Evolutionary Development

Austin Henderson
Scalable Conversations
Berkeley, CA, USA
henderson@rivcons.com

Jared Harris
Scalable Conversations
Berkeley, CA, USA
jed@mindloom.com

ABSTRACT

The development of large software systems using the practices of "open source" are already telling examples of World Machines. We should learn what has made open source practice so effective. Our conclusion is that open source is supporting the processes of evolution with the aid of powerful technology. Our position is that these practices can be applied beyond the narrow field of software development. (A position paper for Critical Alternatives 2015 workshop W7: World Machines.)

Author Keywords

Evolution; software development; open source; evolutionary development.

ACM Classification Keywords

K.6.3 Software Management. Software development.

INTRODUCTION

There are hard problems in the world. Ones that deserve the creativity, skills and effort of many people. Ones that will take exploring lots of alternatives, trying and learning, combining parts of different solutions to create even better solutions. Solutions will vary depending on circumstances, but different solutions have to be compatible enough to deliver more benefits than conflicts. The process of finding, sharing and implementing solutions needs to work when it gets large, even world-wide. We have been saying for some time [1,2] that problems like these require: local *responsiveness*, non-local *coherence*, and the ability to *scale*. And we have wondered how technology can be brought to bear to improve our processes on these dimensions.

Conversations that span large groups and large distances in time and space are not new. They go back at least to the invention of writing. They include the "great books" conversation that spans the time from Homer to the present, and to some degree the entire literate population of the world today. Less obviously but just as significantly are the collective "making" of skilled practices such as those of agriculture, practical geometry, architecture, schools of design, disciplines of engineering, the law, and governance.

With each advance in communication — speech, writing, printing, rapid long distance messaging, broadcasting, digital networks — we have seen new possibilities for large scale social practices to emerge, diversify, and jell into new social institutions. Some argue that these advances have always been on an exponential curve, but until recently the rate of change has been slow enough that the shape of the curve was hard to perceive. In our lifetimes, however, the rate of change has accelerated to the point where it is clearly driving the evolution of our institutions.

To deal with this rate of change we need to consider *how* advances in communication lead to change in social institutions. The development of new communication tools is entwined with the development of new practices and new institutions. For example, writing was not invented simply to communicate better. Our earliest examples of cuneiform record accounting data such as quantities contributed to and withdrawn from granaries. This writing enabled a far more persistent and accurate community memory, and in turn led to new, more abstract kinds of rights and obligations.

Over the following several thousand years, evolving communication technologies opened up the possibilities for new social institutions, but then the institutions had to grow in the new territory, realizing some possibilities and blocking others. These technologies enabled major new growth in such domains as law, tax collection, definitions of land ownership, military organizations. They also enabled the emergence of science, engineering, mathematics, philosophy, history and literature through the large scale literate conversation around the Mediterranean.

Improvement in communication has enabled institutions (e.g., the Roman Empire and the Catholic Church) to achieve considerable coherence over large areas and significant time periods. However this large scale coherence has come at a high price. For the last few millennia, nearly all institutions that have been carrying out large scale, complex, interdependent tasks have imposed strong control through hierarchical organization. Although there are a few important exceptions (e.g., the development of modern science), the pattern has been so close to universal that we tend to take it as a natural law: "Big, complex projects require strong hierarchical governance."

However, with the advent of modern communications and computing, this "law" no longer holds. New governance possibilities have opened up, and new kinds of institutions have grown up in this new territory. The exponential improvement in communication, storage and computation in the last few decades has led to the emergence of a radically new kind of institution, generically called "open peer projects." These allow large and diverse groups to form, to discuss and experiment with alternatives, and to synthesize and distribute their results, with no one in charge and giving orders. The speed and quality of large scale projects has increased, and their effective cost has fallen dramatically.

Wikipedia—the defacto encyclopedia at least for English-speakers— is perhaps the best known example, based on on-going contributions from hundreds of thousands of contributors all over the world.

A somewhat less well-known, but even larger and more powerful, example is *open source software development*. Open source projects are based on voluntary collaboration among as many as a quarter of a million software developers. Open source practices provide ways to *explore alternatives*, *manage projects*, and *coordinate* between people who may never meet, and indeed who may not even speak a common language.

These new institutions have demonstrated that large complex projects can succeed without any centralized control, and furthermore can beat the productivity, range of exploration, coherence, scale and speed of response of comparable centrally-managed projects, all at immensely lower cost. Further, they have shown a superior ability to address issues where the solution, and even the desired goal, is not known at the outset and indeed is developed as everyone proceeds. These successes rest on practices involving *concurrent independent exploration of alternatives* that is later *unanticipatedly merged* into systems that are more coherent than any of the explored threads.

Wikipedia and Open Source Software projects are visible and commonly-known examples of such *open peer projects*. Others include mapping (Open Street Map), molecular biology (Gene_Wiki), research mathematics (PolyMath), and documentation (ArXive):

We believe that many domains of society could benefit from open peer practices and tools. For example: climate models, economic models, curation and analysis of test data (e.g., drug trials, education), legal code, historical archives, models for printable objects.

Note also that in many of these domains, the results to some extent *are* the summarized debate. For example, the legislative debate surrounding a law, or the issues raised during analysis of drug trials, are essential context for understanding and interpreting the resulting decisions. This is in contrast to open source software development where the *work* of creating and debugging code is quite separate and distinct from the work of *managing* code development. This makes the open history of peer projects even more valuable for this kind of domain. We also suggest that even software development might be improved by extending the code-management practices to include the discussions within which it is embedded.

All of these peer projects have been, or would be, enabled by better communication, storage and computation, but that is not sufficient. The projects also depend on using these better resources through open peer practices. Furthermore, they depend on automated support for multiple concurrent interacting development. In this pattern, things advance much the way a wave does, with every segment of the wave responding to local conditions (responsive), spreading and merging with and altering others, and the wave as a whole seeking some or many points of equilibrium (coherence). And waves get large (scalable).

In a different sense, this pattern of development has been around much longer even than writing, longer than any of our technology, and indeed longer than homo sapiens as a species. Open peer development proceeds through local changes, diffusion of the changes, competition between alternate branches, merging and consolidation. This very same pattern can be seen in the evolution of species and ecologies: mutations are the local changes, and the processes of diffusion, competition and consolidation are the evolution of populations. The development of cultures and languages also follows the same pattern, through the local change, diffusion, competition, merging and consolidation of linguistic artifacts, including memes, practices, vocabulary, and syntax. Science too fits the pattern, with decentralized generation and review of experimental results, and development and revision of models, theories, and laws. We therefore have come to refer to this pattern as *evolutionary development*.

As we have said, evolutionary development is hardly new. What *is* new about the emerging open peer practices of evolutionary development is that they make this powerful form of development *explicit*, *accountable*, *faster* and *more efficient*. Although this is only a change in degree, the effect is great enough to cross a threshold, dramatically increase the speed and diversity of development projects, and repeal of the "law of centralized control".

What, then, makes the evolutionary development practices of open peer projects so effective? Our analysis of existing examples has lead us to the following preliminary conclusions:

- Complete record: Contributors can learn from their history; which means they are able to review history; which means capturing history as it happens. Ideally they get a complete archive of all changes, which records every significant move, but does not get in the way of continuing to change things.
- Independent activity: Contributors can work independently without being tied to some central organization; and they can merge their work with the work of many others with whom there has been no anticipated plan to merge, no intervening communication, or even any awareness that their activities are happening and related.

- Archive of discussions: Contributors can capture issues that concern them, share them with others, discuss them (and capture the discussion for later review), and propose and negotiate relationships between them. The results are accessible to anyone, whether old hands or newcomers.
- Sharing: Contributors can package up pieces of work that they have been doing, together with its history, and hand it off to others.
- Voluntary participation: Each project can attract and use volunteer contributions. Projects let anyone explore the work that has been done, observe current work in progress, and participate whenever they want. This means that the process is not driven by schedules or by assignment of tasks.
- Concurrent development: Each project can ensure that that no participant can cause substantial damage to the whole, or hold up the show. Contributions are made in a ways that can be tracked, evaluated and integrated optionally and reversibly.
- Testing: The project can test whether a given change improves things or makes them worse. Further, these tests are retained and reused to ensure that subsequent changes continue to leave things functioning properly.
- Open-ended content: Contributors can hold open-ended discussions of issues that span specific parts of the existing system, or that fall outside structure of the work itself.

This may seem demanding — perhaps even utopian. But these requirements are routinely met by thousands of open peer projects, involving hundreds of thousands or even millions of contributors worldwide. These projects already build and maintain many important components of our technical environment, and they are evolving into an essential component of supporting our economy and society

These requirements are met by a combination of highly evolved institutional practices, and tools that support and, increasingly, directly embody those practices:

- Version management: The capture of history, packaging of results, merging of multiple contributions, tracking specific changes and so forth is supported by version management. Originally based on manual practices, these practices have now largely been built into version management software.
- Conversation capture: Tracking of issues, discussion of designs, planning complex activities, maintaining institutional memory of previous discussions, are supported by ticketing systems, email archives, discussion forums and wikis. Here again, these tools evolved through increasing automation of evolving practices, when those practices became increasingly routine

We believe the potential benefits of open peer projects are enormous and that it would be extremely valuable to open up these "best practices" to the widest possible audience and range of applications.

At the same time, our analysis of discussions in open peer projects, and in Open Source projects in particular, suggests that there is much more that can be done to make these discussions more effective and efficient. We aspire to support collaborations that are: open, versioned, multi-local, coherence conducive, extensible, multi-grained and domain-spanning.

We are proceeding by building prototypes, integrated at three layers (*storage*: for holding of the stuff of conversations, *domain*: for the concepts and substance of conversations, and *interaction*: for manipulating the domain material). We are seeking to make each layer a user-accessible, construction-oriented, evolvable tool-set; and the whole platform to be evolvable and open.

We are currently prototyping construction sets for accessible versions of some domains of open source practices (e.g., ticketing). We are building our initial prototypes in JavaScript so that they will be accessible to everyone through web browsers. We are using emerging reactive frameworks, server-side JavaScript, and document databases to get things running quickly. We expect the results will be portable to any modern platform.

Our medium-term goal is to evolve from the prototypes into production-grade construction sets, with a fairly wide range of pre-build functionality.

Our long-term goal is to contribute as much as possible to the broad movement to integrate peer project practices, infrastructure and institutions into the lives of everyone on the planet, so that humanity will have a better shot at working on and through the truly hard problems we face.

REFERENCES

1. Jed Harris and Austin Henderson. 1999. A better mythology for system design. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '99). ACM, New York, NY, USA, 88-95. DOI=10.1145/302979.303003 http://doi.acm.org/10.1145/302979.303003xxxx.
2. Jared Harris and Austin Henderson. 2012. Coherence and responsiveness. interactions 19, 5 (September 2012), 67-71. DOI=10.1145/2334184.2334199 http://doi.acm.org/10.1145/2334184.2334199