Jump-starting the Computer Revolution That Hasn't Started Yet

John Repko, Pratap Kotala
Computer Science Department
North Dakota State University
1340 Administration Ave, Fargo, ND 58105
john.repko@ndus.edu, pratap.kotala@ndsu.edu

Abstract

Every smartphone has more computing power than the systems that took us to the moon, but where is the “computer revolution?” Consider the bright minds that fill our first-grade classrooms. We then teach them vocabulary -- to give them the colors to paint their stories. Then we teach them grammar to shape their stories, and we don't stop there. We also teach them the arts and literature so that they can build on lessons from the past.

So where is the “computer revolution?" We have the technology, but we lack the Design: a common understanding of the vocabulary and grammar that engages computers to serve our world. We also need Art - the rhymes, metaphors, and similes that bring ideas into focus. We can jump-start the computer revolution by teaching technology framed by Design and Art. We present first results from this approach with suggestions for advances to the curriculum.

I. Introduction

It wasn't that long ago that "procedural programming" was the primary province of computing. The Pascal and C programming languages ruled the universities, and Modula-2 was presented as the logical evolution of that world. Pascal and C were excellent programming languages, but their use has remained in a narrow technical community outside of everyday, non-technical, general purpose applications.

The problem was that too many interesting problem domains didn't match Pascal or C syntax and trying to mold challenging problems into strict typing and rigorous syntax was just too hard. "Object-oriented" programming arose as a more logical match for many programming domains, and the web programming that followed swung to approaches that more closely match web formalisms and protocols. Java and C++ succeeded Pascal and C, but more general purpose uses have remained out of reach because most of our world still does not map into the simple types and objects that comprise the object-oriented world.

The problem appears in starkest form if we consider the advances in computer hardware that have emerged over the past 30 years, and then contrast them with the lack of software progress over the same that comprise the object-oriented world.

The problem appears in starkest form if we consider the advances in computer hardware that have emerged over the past 30 years, and then contrast them with the lack of software progress over the same period. If we update Rob Pike's [1] polemic on system software, it's clear that present-day computing devices have hundreds of times the processing power, memory, and networking bandwidth that their predecessors had at the dawn of the Internet age. Computing language advances are far more modest: Java and C++ have supplanted Pascal and C[2], and Python is jostling Java today. We have also seen niche languages arise, such as Rust, Kotlin and Go, or more exotic complements in Haskell and Clojure[3]. We haven't advanced new languages because we haven't sought to solve new problems -- even "big data" advances merely apply the same familiar linguistic tools to vastly more scaled hardware.

If we want a "computer revolution" we need to bring everyday, non-technical applications into the fold, and we need to advance beyond the notion of programmers as the "high priests of the low cult" [4] in a narrow, technical community. We have the technology. To advance our thinking and our teaching, in this analysis we seek to mesh that Technology with Design (to make new things work) and Art (to reach new audiences).

II. Why This is important

It's easy to imagine that "We have benefitted from generations of progress in technology. Isn't that enough?" We offer that is not enough, and with the fait accompli that is history, it is easy to underestimate just how much impact that Design and Art have in advancing our technologies and in shaping our modern world.

Consider a case from history -- the story of Martin Luther and the 95 Theses that led to the Protestant Reformation. The 95 Theses is a story that many will grasp (perhaps vaguely) from earlier classes in History: Luther had specific complaints with the prevailing Catholic Church, he wrote those complaints as the "95 Theses" and tacked them up on a church door in Wittenburg, Germany, and the Reformation just happened. The actual history was far richer:

October 1517 -- Martin Luther (reportedly) tacks his 95 Theses up on the wall in Wittenberg. Written in Latin, it's likely that few beyond the Catholic clergy could even read them.[5]

January 1518 -- Martin Luther's friends conduct a re-Design, and rewrite his 95 Theses in the vernacular -- German[6]. Instantly the local audience could read the Art of his arguments, and he became a local sensation.

February 1518 -- Like The Beatles at the Cavern Club[7], word starts to spread, but Luther needs "airtime." Enter Technology, in the form of the Gutenberg printing press.[8] Better than "Top 40!' Within 2 weeks the 95 Theses are known all over Germany, and The Reformation was at hand ... almost.

March 1518 -- 1563 (Councils of Trent) -- As part of a counter-reformation, the Church forbade anyone in their jurisdiction from printing vernacular German on printing presses.[9]

Summer 1518 and thereafter -- As printing vernacular could lead to prosecution, vernacular-printers skirted off the edges of Church domain, taking their printing presses with them. Since 95 Theses was their bestseller, they had no choice but to print it, now in new, expanded locations. By midsummer, the 95 Theses were known throughout Europe, and The Reformation was underway!

Protestant services were very different than their stricter Catholic counterparts. Music was a vital feature of the services, and Martin Luther himself is still known to this day for his music (the hymn "A Mighty Fortress Is Our God" was written by Luther). A pipe organ became a central focus of Protestant services. But who had the skills to fashion pipe organs? A skilled guild of high-tech manufacturing arose to support the music of the Reformation, and these precision-manufacturing skills drove the industrial revolution that still shapes our world today.[10]

Among the wonders here was the unification of Technology, Design, and Art. Here Art came first - if Martin Luther had not fashioned compelling arguments - the art of language - the Reformation would not have initiated as it did. Design came second -- re-fashioning the 95 Theses in the vernacular and as a popular sermon, made the Art serviceable to a larger (ultimately global) audience. Technology, in the form of the printing press, broadcast the 95 Theses and sermon to a pan-European audience, and the rest, as they say, was history.

Any of Technology, Art, or Design can be a wonder to behold, but all three must combine to change the world we live in. In programming classes we teach technology - computers and programming languages can be a great start. But they are only a start -- we also need to teach Design and Art if we are to jump-start the "computer revolution" we are seeking. Technology, Design, and Art have been the combined focus of many class sections that we will discuss in this paper.

III. Initial class - Impact of Technology, Art & Design

Design -- Design is the thing that changes your life. Steve Jobs was revered for designs, which he defined as: "Design is not just what it looks like and feels like. Design is how it works."[11] To emphasize this point we may note that the Apple iPhone is only a modestly capable telephone. At the same time the iPhone was a breakthrough Internet computer, and its functions as an alarm clock and flashlight all provide value as well. “Design” is all of these things.

Technology -- Technology is what allows “Design” to happen NOW. Fast and low-power ARM processors, CMOS low power memory, Corning Gorilla Glass™ — without these things you’d have an Apple Newton, not an iPhone or any of the better Android phones.[12] Steve Jobs scores here as well — he took an extremely aggressive interpretation of when “Now” was.

Art -- Art is what cuts through the din and clutter of life. Muhammad Ali was a great boxer, but his work as an artist is why we remember him. Ten thousand hours (according to Malcolm Gladwell[13]) is what it takes to make your skill an art. The Wright brothers made their historic flight on December 19, 1903, with world-changing technology (a lightweight engine to power the flight) and design (U.S. patent 821,393[14] claimed the invention of a system of aerodynamic control that manipulated the flying machine's surfaces), but no Art. It was years later before the media noted their achievements:[15]

In years to come Dayton newspapers would proudly celebrate the hometown Wright brothers as national heroes, but the local reporters somehow missed one of the most important stories in history as it was happening a few miles from their doorstep. James M. Cox, publisher at that time of the Dayton Daily News (later governor of Ohio and Democratic presidential nominee in 1920), expressed the attitude of newspapermen—and the public—in those days when he admitted years later, "Frankly, none of us believed it."

History offers many examples of the union of technology, design, and art, and the Bemidji State University course TADT 2100 - "Impact of Technology, Art and Design" was the first to present an integrated, episodic course in the history of technology, art and design. The Impact of Technology course was a general liberal education elective course offered to undergraduates in the School of Technology, Art and Design at BSU. The three disciplines were presented in a unified format, so that the systemic nature of technology, art and design could be illustrated. It was the intention that students who complete a semester of unified Technology, Art and Design learn that each of Technology, Art and Design must be properly fashioned for the success of any venture. The class was conducted to advance three specific goal areas: Goal Area 1: Communication, Goal Area 2: Critical Thinking, and Goal Area 8: Global Perspective.

Impact of Technology was taught in both in-classroom and online sections. Classes employed the modified case method, where each class section presented a historical business case for which students were asked to identify the contributions of Technology, Art and Design, determine what the expected outcome was, and suggest how the protagonists in each case might have better applied the technology, art, and design that was available to them. The class culminated with student projects in which they would declare in innovation (an advance in Technology) and describe their plans to integrate Design and Art to bring that innovation -- now a full product -- to market.

The class was popular and extended the Technology, Art & Design department ideas out into the general university population. In each of the terms that it has been offered it has been oversubscribed in both online and in-classroom sections.

IV. Teaching Technology - Computer Packages

Technology, Art and Design have now been integrated for almost a full year at North Dakota State University in the CSci-114 "Computer Packages" course. The course is a general requirement, and the students are predominantly first-year college students who take the course to a) fulfill general requirements as well as b) learn application-specific techniques for the Microsoft Office Applications: MS Word (word processing), Excel (spreadsheet), PowerPoint (presentations), Access (database) and Outlook (email).

Here the course has been presented in a series of 16 (twice per week) computer labs where the students work through a pre-planned series of exercises defined and given by Cengage under the name MindTap™. We have expanded the lab curriculum by adding a short, 15-20 minute introduction to Design and Art in each section, where the specific additions are chosen to supplement the particular application under review. The class schedule and additions then took on the following form.

Introduction to Computers

- Intro to Technology, Art and Design
- Sample case presentation - Martin Luther & the Reformation
- The Evolution of Computers - the "Zen of Presentation"[16]
- Introduction to Research - Google search tips

Microsoft Word Training

- Introduction to Microsoft Word
- Kurt Vonnegut on Storytelling[17]
- David Foster Wallace on Writing in the Age of Facebook[18]
- MS Word Readability Tools - Great Gatsby Ch. 4[19]
- Writing templates - resumé, press release, elevator pitch

- Win and loss notices - examples from Expo 2023 Minnesota
- Principles to guide document design

Microsoft PowerPoint Training

- Introduction to Storytelling - Patagonia and "Creation Myth"
- Storytelling - Garr Reynolds TED Talk - Kyoto
- Storytelling - Christopher Booker The Seven Basic Plots[20]
- Storyboarding / Storylines - Jaws and Dead Poet's Society
- Storytelling - the attention curve, horizonal / vertical logic

Microsoft Excel Training

- Introduction to Microsoft Excel - visual languages
- Analysis - "Margin Call", vLookup and hLookup[21]
- Excel for Analysis - Intro to Options and Derivatives
- Great spreadsheet examples (Scrabble, word cloud, etc.)
- Spreadsheets and forecasting, seasonality

Microsoft Access Training

- Introduction to Databases and MS Access
- Relational databases
- Beyond relational - "big data" and NoSQL

Microsoft Outlook Training

- Introduction to Email and MS Outlook
- Email templates and etiquette

The class curriculum was expanded on the idea that it makes no sense to teach a student how to make a 3/4 point, double-lined, gray #6 border around a picture in a document without first explaining why they might want such a border in their document. The technology of Microsoft Office applications can then support the art of what they want to say, and the design of how they want to say it.

V. Student Responses

When offered at Bemidji State University as TADT-2100 - The Impact of Technology, Art & Design, the course was a favorite, oversubscribed, and drew a wide variety of students from across all fields of study. Students from varying degree programs completed the course, with Technology, Art and Design students generally performing well, but with some of the strongest performers hailing from as disparate fields as Nursing and Marine Biology. Additionally, final student presentations were quite inventive in their application of Art and Design to their chosen Technologies.

The Technology, Art, and Design expansion was then applied to the teaching of the CSci-114 "Microcomputer Packages" course at North Dakota State University. Here the student population was similarly diverse, and (as at Bemidji State University) the course was predominantly composed of first-year students. Here, though, the course was a general requirement, and nearly all of the students in the class enrolled in it to meet that requirement. Otherwise, the students were of widely varying backgrounds and brought a diverse set of expectations to the course. The students were surveyed to characterize their experience and expectations, with the results presented in Figure 1.

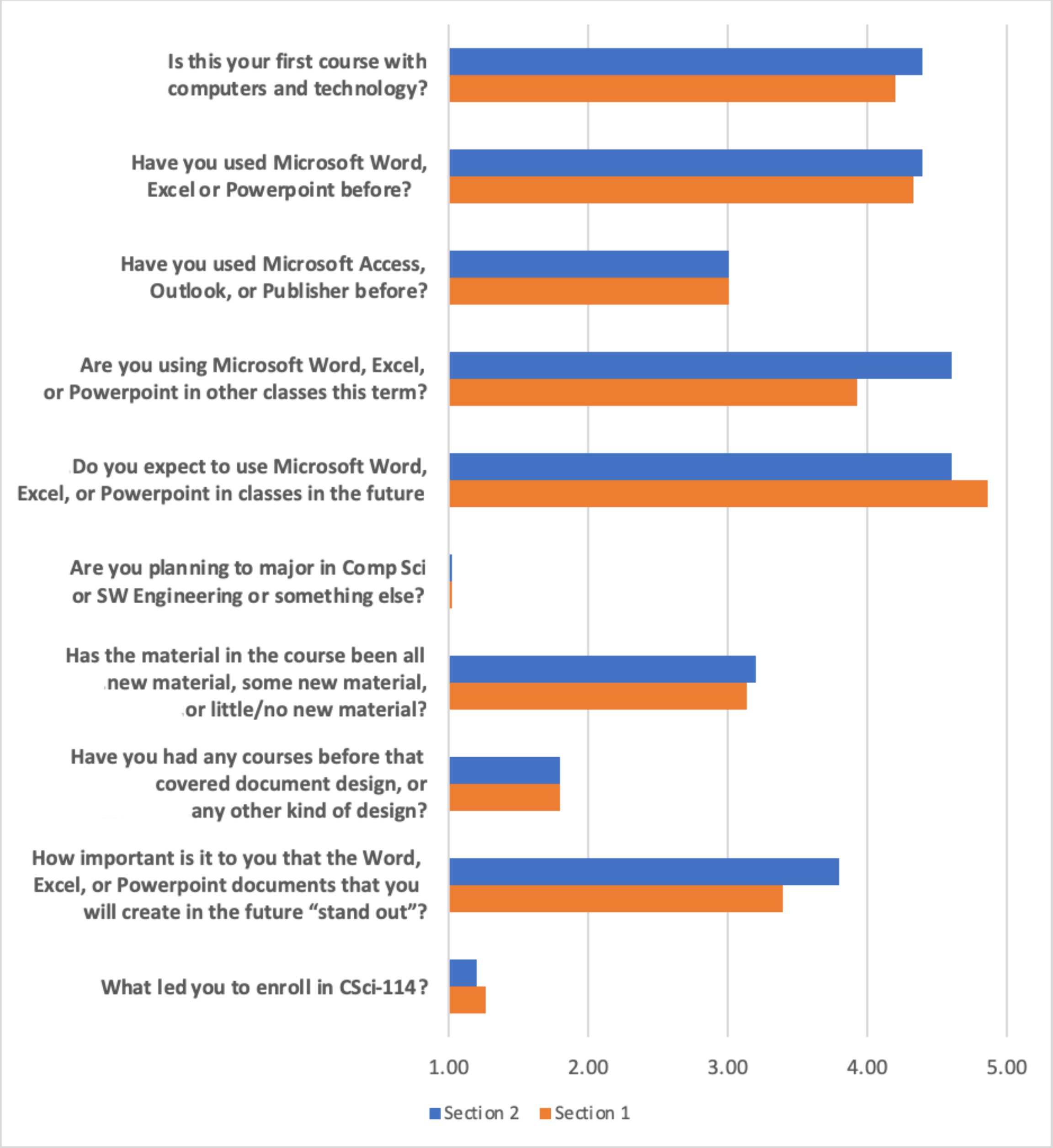

Figure 1. Survey Results from Students in Computer Packages

In Figure 1 above presents the results of a Likert-scale set of questions that assayed the students' general familiarity with technology, and whether Design and Art were part of their backgrounds. In the survey questions above, a score near 1.0 indicates a negative response, 3.0 a noncommital response, and scores near 5.0 show a generally affirmative answer. The survey showed the following salient results:

Technology Familiarity

For most students this was their first course with computers and technology, but most were already familiar with the most popular applications in Microsoft Office, and half were familiar with Office's less commonly-used applications.

Expectations

Almost all students expected to use Word, Excel, and Powerpoint in the future, even though none of the students surveyed were planning to major in Computer Science or Software Engineering.

Innovation

The course presented a modest amount of new material, and few students had covered document design any previous coursework.

Art - Standing Out from the Crowd

Students surveyed felt that it was modestly important (3.4 and 3.8 out of a 5-point Likert scale, where 5 = "Very Important") to have their future work stand out -- this may be explained by point 5 below.

5) Why Did You Enroll in CSci-114?

Practically all of the students (23 of 25) surveyed enrolled in the course because it was a requirement, the two exceptional students enrolled because they had friends enrolled in the class.

Students coming into the Microcomputer Packages course generally have some familiarity with personal computers and desktop applications, and they modestly want their documents to show well, but they had no higher expectations for computing and chose to enroll in the class principally because it is required. As presented, the course will give them better skills with desktop applications and a broader appreciation for the contexts in which to apply those skills. As the students indicated no higher aspirations for Computer Science or Software Engineering, we cannot expect that it will make them otherwise better at computing.

VI. Curriculum and Proposed Changes

In the Fall 2018 term, CSci-114 Microcomputer Packages course featured exercises in MS Word, PowerPoint, Excel, Access, and Publisher; for Spring 2019 Publisher (desktop publishing) has been replaced with Outlook (electronic mail). The principal focus of these applications is content creation, with accompanying background in data management (Access) and communications (Outlook) in Spring 2019. The following sections describe the class structure and the software applications that were taught as part of the course.

Introduction

The essential point in the class introduction on the Reformation is that history doesn't progress by happenstance -- but changes in human affairs take place when there is a story to tell (Art), the story is fashioned (Design) to reach an audience, with advances (Technology) that make the story universally available at that time.

MS Word - Art and Design

Word was introduced by shifting focus from the "fonts and formats" that Cengage favors to the more general concept of storytelling. Here recordings by writers Kurt Vonnegut and David Foster Wallace provided their tips for capturing an audience with a story. The subsequent sections on MS Word reinforced the suggestions: that the point of the writing is to deliver a compelling narrative, and the formatting tools in Word are used to make it easy for the audience to capture and comprehend the critical points of that story.

MS PowerPoint - Art and Design

PowerPoint is infamous for "bullets to boredom,"[22][23] but here too the focus is shifted to meta-level, where the story determines the PowerPoint features that are most beneficial to telling that story. The class introduced the seven basic story types (foundational myth, rags-to-riches, the quest, overcoming the monster, etc.) to provide a structure for the content of the PowerPoint exercises. The class also introduced "horizontal logic" (the notion that reading just the headlines of a PowerPoint presentation should be sufficient to capture the essence of a story) and "vertical logic" (the notion that the content on a slide should flow up to support the headline) to structure the narratives for each PowerPoint presentation.

MS Excel - Art and Design

Excel is a spreadsheet application that provides an entry to the world of numbers and finance. The class introduction featured the trailer from the film "Margin Call" (a 2011 film drawn anecdotally from the 2008 collapse of Lehman Bros.), where modest spreadsheets were prominent as the harbinger of the economic ordeal that followed. The Art and Design of the class also offered a high-level introduction to options pricing and more general forecasting, reflecting the history that the arrival of Lotus 1-2-3 and the IBM PC enabled the takeover boom of the '80s[24]. The class also offered an introduction to "big data," with examples that showed that the million-row limit introduced with Office 2007 (and all subsequent MS Office releases) provides access into the analysis of data sets of previously-unthinkable size.

MS Access - Art and Design

The focus on data and analytics continued in the class sections for MS Access and relational databases. Databases are a difficult topic to provide supplemental Art and Design for - the domain-specific language most commonly used for data access (ANSI SQL) is dry and generally beyond the interest of first-year non-technical students. To establish the Art and Design in even so purely-technical a topic, students were presented the story of Dr. Brian Druker and the "big data" analytics that lead to the discovery of a successful treatment for a dread disease. With the product advances noted in the Excel section, most applications at student-level can be tackled successfully with a spreadsheet rather than a database. The "big data" presentations were created to show students the evolving Design and Art arising from the analytics of this social-media age.

MS Outlook - Art and Design

We closed the class with a look at the epistolary form, the art of exchanged messages and modern business etiquette of email and texting. Design is a valuable addition here, as email and texting have become increasingly common and accepted as a form of business and professional communication. Here Inc. Magazine's "15 Email Etiquette Rules Every Professional Should Follow" and Email Etiquette for Today's Business Student[25] provide essential framing to avoid the types of communications faux-pas that are common humor threads on the Internet.

These additions in context were well-received by the students, but they did not extend into the specific Cengage exercises, labs, projects and exams for each application that comprises the course. Students complete editing tasks for particular document types, but the examples in computer training did not explore the document templates for each software application (e.g., resumés, press releases, contracts or nondisclosure agreements in Word, budgets in Excel, pitches in PowerPoint, etc.). The formatting changes that the students were given to complete were devoid of context -- they made specific changes to documents without being provided an explanation of why those changes should be sought or were desirable.

In future classes, we will seek to advance the integration of Technology, Art and Design with the following changes to the curriculum:

1) Employ Specific Document Types

Most common documents from Microsoft Office applications have specific, known syntaxes and formats: resumé format, contract format, NDA format, Excel budget, Powerpoint "pitch," etc. Changes to these documents are not random, and each document template follows a grammar that is common to all of that document type. Students need to be taught not only how to complete specific transformations, but why those transformations should be sought in the first place.

2) Define the Specific Document Type

Most common documents have a known and understood format and syntax, often referred to as "boilerplate." Changes to this meta-level design/document grammar are not taught in the course but could be a crucial part of understanding what kinds of edits and changes are appropriate for a document of a given type.

3) Course-wide Document Narrative

There is little connection between the exercises for Word, the PowerPoint exercises that follow the Word section, nor the Excel exercises that follow the PowerPoint section. As we teach the technology behind each application, we can use it to advance the meta-level design of a more general narrative and use that narrative to select the most appropriate document types and tools for the delivery of that narrative.

4) Next Steps - How Should the Story Be Told?

As presently delivered, the format of the Microcomputer Packages course is declarative - Cengage presents a series of steps to be completed, and the students are graded by how precisely they complete those steps. This declarative format is a beneficial approach for establishing a limited context and teaching specific techniques such as highlighting and formatting within that context. The next logical step in the evolution of Microcomputer Packages thinking is the introduction of choice: what story is the student given to tell, and what are the most appropriate ways to tell that story?

One of the benefits of the design of the Impact of Technology course was that students were expected, in their final presentation, to combine Technology, Art, and Design to present a case study on the combination of technology, art, and design. Microsoft Office applications are only an introduction to the rich world of technology tools for storytellers, and leading students to ask the question "How should we present this?" opens them up to consideration of tools beyond their experience, such as Tableau Software for analytics or Prezi for presentations. The introduction of the features and functions of Microsoft Office is a valuable offering for students, but with periodic version updates, the changes in Office functionality, and the bonanza of richer tools available may make the introduction to that greater world of analytics and storytelling a more valuable contribution to students.

5). Looking Beyond - What is the Story Here?

Presenting technology declaratively in a course such as Microcomputer Packages is appropriate as a start and is beneficial for taking a student population of widely varying skills at the beginning of the course and getting everyone to basic proficiency with the tools. As noted in Next Steps, it is worthwhile to introduce the follow-up question: How should the story be told? There is also a second follow-up question, a question that represents the foremost gap between the computer skills we teach and the computer revolution we seek: What is the story here?

Today we generally teach technology as an extension of the scientific method, in which a hypothesis is proposed, data is gathered, and analysis is performed to establish the truth or falsity of the hypothesis. We teach Software Engineering in much the same fashion: In our ideal software project we create requirements, create software to meet the requirements, and confirm that the software meets those requirements correctly. Performing these tasks correctly and successfully is the essence of Software Engineering.

But what if this basic, intuitive approach is wrong? Consider the case of Dr. Brian Druker noted above. To find a cure for the dread disease (in this case chronic myelogenous leukemia), Drs Lydon and Druker examined a vast pool of data, found outliers in that data, built and tested hypotheses and refined a particular outlier to produce the drug Gleevec.[26]. Drs Lydon and Druker performed exploratory programming with vast data, but without a key objective or specific requirements. Their software followed where their intuition led. We will jump-start the computer revolution when we can all explore data, conduct inquiries and write programs where our requirements follow our intuitions, rather than leading them. In Hackers and Painters[27], Paul Graham writes

> “For example, I was taught in college that one ought to figure out a program completely on paper before even going near a computer. I found that I did not program this way. I found that I liked to program sitting in front of a computer, not a piece of paper. Worse still, instead of patiently writing out a complete program and assuring myself it was correct, I tended to just spew out code that was hopelessly broken, and gradually beat it into shape. Debugging, I was taught, was a kind of final pass where you caught typos and oversights. The way I worked, it seemed like programming consisted of debugging. For a long time I felt bad about this, just as I once felt bad that I didn’t hold my pencil the way they taught me to in elementary school. If I had only looked over at the other makers, the painters or the architects, I would have realized that there was a name for what I was doing: sketching. As far as I can tell, the way they taught me to program in college was all wrong. You should figure out programs as you’re writing them, just as writers and painters and architects do."

Software Engineering does not generally deal with the idea of a "sketchpad" very well. The term "hacking" is often a pejorative, when hacking (in the sketching sense) is merely a means of establishing greater intentionality between the domain we are working in and the syntactic limitations of the programming languages that we employ to work in those domains. We might get closer to that revolution by shrinking the gap between the language and the problem domain.

The goal in establishing the What is the story here? question takes us to a higher objective in teaching about microcomputer applications, where the point is the narrative we have to tell, and the techniques we teach are tools to identify the best story to tell, and methods to tell that story better.

VII. Conclusions and Lessons Learned

CSci-114 Microcomputer Packages is a rich course that provides an excursion of remarkable depth into the features and capabilities of Microsoft Office applications. As Microsoft Office is now in its 16th major version and has been on the market for almost 30 years, it's not surprising that each of the applications has a wide range of features and functions. With its Cengage computer-aided testing, the course offers broad coverage of application features, and the Fall 2018 and Spring 2019 classes are the first to introduce Design and Art as a broader context for the use of the Microsoft Office applications.

This context is critical if we seek to enable a more productive use of these tools as part of a more general computer revolution. Students need to grasp the functionality of the remarkable computational devices that are available to them, but they also need a broader context to seek devices and applications to do more for them. The applications taught in Microcomputer Packages exist principally as aids in communication and students will benefit from mastery of these tools as they are currently used. A computer revolution will follow when students and non-technical users learn to look beyond these applications to evolve their worlds and to tell their stories.   History was made when Martin Luther's Art was extended out to Design (writing in the vernacular) and Technology (broadcast via the printing press) to tell his story. More recently, the "summer blockbuster" was born when Steven Spielberg turned to Design and Art (redoing his narrative through storyboards, dialog, and music) to compensate for failures in technology (that his mechanical shark did not work) when making the movie "Jaws." The delays that accrued as he to turned instead to music (John Williams' score) and storytelling (e.g. the tale of the USS Indianapolis) took him past his planned Holiday 1974 release date out to Memorial Day, 1975. The "summer blockbuster" was born, and as Spielberg later noted, "The shark not working was a godsend."[28]

To jumpstart the computer revolution, we must re-engage technology with design and art in our daily affairs. As we look beyond what our current tools can do to build our stories and communicate with each other we will trigger and drive a new generation of computer and communication capabilities -- and the "computer revolution" will be underway.

References

[1] Pike, R. (2000, February). Systems software research is irrelevant. In Talk. CS Colloquium, Columbia University. URL: http://herpolhode. com/rob/utah2000. pdf (visited on 12/09/2010).

[2] Delorey, D. P., Knutson, C. D., & Giraud-Carrier, C. (2007, June). Programming language trends in open source development: An evaluation using data from all production phase sourceforge projects. In Second International Workshop on Public Data about Software Development (WoPDaSD'07).

[3] Meyerovich, L. A., & Rabkin, A. S. (2013, October). Empirical analysis of programming language adoption. In ACM SIGPLAN Notices (Vol. 48, No. 10, pp. 1-18). ACM.

[4] Booch, G. (2013). Deus ex Machina. IEEE Software, 30(6), 14-16.

[5] Eisenstein, E. L. (1980). The emergence of print culture in the West.

[6] Holborn, L. W. (1942). Printing and the Growth of a Protestant Movement in Germany from 1517 to 1524. Church History, 11(2), 123-137.

[7] Cooper, L. E., & Cooper, B. L. (1993). The pendulum of cultural imperialism: Popular music interchanges between the United States and Britain, 1943–1967. The Journal of Popular Culture, 27(3), 61-78.

[8] Dixon, C. S. (2017). Luther's Ninety-Five Theses and the Origins of the Reformation Narrative. The English Historical Review, 132(556), 533-569.

[9] McNally, R. E. (1966). The Council of Trent and Vernacular Bibles. Theological Studies, 27(2), 204-227.

[10] Cantoni, D., Dittmar, J., & Yuchtman, N. (2018). Religious competition and reallocation: the political economy of secularization in the Protestant Reformation. The Quarterly Journal of Economics, 133(4), 2037-2096.

[11] Serrat, O. (2017). Design thinking. In Knowledge Solutions (pp. 129-134). Springer, Singapore.

[12] Korkeamaki, T. P., & Takalo, T. (2012). Valuation of innovation: The case of iPhone.
[13] Gladwell, M. (2017). Outliers: the story of success. Audio-Tech Business Book Summaries.
[14] Wright, O & W. (1906). 821393. Retrieved from http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&p=1&u=%2Fnetahtml%2FPTO%2Fsearch-bool.html&r=1&f=G&l=50&co1=OR&d=PALL&s1=0821393.PN.&OS=PN/0821393&RS=PN/0821393.
[15] Tobin, J. (2003). To conquer the air: The Wright brothers and the great race for flight. Simon and Schuster.
[16] Reynolds, G. (2011). Presentation Zen: Simple ideas on presentation design and delivery. New Riders.
[17] Baldwin, M. (1986). The Way to Write Short Stories. Elm Tree.
[18] Wallace, David F. (1996), Radio Interview with Christopher Lyndon. The Connection.
[19] Fitzgerald, F. Scott. MS, Fitzgerald, F. Scott (Francis Scott), 1896-1940-Manuscripts. Princeton University. The Great Gatsby. Princeton University Digital Library, 2010.
[20] Booker, C. (2004). The seven basic plots: Why we tell stories. A&C Black.
[21] Werner, A. (2014). 'Margin Call': using film to explore behavioural aspects of the financial crisis. Journal of Business Ethics, 122(4), 643-654.
[22] Norvig, Peter. "PowerPoint: Shot with its own bullets." The Lancet 362.9381 (2003): 343-344.
[23] Tufte, Edward. "PowerPoint is evil." Wired Magazine, Septembre (2003).
[24] Deal, T. E., & Kennedy, A. A. (2008). The new corporate cultures: Revitalizing the workplace after downsizing, mergers, and reengineering. Basic Books.
[25] Agnew, D. S., & Hill, K. (2009). Email etiquette recommendation for today's business student. Academy of Organizational Culture, Communications and Conflict, 1.
[26] Lydon, N. B., & Druker, B. J. (2004). Lessons learned from the development of imatinib. Leukemia research, 28, 29-38.
[27] Graham, P. (2004). Hackers & painters: big ideas from the computer age. " O'Reilly Media, Inc.".
[28] Niven, D. (2014). It's not about the shark: How to solve unsolvable problems. Macmillan.

John Repko is a Graduate Teaching Assistant of Computer Science at North Dakota State University. John's contributions include innovations in visual languages that led to Steve Jobs' NeXT machine and the software that created the Internet, and advances in hosted software delivery at Oracle that led to the Cloud we all use today. John was formerly a VP at Oracle, CTO at Apropos Technology, and the founder several startup companies, including Pikasoft LLC and SMFSP LLC. He is currently a PhD candidate in Software Engineering at North Dakota State University. John has a bachelor's degree in Chemical Engineering from MIT, a master's degree in Computer Science from Rensselaer Polytechnic Institute, and an MBA from Stanford University Graduate School of Business. John was also elected to Pi Mu Epsilon, the national honorary mathematics fraternity.

Dr. Pratap Kotala is a Senior Lecturer in Computer Science at NDSU, Fargo. In over 20 years of teaching, he has taught variety Computer Science courses at undergraduate and graduate level in traditional as well as online/web-based setting. Pratap also has two years of industry experience as an IT consultant for Blue Cross Blue Shield of Florida. His research interests include Software Engineering and Cybersecurity. Dr. Kotala obtained Ph.D. in Computer Science from NDSU.

He is also an instructor and course coordinator for CSci 114 course at NDSU for over 15 years. Dr. Kotala is currently part of the North Dakota K-20 Cybersciences and Cybersecurity Education task force.