ISR Institute for Software Research
University of California, Irvine

A History of Software Engineering in ICS at UC Irvine

Richard N. Taylor
University of California, Irvine
taylor@uci.edu

October 2018
ISR Technical Report # UCI-ISR-18-5

Institute for Software Research
ICS2 221
University of California, Irvine
Irvine, CA 92697-3455
isr.uci.edu

isr.uci.edu/publications

A History of Software Engineering in ICS at UC Irvine

UCI-ISR-18-5

Richard N. Taylor

October 2018

Introduction

A history of software engineering in ICS at UCI is a bit of a name game. The term "software engineering" was effectively coined in 1968, the same year that ICS began. But what research topics constitute software engineering has always been somewhat fluid. Nonetheless, the very first PhD student graduated by ICS was William Howden, who throughout his subsequent career as a professor was focused on a core SE topic, software testing.

Software for the Internet has a long and strong history in ICS. From the creation of DNS by Paul Mockapetris through to the HTTP/1.1 protocol, the REST architectural style, and the Apache web server, software engineering concerns and approaches have been at the forefront. Indeed, a hallmark of SE research at UCI has been its interdisciplinary character; many of the key results were the consequence of looking at key software challenges in applied circumstances, such as the Internet, or as the result of working with researchers in other fields, such as human-computer interaction.

Major software engineering topics at Irvine have included software reuse, programming environments, integrated software engineering environments, process, software safety, analysis and testing, security, design methods and tools, open source software, and software architecture. These topics have been pursued by two dozen faculty over the 50 years of ICS and have been the topic of over 100 PhD dissertations.

These diverse pursuits were enabled by strong grant support from DARPA, the National Science Foundation, other government agencies, and many corporate friends and sponsors. A broad community focused on software engineering grew up around UC Irvine and has been supported institutionally by UCI's Institute for Software Research, and its predecessor, the Irvine Research Unit in Software (IRUS). Community events became a hallmark of UCI, ranging from one of the first software process improvement networks (SPIN), to topical workshops, to leadership and hosting of major international software engineering conferences.

The impact of the software engineering work at UCI and the broad community of participants has been felt internationally. HTTP/1.1 is a daily tool in billions of people's lives; UCI personnel significantly advanced software development practices and tools that are stock in trade for all professional developers, such as configuration management. New companies have sprung from UCI SE work, and a legion of developers and researchers have graduated.

The future of SE at UCI looks as bright and promising as the past has been accomplished. PhDs are now awarded specifically in Software Engineering. A new Master of Software Engineering degree is now in place. An undergraduate B.S. in Software Engineering is a popular major. And another generation of faculty has assumed leadership positions.

The following brief account highlights some of the key events, topics, and people who have contributed to software engineering's rich legacy from ICS at UC Irvine.

Faculty

Over twenty faculty have led Software Engineering in ICS at UCI over the past 50 years. Given the interdisciplinary character of SE work at UCI during that time, along with changing faculty interests and emphases, no rigid number of participants can be identified, but the accompanying chart shows the major figures involved during our history.

Primary leadership in the field termed "software engineering" during the formative years of the '70s and early '80s must be credited first to Peter Freeman and then to Thomas Standish. Freeman's work during that time focused especially on design methods and tools, software reuse, and software engineering education. Standish led innovative work in programming language design, program manipulation, rapid prototyping, and program understanding which eventually led to UCI's extensive contributions to integrated software development environments. Concurrently UCI was beginning its influential work in "Internet" software, with key contributions beginning with Dave Farber and Einar Stefferud, a resident researcher.

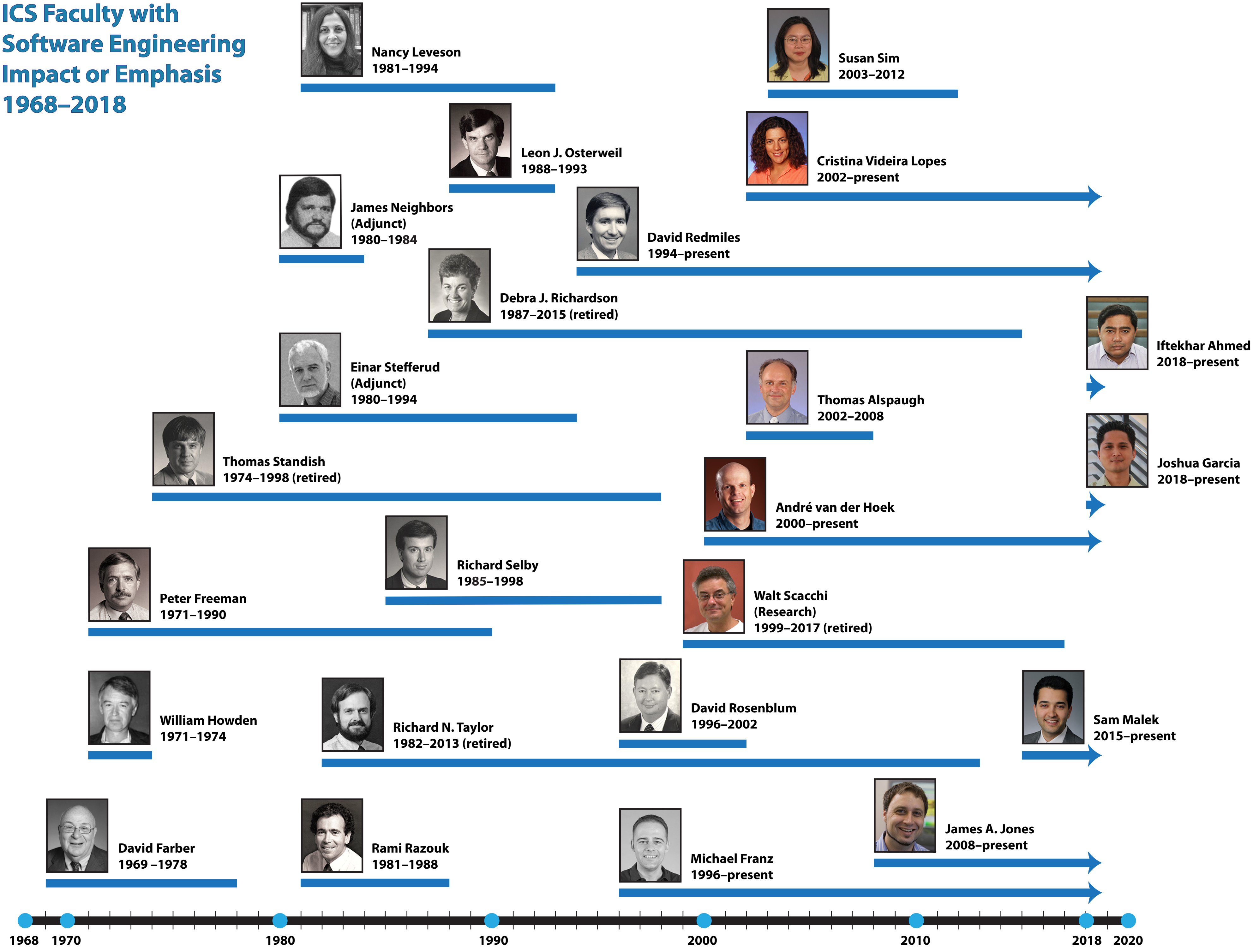
The 1980s were the explosive growth years of SE at UCI, just as SE was emerging on the global stage as a major computer science research area. Jim Neighbors joined the faculty as an Adjunct professor in 1980 following completion of his PhD dissertation under Peter Freeman, to pursue studies of component-based software engineering and domain analysis. Nancy Leveson joined the faculty in 1981 and would go on to establish the subfield of software safety. Rami Razouk joined the same year, bridging work between distributed systems and SE. A strong analysis theme ran through both of their research agendas. Richard Taylor joined in 1982, drawn by Standish's work in programming environments and the strong group of PhD students who were increasingly calling UCI their home. His work focused on integrated software engineering environments, hypermedia, and, ultimately, software architecture. With rising grant support, this core group was soon strengthened by the additions of Richard Selby in 1986 and Debra Richardson in 1987. Selby brought a focus on empirical research and software metrics, while Richardson focused on analysis and testing. This critical mass in SE then became the magnet that attracted Leon Osterweil to join the group in 1988, just as his pioneering work in software process was lighting an international fire.

The 1990s were years of continued growth, faculty turnover, as well as years of dramatic technical impact. The technical activities and impacts are described separately below, but it deserves highlighting here that it was the result of a large, diverse, but also remarkably cohesive and collegial group of faculty that made these results possible. Highly collaborative work during this time was aided by the additions of David Redmiles and David Rosenblum to the faculty in 1994 and 1996, respectively. Redmiles was key in enhancing the ties with researchers in computer-supported cooperative work and human-computer interaction. For instance, Mark Ackerman, who joined ICS in 1992, contributed to the SE group, as did Gloria Mark later in the 2000s. Rosenblum brought strength not only in analysis and testing, but also in the intersection of SE and distributed systems. 1996 also saw the addition of Michael Franz to UCI, whose focus was on compilation and, later, security in language platforms. The decade closed with the hiring of Walt Scacchi in 1999 as a full-time researcher. Scacchi brought renewed strength to work in process, further developed ties with researchers in management, and would go on to do groundbreaking work in computer games and provide leadership in the open source community.

The 2000s were a time of increasing technical diversity within SE. André van der Hoek joined in 2000, having done capstone work in configuration management that led to new work in the emerging field of coordination in development. 2002 saw the addition of Thomas Alspaugh and Crista Lopes to the faculty, Alspaugh having done work in requirements engineering, and Lopes with seminal work in aspect-oriented programming and then later in program design, mining software repositories, and in game technology. Susan Sim joined in 2003, bringing additional focus on requirements engineering. James Jones was the final hire during this first decade of the new millennium, coming in 2008 with an outstanding record of results in software testing. The current decade has seen three additional hires in software engineering, with Sam Malek in 2015, and Joshua Garcia and Iftekhar Ahmed in 2018. Malek and Garcia have particular strengths in software security and software architectures and Ahmed in analysis and testing.

While the faculty have led the charge and set the agenda, the remarkable cohort of PhD graduates from the SE group at UCI is equally responsible for the outstanding technical successes achieved. By 2018 there have been over 100 PhD graduates in software engineering and closely aligned fields. These graduates have gone on to highly successful careers in academia and industry around the world, ranging from full professorships in leading universities to founders of new companies to senior managers and technologists in influential companies.

ICS Faculty with Software Engineering Impact or Emphasis 1968–2018



Nancy Leveson
1981–1994

James Neighbors
(Adjunct)
1980–1984

Einar Stefferud
(Adjunct)
1980–1994

Thomas Standish
1974–1998 (retired)

Peter Freeman
1971–1990

William Howden
1971–1974

David Farber
1969 –1978

Leon J. Osterweil
1988–1993

Debra J. Richardson
1987–2015 (retired)

Richard Selby
1985–1998

Richard N. Taylor
1982–2013 (retired)

Rami Razouk
1981–1988

David Redmiles
1994–present

David Rosenblum
1996–2002

Michael Franz
1996–present

Susan Sim
2003–2012

Cristina Videira Lopes
2002–present

Thomas Alspaugh
2002–2008

André van der Hoek
2000–present

Walt Scacchi
(Research)
1999–2017 (retired)

James A. Jones
2008–present

Sam Malek
2015–present

Iftekhar Ahmed
2018–present

Joshua Garcia
2018–present

Technical Impacts, Achievements, and Activities

Every dissertation has an impact. So does every published paper, every grant, every collaboration, every conference, every workshop. Yet in hindsight some dissertations, some papers, some grants, and some activities seem to especially sparkle, and are noteworthy for their subsequent impact both within the international software engineering community and, more broadly, in the technical world as a whole.

Impacts often have to be contextualized. With contemporary knowledge it is sometimes difficult to appreciate how difficult, or how forward-thinking, an accomplishment 20, 30, or 50 years ago was. Many technologies we take for granted today required profound, novel, counter-intuitive thinking when the core ideas were first mooted. The following paragraphs both provide a little context and then highlight some of the remarkable achievements of SE at UCI-ICS over the past 50 years.

The 1960s and 1970s.

The formation of software engineering as a discipline in the ’60s and ’70s took place in a contentious context. On the one hand industry was forging ahead in using software in increasingly audacious tasks, such as landing a man on the moon. But within academia “software” was struggling. Formalists with a mathematical inclination demanded rigor and proofs, while electrical engineers counted clock cycles and linked software thinking with low-level architectural concerns.

The great achievement of the era, one significantly led by UCI, was recognizing that software engineering was an intellectual discipline on its own, not inexorably tied to hardware concerns, or existing as a minor branch of mathematics. Rather, SE researchers identified the intertwining of design, management, human factors, programming, and quality engineering as equal partners in the new discipline.

UCI contributed significantly to this achievement, largely due to the efforts of Peter Freeman, whose work individually and in concert with Tony Wasserman (UCSF/UCB) brought design methods to the forefront of discussion and, in particular, led community-wide efforts to develop software engineering education. The first graduate class in software engineering at UCI began with a proposal from Freeman in 1974—just six years after the famous NATO Garmisch conference. That year also saw the publication of one of the earliest books in SE, Freeman’s *Software Systems Principles.* In 1976 Freeman and Wasserman organized an influential seminar at UCI on SE Education, a meeting attended by many leading SE researchers of the era. Its influence was greatly aided by proceedings published by Springer Verlag (which, at the time, was probably the most influential book/proceedings publisher in computer science).

Perhaps the most significant impact of this line of work came as a result of collaboration between Freeman, Wasserman, and Dick Fairley (Texas A&M), with the development of a model curriculum for SE, published in the second ICSE, which became the foundation for most SE curricula nationwide for several years following. Freeman’s ACM and IEEE Fellow Awards were based on this influential work.

The 1970s was also a decade of rapid advances in programming languages and programming support systems. While SE, broadly, was finding its full scope, work in language systems progressed with remarkable technical achievements. One particular strand of this work was in interactive program development and manipulation, drawing from work in extensible programming languages and (what would later come to be called) refactoring. Xerox PARC was a key leader in this area, with its innovative Interlisp environment. UCI brought together key ideas from this strand of work with another effort underway: the DoD's sponsorship of the creation of a new, mandated programming language, informally "DoD-1," which later came to be known officially as Ada.[1]

The insight was this: the powerful programming capabilities illustrated by Interlisp could be enhanced and brought to bear on the strongly typed, statically scoped language the DoD was promulgating. The DoD, in fact, was driving the community to produce an environment to support development of Ada programs. Thomas Standish played a key role in advancing this idea with UCI's hosting of a "DoD-1 Environments Workshop" in 1978. This community-wide effort led to creation of the "Sandman" and "Stoneman" requirements documents for such an environment. Standish not only helped the requirements to move along, but showed how they could, in part, be met through his innovative Arcturus environment, a project that spanned across into the 1980s.

A final key technical impact from work in the 1970s was development of approaches for aiding software reuse. This topic was pursued both by Standish and Freeman, with Standish keying off of the environment topic, and Freeman with his PhD student Jim Neighbors pioneering domain analysis and reuse at higher levels of abstraction than programming.

At the close of the '70s software engineering was very well established at UCI. With a broad technical focus, publications, software, and curricular leadership, SE took its place as one of the original five focus groups of the ICS Department.[2]

The 1980s

The computing industry in the '80s continued its focus on large-scale, custom software applications. The DoD, DARPA, and the associated industry were dominant players, for until the latter years of the decade this was still the era of the Cold War. PC's and desktop computing were in their infancy, and e-commerce was just beginning to emerge. Computing in academia was beginning to ossify a bit, centered around maturing fields such as operating systems, theory, compilers, and databases. A key factor in advancing research, however, was the cost of computing equipment. Prior to the '80s the key enabler for leading departments was access to cutting edge—and expensive—machines. But the '80s saw the beginning of the leveling of the playing field.

As the decade began Freeman and his students continued their work in software reuse, with particular emphasis on domain-based approaches. A cohort of students graduated, taking these concepts into other universities and, ultimately, into the private sector, including two of the first start-ups to emerge from UCI's SE work: Semantic Designs, founded by two of Freeman's PhD students, Ira Baxter and Chris Pidgeon, and Bayfront Technologies, founded by Jim Neighbors.

While not "software engineering" research accomplishments per se, two major achievements from the early '80s set the stage for some of the highlighted SE achievements of the '90s discussed below. Specifically, Paul Mockapetris, who graduated in 1982, developed the Domain Name System (DNS), which is key to the everyday functioning of the Internet. Working at the same time, Marshall Rose, who received his PhD in 1984 with Rami Razouk serving as his advisor, researcher Einar Stefferud, and staff member John Romine made key contributions to the mh mail system, the MIME standard, and the POP and SMTP mail protocols. Email on Unix systems was powered by mh for decades and MIME, POP, and SMTP are still in use. The tradition established by these two landmark UCI efforts of contributing to application-level Internet technology prepared the way for the software engineering research of the '90s described below that led to the modern WWW.

[1] Irvine Compiler Corporation was a spin-out from Standish's work with Ada, incorporated in 1981.

[2] Information and Computer Science (ICS) existed on the UC Irvine campus as an independent department from its founding in 1968 to it becoming a full-fledged School of Information and Computer Sciences in 2002.

While software testing and analysis was a broadly productive field in the ’80s, UCI’s Nancy Leveson brought a novel and highly influential aspect to light: software safety. Safety analysis applied fault trees, Petri Nets, and allied techniques to bear not on showing that software did what it was supposed to do, but rather to show that at least it did *not* do anything drastically wrong. Her work with safety analysis broadened to include emphases on requirements engineering, management practices, and human interface design. This broad, integrative approach to engineering became typical of SE work at UCI, reaching across artificial disciplinary boundaries to achieve significant new results. The safety work at UCI reached widespread acclaim with the analysis of the Therac-25 accidents, work performed by Leveson and her PhD student, Clark Turner. Also notable from this era and branch of work was the debunking of N-version programming, which was joint work with John Knight from the University of Virginia.

The environments work begun by Standish in the ’70s blossomed in the ’80s. With Richard Taylor joining the faculty and the Arcturus project, it soon began to take on broadened scope. The DARPA-sponsored drive towards Ada Programming Support Environments (APSEs) led Taylor to co-found the Arcadia project in 1984. This project, a collaborative effort that started with SE researchers at UCI, the University of Colorado Boulder, and the University of Massachusetts, Amherst, would go on to be one of the most influential software development environment projects. Arcadia continued as a project well into the 2000s and expanded to include researchers at additional universities and multiple industrial partners. The key focus that Taylor and the other researchers brought was a belief that all SE activities were intertwined and could be supported by an integrated development environment. The facilities that developers have today in environments such as XCode, Eclipse, and Visual Studio all had their origins in the many environment projects of the ’80s. With Arcadia, programming was no longer seen as the chief or only focus of development, but rather was one part of an effort that encompasses management, requirements, analysis, testing, deployment, configuration management, and more. Arcadia also served as a bridge from the somewhat insular world of Interlisp and Smalltalk to the commercial development world, and especially to the aerospace community, which was being compelled to move to Ada.

A critical enabler for Arcadia was large-scale (for the time) funding from DARPA and the National Science Foundation. Happily, through the ’80s and ’90s DARPA was flexible with its investigators, allowing them to pursue promising leads, even if those were not strictly within the purview of the sponsoring contract or grant (more about this below, in the ’90s).

As significant as Arcadia’s technical goals for tool integration were, Arcadia perhaps more significantly emerged as a model for multi-institutional collaborative work. Many universities had programming environment projects, but the scale of Arcadia was substantially greater, and placed it as a peer of other national-scale collaborative projects such as the European Software Factory. Arguably the essential secret of its success was brutally honest technical review meetings followed by long, pleasant dinners with excellent wine. Equally important was the commitment of the investigators to “the idea of Arcadia,” as opposed to formal boundaries based upon grant support. This ambiguous border to Arcadia enabled many researchers to participate in technical exchanges, even when specific contract support was not in place. Such collaborations continued for many years.

One particular stream of research within Arcadia was led by Leon Osterweil and his focus on software processes. His seminal paper on process programming was presented in 1987, the year before he joined UCI’s SE faculty. The technical buzz created by that paper led to intensive focus on process in the later years of the ’80s, and well into the ’90s. The paper’s impact was recognized at ICSE 1997 with the retrospective Most Influential Paper (MIP) Award.

A truly transformative and enabling step for SE at UCI in the 1980s was the winning of an NSF Coordinated Experimental Research grant. Led by Richard Taylor, the grant was awarded in 1986 for the establishment of "A Laboratory for Software Research." Coinciding with the release of Sun Microsystems' new "lower priced" workstations, this grant provided the equipment that enabled UCI to pursue research on a footing equal to the heretofore privileged few universities. With bitmapped displays, X windows, Unix, 24Mb memory, and a three-button mouse, it was a heady time!

Lastly, another key enabler of SE research at UCI in the '80s were UC MICRO grants. These grants were exclusively for joint university-industry projects, with industry contributions matched by the University of California, and had overhead waived. With strong aerospace and device industries in California, these grants fostered increasing ties with the university. Such collaborative work meshed perfectly with the integrative focus of work by the SE researchers and with the model of collaborative efforts exhibited by the Arcadia project.

The 1990s

The key achievements of the 1980s led to the explosion of results and impact that came from the 1990s. Across the nation software engineering grew in stature and in number of participating researchers. Grant money was abundant, DARPA was exciting, collaboration was the norm. Computing devices became ever faster and more interesting; PCs and Macs came to be ever-present, ever more capable, and affordable. E-commerce started its explosive growth, the Internet enabled "everything," and who can forget the IPO boom at the close of the decade?

From the perspective of a SE researcher, however, perhaps the key transition was from an almost exclusive focus on development of custom applications by large contractors to development of speculative, commercial applications, many of which were designed to run on the now-common and inexpensive desktop machines. For that world, "requirements first" no longer made complete sense. And then there was the Web.

The Arcadia Project continued to be the primary context for SE work at UCI throughout the 1990s. The process research that Osterweil brought to UCI had wide-ranging ramifications. Notable was a Symposium on Software Process in December 1991 followed by the establishment in 1992 of the Southern California Software Process Improvement Network (SPIN). SPIN was a joint university-industry initiative to explore the newly emerging ideas about process and how they could be applied and assessed in the commercial world. This SPIN was the second in the nation, and involved dozens of companies, where representatives came to UCI for meetings on a monthly basis. The meetings continued at UCI for the decade. Joint work with industry in examining software processes in practice also led to the 1995 publication of the widely read *Microsoft Secrets*, jointly authored by Michael Cusumano of MIT and UCI's Richard Selby.

The excitement around "process" helped lead to the 1990 formation of the Irvine Research Unit in Software (IRUS), which went on to sponsor other regular industry-university meetings and partnerships, such as the Software Engineering Tools and Technology forums and the Bay Area Roundtables (BART). The BART meetings continued regularly throughout the decade, bringing exposure of UCI advances to the then-burgeoning Bay Area software community. Among other technical symposia, four California Software Symposia were held, jointly sponsored by USC's Center for Software Engineering. At the end of the decade IRUS "graduated" to become the Institute for Software Research (ISR), about which more will be said below.

From the outset Arcadia had a major focus on software analysis and testing.[3] One key and early result was Debra Richardson's development of specification-based test oracles. Published in 1992, this work was recognized in 2012 with an ACM SIGSOFT Impact Award, reflecting the importance of the contribution. (This award followed an earlier key test-of-time award for Richardson, based

[3] Development of many language processing and analysis tools was strongly aided by the project's professional programmers; there were as many as six on staff at one point.

on work that took place just before she joined the UCI faculty. Her 1985 paper "A Comparison of Data Flow Path Selection Criteria," written with her then-colleagues at the University of Massachusetts, Amherst, won the 1996 ICSE MIP award.)

One focus that Arcadia did not have from the outset nonetheless yielded what is unquestionably the project's major impact: leadership and development of the HTTP/1.1 protocol, the Apache web server, the Apache Software Foundation, and WebDAV. In the late '80s hypermedia was an active research area and Taylor led Arcadia's Chimera project in the development of novel ideas concerning hyperlinks as first-class relationships. Contemporaneously, the WWW was beginning at CERN in Switzerland. Ideas started coming together when Roy Fielding became Taylor's student and started actively working within the Web project. Fielding soon assumed a leadership role that would see him be first author of the HTTP/1.1 protocol, co-founder and developer of the Apache web server[4] (httpd) project, and founding Chairman of the Apache Software Foundation. DARPA/IPTO, as the primary funder of Arcadia, proudly took credit for these world-changing results at its 40th anniversary meeting. The irony is that Taylor, as UCI's Arcadia PI and Fielding's advisor, had to keep the Web work under wraps, protecting it since hypermedia research was not part of the Arcadia research agreement. It was "simply engineering" that others were working on…why should DARPA be involved? Thankfully DARPA directed with loose reins, and the world is better for it.

Not only was hypermedia not part of Arcadia's original charter, neither was leadership of the open source software movement. Yet the Apache Software Foundation (ASF) emerged from UCI's work, in collaboration with many others, and Fielding became the founding Chairman. Later, another of Taylor's students, Justin Erenkrantz, served as ASF President. While organizationally the ASF is interesting, perhaps more so is the concept of open source, and especially the practices and techniques that enable its effective application. In 2000 Fielding co-authored an ICSE paper on open source development in the Apache project, which was awarded the 2010 ICSE MIP award. Walt Scacchi, having joined the UCI SE community, picked up both the process and open source threads and began a long period of leadership in analyzing and leading the open source community. This started with his organizing and hosting two F/OSS workshops at UCI and the subsequent establishment of the OSS conference.

Irvine's open source contributions were not limited to Web-related technologies. At the same time as the web work was progressing, the nascent field of software architecture was growing. One early result came from David Redmiles' student Jason Robbins, who produced the widely used Argo/UML development tool which had context-sensitive support for making decisions. In 2003, ArgoUML won the Software Development Magazine's annual Readers' Choice Award in the "Design and Analysis Tools" category, and in 2010, at the Automated Software Engineering Conference, the 1996 paper on ArgoUML won the Most Influential Paper award.

Software architecture, of course, became a major focus of research at UCI. With major support from DARPA's Domain Specific Software Architecture (DSSA) program, key results came in architectural styles, architecture description languages, and architecture-based development environments. Technically, one distinctive focus of the UCI work was on event-based systems. While more programming-focused event-based architectural styles, such as C2, emerged, the focus on Internet-scale event notification had the greatest impact. David Rosenblum played a key role in this, working with Alex Wolf at the University of Colorado Boulder on the Siena event observation and notification system, work which resulted in a SIGSOFT Impact Award in 2008 for their paper

[4] Apache received the 1999 ACM Software System Award; Fielding accepted the award at the ACM awards banquet that year on behalf of all the cited developers.

published in 1997.[5] Rosenblum went on to help launch PreCache, a startup company developing technology in the area of publish/subscribe networking, where he served as CTO and Principal Architect from 2001 to 2003. Additional Internet-scale notification efforts were also pursued, and IRUS hosted several national workshops on the topic, under the banners of WISEN (Workshop on Internet Scale Event Notification, 1998) and TWIST (The Workshop on Internet Scale Technologies, 1999 and 2000). A direct outcome from this line of work was yet-another startup, KnowNow Corporation, co-founded by Taylor's student Rohit Khare in the dot-com boom era of the year 2000.

A key technical benefit of event-based architectures is their adaptability. UCI's SE group produced a seminal work in the field, led by Taylor's student Peyman Oreizy. Run-time dynamic adaptation based on reasoning over an explicit architecture was the focus of their paper which appeared at ICSE in 1998, and which won the ICSE 2008 MIP Award.

The other architectural style for which UCI is justly famous is REST, REpresentational State Transfer, the style underlying the design of HTTP/1.1. Indeed, the underlying technical focus of all the work that led to the WWW results was *architectural*—designing for change, designing for scale, designing for multiple non-functional properties. The subject of Fielding's PhD dissertation, REST emerged from the interdisciplinary milieu of UCI,[6] as it draws inspiration from the coordination/collaboration work of Mark Ackerman, the hypermedia community, software architecture, and the network community. Fielding and Taylor's paper on REST, which appeared in 2000, was recognized in 2017 with an ACM SIGSOFT Impact Award. REST-based architectures are now prevalent in Internet-based applications, with many books and how-to guides available to guide practitioners.

The final major contribution from the software architecture work of the 1990s was Neno Medvidovic's paper on architecture description languages (ADLs). Development of ADLs was rampant in the 1990s and Medvidovic's work brought some much-needed rationality to the field. The paper has now been cited over 2700 times and was the most cited paper from the year 2000 in all of software engineering.

The excitement and ferment of the 1990s led to the dot-com boom at the end of the decade. As noted above, UCI spun out several companies. In addition to those already mentioned, Amadeus was spun out in 1994, led by Richard Selby, having a focus on metric-based support for continuous process improvement. Endeavors was started in 1998 by several of Taylor's students, having its initial focus on automated support for workflow.

The 2000s

While the 1990s began with the cheer accompanying the end of the Cold War, the 2000s quickly were dominated by the effects of the 2001 terrorist attacks. The consequences for software engineering and computer science research generally were not immediately recognized but were in fact quite significant. The consequence was that DARPA and the military funding agencies moved from supporting long-range, inventive, and exploratory research to short-range, highly-focused, metric-assessed projects. The National Science Foundation assumed new prominence as a result, but it was unaccustomed to such free-ranging research programs, and was further bound by numerous, well-intentioned but ultimately straitjacketing constraints.

Nonetheless major new research results were in the offing, some directly derivative of projects originating in the '90s, but others new, and sometimes responsive to the new national focus on the

[5] While at UCI, Rosenblum also won the ICSE 2002 MIP Award for work he did while at Bell Labs in 1992, on programming with assertions.

[6] Notably, that interdisciplinary milieu was facilitated by the co-location of all the SE students, faculty, and technical staff in the ICS2 building complex.

terrorist threat. Still other topics emerged from the broad, integrative view of software engineering held by the research group from the beginning. In particular, a strong socio-technical view of SE was taken. This view led to strong interactions with researchers with a primary focus on workflow, awareness, and collaboration topics. UCI's historic focus on distributed and decentralized applications, such as the Web, led members of the group to a somewhat surprising but prescient niche topic: computer games. Whether for education, entertainment, or "strictly for work," games emerged by the end of the decade as a frequent topic.

While the Arcadia project ended in the early years of the decade, the software architecture focus continued with many new results. This stream of work culminated in 2009 with the publication of the first comprehensive textbook on software architecture, *Software Architecture: Foundations, Theory, and Practice*, by Taylor, Medvidovic, and alumnus Eric Dashofy. For the first time a comprehensive and consistent set of definitions were produced for the field, supporting its unification.

The long-running theme of software analysis and testing research was invigorated by the addition of Jim Jones to the faculty. Jones is perhaps best known for the creation of the influential Tarantula technique that spawned a new field of "spectra-based" fault localization. For this work, published in 2002 as "Visualization of Test Information to Assist Fault Localization," he and his co-authors at Georgia Tech were awarded the prestigious ACM SIGSOFT Impact Award in 2015.

Between 2004 and 2006 André van der Hoek and his student Emily Oh Navarro received two National Science Foundation grants to support research and development of games for teaching simulated software engineering processes to undergraduates. The role-playing game, SimSE, that emerged from this project went on to win the 2009 Premier Award for Excellence in Engineering Education Courseware. Another educational game that emerged from this project was described in a 2005 paper by Alex Baker, Oh Navarro, and van der Hoek, "An Experimental Card Game for Teaching Software Engineering Processes," which went on to be recognized as one of the ten most cited papers of *Journal of Systems and Software* in 2015. (After graduation Baker went on to co-found VisiTrend in 2011. VisiTrend started as an R&D services company catering to the DoD with a focus on the integration of machine learning and visualization with applications in several domains including cybersecurity.)

This work on software engineering and games was an early aspect of the flood of games-related work to emerge from the SE group. Some of the work, led by Walt Scacchi, emerged with the support of a $3M grant from the NSF, entitled "Decentralized Virtual Activities and Technologies: A socio-technical approach." This project investigated different kinds of software development processes and social interaction practice that spanned multiple organizations. Contemporary open source software development projects were a key exemplar. Attention was also directed to study and prototyping of computer game-based virtual worlds as media for modeling, visualizing, and simulating such processes and practices.

Crista Lopes brought a new focus to SE at UCI in the 2000s, mining and searching Internet-scale software repositories. She came to the UCI faculty as co-inventor of aspect-oriented programming[7] and as one of the original designers of the AspectJ programming language. In her mining and searching work she developed new techniques for structure-based search of code repositories. This technique, now widely cited, enables code search that goes beyond conventional keyword-based techniques.

The focus on building a community of software engineering researchers that began with the Arcadia project in the early '80s reached a major new level in the 2000s with the establishment of the

[7] The original paper on aspect-oriented programming has been cited over *9000* times according to Google Scholar.

Institute for Software Research, as initially noted above. ISR has been the only institute in the University of California system devoted to software engineering research. Beginning with seven faculty members in 1999, by 2018 there were thirty-two faculty and faculty affiliates from fourteen institutions.

ISR was a key enabler for faculty, students, and projects throughout the 2000s and on into the 2010s, when it was transitioned from a UC-wide entity to an ICS School center in 2018. ISR's contributions were not limited in effect to just UCI and ISR-affiliated researchers, however. ISR provided critical support to the major international software engineering conferences, including injecting a level of professionalism previously unseen, but which is now expected. IRUS, ISR's predecessor, began this community service with contributions to ICSE '94, ICSE '97, and then especially with the organization of the ACM Work Activities and Coordination Conference of 1999. As ISR, major support began with the Foundations of Software Engineering conference in 2004 and the Automated Software Engineering conference of 2005. A long roster of technical symposia and workshops also belong on ISR's roster of contributions.

The 2010s

The 2010s are upon us and, for research results, the test of time for impact needs, well, time. The context for software engineering work continues to change, though. As software reaches into every area of our day-to-day lives and mediates our activities and even our relationships with friends, family, and co-workers, new ways of thinking about software are essential. The decade has seen mobile devices become ubiquitous, with billions of smartphones and tablets used worldwide. The mobile computing revolution brought about a new set of software engineering challenges, such as limited energy resources on mobile devices calling for new software engineering methods to treat energy efficiency as a first-class software quality attribute. Other currents include the focus on machine learning and big data, the continuing growth of the game world, and cloud-based development.

Computer security has emerged as one of the greatest challenges. There is recognition that innovative point solutions in, e.g. cryptography, are insufficient for securing applications and data. Rather, a holistic approach is required, demanding that security experts work jointly with application designers, database experts, human-interaction experts, and more. Happily, such an interdisciplinary view is completely consistent with UCI's historic approach to SE. One exemplar of this approach is the work of Sam Malek and Joshua Garcia, which addresses mobile computing and vetting security and privacy properties of an application of unknown provenance. Supported by a series of grants from the Department of Defense and the intelligence agencies, Malek and Garcia have produced a tool, RevealDroid, which has been adopted for use by the Department of Homeland Security for vetting mobile apps.

UCI has continued to lead efforts in advancing software architecture research in the 2010s. Malek and Garcia, together with researchers spanning several institutions, including ICS alumnus Neno Medvidovic, supported by an NSF grant, are developing a first-of-its-kind integration framework for assembling architecture-related techniques and tools with the goal of enabling empirical research in the context of software maintenance.

UCI's focus on software design research has resulted in several notable achievements. The decade began with UCI hosting an NSF workshop focused on studying professional software design. Subsequently André van der Hoek's student, Nicolas Mangano, won the 2014 ACM SIGSOFT Outstanding Doctoral Dissertation Award for his work on the Calico early-phase software design tool. Mangano then went on to start up a new company, Sketch Together, based on this research, in 2013. Further, in 2016 van der Hoek published a book with Marian Petre of The Open University, titled *Software Design Decoded: 66 Ways Experts Think.*

In a similar vein, Crista Lopes published *Exercises in Programming Style*, in 2014. This book was declared "Best Programming Book of the Decade" by the Software Development Times. Using a simple computational task (term frequency) to illustrate thirty-three different programming styles, the book helps readers understand the various ways of writing programs and designing systems.

Lopes also became a key contributor to the open source software-based OpenSimulator project, a virtual world development platform comparable to and interoperable with Second Life. She extended OpenSim to support hypergrids, extensible virtual worlds that can be composed into large-scale, networked virtual worlds. Her software development efforts were responsible for the production and release of the Diva distribution of OpenSim. She also served as co-organizer for the OpenSimulator Community Conference in 2013, 2014, and 2015. Her contributions were recognized with the Antonio Pizzigati Prize in 2016.

With regard to software in society, Redmiles' research group investigated factors contributing to trust development, which included awareness of collaborators' activity as well as affective factors built on informal interactions. Extensive field work by researcher Ban Al-Ani, including many interviews, provided the basis for understanding these factors. Game-theoretic modeling techniques, explored by then student Yi (Oliver) Wang, demonstrated how trust was greatly affected by informal communications, termed "cheap talk."

From 2007 to 2017, the Naval Postgraduate School's Acquisition Research Program supported the research of Walt Scacchi and Thomas Alspaugh in which they investigated the acquisition and implementation of open source software in high security military command and control systems. They examined the effects on open software system architectures when mixing components subject to different Intellectual Property (IP) licenses and determined that many license schemes mediate the efficacy of cybersecurity mechanisms when applied to independently developed components versus when applied to those integrated into an overall system architecture.

The year 2011 saw ISR provide its greatest level of support to the international software engineering community yet, through its leadership and management of ICSE 2011, held in Honolulu, HI, as well as its support of SPLASH / OOPSLA 2011 held in Portland, OR. ISR was similarly responsible for the highly successful International Conference on Global Software Engineering (ICGSE 2016) held at UCI.

Institutionally there have been several major SE-related accomplishments. After decades of degrees in software engineering being awarded under the general header of "Information and Computer Sciences," in 2011 the University of California granted the UCI campus the right to award MS and PhD degrees under the title of "Software Engineering." Similarly, the campus endorsed the formation of a BS in Software Engineering the same year, so SE at UCI is now rightfully acknowledged as a full-fledged discipline. Then, in 2018, UCI began its self-funded "Master of Software Engineering" program, targeted at practitioners. And it is only 50 years since the NATO Garmisch conference.

This is a story "to be continued."

Acknowledgments

I gratefully acknowledge the invaluable contributions from many people, particularly Peter Freeman, Jim Neighbors, Sam Malek, André van der Hoek, Walt Scacchi, Lee Osterweil, and David Redmiles. Debra Brodbeck provided invaluable assistance with many of the details and especially with the timeline. Like any history, though, errors and omissions are solely the responsibility of the author.