Software Not Provided:
Challenges, Experiences, and Opportunities in Supporting Communities and Democracy

Todd Davies, SSP, Stanford
Jerry Feldman, CSEE, Berkeley and ICSI
Under what conditions is software produced?

- Money motive
- Scratching your itch
- Easy to write
- Similar to something else that exists
But what about software that is...

- unprofitable
- not widely desired by developers
- hard to write/service

AND

- novel
For instance, software that is...

- designed for people who have little to no money
- designed to meet particular needs of those who are not computer experts
- complex, either in the tasks it supports or the nature of what is needed
- not already available
Questions for discussion

- What software do under-resourced people need?
- What types of tools do nonexperts need but experts not need?
- What features make software hard to write?
- What software is technically feasible but not available?
A candidate application area for underprovided software:

- community computing software especially aimed at meeting the needs of low-income communities
Subspecies of Community computing: Community-based software

- Is the software developed by, as well as for, the community?
- Do community members have control (i.e. ownership) over both technology and content?
- Is community benefit the primary goal that drives the availability of the network (as opposed to motives of an external provider)?
Examples of community computing

- Community networks
- EPA.Net
- The Election Incident Reporting System (EIRS)
- Online deliberation platforms
EPA.Net

- Project description
- Funding model: corporate philanthropy, one-time gift to CTC
- History and contribution to the community
- Challenges
Project description
- Funding model: joint grant to two nonprofit advocacy groups
- History and contribution to democracy
- Challenges
Deme

- Project description
- Funding model: university leverage grant to support student programmers
- History so far
- Challenges
- More at Groupspace.org