 6 months capacity max
• Why not S3, etc?
• Users don't linger (hurrah!) - find, make, checkout (~15 pageviews)

• International responsiveness is very important

• but growth rates are relatively predictable
• US is our biggest market, but
• US performance was lower
• US Googlejuice was way lower
• US conversion was lower
• virtually hosted squid caching proxies in NY
• 90% US traffic served from the US
• Googlejuice jump (but down slightly in UK!)
• model extends to other countries, flexes with demand
• <100 USD/month!
That sounds easy?
• Customer service (cost, quality)
• Distribution
• Physical production
• Quality control
• http://www.moo.com/service

• Actually feeds into Salesforce.com

• All other customer service tools are web-based and home grown
• 2 hours to implement (September 18th 2006)
• Cheap (a few hundred dollars/year)
• Reliable
• Ugly, clunky, web 0.5 interface
• Slows down CSRs, hinders their ability to serve excellently
## Order form

<table>
<thead>
<tr>
<th>Order ID</th>
<th>290981</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order reference</td>
<td>0500322209</td>
</tr>
<tr>
<td>Created</td>
<td>2008-10-21 15:28:29</td>
</tr>
<tr>
<td>Fulfilled</td>
<td>No</td>
</tr>
<tr>
<td>Payment reference</td>
<td>N/A</td>
</tr>
<tr>
<td>Terminal</td>
<td>No</td>
</tr>
<tr>
<td>Dispatched</td>
<td>No</td>
</tr>
<tr>
<td>Cancelled</td>
<td>No</td>
</tr>
<tr>
<td>Returned</td>
<td>No</td>
</tr>
<tr>
<td>Internal</td>
<td>Yes</td>
</tr>
<tr>
<td>Priority</td>
<td>No</td>
</tr>
<tr>
<td>Shipping products</td>
<td>Yes</td>
</tr>
<tr>
<td>Promotion</td>
<td>moointernal</td>
</tr>
</tbody>
</table>

### Order Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Product</th>
<th>Size</th>
<th>Price</th>
<th>Quantity</th>
<th>Total</th>
<th>Dispatched</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print</td>
<td>Flickr Business Cards</td>
<td>50</td>
<td>€0.00</td>
<td>1</td>
<td>€0.00</td>
<td>No</td>
</tr>
</tbody>
</table>

### Items

<table>
<thead>
<tr>
<th>Item</th>
<th>Status</th>
<th>Updated</th>
<th>Barcode</th>
<th>Batch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flickr Business Cards (50)</td>
<td>cropped</td>
<td>2008-10-21 15:32:15</td>
<td>000004177609</td>
<td>Unassigned</td>
</tr>
</tbody>
</table>
• At launch:
  • 1 product, 1 price, 1 currency, 1 shipping method, 1 shipping rate

• Now:
  • 23 products, 1 price, tiered multi-item discounts, 3 currencies, 2 shipping methods, variable discounted bulk shipping rates, 5 languages

• 495,038+ combinations

Hidden Complexity
• Global Airmail
• Vastly different local performance
• Weight/size limitations
• 2% of everything goes missing
• Returns
• Used to outsource everything
• now we buy commodity printed sheets
• laminate, cut, pack, ship ourselves
• Why? QUALITY CONTROL, (cost), Turnaround time
• Every pack gets looked at by a human (not infallible, but...)
• One day, may insource printing

Finishing
• Conceived in London
• Designed in Oakland
• Made in China (via Hong Kong)
• Lead time on proofs: weeks
• Quality control
Quality Control
• Top level measures
• Nagios, MRTG
  • Orders/hour graphs, alerts
• The Big Board
• Daily email, all employees
The Big Board
• AJAX breaks ability to test
• Authenticated APIs break ability to test capacity/performance
• IE6 (2001) is *still* a bitch
• Emails and Daily summaries of
  • PHP errors
  • PHP Fatal errors
  • (inc any SQL errors)
  • Apache errors
  • always save URI, browser details, stacktrace for these
• Automated testing, using WATIR (Web automation testing in Ruby)
• ~400 tests run for rollout
• Ajax vs Flash vs PDF
• key thing: User gets what they ordered
• Low cost product: error rate must be minimised

• A small % of CS queries can eat all your margin

• Want to deliver most sympathetic CS
• Calibration test sheets with every print run
• All orders, batches tracked by barcodes
• Error rates in Finishing closely monitored
• Rapid turnaround for reprints
• Every product has a control card
• but we like to make them fun
• Pitfalls of using a lot of APIs
• Authentication, scaling, performance
• Who builds?
money, atoms
1. write-only API is for MOO Pack creation
2. Allow 100% flexibility in text and image layout on a pack of cards (eventually)
3. But make simple templates the default (now)
4. Be as easy to use as the Flickr API (or nearly so)
5. Provide decent preview tools for developers
6. **Share revenue with API keyholders (after 1.0)**
1. Register for an API key (http://www.moo.com/api)
2. POST XML document to moo.com/api/api.php
3. Redirect users to MOO for checkout
4. Can specify return_path to bring user back to your site at the end
<?xml version="1.0" encoding="UTF-8"?>
<moo xsi:noNamespaceSchemaLocation="http://www.moo.com/xsd/api_0.6.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <request>
    <version>0.6</version>
    <api_key>88888888-4444-88888888-88888888-4444</api_key>
    <call>build</call>
  </request>
</moo>
<?xml version="1.0" encoding="UTF-8"?>
<moo xsi:noNamespaceSchemaLocation="http://www.moo.com/xsd/api_0.6.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <request>
        <version>0.6</version>
        <api_key>88888888-4444-88888888-88888888-4444</api_key>
        <call>build</call>
    </request>
    <payload>
        <chooser>
            <product_type>minicard</product_type>
            <images>
                <url>http://whitelabel.org/barcamplondon5.png</url>
            </images>
        </chooser>
    </payload>
</moo>
STEP 3: Redirect user to MOO checkout
1. Simplest format: drop up to 200 images, user chooses from within those.

2. Preselected images

3. Preselected images with specified crop, zoom data

4. Variable text, text colours, bold, etc
- can specify all text fields on all cards individually
  - 3 fonts (serif, sans serif, fixed)
  - **bold**
  - alignment (depending on product)
  - fully variable colour per text line
  - 600 different colours in a 100 pack of minicards
- avoid pale colors for text lines, especially yellows
**Examples:** variable text
• http://www.moo.com/api/validator.php
  • validates XML document
  • feedback
  • generates pdf preview
• Dan Coulter’s MOO-Builder library for PHP
• supports all features of the API
• totally easy to use.
Basic Usage

```
<?php
require_once 'lib.moo.php';
$moo = new MOOBuilder('[insert your API key here]');
$moo->addDesign($image);
$moo->setProductType('minicard');
$moo->redirect();
?>
```

http://dancoulter.com/projects/moo-builder/
Advanced Example (creating designs)

```php
<?php
require_once 'lib/lib.moo.php';
$moo = new MOObuilder('[insert your API key here]');
// We don't want to send the user to the "chooser" on MOO's page
$moo->isChooser(false);

$d = new MOOdesign();
$d->addImage('http://ssdn.us/images/b4b_minicard.jpg');
$d->addText(array('string' => 'Blogs For Bands', 'bold' => true, 'id' => 1));
$d->addText('WordPress Meets Music', 2);
$d->addText('http://blogsforbands.com', 3);
// I need to credit and link to the Creative Commons photo I used on the frontpage.
$d->addText(array('string' => 'CC Photo by Bertrand http://is.gd/gMU', 'align' => 4))
// We still have to add the design to the MOOBuilder object.
// You can add as many designs as you need.
$moo->addDesign($d);
$moo->redirect();
?>
```

http://dancoulter.com/projects/moo-builder/
• **Ruby**: [http://gems.github.com/list.html](http://gems.github.com/list.html) (James Darling)

• **PERL**: [http://search.cpan.org/dist/Net-Moo/](http://search.cpan.org/dist/Net-Moo/) (Aaron Straup)

• **.NET**: [http://www.unfinitydesign.com/blog/2/](http://www.unfinitydesign.com/blog/2/) (Nathan Kleyn)

**COBOL coming soon**
Ten Word Review
moolulu.com

Make awesome Moo cards from your Lulu books!

They're small, colourful, high quality and, above all, unique! Moo cards are a great way to advertise your books and your authoring prowess. Be proud of your Lulu creations, create some Moo cards with them today!

Enter your Lulu storefront name to get started:

Go

How do I find my Lulu storefront name?

about

http://www.moolulu.com
• 15% Revenue Share to API key holders (we will pay retroactively!)
• Business cards
• Fully variable text and imaging positioning
• Scaling, QA issues for a physical business are fundamentally different from a typical Web 2.0 business

• Must be tailored to maximising the customer experience whilst minimising costs
Questions?
© Adam 'Apelad' Koford http://www.moo.com/designs/designers/adamkoford/

Stefan Magdalalinski, stefan@moo.com