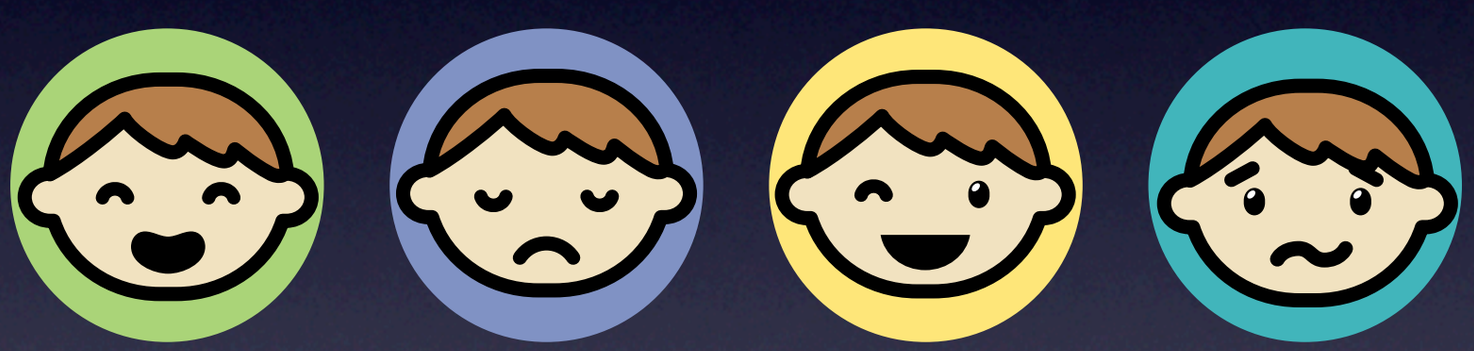
Pownce

Practical Lessons We Learned



Leah Culver

Co-founder and Lead Developer

Pownce

Social messaging application

Developed in 4 months

Invite-only launch in June

We've learned a lot ...

Lesson:

Think about technology choices

- We could pick anything!
- Social as well as technological reasons factored into our decisions
- Took risks
- Open to new technologies

Why Django?

Django is a Python web framework

- Yay! Web frameworks!
- Documentation and readability
- Auto-generated admin
- Active community
- Framework open to growth

Think about tech choices...

Why S3?

Amazon's Simple Storage Service

- Pownce files are stored on S3
- Less maintenance for Pownce
- Inexpensive
- Been very reliable so far

Think about tech choices...

Why AIR?

Adobe Integrated Runtime

- Works on both PC and Mac
- Easy to develop
- Encourages good UI
- Lots of good buzz

Think about tech choices...

Lesson:

Do a lot with a little

- Pownce has a tiny team
- One website developer
- Self-funded
- Short deadlines

Small Teams

We wear many shoes.

- Multiple roles
- Learn quickly
- Dedicated

Do a lot with a little...

Open Source Tools

Plenty of web application help

- Someone has solved this problem before
- ... and they're probably smarter than me
- Lots of tools available
- Free to use

Do a lot with a little...

Use Your Resources

Get some help.

- Documentation websites
- IRC
- Network and learn from friends
- Exchange knowledge with other sites
- Participate in communities

Do a lot with a little...

Lesson:

Be kind to your database

- Pownce's database is its main bottleneck
- One MySQL database
- Responding quickly to slow queries has helped keep Pownce running
- Few simple tips...

Caching:

“I’ve already done that.”

- We use memcached
- Caching at page and object / list level
- Cached our static pages since launch

Be kind to your database...

Queuing:

"I'll do that later."

- Taking a (shorter) note of a (longer) process to do later
- We "send" notes via a job queue
- Need to improve our queuing system and add more processes

Be kind to your database...

Limits and Pagination:

"I don't need to do ALL of that."

- Notes list, friends list, recipient lists...
- Good user interface as well
- Django Paginator object is a good starting point

Be kind to your database...

Index:

“I’ll mark that to find it later.”

- We had to re-think how we were accessing our data
- Friend searching is a prime example of where good indexing can improve performance

Be kind to your database...

Avoid Complexity:

"I won't make the db do that."

- Some queries are just too complicated (for a new web app)
- Consider if they're actually NEEDED
- Usually good to avoid abstract or conceptual data display

Be kind to your database...

[JARRING CHORD]

Expect Anything!

Lesson:

Expect Anything

- Young sites can run into many problems
- Need to respond quickly
- Can’t prepare for everything
- Every web application is unique

Keep Backups

Because stuff happens.

- Use version control
- Have a system to revert code changes
- Track dependencies and updates made
- If developing locally, backup personal work

Expect anything...

Duly Noted:

Keep lots of data.

- Stats to monitor
- Quantitative measures
- Pretty graphs

Expect anything...

Community

Keep in touch with your community.

- Let users know what you're working on
- Respond to individual bug reporters
- Inform users of bug fixes and new features
- Be careful about asserting deadlines

Expect anything...

Friendships Matter

Social sites are all about friends.

- Strive to make it easy to establish, maintain or break relationships
- Accurately represent user relationships
- Online friends have real-world effects
- Don’t mess this up!

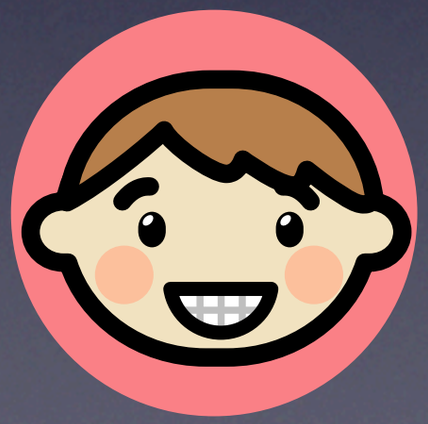
Expect anything...

A feature that matters.

Where did my friends list go? I don't see my friends on the left-side of my profile anymore. I can only see their names when I sent links, etc.

☹

(from Satisfaction)



Expect anything...

Prepare to Scale Up

It's a good problem to have.

- "Don't prematurely optimize"
- ... unless you work with Kevin Rose
- Design for success
- Accept that your code will change

Expect anything...

Lessons Learned

Think about technology choices

Do a lot with a little

Be kind to your database

Expect anything

Thanks!