Systems Development

David West
Transcendence Corporation
College of Santa Fe
dwest@csf.edu
profwest@fastmail.fm

Abstract

This article defines a new discipline, Systems Development, and calls for a radical change how students master this discipline. The article begins by acknowledging the difficulty of introducing educational reform, describes Systems Development, sets mastery goals, outlines a curriculum, and describes an ideal learning environment. Two efforts to realize the objectives set forth in the article are described: the first is historical and includes lessons learned; the second is just beginning and includes future directions.

Background:

Academic and professional journals seem to be filled with articles expressing a strong sense of dissatisfaction with both the academic discipline of computer science and the state of the computing profession.

A common prescription for this malaise is some kind of educational reform. Proposed reforms might address content (more math is a special favorite) or pedagogy. In either case the premise seems to be better education will produce better practitioners.

Calls for educational reform in general have, probably, been heard from the day that Socrates first sat across the log from Plato and Confucius first outlined forms of proper instruction. Some advocates of educational reform have gained widespread recognition; John Dewey[i] and Mortimer Adler[ii] come to mind. Most (Ted Sizer[iii] and Ken Wilson[iv] for example), are recognized mostly by other reformers. Universities and Colleges have attempted reform, even going so far as to create separate colleges[v] to embody alternative approaches.

Peter J. Denning,[vi] Roger Schank,[vii] Robert L. Glass,[viii] Pete McBreen,[ix] and Richard Gabriel[x] are notable advocates of reform in the area of computer science education. Attempts have been made to implement the ideas of some of these reformers, with varying degrees of success. Neumont University in Salt Lake City (www.neumont.edu) is an example that is successful on the ground but, as yet, lacking the imprimatur of full accreditation. Roger Shank is, apparently, quite successful in developing and selling story based curricula – although it seems directed more to vocational studies. Perhaps the most successful example of educational and curricular reform in engineering (including computer/software engineering) is Olin College.[xi]

There are commonalities among the various reform advocates: experience-based learning; studio-arts derived pedagogy; pragmatic / practitioner based learning; principles-based rather than technology-based curricula, re-thinking calendars, and re-modularization of the curriculum are examples. Many of these ideas have been adopted in the proposal outlined in this paper.

Some common criticisms of reform proposals include:

- Proposed curricula are too vocational or too "faddish."
- Insufficient math and classical science (physics). A corollary: design and human factors courses are "too soft."
- Not enough Art (or English, or History, of my subject area). This is really little more than turf protection.
- Not enough room for prerequisites.
- Students can't learn "that much" or "that fast."
- And the old stand-by, "we tried that and it didn't work."

A different kind of objection is grounded in the redefinition of public education in the United States at the end of World War II. Demobilization (and its direct consequence, the "Baby Boom") mandated rapid expansion of the ability to deliver all levels of education, from preschool to graduate school. Veterans, women suddenly displaced from the labor force, and (a few years later) their children created an unprecedented demand for schools.

To accommodate this demand, a 'factory production' model of education was put in place and it still prevails. Byproducts of this factory model include fixed calendars (quarters / semesters), fixed credit-hour / face-time ratios, and an emphasis on standards and standardized testing. (The fascination with "professional certifications" is a corollary of the standards mindset.)

Factory-based education delivers massive amounts of inexpensive standardized "product". It does so at the cost of minimizing "options," inhibiting change and severely limiting the 'volume' of learning to a kind of lowest common denominator.

Most universities offer core curricula that are 90% common with all other universities offering that curricula. Electives are on the books but are sporadically offered. Universities are notoriously slow to add new courses to their curricula let alone retooling the entire factory!

Despite almost overwhelming evidence that the factory model is fatally flawed, adherents continue to insist it has produced, "the Greatest Educational System in the World." Any reform that challenges the implicit factory model of modern higher education in the U.S. is rejected as elitist, expensive, and impossible.

Despite the mixed success of our predecessors and in spite of the expected criticisms, a clear need for reform remains: particularly in the area of computing.

Limning the Territory

The ACM-AIS-IEEE Computing Curricula 2005 Overview Report[xii] addressed the growth and specialization in the field of computing since the 1960's. What had been a fairly precisely delineated field with three focal points (electrical/computer engineering, computer science, and information systems) has become a complicated overlapping field that includes Electrical Engineering, Computer Engineering, Computer Science, Software Engineering, Information Technology, and Information Systems.

Nascent in the ACM document is the definition of another discipline – Systems Development (SD).

Systems Development overlaps several of the ACM defined fields: incorporating programming from computer science; software, process, and tools from software engineering; and an applications focus from information systems. SD additionally requires extended knowledge of systems

(General, composite,[xiii]" and complex), a deeper understanding of the history of computing and science, significant understanding of business and management, and multiple topics in communication, philosophy, anthropology, and more.

SD, as presented in this paper, emphasizes craft, design, creativity, and innovation instead of the formalism that is the core value of computer science.

Systems Development should be called "Reality Construction"[xiv] because it is based on the fundamental premise that everything a Systems Developer does is necessarily integrated with, and necessarily changes (reconfigures), an existing complex system - the human, organizational, socio-political, and cultural system we call reality. SD delivers altered states of reality.

A full discussion and justification of this neo-discipline is not the focus of this article. The purpose of this discussion is to clarify and focus our intent. Although we started the article with the implicit notion that we wanted to reform "computing," our real aim is to install a new discipline in academia: Systems Development.

Mastery

> "*The ideal architect should be a man of letters, a skillful draftsman, a mathematician, familiar with historical studies, a diligent student of philosophy, acquainted with music; not ignorant of medicine, learned in the responses of jurisconsults, familiar with astronomy, and astronomical calculations*."
>
> Vitruvius 25 B.C.

Vitruvius' architect had to be a polymath – had to have a significant degree of competence in all fields of extant knowledge. Master Systems Developers are not true polymaths but they will be expected to know far more, about a range of subjects that exceeds what was expected of Vitruvius' architect.

In the next section a curriculum for SD will be outlined, but first a brief discussion of mastery

Mastery is defined as knowing something about every aspect of a body of knowledge and a lot about a subset of that body.

Mastery is achieved and evaluated in stages:

1) Rote understanding of concepts and vocabulary – able to pass the kind of objective test typically used to grant basic certification.
2) Ability to apply the knowledge when working with others who have higher degrees of mastery.
3) Ability to apply the knowledge when working alone.
4) Ability to apply the knowledge in a novel context.
5) Ability to mentor others.
6) Ability to produce teaching materials – teach groups rather than one-on-one mentoring.
7) Ability to make an original contribution or extension to a given area of knowledge.

A master would exhibit stage 4 abilities for the body of knowledge and stages 5-7 for a significant subset. The subset would reflect individual interests and ambitions and would roughly correlate to tracks and concentrations in typical curricula.

Note that, excepting stage one, mastery is always based on one's ability to actively do something, not simply and passively know something. This bias is an echo of Giambatisa Vico (1668-1744) who asserted, "We are sure only of that which we build." Doing is also central to experiential learning, project-based learning, and story-based learning, ideas that are central for the proposal that follows.

This definition of mastery erases the distinction of "education for practice and education for research." The goal is education for mastery with a mix of breadth and depth that allows one to become a generalized or specialized practitioner or a specialized researcher. Everyone will have the breadth of knowledge expected of a candidate for a Ph.D. program. Everyone will also have advanced to stage seven (a refereed publication or conference presentation) in a more narrowly defined area. Anyone that has achieved this kind of mastery is prepared for the rigors of doctoral level research, should they so choose.

Curriculum

The metaphor of a mosaic is useful for outlining the proposed curriculum. A mosaic is made up of discrete tiles – a curriculum is composed of discrete knowledge elements[xv]. It is important to note that each "tile" represents more than knowledge – it also includes aspects of practice (application of the knowledge), skill, and experience.

As is true with any pictures, the mosaic exhibits themes – patterned relationships among subsets of the tiles comprising the mosaic. Perhaps there are even representations, identifiable figures or iconic images, themselves made up of a small collection of tiles.

Each tile has an associated color, indicating the discipline or field most associated with the knowledge represented by that tile. Units of philosophy, anthropology, literature, mathematics, history, science, etc. - are distributed across the entire mosaic, showing at-a-glance the relationships of knowledge across disciplines.

Integrating the Liberal Arts, Math, and other disciplines in terms of discrete knowledge elements creates a much more meaningful curriculum. Instead of being expected to know *something* about X, students are expected to know *this* about X.

We might also use a center-to-edge color gradient within each tile to represent the depth of understanding implicit in even a stage one mastery – a kind of degree of difficulty alert: the sharper the gradient the greater the difficulty.

Themes and discrete images within the mosaic graphically capture an analog of the "Great Principles" approach to computing advocated by Peter J. Denning[xvi]. Macro-themes roughly equivalent to his "mechanics, design, and practice. More focused themes roughly equivalent to "computation, coordination, simplicity, innovation, application,' etc.. Highly discrete themes, captured in a small number of related tiles, would be analogous to Deming's coding structures, relational databases, architecture, networks, and similar "concrete" topics.

The mosaic provides an immediate gestalt view of the entire domain to be mastered. Themes[xvii] provide focus without sacrificing relationships to elements outside of that focus. Colored tiles show contributions from all disciplines – in context, with cross-disciplinary relationships obvious. Students not only know that they have to know Z, they know why, and they know how to apply Z as they learn it.

How big is this mosaic – how many tiles?

At New Mexico Highlands (experience report below) we defined a little over 500 "competencies," that did not include liberal arts and other required courses. Experience showed this number was insufficient.

A typical undergraduate program requires 128-136 credit hours and a Masters degree requires an additional 36-48. Using the greater number (184) and assuming 5 discrete knowledge units are learned in each hour suggests a mosaic consisting of 920 tiles; for a traditional education.

Roughly 25% of this number can be eliminated on grounds of redundancy: loops in Java and loops in C++ constitutes one bit of knowledge, not two[xviii]; and, the same unit of knowledge, with elaborations is seen in a succession of courses. This leaves us with a mosaic consisting of 690 tiles.

I would expect the mosaic for Systems Development to, at least, double that number because the SD curriculum expands both technical and liberal arts expectations.

The numbers provided should be regarded as illustrative, not definitive. There are too many assumptions involved to make them truly meaningful. The point the numbers illustrate is, however quantified, a Master of SD will have achieved a stage 4 understanding of almost twice the units of knowledge expected in a traditional education.

Although few would deny the possibility of that many tiles (as educators we are constantly battling the need to add more subject matter to our curriculum); most would raise the purely pragmatic objection – you cannot put that much breadth and depth into a curriculum and expect anyone to master it in the 6 years typically allotted to completing a Bachelors plus Masters degree program.

This objection, however, is not grounded in any understanding of how much or how fast individuals can learn. Instead, it is grounded in the assumptions behind the 'factory model' of education described earlier. The problem is not that it cannot be done; rather, it is, it cannot be done unless we can escape the factory constraints of semester, course, lecture, and book chapter.

Returning for a moment to the topic of mastery. If we allow the tint or shade of each tile to vary as an indication of stage of achievement (an almost transparent tint indicating achievement at stage zero and a deep vivid tint achievement at stage 7) the mosaic also provides a snapshot of an individual's progress towards mastery.

To graduate everyone is expected to be at stage 4 for every tile. They are also expected to be at stages 5 through 7 for some percentage of the tiles as befits their individual interests / specialization.

This snapshot of individual progress is one part of the student assessment package. Other aspects of assessment include ongoing peer and instructor evaluations, evaluations from visiting Mentors (recognized professionals), and a portfolio of completed work. [Program assessment would aggregate these snapshots and determine which tiles were least effectively communicated, how many tiles were completed per unit of time, and how well the topics represented by each tile mapped to career and academic success for our graduates.]

Individual tiles, even collections of tiles that represent themes, are not packaged into courses!

The student (and the faculty) confronts the entire mosaic in its entirety from matriculation to graduation. Of course, at any given moment one's attention must be focused on some subset of the mosaic.

Education is time-boxed and each individual student may elect any subset of tiles to be the focus of their learning during for the duration of that time-box. Usually this choice will reflect a particular project with which the student is involved as a participant – as a developer. Such an election of focal tiles is an "individual learning plan (ILP). This organization of the learning is akin to the agile approach to development – all the tiles in the mosaic represent the "product backlog" and the set of tiles in each ILP represents a "sprint backlog."

(Individual tiles can, of course, appear more than once in an ILP. The first time the goal may be stage 2 mastery, the next time, stage 4.)

Beyond stage one – each degree of mastery must reflect actual performance – work completed in the context of a development project. It is therefore the responsibility of the institution offering this program to assure sufficient projects with sufficient scope to allow everyone to work on all tiles within a reasonable period of time.

So far, I have said several things about the curriculum while avoiding saying anything explicit about its actual content. The reason being – curriculum development is a work in progress. A future article will be written when the content is fully defined and can be articulated and defended.

It is clear that the curriculum will be at least double in terms of content than what is expected in a typical university educational program. The question will immediately arise, "how do we expect to deliver (and students master) this scope without also doubling the time required to earn a degree?"

The answer to that question is grounded in a different kind of educational process and environment – the Bottega.

Bottega (the classroom)

In order to master the entire scope of the proposed curriculum a different kind of learning environment is required. This environment must support continuous learning (not limited a fixed number of hours over the course of a fixed length of time). It also must allow for multiple concurrent threads of individualized learning (ILPs). And, there must be numerous, meaningful, work activities that provide immediate focus and require immediate application of what is being learned.

A bottega[xix] similar to the workshop where Leonardo worked and studied would be ideal. A bottega provides:

- A "storefront" where goods and services are produced and delivered to paying customers.
- A workshop simultaneously engaged in the craft, in building the tools and discovering the techniques that advance and support the craft, and teaching that craft to apprentices.
- A place noisy with multiple projects and activities; walls and benches covered with works in progress and exemplars of the craft.
- A place filled with the tools of the craft (add computers and digital displays to the easels, brushes, hammers, chisels, carving, forges, kilns, model making, etc. tools found in a typical bottega). With room for lounging, and eating facilities as well.
- An intellectual center that is a "must visit" for masters, scientists, and thinkers visiting the area. A center overseen by local masters and journeymen.
- A fountain of innovation and creativity!
- An environment and atmosphere that is very self-consciously multi- and inter-disciplinary; that mixes theory and practice almost without differentiation.
- A place full of music, conversation, laughter, and even loud argument.
- A place to share food and drink, perhaps sleep, and even love[xx].

It would be difficult to imagine an environment more alien to the typical university classroom or laboratory. An exception might be the art studio. A bottega consciously mimics the ideal workspace advocated for agile software development: open, lots of whiteboards and communication artifacts, movement, collaboration, and socialization.

The activities in the bottega drive the learning that takes place in the sense that they provide the context in which discrete units of knowledge are acquired and applied. There is an obvious need to provide a set of projects in a specified interval of time that are sufficiently diverse to assure all competencies can be acquired by each student.

To be maximally effective, students need to spend large blocks of time in the bottega; ideally most their time. A situation requiring students to leave the bottega to attend classes on other parts of the campus creates an obvious conflict. Part of the reason for integrating discipline specific knowledge in the curriculum mosaic described earlier is to allow for the entire curriculum to be delivered in one physical space – the bottega.

Not only is the curriculum itself integrated; that integration is reinforced by the mode of learning and the physical context of learning. The connections across disciplines are physically manifest.

This does not imply a common pedagogy for all elements of the curriculum. It only requires that all teaching and learning take place in the same environment.

The bottega operates as a "one room schoolhouse" and everyone, regardless of degree of mastery, is in the same place and exposed to everything in that environment. If something occurring in one part of the room sounds interesting or relevant, the student is free to roll their chair across the floor and participate.

In this kind of environment a lot of knowledge is absorbed non-consciously, almost by osmosis. When the time comes that an individual needs to consciously acquire and apply a bit of knowledge, that process is accelerated by virtue of the reservoir of non-conscious "background learning" they have already completed.

Personal responsibility, deadlines, and understanding and satisfying customers are all aspects of being a master. Commercial product development provides the educational and experiential opportunity necessary for students to achieve those aspects of mastery.

In the classical bottega, students paid a fee to the botegga in order to become apprentices – a fee that covered the costs of their education. As apprentices they also received compensation for their work – in proportion to their ability to contribute.

In the contemporary bottega, wages should be seen as a form of financial aid – work-study with the advantage that the work is directly relevant to the student's course of study.

Even the most novice individual in the bottega can make a meaningful contribution – a point born out by our experience at New Mexico

Highlands University. Even in the case of extreme deadline pressure, teams can and will integrate the work of the novice to advance the teams agenda – the novice does not have to be a distraction or a hindrance.

The bottega environment also changes the nature of competition and cooperation among students. It is impossible to be a "free rider" in team projects or to disguise the nature of individual contributions to team efforts. Cooperation therefore increases dramatically as individuals learn effective teamwork.

Competition in a traditional classroom tends to be of the type seen in zero-sum games: as there are only so many "A" grades to go around.

Competition in a bottega can be intense, but it is redirected. "I can learn this faster than you," or "my design is more elegant than yours," are challenges that might be heard in a bottega. This is competition, but competition without winners or losers. Individual differences are exposed and celebrated, a kind of competition, but those same differences almost guarantee that I can win in one area, you in another.

It is also important to remember that mastery has been defined, in part, in terms of your ability to help, mentor, and teach your peers. Competition is thereby transformed into a form of cooperation.

A bottega is a very different kind of world – and necessarily requires a different kind of culture than is found in the typical classroom. Fostering this kind of culture is an essential task for the masters that oversee the bottega environment.

Can this idealized environment be created, maintained, and can it work?

Experience Report – New Mexico Highlands University

In August, 2004, 34 students, freshmen through graduate students, entered a unique program in Software Development at New Mexico Highlands University.

The degree program had no courses. (Token SD 100, 200, 300, and 400 courses with variable credits allowed the registrar to record grades and map to tuition.) Students were expected to master 500+ competencies in their "major" plus the usual general and liberal arts course requirements in order to graduate with a Bachelor of Arts degree.

The program was apprenticeship-based: everyone in the program was able to earn money working on real-world software delivered to paying customers with real-world timelines. Wages ranged from $7.50 to $20.00 per hour depending on the level of mastery of the student.

Fifty percent of the students were female (extraordinary for a computing degree) and 60+ percent were minority (consistent with the overall makeup of the University student body).

Two full-time faculty, Pam Rostal and Dave West, designed and operated the program. Internationally known master developers visited the program frequently, working with students (pair programming) and contributing their expertise via hands-on mentoring and conducting other types of learning sessions.

All development work and instruction took place in a "one-room-schoolhouse" with overt instruction occurring on an as-needed basis in very intense, short, sessions. Everyone was involved in these learning sessions whether the topic was “graduate level” or “elementary.”

Our studio consisted of three connected rooms, island work areas, wall-to-wall whiteboards, hard floors and wheeled chairs. Couches, a refrigerator, and a microwave provided some nominal but much used amenities.

Both the software being developed and the learning taking place was organized along the lines of an agile project: backlogs, iterations, spikes, pair programming / pair learning, test-driven[xxi], daily builds, and retrospectives.

Significant time was spent fostering “culture and community.” Ceremonies were used to initiate people into the community and advancements in apprenticeship levels. Retrospectives were always preceded with a pot-luck meal, and various social activities (e.g. ski trip) took place. Colored shirts denoting apprenticeship level (there were four levels) and the ubiquitous presence of the programs logo assisted in creating a sense of identity and belonging.

X bar times 10 alluding to the goal of graduating professionals ten times better than average. Software is our medium, Craft (Mastery) is our goal, People are our focus, Systems is our perspective, and Agility is our process.

Four students left the program the first week – unable to make the time commitment necessary, and two non-traditional students left the program after the first semester because of work conflicts. Three new students joined the program the second semester.

Our freshman retention rate was 100% between first and second semesters at a University that averaged less than 50% freshman retention to the second semester of the first year. Forty additional students – mostly freshmen – had committed to enter the program in the fall of 2005.

Seven students (all but one would have been considered seniors or graduate students in traditional programs) in the inaugural class were offered full-time employment by the end of the first year, 2 were accepted to graduate school, and one was able to concurrently fulfill requirements for a Masters degree in computer science.

Eight students co-authored papers (with each other or with faculty), and delivered presentations at refereed conferences, OOPSLA in Los Angeles and Agile in Denver, that year.

Lessons learned from the first (and only[xxii]) year the program was in place are, unfortunately, mostly anecdotal or idiosyncratic because of the short duration of the program; but they include:

- Students learned far more, in shorter time, than expected.
    - Two students entered the program with zero computer knowledge (did not know how to cut-and-paste) and were able to make valuable contributions to a Java / J2EE project within two months. One was able to mentor others joining that project.

    - Student ILPs increased from an average of 20 competencies the first iteration to 50 by the last iteration.
- The depth of learning and understanding exceeded expectations. Students were able to engage professionals in discussion of their work and their paper presentations at professional conferences. We received numerous comments regarding the outstanding quality of our students from conference attendees.
- All but two students advanced at least one apprenticeship level the first year (acquired and demonstrated roughly a quarter of the defined competencies).
- Students rapidly evolved a process of self-directed learning – using the Web to find answers (including Googling error messages) and information.
- Students selected, installed, and configured their own tools – database, wiki, configuration management, editors, etc.
- The "strong sense of community," "human values (software by humans for humans)," and the "open collaborative learning environment" were the factors women students consistently mentioned as the primary reasons for entering and staying with the program.
- Student's learned teamwork and project management skills concurrently with technical skills.
- Students managed all aspects, (except initial solicitation), of all projects, not faculty.
- Computer Science faculty that had opposed the new program became active participants the second semester.
- Students averaged 30 hours a week in the studio.
- The "noisy" environment was not a distraction – it was an asset.
- The quality of student work far exceeded what would have been done for a typical homework assignment.
- It became apparent that the original set of competencies needed significant revision and extension. It was limited because it was largely the effort of two people, it needed finer granularity, and it needed more content from other disciplines[xxiii].
- Both Pam and Dave averaged 70+ hours a week in the studio. (This time commitment would be unsustainable in the long run.)

There were problems, mostly of the sort that might be expected in any inaugural effort, but overall the program met or exceeded all of our expectations. We ended the year convinced of our philosophy and our approach.

When the program was cancelled, students organized a letter writing campaign, arranged a meeting with New Mexico Governor Richardson's staff, and solicited endorsements from some of the nationally prominent individuals they had met at conferences. This was to no avail, but it reinforced the conviction that the program should continue, and efforts were initiated to replicate the effort at another college.

Moving Forward – The College of Santa Fe

The College of Santa Fe is a small (less than 2,000 students) liberal arts college with very strong programs in the fine and performing arts.

A first approximation of the program set forth in this article was approved and initiated in the fall of 2007. Dave Thomas[xxiv] was instrumental, both in terms of support and leadership, in completing the discussions that led to this beginning.

The degree program in place was far more of a compromise with traditional academia than the program at Highlands. The most radical feature of this degree is the fact that 45% of the major is completed in studios (bottegas).

Two years of reorganization and dealing with financial difficulties have impeded evolutionary plans for the degree program; but a recently established institutional stability makes it possible to proceed. The degree will be offered by School of Business, Innovation, and Technology at the College of Santa Fe.

In the fall of 2008 the existing program will be redefined (and approved by academic committees and the administration) as a five-year (matriculation to graduation) Master of Science program.

In the courses, and especially the software studios, every effort will be made to move students towards the competency / bottega models proposed above.

Concurrently, *Transcendence Corporation* working with the *Santa Fe Complex* (http://sfcomplex.org), will convene a workshop for interested educators and professionals to define and articulate content for the full "curriculum mosaic" discussed above[xxv]. The competency-based curriculum will be mapped to a course-based framework to allow students to transfer credits from this program to other universities (and to satisfy advanced placement guidelines for participating high school students).

The *Santa Fe Complex* will be one of our first botegga environments – a role that they already play in the areas of applied complexity, visualization, agent-based modeling and the intersection of art and science.

We expect 50 full time students to join this program in the late spring / early summer of 2009 including at least 10-15 high school juniors and seniors (fulfilling State mandated Advanced Placement educational requirements) and 10 "transfers" from the degree program at the College of Santa Fe.

At the outset we will probably not have a full apprenticeship component in place. Paid projects at Highlands came from a number of local companies and we intend to renew those connections. We also have ambitious plans to seek projects from corporations nationwide.

We will be able to offer clients essentially the same financial savings as off shore development without a lot of the inconvenience and problems. Involved companies will have opportunities to work with – and eventually recruit – students that have worked on their problems; in the bottega, via internships, and externships[xxvi].

We are also in discussion with corporations and various granting agencies to provide funds for projects we define ourselves (necessary to assure coverage of all tiles in the curriculum mosaic).

We expect additional funding for apprenticeship activities from national and international consulting companies. At present most of the large firms face significant challenges in finding and recruiting qualified applicants. Several have established their own in-house training programs and boot camps to provide university graduates with the additional knowledge and skill required to be successful. Their interest in our program, and willingness to support it, stems from the prospect of recruiting employees that are effectively journeymen, not entry level, and who have a skill set that cannot be duplicated off-shore.

Notes and References

[i] Dewey, John. Democracy and Education. Echo Library. 2007.

[ii] Adler, Mortimer J. The Paideia Proposal. Touchstone. 1998. (Paideia is "the process of educating man into his true form, the real and genuine human nature." Jaeger, Werner. *Paideia: The Ideals of Greek Culture*, vols. I-III, trans. Gilbert Highet, Oxford University Press, 1945.)

[iii] Sizer, Theodore R. Horace's Hope: What Works for the American High School. Mariner Books. 1997.

[iv] Wilson, Kenneth G. and Bennett Daviss. Redesigning Education. Teachers College Press. 1996.

[v] George Mason University and the University of Arizona are two examples.

[vi] Denning, Peter J. "Educating a new engineer". ACM *Communications*. December, 1992.

[vii] http://www.socraticarts.com

[viii] http://www.robertlglass.com

[ix] http://www.mcbreen.ab.ca/

[x] http://www.dreamsongs.com

[xi] http://www.olin.edu

[xii] Computing Curricula 2005 – The Overview Report. Copyright © 2006 by ACM and IEEE.

[xiv] I first encountered this phrase and idea in Christiane Floyd, et. al.'s book, *Software Development as Reality Construction.* Springer-Verlag. 1992.

[xv] The term "competencies" has often been used as a label for such discrete bits of knowledge. Unfortunately, the term has some negative connotations for some and so its use here is minimized.

[xvi] "Great Principles in Computing Curricula," invited talk. Denning, Peter J. SIGCSE'04, March 3-7, 2004.

[xvii] I see this mosaic in terms of abstract rather than representational art – a theme would be a bounded subset of tiles that delineated a focus without eliminating the connection to the whole.

[xviii] Peter Denning talks about "technology driven curriculum," where courses in different programming languages are offered because they represent different "technologies" or tools and not because they deal with fundamentally different kinds of knowledge. This is one kind of redundancy.

[xix] The core of my description of a bottega comes from Fritjof Capra's wonderful book, *The Science of Leonardo*. Doubleday. 2007.

[xx] The Greeks considered Paideia to be carried out by the aristocratic class, who were said to have intellectualized their culture and their ideas; the culture and the youth are then "moulded" to the ideal. Starting in archaic times, love played an important part in this process as adult aristocrats in most cities were encouraged to fall in love with the youths they mentored – Wikipedia.

[xxi] Giving students tasks they did not have the knowledge to complete is analogous to writing a test that cannot be passed because the code has yet to exist.

[xxii] It is essential to talk briefly about why the program was in place for a single year – to make it perfectly clear that termination had nothing to do with the program itself. It is not a pretty story, and it would be hard to adequately describe the perfidy of the individual responsible. Two months after the program was approved and budgeted by the Board of Regents, that same board hired a politician with no educational background or expertise to be President. According to Board minutes - the primary objective for this President was to change the ethnic composition of the faculty. To do so, the President did everything possible to pressure tenured "carpet bagger" faculty to leave, unjustly denied tenure to faculty, illegally hired administrators, and much more. His actions earned sanction of the University by the AAUP, a multitude of lost lawsuits costing the University millions. He was finally fired and is

currently under indictment for allegedly receiving kickbacks from a federal construction project while a state senator. The faculty leading the new program were on the President's hit list. When it became obvious that his ongoing efforts to destroy the program were failing, he ordered it terminated – without consultation, advice or consent of the faculty, Board of Regents, or students.

[xxiii] This content had not been included because students were required to take the liberal arts core, plus a number of supporting courses in other departments in traditional formats.

[xxiv] Carleton College, formerly of Object Technology International and IBM, and currently of Bedarra Research.

[xxv] If any readers are interested in participating, please email me. At this point some of the expenses of attending will be defrayed, but not all. We are working to obtain funding to cover all costs of participating.

[xxvi] Externs complete dedicated work on the same kind of projects as interns, with the added benefit of completing part of the work in the botegga under the tutelage of master educators and professionals.