

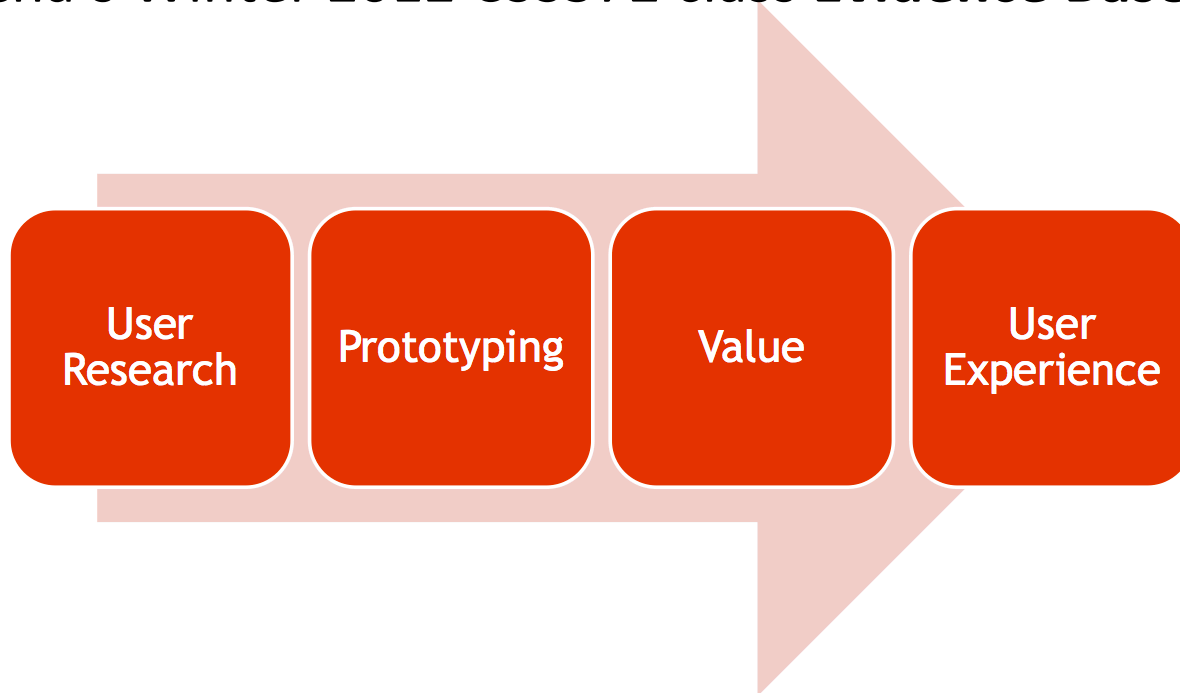
# Phix the Bookfinder

Architects United

Dan Becker & Dave Langer

# Building on previous work

Bookfinder problem space and solution concept was initially explored by Yulana Shestak, Ren Wu and Dan Becker in David Socha's Winter 2012 CSS572 class **Evidence Based Design**



No recycling—CSS590 gets 100% fresh code & written material 😊

# Description: Who Phix is for, what it isn't, and what it is

**Readers** don't want to **buy** a book, or **borrow** a book – they do these things because they **want to read books**.

Phix is **not** a library lending app, or a book buying app.

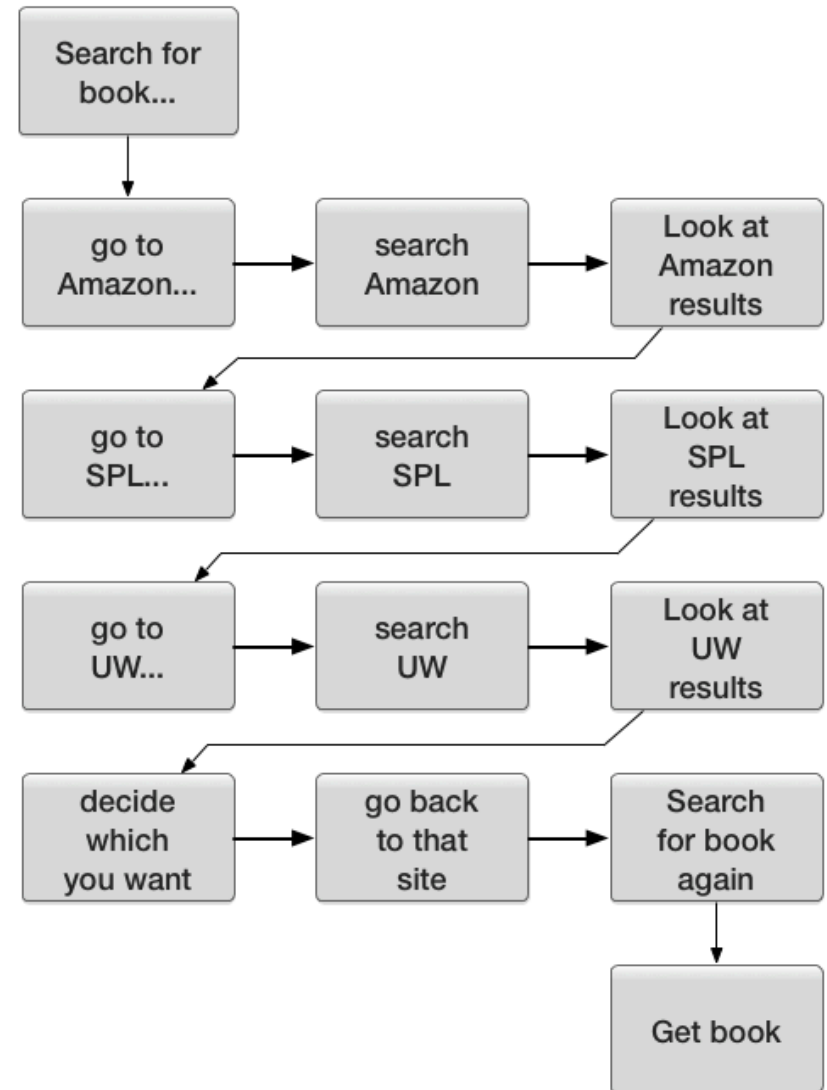
It's **not** social networking, recommendations, or ratings.

**Phix helps readers find the books they want to read.**

**Search once with Phix** to find a book in **multiple** online **stores** and **libraries near you**

# Rationale: Why Phix is needed

- There are lots of ways to get books, but searching them all one after another is a hassle!
- Not searching libraries is like “buying the cow when you can get the milk for free”
- Other book search options exist, but not good enough:
  - Too much complexity
  - Treat buying vs. borrowing as two separate searches



# Rationale: Why desirable?

## Phix addresses bookfinding pain points

- Reduce Complexity

- Focus on user path through core use case



- Increase Coherence

- Combine libraries and bookstores
- one search: one set of results
- better information: better decisions
- Phix gets you from searching to getting fast
- Send user to source to complete transaction






# Rationale: Why Mobile

- Often out of the house when you find out about a book you want to read
  - Out with friend, at work, shopping
- Use **location** to make the app **smarter** and **easier**
  - automatically finds the nearby libraries
- Future version could use **camera** for barcode scanning or cover search
  - Improves ease of use in some use cases – but not all
  - Those features are too big to add to project scope

# Schedule & Feasibility

RUP Phase	Status/Schedule	Milestones
Inception	Complete	Core use case defined Key benefits identified
Elaboration	Complete	APIs and libraries selected; proof of concept built Solution architecture and data model determined User flows and use cases outlined, prioritized User Experience design defined
Construction	Week 5	ViewModel (VM): Google Books search View (V): design views for search, results, detail, map
	Week 6	VM: Google Books book details V: implement search and results
	Week 7	VM: Worldcat search V: implement detail
	Week 8	V: implement map V/VM: refine UX/UI from early user feedback
Transition	Week 9	Release for class testing
	Week 10	Refine based on class feedback
	Week 11	Final release, Demo video

# Technical Approach

Component / API		Capabilities Provided
Google Books API		Book search Author, Title, ISBN Book cover thumbnails Ratings Description
WorldCat API		Online Computer Library Center (OCLC) book number
WP7 Location Services		User location
Bing Maps API		Convert user location to ZIP code Provide maps to local sources
WorldCat HTML servlet		Identify nearby libraries that have a book based on ZIP code

The solution is highly constrained by restrictive terms of service on APIs and platforms. Many features which are technically feasible and highly desirable to users are simply not allowed – for example, Amazon’s API restrictions block Phix.



# Phix by Architects United: Summary

**Desirable:** Phix gets readers the books they want - fast, easy, and free

**Mobile:** uses location information to automatically search nearby libraries – and map them

**Feasible:** several well-documented APIs provide most data; remainder can be scraped

**Realistic schedule:** proof of concept code already written; modular architecture allows sequential delivery of features; deliverables arranged into logical and achievable sequence