Is Agile Old School?

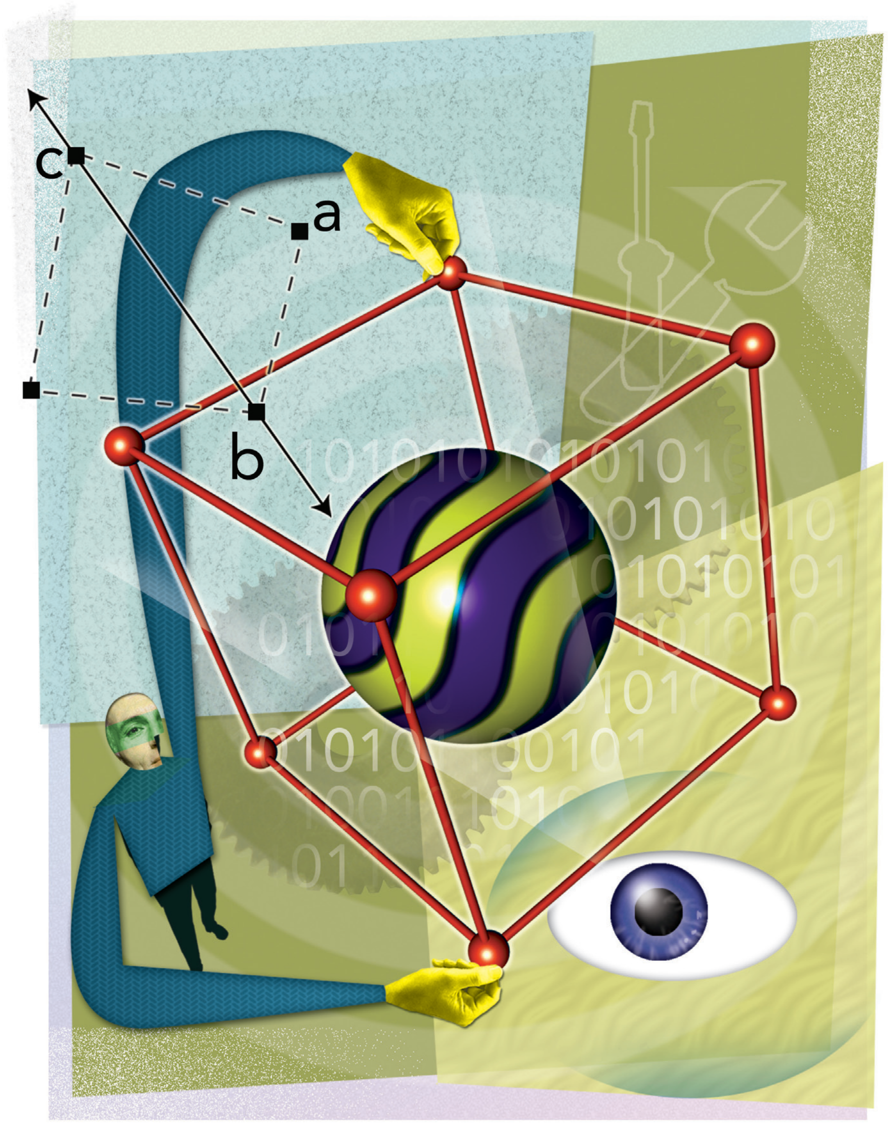
8:31am, Nov. 26, 1968. I drove my 1967 Volkswagen Bug into the upper Parker Street parking lot of the old mill building. It was a crisp fall day. The dried leaves crunched under my feet as I walked from my car into Digital's then one and only executive office and manufacturing facility. I went into the cafeteria, grabbed a can of Fresca and a grilled English muffin, and sat opposite Al along one of the many beat-up old wooden folding tables, the kind used for trade shows but without the nice drapes. "You know, Jim, we only have today to get the project done and turn it over to the owner tomorrow for his final acceptance," said Al.

9:31am John Glavin, manager of the programmers, sat down at the end of a long, makeshift conference table. (It was 1968. Digital did not spend a lot on accoutrements.) On each side of the table sat about 10 programmers. John was just finishing an around-the-table poll of his 20 programmers. Each morning, every one of them had to stand up and speak to the state of the current project—what they did yesterday, what they intended to do today, and to request any help they might be needing. It was now my turn. John turned to me and inquired, "Well, Jim, did you get that cost center analysis program to compile last night?"

10:31am On each side of the aisle leading to the business computer room, there was a row of five whitewashed, plywood offices. Each office had two gray metal desks (ex-Army), one for each of the programmers. I was sitting at my desk writing occur statements in Fortran when Al said, "I know you believe there is a bug in the Fortran compiler, but John is not buying it," said Al. "Anyway, you are not going to be able to get to those guys. They are walled

A look back at programming in the '60s shows that agile software development isn't all that new

By Jim Johnson



off. I was asking around and I heard there is one guy named Gordon who only works at night. He might look at it for you, if you are around on third shift. They say he is kind of eccentric."

11:31am In those days, programmers did their work in teams of two and generally worked on one major application and some minor ones. They were more than programmers: They gathered requirements, created flow charts, wrote their own tests, demonstrated results, and wrote user and operator instructions—all the documentation—and met with the users. Al was the programmer for the cost center applications. His office buddy was a guy named Art, who had just left a few weeks before for a job at Arthur D. Little. Art had been working on a cost center analysis program for a few months before he left. It was written in Fortran. But Al, like all the rest of the programmers, was a COBOL programmer. I had taken two semesters of Fortran at Lowell University and three weeks previously had been drafted out of operations to take over the cost center analysis program. John poked his head around the office door and said, "We're going over to the Nathan Street Diner to grab some lunch. You guys want to come?"

12:31pm Eight of us sat around a black oval table with somewhat-matching chairs. In front of me was the blue plate special: Salisbury steak, French fries, and green string beans, all covered in thick brown gravy. We talked mostly about what each of us was doing for Thanksgiving. John said in a side conversation that he had talked to Peter. Peter was the owner of the project I was working on—my customer. He had told John that his CFO was really interested in seeing how the spreadsheet worked, that it would be a real help if they could have it in December for the coming year's budgets. Peter had also told John that he understood he could not get everything in the first release, but that he would really like the program to be able to read and write the data using DECtape. "Hey, I'll come by around 6:30 to see how you're doing," John told me. "After that, we can go to the package store and I'll help you pick out a nice bottle of wine to bring to your future in-laws' Thanksgiving Day dinner. You're 21 now—you can even buy it on your own."

1:31pm I had no idea how to program the application to read and write DECtapes. Not only that, but no one in my group knew how to create code to read and write DECtapes. They were all COBOL programmers for the business systems, which ran on a Burroughs 300. My application was the first and only business application in Digital that would run on Digital computers—first on the PDP-6, but targeted at the new departmental computer, the PDP-8. I took the manual and started to read how to program with Fortran reading and writing DECtapes. Although the business programmers had access to the systems to compile and test their programs, the system I needed was in the Fishbowl. The Fishbowl was located just inside the left side of Digital's Main Street entrance. It was about the size of a small house, 30 feet long and 25 feet wide. It had glass on three sides and you could look into it from the Main Street lobby. During the day, the Fishbowl was used by sales and service to woo clients to the new machines. I could not access it until after 5 p.m.

2:31pm I heard a scream and yelling. It was Walter. “They dropped my deck,” he yelled. “They dropped my deck,” he cried, and then, almost in tears, he whispered again, “They dropped my deck.” For COBOL programmers in the late ’60s, dropping a program deck was a traumatic event. Sure, the last few digits of the deck were serial, but that was only for the first pass. Changes to the application, errors, and jammed and worn-out cards meant the deck had to be reassembled by hand. Given that most programs were 200 to 300 cards, it was not a trivial task. Walter and his officemate would spend the next few hours collating and putting the deck back in order, one card at a time.

3:31pm I was still coding, but I was thinking about the business programmers. Programmers got access to the computer mostly during the day, in between small production runs. Heavy-duty production was run at night on the second and third shifts, when the programmers were either home watching (black-and-white) TV, sleeping, or both. The programmers would come in two at time and hand the deck to the operator. The operator would punch in the boot code using on and off switches. The card reader would read the first instruction and compile the deck. If the deck compiled, it would ask for input on the ASR-33 teletype machine. Most of the time it did not compile, and the printer would spit out a core dump with Level E error messages. The programmers would read the dump, flag the errors, pull the cards, mark them up, and go to the keypunch machine to make corrections. They would go through the same process over and over until the compile was clean.

4:31pm In those days, programmers wrote code on paper. If you were fortunate, a keypunch operator would keypunch your work from a coding sheet. I was through coding, but I was not fortunate. My code needed to be put onto paper tape, not punch cards. I went to an open teletype machine and started typing my code onto punched tape from my programming sheets. Both my program and the programs of the business programmers were small. We only had 32,000 bytes of memory to store our code, and virtual memory was yet to be invented. Therefore, compiled code needed to fit into these 32,000 bytes.

In those days, programmers wrote code on paper. If you were fortunate, a keypunch operator would keypunch your work from a coding sheet.

5:31pm As a computer operator, I worked the second shift and went to school during the day. Now, although I was still at school, I was a programmer and able to keep the same schedule. Therefore, the restriction of access to the Fishbowl was not a problem for me; in fact, it was a benefit. At the moment, though, school was on break for the Thanksgiving holiday and I was working the day shift, or least that is what I thought. Now, sitting at the teletype console, I first needed to add my new code to the current code. You did this by creating a new paper tape. First you would read the tape one line at a time and punch new tape. You need to stop the tape at the right time to change the input tape. Often the tape would break and misread, and you would need to splice it. It took me about an hour to get a clean tape and run my first compile of the night.

6:31pm I was going over my error report and core dump when John dropped by the Fishbowl. John asked me a few questions, and frowned when I told him I had yet to get a clean compile. “Let’s take a break and get that bottle of wine,” he suggested. We walked to the Maynard Package Store, which was located on Main Street, two blocks from the old mill. “Mateus,” said John. “It always goes good with turkey, and at $1.50 a bottle, it’s in your budget.”

7:31pm Back in the Fishbowl, I was just about to try my second compile. Correcting errors on paper tape was even more tedious than typing the code in. First you had to copy the tape to the point of correction. This meant reading the paper tape, stopping it at exactly the right place, and punching onto the new tape. Being dyslexic as well as digitally challenged meant lots of punch-outs. Punch-outs were deletes in paper tape made by punching holes across the tape. Both the ASR-33 and the PDP-6 would ignore punch-outs. My program on paper tape was now about five feet long. I booted the program in with a set of toggle switches on the front of the PDP-6. The tape read in pretty fast and you had to make sure it was straight; otherwise it would curl, and then break.

8:31pm Still no clean compile. My poor typing and bad correcting of the errors I had already found were now causing more errors. The day before, Al and I had met with Peter. Programmers in those days met with clients, sometimes on a daily basis. They would show them results, get test data, and offer feedback on a project’s progress. Art had designed this program to be a set format, with 14 columns. The first column was the cost item, the next 12 were for each month of the year, and the last was a total column. The columns were five digits long, so January was a “space Jan space,” and so on for the rest of the months. That meant the amounts had to be $10,000, or $10 million, if you left out the thousands. Peter wanted six digits instead of five.

Buyers Guide: Requirements Management Tools & Services

Company	Focus Areas	Product(s)	Service(s)
3SL	AC, APR, ARI, B, H, IA, ME, RCA, RCL, RHT, RV, T	x	
Accept Corp.	AC, C, DS, H, IA, INT, ME, RCA, RCL, REP, T	x	x
Axure Software Solutions	C, RCA, RE, V	x	
BITPlan	RCA, H, RHT	x	
Blueprint Software Systems	AC, ARI, B, C, INT, ME, RD, RE, REP, RHT, RV, S, V, W	x	x
Borland, a Micro Focus company	B, C, IA, RCA, RD, REP, RV, T	x	x
CaseComplete	AC, C, H, IA, ME, RCA, RCL, RD, REP, RHT, RV, S, T	x	
Compuware	APR, ARI, B, H, IA, INT, ME, RCA, RCL, RHT, T, V	x	x
eDevTechnologies	C, ME, RCA, RCL, RD, REP, T, W	x	x
Elsinore Technologies	H, RCA, REP, T, W	x	
Future Tech Systems	ME, REP, RHT, S, T, V	x	x
Gatherspace.com	C, H, RCA, RCL, REP	x	
Goda Software	B, C, H, IA, INT, RCA, RE, REP, RHT, RV,T, V	x	x
IBM	AC, B, C, DS, H, IA, ME, RCA, RCL, RD, REP, RHT, T, V, W	x	
iRise	C, ME, RCA, RD, RHT, RV, S, T	x	x
Jama Software	B, C, ME, RCA, RD, RE, REP, RHT, T	x	x
KollabNet	AC, H, IA, ME, RCA, RCL, RHT, RV, T		
Kovair	APR, B, C, DS, H, INT, ME, RCA, RCL, RD, RE, REP, T, W	x	x
MKS	AC, B, C, H, IA, ME, RCA, RCL, RE, REP, RHT, T, W	x	x
Pathfinder Development	RCA, RCL, RD		x
Polarion Software	B, C, H, IA, ME, RCA, RD, RE, REP, RHT, T, W	x	x
Projectricity	H, IA, ME, RCA, RCL, RHT, T	x	
Prometeo Technologies	C, IA, ME, RCA, RCL, REP, RHT, T	x	
Rally Software	H, IA, ME, RCA, RCL, RHT, T, W	x	x
Ravenflow	AC, APR, ARI, H, ME, RCA, RCL, RHT, T	x	x
Ryma Technology Solutions	DS, H, ME, RCA, RCL, RD	x	x
Serena Software	AC, B, C, H, IA, INT, ME, RCA, RCL, RD, REP, RHT, RV, T, V, W	x	x
Sophist	AC, INT, ME, RCA, RCL, REP, RHT, RV, T, W	x	x
Sparx Systems	H, IA, ME, RCA, RCL, RHT, T	x	
TargetProcess	C, INT, PM	x	
TechExcel Inc.	IA, RCA, RCL, RD, RHT, RV,T	x	x
ThoughtWorks Studios	C, PM, V	x	
VersionOne	PM	x	
Visure	B, C, INT, RCA, RD, RE, REP, S, T,W	x	x
Vitech Corp.	AC, C, H, IA, ME, RCA, RCL, RD, REP, RHT, RV, S, T	x	
Workspace.com	C, RCA, RCL, REP, RHT, S, T, V, W	x	

Key to Focus Areas

AC	Ambiguity Correction	RCA	Requirements Capture
APR	Automatic Parsing of Requirements	RCL	Requirements Classification
ARI	Automatic Requirements Identification	RD	Requirements Definition
B	Baselining	RE	Reuse
C	Collaboration	REP	Reporting & Analysis
DS	Decision Support	RHT	Revision History Tracking
H	Hierarchies	RV	Requirements Validation
IA	Impact Analyses	S	Simulation
INT	Integration	T	Traceability
ME	Multiuser Environment	V	Visualization
PM	Project Management	W	Workflow

Bold= Sponsor

9:31pm The Maynard House of Pizza was located directly across from Digital's Main Street entrance. They would make you a submarine sandwich and put it in the pizza oven. It came out hot and crispy. The House of Pizza also sold Coca-Cola in the traditional glass bottles. I was eating a meatball sub and drinking a Coke while poring over the error printout of my third compile. The program also had a set number of rows, which was also 14, thus creating a perfect square. Peter had no problem with the column constraint, but we wanted the rows to be variable. Most of the errors were in this section of code.

10:31pm I had my first clean compile in days! The new program was stored on DECtape, and I loaded it into memory using the toggle switches. Now, the hard part: testing to see if it worked. The teletype asked, "What period?" I typed in "1969." "What cost center?" I typed in "101." "How many rows?" I typed in "4." "What is your first item?" I typed in "Labor." "Jan?" I typed in an amount. "Feb?" I typed in another amount. I typed in a whole spreadsheet and at the same time punched it onto paper tape. I could read in the paper tape as input for my next test. I printed out the spreadsheet.

11:31pm I used 1s so it was easy to add up the rows and columns in my head. I found a number of conditions in which the total did not add up properly. I went through my source code, made the changes to the paper tape, compiled, got errors. I corrected errors and compiled again. Finally I got another clean compile. A few people had been in the Fishbowl in the early evening, but now there was only one other person and he was leaving. We had talked a little shop earlier about the various projects we were working on. I asked if he knew a guy named Gordon who worked on the Fortran complier. He said, "I worked with him awhile ago, but he turned into a real strange person. He only cut his hair once a year, on May 1st. He also only ate white food." He agreed that Gordon worked during "off hours," but did not know where to find him.

12:31am I needed to test for larger and more complex spreadsheets, but there was no adding machine in the Fishbowl. The only people who had adding machines were accountants, and their doors were locked to me. Then I remembered that Benny had an adding machine. Benny worked in the business computer room as the data control manager. All input and output went through him. Benny also did quality control on reports and kept track of logs and timesheets—which is why he had an adding machine.

1:31am The old mill was a sprawling complex of disjointed, connected buildings. If you could place two rooms farthest apart from each other, they would be the Fishbowl and the business computer room. I ran another two dozen spreadsheets, all with different amounts and rows. Then I took the printouts and marched off to the business computer center. The third-shift operator was a kid named Eric. Eric had joined Digital two months ago and had a five-digit badge number. (My number had four digits and Benny's had three—he was an old timer.)

Heavy-duty production was run at night on the second and third shifts, when the programmers were either home watching (black-and-white) TV, sleeping, or both.

2:31am In the 1960s, adding machines were purely mechanical and looked more like the old-fashioned type of cash register that you might find at an old country store. I took my two dozen spreadsheets and added up the columns and rows. Twenty-two of them added up perfectly; two did not. So, what did those two have in common, and not in common, with the others? First, they both had more than 14 rows; but four others that were OK had more than 14 rows. Second, they both had an odd number of rows. But half the others had an odd number of rows. Third, they had amounts that totaled to six digits. But three of the others had totals of six digits. None of the others had all three conditions.

3:31am It turned out that when all three conditions existed, the program would add the last row twice. I went to the Fishbowl and printed out my last version of source code. I went line-by-line through the code, and could not find anything. I needed to find Gordon. I went to the guard station and asked the guy at the desk if he knew where I could find Gordon. "What's his last name?" he asked, but of course I did not know. The guard said he was new to this job, but another guard now doing the rounds would be back in a few minutes and he could probably help. I took out my code pad and started to write a workaround. The roaming guard showed up a few minutes later. He knew Gordon and said he had just seen him working at his bench. "Oh, I see quite a lot of him," he added, with a wry smile, and gave me directions.

4:31am I wove my way through the old mill, past half-completed machines, test machines, desks, and paper everywhere, until I reached what I believed was Gordon's lair. A man was sitting there in a tall director's chair, the kind with the canvas back and seat. In front of him was a long bench strewn with machines and hardware. There was also a tall glass of milk and a plate of marshmallows. Draped over one of the machines was a shirt and pants. Under the director's chair was a pair of shoes and some socks. When I was near, I called out Gordon's name and he stood up in all his naked glory.

5:31am Now back at Benny's desk, I went to work on the instructions and documentation. Gordon had taken my code and said he would look at it and that I should come back in an hour or so. He speculated that the code might harbor a compiler bug, but doubted it, saying that most likely it was just a logic error. He was a tall, lanky guy, and clearly had nothing to hide. He liked working at night, he said, when he could be himself and no one was around to bother him.

6:31am As I headed back to Gordon, I wondered what I would find. He had found the problem, he said, and fixed it. He added that he had made some changes to make the program run more efficiently, and handed me a DECtape. "It looks like a bunch of rows and columns," he noted. "What good is it and who uses it?" I explained that it was for accountants. I didn't ask, and he didn't say, if the compiler had had a bug or not. I went back to the Fishbowl, ran a bunch of new tests, and took them back to Benny's desk. They all added up. I finished the instructions.

7:31am I was spreading grape jam on my grilled English muffin when Al came into the cafeteria. I told him I had just finished up. I gave him the compiled program on DECtape, the test input paper tapes, and the instructions. We talked about Gordon and the events of the last 24 hours. He told me to go home and get some sleep. I went to the parking lot and got in my car. You know, I thought, I will just sit here for a while. I put the seat back and closed my eyes. SW

Jim Johnson is the founder and chairman of the Standish Group. He has been involved in the computer industry for more than 40 years and has a long list of published papers, articles and speeches. Visit www.standishgroup.com.