Effective organization for uncertain collaborations: Lessons from Free (Libre) and Open Source Software Development Teams

James Howison, PhD Candidate
Syracuse University, School of Information Studies
Advisor: Kevin Crowston
Dissertation Proposal
More at james.howison.name & floss.syr.edu

How can collaborations of serious complexity and value be accomplished working with unreliable partners? Collaboration is hard enough across organizational, temporal, motivational and geographic discontinuities without adding discontinuities in effort and commitment amongst the collaborators themselves.

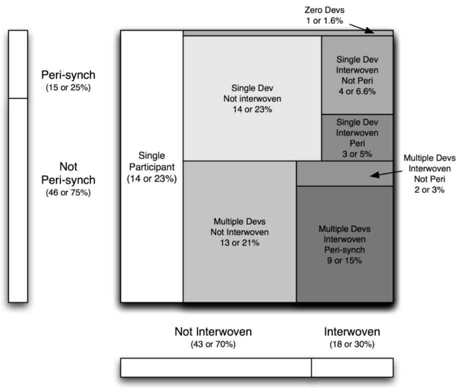
Yet some (not all) Free (Libre) and Open Source Software (FLOSS) development succeeds in building complex software in situations that begin with highly uncertain future commitment.

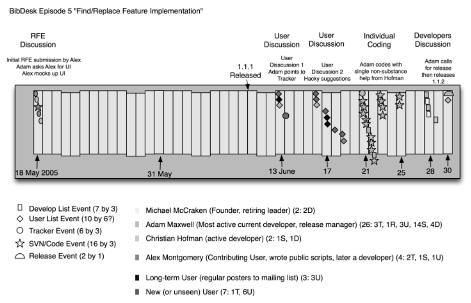
- How do FLOSS projects organize to succeed over time?
- How do their challenges and responses change over time?
- Which factors do we need to pay attention to when hoping to apply FLOSS organization techniques?

	Emerging	Growing	Coping
	Success breeds success Goal accomplishment keeps participants while the application grows in usefulness and attractiveness		
Key Challenges Emergent organization visible in 'reactions to situations', rather than plans and procedures.	Attracting • Autonomy and goal accomplishment linked to satisfaction and motivation • Being 'blocked' in volunteer work is seriously de-motivating • Organize for attraction and retention	Coordinating • Growing complexity increases dependencies • Growing audience demand attention	Defending • Influx of attracted participants threaten bloat and loss of focus • Exploitation opportunities threaten 'social contract'
	Change brings conflict Participant's mental models of the project and its challenges change at differing rates		
Resources for responding	• Degrees of organizational freedom – Flexibility in goals, tasks, deadlines – Lack of 'external audience' – 'Flow artifacts' (low instantiation costs)	Increasing commitment Application works well enough to be indispensable	Authority Past contributions and commitment define leaders
	Organizational culture is path dependent Practices build up over time, early practices remain 'default responses'		
Practices Build up over time, rather than replace. Path dependent organizational cultures	Organize for Accretion Open licenses ensure that participants can rely on previous work (can't be withdrawn) Low interdependence achieved through deferring tasks, restructuring actor-actor dependencies to actor-technology Asynchronous activities with visible archives create opportunities for help and participation	Build commitment Successful interdependencies build trust, prompting more reliance, mostly dyads Growth in 'risky' reliance: investing effort that only pays off if others act as expected Pacing tied to periodic deadlines (release schedules)	Erect Barriers to entry Institutionalization Quality 'walls'(harsh interpersonal language, especially to 'out-group') Physical meetings test commitment Faster pacing of work (excludes the non-commited)
	Projects with 'limited potential markets' may succeed without facing the challenge of defending		

Pilot Study

- Participant observer in BibDesk for three years
- Fully analyzed one month of archives
- 61 Episodes
- Clear evidence of accretion
  - Highly independent work
  - Relatively short episodes (between releases)
  - Complex features deferred until easier to build
  - Low 'risky' reliance behaviors (only 1 episode)
  - Most intra-episode cooperation is problem solving assistance (helping reliance)





Study Design

- Study full project lifetime of 3 projects using archival evidence
  - May take samples over lifetime to reduce load
- BibDesk (participant), Gaim (large and successful), Fire (fails after 4 years) Arrange all archives (Lists, CVS activity, Bug Trackers) into Episodes linked by topic made up of Events performed by Person at a Time
  - Iteratively reduce episodes, retaining links to archives
- Code episodes for practices described above
- Measure productivity over time (SLOC or features produced)
- Examine timeseries of practices, compare to predictions above and to productivity measures

Reliance: Types and responses

- **Backward looking reliance**
  - "I would never have tried this if you hadn't built the groups feature"
- **Helping reliance**
  - "Can anyone help me understand this?" Often not met, creates opportunities for peri-synchronous interaction
- **'Risky' reliance**
  - Activity is 'blocked' unless others act as expected
  - "I'm stuck, can anyone check this file in for me?"
  - "Unless SVN is accessible I un-volunteer"
- **Responses**: Defer task, restructure dependencies (eg SVN, re-factoring code), build reliability through shared knowledge of motivations (extrinsic) and institutionalization