Dynamic Link
Christian Perspectives on Software Development
Issue 1 2007-2008

Welcome!

Thank you for taking the time to read Dynamic Link: Christian Perspectives on Software Development.

This journal represents the participation of over 80 Christians involved with information technology in roles as software engineers, business analysts, consultants, software testers and development managers. This special group of people donated their time to provide you their perspectives on the connection between their faith and vocation.

The Vocation Venture Fund of Calvin College, supported by the Lilly Endowment, provided the financial support for this publication. Initially, we contacted software developers who were professed Christians and talked to them about the influence of their faith on their career. A summary of that is provided in this journal. We would have stopped with that one article, but what surprised us in that original project was the strong desire many had to share and talk about this aspect in their work life. Apparently, there are very few forums to discuss software development in a Christian context. So, we extended an invitation and, as you will see, were blessed with articles written from the heart. We encouraged our authors to maintain a casual style of writing expressing their personal views.

Dorothy Graham, a well respected software testing consultant in Europe and the United States, summarizes how she and her associates have integrated their faith into their consulting business model. Mike McIntosh, a 23 year veteran in multiple software development roles and now with Amway/Quixtar, shares a valuable lesson on overconfidence in software engineers. Inspired by 30 years as a contractor, Kenneth Shafer's sleuth analogy leads to an interesting discussion of Christian stewardship present in the act of software creation. James Stewart, a free lance developer, reminds us of our Christian calling to deliver quality in all that we do to create a positive impact in the daily lives of our users.

Also, two recent Calvin graduates share their experiences. Jared Staal, who participated in the original project by conducting phone interviews, shares his observation of an almost symbiotic nature between one's personal work environment and faith. Remington Steed, a developer for the Christian Classics Ethereal Library (www.ccel.org) since he graduated last year, reflects on how his undergraduate internship revealed software development being another activity supporting the great commission.

Our goal is to create a dialogue. This journal is the start. We pray that you will consider sharing and investigating issues of faith as you progress in your careers.

Patrick M. Bailey
Department of Computer Science
Calvin College

Table of Contents

CALVIN
College

Educating *for* Vocation

Funded by the Vocation Venture Fund

Christian Faith in the Software Industry

by Dorothy Graham,
Grove Consultants, UK

Dorothy Graham, a Calvin graduate, is a leading expert in the area of software testing. Dorothy talks about how her faith guided her career and how its integration into her and her colleagues' daily work has been formalized.

When I was a student at Calvin, I never dreamed that I would end up starting a business, let alone one that has been as successful as Grove Consultants is!

I don't remember much about my study of Calvinism (my father, a theologian who taught at the Seminary, would not be pleased to hear this!), but I do remember that faith should be integrated into all aspects of life. This I have always tried to put into practice.

I took teacher training at Calvin (including practice teaching) and got a Master's degree from Purdue University. Rather than go into teaching, I decided to see "what the real world was like". I worked as a software developer for 2 years at Bell Labs in New Jersey, in a group that tested the work of the development group. I then moved to England (having married an Englishman whom I met at Purdue), and worked as a programmer / software engineer / software development team manager for seven years. I paid more attention to testing than many other people seemed to at this time. I was always a church member, but didn't think a lot about how my faith directly influenced my work. In fact I can remember one occasion when I was rather unkind to a contractor when his renewal came up. I concentrated too much on a recent problem rather than on his long-term performance. I have always regretted this and wished I had been more Christian in my attitude towards him.

A new job opportunity arose to become a trainer for The National Computing Centre, which conducted training courses for people in industry. This appealed to me as it used both my technical knowledge and my interest in teaching. Initially full-time, it became a part-time contract position when our daughter arrived, which was ideal for me. I worked six to eight weeks a year presenting training courses. As our children grew, I enjoyed doing more and more training work, both for NCC and others. Realizing that software testing was a much neglected area and being biased towards testing by my experience, I helped develop new training courses in software engineering and testing. I was doing well, my work was appreciated, and I really enjoyed this work and my desire was to expand this career.

Does God really want me to do this?

In 1990 we went to "Spring Harvest", a week-long retreat put on by the church. There I felt very disturbed about whether my ambitions were taking precedence over God's plan for my life. I didn't like the idea that God might want me to sacrifice what I loved doing to do something different for Him. Over the next year I pondered this, and when next year's Spring Harvest came along, I was at last ready to be open to do what God really wanted, and was at last willing to give up my testing career if He was calling me to something else. I prayed for guidance and for Him to show me what He wanted me to do. The message that came to me during this week was very clear to me: God wanted me to continue what I was doing. This was confirmed when I arrived home and discovered to my surprise that my very first conference presentation was scheduled to be not just a track but a keynote!

Dorothy Graham

How Grove Consultants started

My husband Roger and I had moved to a house called "Grove House" in 1987. The following year when I had trouble getting an invoice paid because I didn't have a company name, I chose "Grove Consultants", though the "s" was unnecessary - it was just me.

In 1993, two interesting things happened. My husband took an early retirement package from Manchester Metropolitan University, so I became the breadwinner of the family. My work had been gradually building up over the years to the point that we thought we could survive on what I earned – some trust was involved here! Roger and I became business partners, the "s" became reality, and he has always been a stabilising and sensible influence on Grove.

The other thing was that my friend and fellow testing enthusiast, Mark Fewster, whom I had known since 1989 and had worked with on various projects, asked to join us, so Grove Consultants grew to three people. In that first year, we didn't do enough marketing, and we had some difficult months, but eventually work picked up and we became financially solvent.

When he started, I discussed with Mark how I wanted us to work together, and wrote the first version of what has

become the Grove "Ethos" – a statement of our Christian business principles that is the framework for Grove Consultants, which has grown to include Lloyd Roden (who joined in 1999), Clive Bates (2002) and Julie Gardiner (2006). Through the years, we have often returned to our ethos, *particularly in difficult times*. Each of us tries to work under this ethos not only in principle but in action.

I would like to take you through our ethos, statement by statement, to explain what it means to us. The Ethos is in blue, followed by my comments in italics.

Grove Consultants Ethos

1. What we are

1.1 Grove Consultants is run as a Christian business. Our aim is to serve God by helping people.

This is a "nutshell" description of who we are and what our purpose in life is.

1.2 We are a select organisation consisting of individuals who subscribe to the principles outlined in this document.

People join Grove by invitation, after prayerful consideration and by unanimous agreement.

Mark and I had been saying for a long time, "If we ever got anyone else, it would be someone like Lloyd." When our workload picked up with the new qualifications in software testing, and Lloyd booked onto our second course, we took him out for dinner and "popped the question" to him. With Clive, we had thought he might be suitable for a couple of years, but decided consciously to wait until he asked – which he eventually did. Julie joined us less formally at first (she was a bit younger) but it soon became clear that she should be part of us too.

We do not have any plans to add anyone else, but we are open to God's guidance on this – He will make it clear when the time and the person is right. At various times, other people have expressed an interest in joining us, and we have considered some people more seriously than others. In each case God made it clear to us that the person was not right.

We realise that as a small select partnership, we have much more control over our team than a large organization of many employees would have. This is one reason why we want to stay small.

1.3 We provide specialised expertise in software testing, test automation, inspection and other related software engineering activities. We aim to provide value for money by helping our clients improve productivity and quality through practical workable solutions. We aim to measure the effectiveness of our work.

How does our faith affect our technical work? One aspect is that we want to do good quality work, and build on our reputation for quality. We do not "push" our faith, although we may use an example from church life, or we may wear a "fish" symbol to let people know that we are Christians.

The religious culture is different in the UK. The percentage of people who attend church here is less than 5%, rather than the 50% it is in the US. Being more upfront about our Christianity is not as well accepted in the UK as it might be in the US, so we are discreet. When people ask, we can be more explicit about our faith (which happens sometimes).

1.4 We aim to keep work in perspective with other aspects of our lives, including family life, hobbies and other interests.

To be honest, this is the one we all struggle with a lot! This is more a statement of intention than something we put into practice very well. Of course, we all love what we do, so it is very tempting to work too much. I have noticed that things do work much better for us when we remember to keep our lives in balance.

1.5 We are not "cut-throat", "profit above all" or a "body shop".

Many organizations that are our direct competitors seem to be one or more of these! By a "body shop" we mean an organization that puts anybody on any job. We are limited in resources, but every client gets one of us, not a junior trainee.

1.6 We aim to enjoy our work and our time together.

I think we do achieve this one, at least most of the time!

2. Christian principles as applied to our business

2.1 We will be kind to people, encourage them, look for the best motives for their behaviour, try to understand where they are coming from, not be sarcastic, not go behind people's backs.

It can be a challenge to "love the unlovable" in a work context but this we do try to do. Sometimes there will be someone who comes on a long training course who just seems to rub everyone else the wrong way; sometimes there are people who monopolize our time talking about things that aren't really of great importance or interest to us; sometimes we will hear things that other people have said about us that are not only unkind but untrue. In these situations, it is very tempting to react with anger, frustration and a distinct lack of kindness. Yet we do try to be Christian in the way we respond.

For the most part, we are highly respected within our industry, and known for our integrity. However that doesn't seem to stop "professional jealousy" so we need to accept that we will be criticized, sometimes unfairly. We are confident in our own integrity and support each other if this happens.

2.2 We will be happy to offer free advice and share materials with others as appropriate, even if they are not or are unlikely to become a client.

I had one very rewarding "bread on the waters" experience. A student (whose name I didn't even recognize) came on a training course and in the first coffee break, thanked me for getting him his first job in testing - this was a complete surprise to me! It seems he had first gone for an interview at a company where we provided training and didn't get the job, but they gave him my name. He called me and I gave him some advice, including sending him a paper copy of an encyclopaedia article I had written on testing (this was before the days of email!) He had read this and then went for another interview for a tester and got the job. He brought a copy of the article with him to show me. It was wonderful to find out about this several years afterwards.

This approach also makes good business sense, because unemployed people will probably find other jobs in due course, and they remember that we helped them and then come to us for training or consultancy.

2.3 We will cooperate with our competitors where appropriate. We will not seek or accept any fee as "bounty" for referrals to other companies, nor will we enter into any arrangements whereby a "kick-back" is given for referrals to us from other companies. We will not speak ill of our competitors in public. We will freely refer people to any other appropriate organisations when it is in their best interest to do so.

We have often been asked to refer our clients to other organizations, e.g. a tool vendor or other training organization (specializing in a different area). We decided that money changing hands would inevitably bias us (even subconsciously) so we have always refused any kick-back.

This does give us the freedom to make recommendations completely independently.

We suspect that we lost a major training client for several years because one of our competitors offered a kick-back to the organization that did the training bookings. In recent years, individuals from that client have "fought" internally to get us back, which has been rewarding for us.

2.4 If another organisation or person harms us in some way, we will try to learn from the experience, and we will not harbour a grudge – **we will forgive them**, not retaliate.

We have had a number of "forgiveness practice" opportunities in running our business! There are the companies that "forget" to pay us, the student who didn't pay for the course he attended and seemed to disappear from the face of the earth, a seminar organization that went back on its written word about paying speakers I had arranged, the vituperative outburst from someone who misinterpreted taking editorial control of a shared document as trying to take over the whole content, those who doubt the sincerity of our desire to improve the software testing industry, and others. However, we know that if we can't put the experience behind us through forgiveness, it will continue to fester in our own minds. ***I am convinced that forgiveness is one of the most powerful aspects of Christian faith.***

2.5 We aim to provide ourselves a reasonable income and to make a profit. Our fees should be fair to our clients and to us. Fee levels should allow a level of work to support personal as well as professional goals.

It is important that we do realise that we are not a charity, we are a business! We need to be fair both to our clients and to ourselves.

2.6 We will be honest in all financial dealings, such as the value added tax (VAT) and income tax returns, number of hours worked for a client, expenses claimed, etc. We will pay our suppliers promptly.

While obvious, it is worth stating. We added the last sentence after realizing what a hassle it was for us (actually for Jacqui, our accountant) when companies don't pay promptly. Some companies seem to make it a game – "let's see how many excuses we can invent for not paying these people"! We want to treat our suppliers better than we have been treated!

3. Our specialised expertise

3.1 We should aim to increase our knowledge and capability in relevant areas of expertise. We should always apply and share our knowledge and ideas both internally and externally, to provide an opportunity for feedback and to spark better ideas. We aim to produce and maintain high quality course materials.

We pride ourselves (humbly) on the quality of our work, so this is important for us.

3.2 We should give our time to provide general help and support to the testing community. We should attend and give presentations at conferences and seminars, to help people, to learn and to market Grove.

This is what we sometimes refer to as "charity work" – volunteer or low-paid work that advances the whole software testing community. All of us have given many hours to this kind of work (sometimes recognized, sometimes not). Conferences are good opportunities for us to learn, but also to gain visibility and share our ideas with others. We

don't advertise or do much "sales and marketing". We get most of our work from existing clients or from people who have heard one of us speak.

3.3 We should learn from our clients, our students, other people and our working experiences. We will give recognition to those who we learn from.

Any stories about bad experiences are told anonymously. For any stories about good work, we try to give credit to the people who told us. For example, if one of our licensees has a good idea for improving our courseware, we acknowledge their contribution by name. It is surprising how often people's contributions are either not acknowledged, or their boss takes the credit! We don't want to do that.

3.4 We do not have all the answers. We should avoid arrogance. We should remember that advice is easy to give; making it work in practice is what is difficult.

Every now and then, we get a "humility opportunity" when we are "taken down a peg" after we find ourselves "thinking of ourselves more highly than we ought to think"! We know these are good for us!

4. Our responsibilities to each other

4.1 We will support each other and communicate with each other frequently and effectively.

As a distributed organization (we all work from home), our biggest challenge is communication. Whenever we have problems, this is usually the cause.

We have quite often managed to upset each other, sometimes causing sleepless nights and much anguish. A couple of times we have come near to breaking apart, but when we are together, and can pray together to open our meetings, we realize anew how privileged we are to be working together. We have always been able to go into the depths of our misunderstandings and "put ourselves right" with each other. I pray that this will always continue.

4.2 In our dealings with each other, we will remember the Grove Prime Email Directive :

Regardless of how I am feeling at the moment, and whatever the email says, I will understand and truly believe that the other person communicated to the best of their ability, given the state of their knowledge, their interpretation of what they knew, and the pressures they were feeling at the time.

This is our most recent addition to our ethos, added last October after a meeting that had a lot of misunderstandings and differences of opinion to get through.

We have found that we most often upset each other through emails, sent in haste, in a "knee-jerk" reaction to what we thought the other person was saying (which often wasn't actually the case). Hence we have adapted the Prime Directive from Norm Kerth's book "Project Retrospectives".

This email directive has been very useful when we remember to think of it, and we have suffered the most when we have forgotten about it!

Summary

Our Christian faith is at the heart of our work, as we have tried to summarize in our Ethos. Our faith determines how we deal with other people (clients, students on our courses, our competitors), how we deal with each other, and the activities that we do (e.g. "charity work"). It influences our finances and our technical work. We are fallible and often fail to live up to our ideals. We have been very blessed with rewarding work, stimulating colleagues, lots of fun and financial solvency. We have tried to integrate our faith into our working lives as fully as we know how.

I would be interested to hear your comments and experiences. ■

***Dorothy Graham**, a Calvin College graduate, is the founder of Grove Consultants which is based in the UK and provides advice, training and inspiration in software testing, testing tools and Inspection. Originally from Grand Rapids Michigan, she has lived and worked in the UK for over 30 years.*

Dorothy is co-author with Tom Gilb of "Software Inspection", *Addison-Wesley, 1993, co-author with Mark Fewster of* "Software Test Automation", *Addison-Wesley, 1999, and co-author with Rex Black, Erik Van Veenendaal and Isabel Evans of* "Foundations of Software Testing: ISTQB Certification", *Thomson, 2007.*

Dorothy was Programme Chair for the first EuroSTAR Conference in 1993. At the EuroSTAR conference in Barcelona in 1999 she was awarded the IBM European Excellence Award in Software Testing.

Her contact information is below:

Dorothy Graham, nee Hoekema, attended Calvin College 1962–65 and 1966-67 (year out in England). Grove Consultants: www.grove.co.uk

Software Development and Maintenance as Stewardship

by Ken Shafer

In software development, how much of what we do is creation versus discovery? Ken Shafer believes discovering the answer will bring you one step closer to understanding your relationship with more than the software you build.

Many years ago, in the company of peers at a Software Conference, I attended a workshop by a member of Toastmasters International, an organization dedicated to furthering better public speaking. The workshop leader conducted an exercise in extemporaneous, or spontaneous, public speaking by asking me, without preparation time, the following question: "If you were not a software specialist, what career might you have chosen, and why?"

The answer immediately came to mind, and then to my lips: "Why, a detective, or private detective, of course! After all, I would be performing very similar work. Ferreting out clues, gathering evidence (analyzing the problem), developing a list of "suspects" (tracking down bugs), identifying the culprit (diagnosing the problem), and bringing about justice (making the fix.)"

I still feel that way today, and I still have that point of view about working with software, I've also developed a broader and different way of answering that question. And that is "software specialist as a steward", and the conduct of his career as "stewardship".

What is stewardship? Here is one Miriam-Webster Dictionary definition: "Stewardship: Duties and obligations of a Steward, an employee on a ship, airplane, bus, or train who manages the provisioning of food and attends passengers." And here is another: "the conducting, supervising, or managing of something; especially: the careful and responsible management of something entrusted to one's care."

What is the relevance of that to my career as a professional dealing with software? My first observation is this: although I have acted at different times as software analyst, software developer, quality assurance tester, and maintenance programmer; I have come to feel that I may like maintenance programming the most. This may be contrary to the "popular beliefs" regarding what roles are glamorous.

all work lends dignity to the practitioner

However, I have felt for some time that *all* work lends dignity to the practitioner and that any indignities endured in the workplace are the result of damaged relationships among workers independent of the work to be performed. It is not to any deficiencies in the work itself. However, the real reason that I think I prefer it is that it is the best way to manifest both definitions of stewardship above.

I will explain, and I will use the second definition to help illustrate. Maintenance programmers are charged with "conducting, supervising, or managing of something", the thing "entrusted to your care." That "thing" is the source code of the system that someone else has given you. This immediately creates a sense of humility, because something has been

Ken Shafer

given to you. It also creates sense of excitement, because of the opportunity of discovery about that product. In other words, software maintenance is work performed upon an artifact created by others. Just how much will be made dramatically clear. So the first blessing bestowed upon us is this feeling of humility.

I believe once one understands this with a certain amount of detachment, it becomes easier to see that there is a greater purpose, or social purpose, that is behind the creation of the artifact bestowed upon us. For me, knowing that the software application has that greater purpose represents the triumph of transcendence of the Eternal over the day-to-day hum-drum. In my career I have worked on projects with many different applications: those that brought life-saving medicine to the suffering, economic reconstruction loans to uplift entire populations, or just helping people stay connected through the postal service. When we are stewards, we are stewards not just of the source code, but of the greater social good.

From there we can see another point of view: that software helps us establish connections with other people. In addition to binding us to a community of people, software also binds us to a community of spirit. How does it bind us to a community of spirit? The second definition will help us to see: as stewards, we are attending to

follow passengers on a voyage of discovery. To me, the purpose of that voyage is to discover God's Divine Plan and what our role is in that Plan.

Before I became a software practitioner, I studied graduate mathematics, and a couple of years ago I posted some thoughts on a discussion board about what I saw as parallels between certain schools of thought in mathematics and certain perspectives in software engineering. I called that discussion thread, "Is Design Created or Discovered?" It had to do with a long-time debate among mathematicians as philosophers as to whether mathematical concepts like numbers are man-made inventions or God-given creations. Here is an excerpt from that discussion:

I dusted off my very old "Philosophy of Mathematics" by Stephen Barker, and have pulled out of that book some debate excerpts between the "conceptualists" and the "realists":

> Barker wrote: We may classify "conceptualists" those who hold that there are numbers and that they are abstract entities, but that they are creations of the mind. And we may classify as "realists" those who hold without qualm that numbers as abstract entities literally exist independently of our thinking…Unlike the conceptualist, the realist does not feel that the realm of abstract entities is in any way limited by the mind's poor power to create, for abstract entities exist in and of themselves, not as constructed by the mind…From the realist's point of view, the task of the mathematician may be compared to a voyage of discovery. The mathematician cannot create or invent the objects of which he speaks, but they are there waiting for him to discover and describe them.

Just as the mathematician's task may be compared to a voyage of discovery, so is the software practitioner, especially as maintenance programmer. Figuring out what the program does is certainly a discovery process that does not happen overnight!

In an analogous way, and coming from the "realist" camp as described above, I have come to believe that for any given (software) problem, the Design of the solution for that problem exists independently of our thinking, and that we do not "create" the solution, we merely "discover" it! This point of view is echoed by those who contend that "the design is in the code", or that "the code IS the design". Designs are discovered, not created.

> In addition to binding us to a community of people, software also binds us to a community of spirit.

Fostering such an attitude helps immensely in letting go of one's ego during those design and code reviews! For now we realize we are not really critiquing a creation that is "ours", we are instead assessing our understanding and awareness of the solution, and we are getting help from other people in doing that! It is important to make the following distinction: although the "Design" is pure and perfect, our implementations are always imperfect, because the awareness and the understanding is imperfect, but by working and acting, better awareness and understanding is achieved. This imperfection explains why there is the appearance of a multitude of different ways to code the solution, because each one is an imperfect understanding. Some are better than others, but all of them are connecting us to achieve a better understanding - an understanding that it is not YOUR design, but HIS design. And then we can begin to see how this applies not only to software but to life itself.

For example, software, though an intangible, still manifests itself as a birth, a life, a death and a rebirth. It is in fact what is called the software life cycle. So, if the design is not ours, but HIS, then what is ours? I believe that attitude and stewardship are ours - we are the caretakers.

In conclusion, design is HIS; Stewardship is ours. Software is not an artifact to be "created" where in our mind we "own" the software product. Rather software and its development is a means of cultivating an awareness of who we are and what role we are to play in a Master Plan. ■

Ken Shafer *has been a software professional for more than thirty years. Graduating from Indiana University with an MA in Mathematics and an MS in Computer Science, he has spent his entire career as an IT contractor. Although his work has encompassed all phases of the software life cycle, including Quality Assurance and development, he is most comfortable and most effective in legacy applications maintenance and their enhancement, rejuvenation, and transformation. Ken keeps an active professional life, being a twenty-five year member of the Association for Computing Machinery (ACM), the Institute of Electrical and Electronics Engineers Computer Society (IEEE), the Independent Computer Consultants Association (ICCA), and Phi Beta Kappa honorary. He is married to his college sweetheart, Gayle, and together they continue their quest for the nearly-perfect house with the nearly-perfect dog. Both have yet to be found.*

Against Optimism in Software Development: Optimism v. Realism in Business Software Development

by Mike McIntosh

Software developers have a natural tendency to be optimistic. According to Mike McIntosh, we could be more optimistic about the expected results if there was less optimism and more realism in our practices.

We must take care not to make intellect our god. It has, of course, powerful muscles, but no personality. It cannot rule, only serve.[1]

~Albert Einstein~

What is the relationship of Christian faith to software development? The sum of it is *wisdom*. This wisdom is sturdy with Biblical common sense, based on and shaped by historic Christianity. I believe wisdom has led me away from an optimistic approach and to a realistic approach toward software development. This article will describe why I believe it is a wise move. Despite twenty-three years in IT, I can name only three or four colleagues who are realists. In our field, the great temptation leans towards optimism.

In writing this, I am speaking from my particular experience. That is, I write as a Christian who has participated in or led numerous software development projects in an IT business setting. When I use the phrase "software development" or the word "development," it assumes this context and background.

1 Quoted in *The Instinct to Heal,* David Servan-Schreiber, Rodale Pres, 2003, p. 13.

Optimism

Optimism in software development is, as I think of it, the belief that any business problem has a software solution that is sufficient to solve the problem. Optimism about software is reductionistic and has many facets. First, it almost always underestimates the effort required to design, build, test, and implement a software solution. That is, it believes that business software is inherently simple to build. Second, it mistakes the technical aspects of a business problem as the whole problem, forgetting that at the base any business problem is first of all a *human* problem. Third, it believes when the software is delivered the project is done. It limits a software project to software delivery. It skips over, for example, training business users or capturing "lessons learned" from the project. Because it ignores the results of past work, it is "the triumph of hope over experience."[2]

We are nursed on such optimism. In North America, we are the children of inventors. We helped win two world wars with our technological ingenuity. Today, we have almost a sense of entitlement: we expect technological improvements if not significant breakthroughs, whether in medicine, entertainment or software.

Most of us take an optimistic approach to problems. My mother-in-law asked me to replace a circular nylon washer in her kitchen faucet, which was leaking. I had just replaced washers in a couple of faucets at my own home two weeks before.

2 Borrowed from Samuel Johnson, an 18th century English author. The original quote was regarding remarriage: "A second marriage is the triumph of hope over experience."

Mike McIntosh

"I'll have it done in less than an hour," I assured her. Four hours, three trips to the hardware, and $70 later, she had a *new* kitchen faucet. I had been optimistic.

Here is another story that illustrates technological optimism. Long ago in the 1960s, computer programs and data were stored on paper cards sort of like index cards, only wider. Each card held 80 bytes of data. Twenty years later, though, it was these same types of punch cards that launched my IT career after teaching history. I was hired to maintain business applications that ran on an outdated IBM 1401 mainframe with 16,000 bytes of memory. The source code language was IBM 1401 "Autocoder" that compiled into assembler code.

Our managers determined to re-write these first business applications from this 1401 Autocoder into COBOL. They tossed the original source code, since the re-write would not be difficult.

So they thought. The re-engineering continued even after I left seven years later. Since they had literally tossed the original source code—that is, threw the cards into the trash bin—all they had left was the compiled assembler code stored on punch cards. So they hired staff to learn and maintain applications that had no source code.

It was an odd start and I owe it to my employer's technological optimism. They thought writing (or re-writing, in this case) software was easy. Yet these old programs proved remarkably complex.

Years and years later, I see and hear this optimism all the time despite our employers paying us very well to make intelligent, wise decisions. At design meetings, someone from Marketing asks, "Can we make the website do *this* specific thing ?" Then IT responds, "Sure, we can build that; we can make it do whatever we want." We implicitly start from the position of infinite possibility and work backwards from there when we define our software projects. In so many words, we have just said, "Building software is easy."

Usually our development estimate only considers the time needed for coding, and even this is often a hopeful estimate. Seldom considered are reviewing design, walking through code, testing interfaces between other code, or confirming assumptions with other developers.

Optimism in software development is not without benefits, of course. Management likes a "can do" attitude. No one likes to say "no," and most of us want to look good. Besides, if we meet the deadline, especially if we live thorough eighty-hour weeks and consume copious amounts of pizza and caffeine, we look like heroes. There is also a not-to-be-denied sense of pulling it off with a team that is a reward in itself.

We figure the cost is worth it. Yet do we stop to tally the cost?

The cost can be great. Individuals begin their next project in debt, intellectually, emotionally, physically. If we are tired, it is more difficult to do a creative design or to stand behind unpopular but good decisions. Our courage lags. "Fatigue makes cowards of us all," Vince Lombardi told his football players. Who would ask someone to start training vigorously for a marathon when she had just completed one yesterday? So the next projects start slowly or start late or start poorly.

Our ability to "create" derives from God.

Despite heroics, however, many software projects do not meet their deadlines. Some flat out fail, certainly more than half, perhaps more. The effects on the next projects are the same as above, but now demoralization factors in.

This is, I believe, a realistic assessment of an optimistic approach to development.

Realism

The better part of wisdom is to take a *realistic* approach to software development, an approach that seeks to see things as they actually are. Historic Christianity claims to be the best description of reality, of how things really are. For example, God is infinite, and we are not. Our ability to "create" derives from God. He creates from nothing. We "create" from the things He has created. The Lord has made humans in His image, we have fallen into sin, and the only way out of our mess is through his Son, Jesus Christ. All this is based on Scripture, and the Church with the guidance of the Holy Spirit has summarized this in our creeds, confessions and catechisms. The truth we speak describes what we believe to be reality.

It seems that we should want our lives and work to be congruent with the way things are, the way the universe actually is and behaves.[3] In approaching our development work, then, we can take our cue from our faith. I strive to have our software efforts and products reflect the way things actually are.

3 The way we interpret the universe, of course, is a reflection of our pre-suppositions, that is, our worldview or what we believe.

So perhaps a start of a confession about realistic software development would have the following statements. First, good development is difficult. It acknowledges human and technical limits. It strives for clarity in code, requirements and communication so that the focus is as clear as possible. It is humble because humans have limits and are fallen, and our solutions always fall short of the ideal. It is humble because it is willing to learn from past mistakes as well as successes. I will attempt to tease these out in the next paragraphs.

Software development is *difficult*, despite what optimists think. The difficulty lies in producing a simple solution to a complex problem. Note the word choice: "simple," not "simplistic." "Everything should be made as simple as possible, but not simpler," said Einstein.

We harness creativity and tenacity to pull this off. More than once, I have walked out of design meetings exhausted, as if we had physically wrestled all day with a problem and only at the last moment pinned it down. Other times, the creative juices flow and a simple, elegant solution emerges. Someone who does not understand this confuses "simple" with "easy."

C.A.R. Hoare put it this way: "There are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies and the other way is to make it so complicated that there are no obvious deficiencies. The first method is far more difficult." Our best work is difficult, our best solutions are simple.

A second implication is that development realism acknowledges *boundaries*. We often deceive ourselves—in a virtual world, it seems we can create things from nothing. We are like gods.

How foolish. Limits pervade work and life. Hardware has limits. Programming languages have limits. Our intelligence, energy

and bodies have limits.[4] When humans go beyond these limits, when we expect more from something than it was created for, Scripture calls it idolatry. They cannot bear the weight of our optimistic expectations.

The truth is we do our best work within limits. As mentioned earlier, I maintained assembler programs that ran on the IBM 1401 machine's 16K of memory. I inherited these programs from former 1401-programmers. Their code routinely amazed me by what it could accomplish in that tight, little 16K box.[5] Kind of like pouring Lake Erie into a Coke bottle.

If we are wise, we will consider our project team's limits. We are amazingly made and can do amazing things when we have to. Shame on the manager or project leader, however, that requires or allows regular overtime on projects. "All work and no play makes Jack a dull boy," and a burned out one.

Third, realistic software development strives for *clarity* in communication. With the best intentions, keeping the project team on the same page is hard. When ambition or gossip or politics enters in, the difficulty intensifies.

But we are wise to keep at it, otherwise confusion takes over. Clarity mandates that we state what the software project will deliver and what it will not. Clarity ensures the project's business requirements and rules, deadlines and budgets, are defined and communicated. Clarity sets the right expectations and gives a project team its best chance to reach its goal. Clarity minimizes confusion and permits the work to proceed more rapidly than otherwise. "It is hard to be aggressive when you're confused."[6]

The last implication is regarding *humility*. If we have our eyes open, we choose humility and reject *hubris*.

If we do not, life will do its best to hammer it in to us. We seem routinely surprised by the unintended consequences of software implementations. Routinely! Still, in my weaker moments I optimistically believe our solution will do what we intend, and only that. Once implemented—oops!—who would have thought of this possible outcome. I am mortal after all, and I work with mortals. I can despair, become cynical, or arrogantly blame others, and any of these will blind us. Or I can become realistic and thus humble.

Interesting that *humble* and *human* share the same Latin root: *humus*.[7] To be humble and to be human is to know oneself to be "of the earth" and not one of the gods. The Hebrews knew God made Adam from the earth. Such humility makes us teachable and ready to learn from our mistakes.

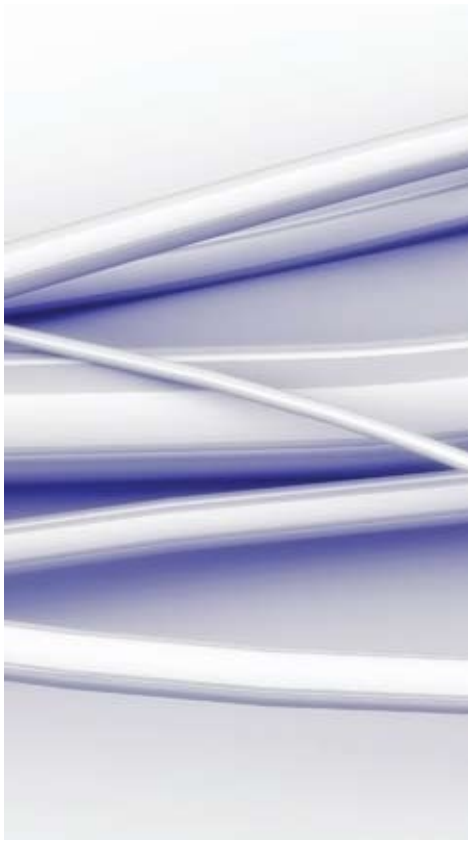
Conclusion

In this article, I have tried to show that although optimism in software development is very common, it is neither wise nor effective. It does not look at the big picture of software development but rather is reductionistic. It believes software is easy to build and so underestimates the effort to build it. It focuses only technical problems and does not see the larger human problems. Optimism assumes when the software is complete, the project is complete.

In contrast, realism is wise in its approach to software development because it strives to do its work with understanding of how things actually are. Difficult is the work to derive a simple solution from a complex problem. It acknowledges that actual boundaries exist: technical, project and human boundaries. It respects these limits. Realistic software development knows clear communication about all aspects of the project is one of the pillars that allows for success. Finally, realism is humble. One's software cannot solve every problem, and it often has unintended consequences. Humility helps us keep our eyes open and makes it possible to be teachable.

May we grow in wisdom and so in effectiveness, and may God prosper the work of our hands and minds to His glory.

Mike McIntosh *is currently doing software project management In Quixtar IT's Information Architecture department. He can be contacted at Mike_McIntosh@Quixtar.com*



4 From what I know of the "transhumanists" or "extropians," they believe through participatory evolution and technology we will be rid of our clumsy, limiting bodies. The Mars Hill Audio Journal addresses cultural issues such as this from the perspective of historic Christianity. On this topic, listen to the interview with Nigel Cameron on the lack of ethical reflection in public policy on technology on Vol. 81 (July/August 2006). www.marshillaudio.org/

5 In comparison, my laptop has a few more zeroes: 2,000,000,000 bytes (2GB) of RAM.

6 What Vince Lombardi said about football also pertains to our craft.

7 I wish I knew where I read this in the past year, but I am indebted to that unknown author for his or her insight.

Software In Its Place

by James Stewart

The purpose to software development may go well beyond additional efficiency in our lives. James Stewart explains some other dimensions and blessings of software creation.

Over the past couple of weeks I've been reviewing a large development project that went significantly over budget. As with so many projects, it seems like the core of the problem was integration: we were migrating from an existing system and there were pieces of that we weren't adequately familiar with; we were building our code on top of an existing tool that had more bugs than we realized; and responsibilities were placed on the UI developers that required them to have a lot more knowledge of the system architecture than was possible.

It is fairly obvious that a Christian involved in software development should be aiming for quality in their work, honesty in their relationships and being who they claim to be. Those are also key components of being a professional and a good human being. If we want to claim something distinctive as Christians working in software development, we should be trying to push beyond those things to something more radical.

Good software does more than simply enable us to perform a task. It helps us to find new ways to approach our work, and as a result can have a profound impact on how we interact with the world. Recently we have seen the rise of so-called web 2.0 applications that open up considerably simpler ways for people to produce and engage with content online. The growth of *flickr.com* coincides with an increase in uptake of digital cameras and makes photography a more socially oriented phenomenon. The development of a tool like *del.icio.us*, which is focused on something as simple as sharing bookmarks, makes researching and sharing information a richer experience. And a website like *etsy.com* allows people to profit from their handicrafts, both financially and with a growing community, and becomes part of a significant cultural shift.

As human beings, creativity and a yearning for community suffuse us. However, as we in the West have become more affluent over recent decades, a strange phenomenon has taken hold where we have pulled away from one another and chosen an easy and isolated pattern of consumption over creation and engagement within community. We see that in the way we choose our vehicles and plan our cities, and also more broadly in how we choose to use the technology available to us. Many people connect the growth of computer technologies with this trend and see computers and the internet as an isolating thing, but properly imagined and carefully put into the appropriate context these technologies have the capacity to deepen, enrich and bring us back to connections we have increasingly let go.

It is perhaps easy to see that potential in the building of start-up companies and the development of new applications, but there is at least some degree to which the same analysis can be brought to the more common work of supporting business needs. One only needs to look at the standard office environment of two decades ago to realize how radically cheaper computers coupled with more powerful contact management and word processing technologies have changed things, let alone the advent of email.

Business software is rarely written to maintain the status quo, and in the engagement of analysts, designers and developers there is the potential to make the working environment more human. We may not always be able to change our financial modeling tools to show the true human and environmental cost of every business decision alongside the financial implications, or change our specifications sheet to explain the ethical decisions behind each stage of the product's lifecycle, but we may be able to free the user up to consider these things because more mundane tasks are taken care of. Or we may simply be able to enable our users to spend more time with their families and less time racing to meet deadlines. And there's always a chance that by making an application programming interface (API) more open or publishing data using certain standards we can enable other people to harness our information in new and exciting ways.

As software developers we need to work to develop an understanding of that power we have to change people's relationship with their work and their environment with each application we are working on. It may be that by simplifying a feature we can save a user time, or it could be that we can use the tools at our disposal to help people learn about the place they are in and the people they are engaged with. Just as a deep understanding of a system's architecture is necessary to build a solid product, so we should take care of our application's place in the broader architecture of human community.

***James Stewart**, freelance developer, now lives in the United Kingdom. He primarily works on web applications using a variety of languages and frameworks, and blogs at http://jystewart.net/process/*

Programming and the Kingdom of God

by Remington Steed

As a recent graduate from Calvin College and one interested in being an IT missionary, I want to share my experience of interning one summer with Wycliffe Bible Translators. Now, certainly, God created us to glorify Him and bear His image always, whether as IT professionals or linguists or artists, whether in a "christian" or "secular" company or workplace. But one article is not enough space to argue that and tell my story. So I leave the former to a better apologist, and I will share what is mine to share.

A few years ago, I spent the summer interning in the small town of Waxhaw, North Carolina. My aunt and uncle had worked there for a number of years with JAARS, the technology support wing of Wycliffe Bible Translators. In the Spring of 2004, with the encouragement of some missionary friends, I decided to spend the summer as a JAARS software development intern, and since my uncle had recently passed away, I hoped to be a blessing to my aunt as well. This small decision had a profound impact both on my "vocation" (my chosen profession) and my "calling" (how I am to glorify God wherever I go).

The Place

At JAARS I discovered a thriving community of local missionaries, former missionaries, future missionaries, and missionaries on furlough. They were everywhere, and each one had amazing stories to tell. I loved listening to the intense, exciting, sometimes danger-filled stories from these folks and the unbelievable ways God stepped into their situations to make Himself known. And it humbled me to hear of the low points in their stories, where they were ready to give up on God because they couldn't see the fruit of their labor. But truly the best thing I received from these missionaries was the challenge to go and do likewise. Their stories made clear the work God was doing, and also the multitude of work left to be done.

There are millions of people out there, in thousands of language groups, who still have no portion of the Bible. They could read a Bible in a trade language, but that would be their second or third language and at best it could teach them information about God, missing the goal of knowing God personally. They need God's Word to speak to their heart so they can really know Him; they need to know He speaks their language. And when these people hear God speak for the first time in the words they know best, incredible change occurs that human power cannot accomplish. Entire tribes have accepted Christ; destructive cultural behaviors, like revenge killings which would have led to tribal extinction, have been wiped out; national governments have recognized the power of God's Word in bringing peace and stability to their lands. You don't have to think very long to create a list of countries that need that kind of change. Just imagine what it will be like when God's Word is preached to every nation.

So the need is great, and the more firsthand missionary stories I heard, the more excited I became to join God's work of building His Kingdom. But there are many ways to build the Kingdom of God: evangelism, church planting, inviting friends to church events, to name just a few. I am a Computer Science major—what can I do to build the Kingdom? I learned many answers to that question during my short time at JAARS.

Remington Steed

The Project

That summer the IT department had ten interns working with about fifty full time staff, and two of us joined the eXtreme Programming team (google "xp" for more info). Why is it called "extreme"? Because this is not your average cubicle-land. One big open room, two tables full of computers and dual monitors, sticky notes of many colors covering the walls in neat categorized columns, toys scattered about. When the first Koosh ball hit my face, I knew good things awaited me there.

If you were to walk into a room like that, wouldn't you expect to find some sort of daycare? That's how it seemed at first, a daycare for programmers, but within a few short days I realized how effective the XP methodology can be. The toys reduce the stress level, the sticky notes keep the various tasks organized and in manageable pieces, and they always work in pairs to help keep focused. Every day there is a brief "touch base" meeting, every two weeks is a longer "goals for the next two weeks" meeting, and when they meet those goals they have ice cream to

celebrate. A fun, exciting, dynamic, and very productive environment.

This software team was, and still is, developing software to aid the work of Bible translators around the world. Think of Microsoft Word on steroids, with specialized formatting for scriptures, supported by a custom keyboarding system that handles very complex scripts which can move right to left, diagonally, wrap in circles, and do other complicated things. Then integrate that program with a suite of other translation tools, like a lexicon of words and cultural notes for the target language, and you get FieldWorks, a very powerful tool for the Bible translator.

The previous collection of software which filled this role (functional but difficult to use) has already doubled the speed of translating a New Testament and, in many cases, simply made it possible to translate. To illustrate the critical need for this software, I present these scenarios: Consider how frustrating it would be to write every word of the New Testament by hand because the software doesn't support your language's complex alphabet. Yes, you might hire someone to help, but it is still slow and prone to error. Or imagine what you would do if the jungle beetles ate the print off of the Book of Mark which you just finished on typewriter. Absurd as this sounds, this was real life for missionaries only thirty years ago.

Today, because of ever-improving technology and support, God's Word is being read, heard, and understood by people from many tribes, many nations, many peoples, and many languages. But if you catch the scriptural reference, you notice something is lacking.

The Purpose

In the book of Revelation, we catch a glimpse of heaven; we see a host of people worshiping before the throne of God, people from every nation and every language. Jesus said in Matthew 24:14, "*And this gospel of the kingdom will be preached in the whole world as a testimony to all nations, and then the end will come.*" That is the completion of the Kingdom of God, and Wycliffe and many other missions have dedicated themselves to that goal. Would a vision statement like that help you write better code? Would it keep you on task, drive you to minimize distractions and use your time wisely? Would it make your job easier? Maybe yes, and maybe no. Whether glorifying God is the mission statement of your company or only yourself, the mission only helps if you make intentional choices towards accomplishing that mission. And this task of intentional living belongs not only to the missionary, but to every follower of Jesus Christ.

I urge you to look beyond the small hard bricks of Christian duty and see the grand temple that God is building

So, what can an IT Professional do to join in building the Kingdom of God? Certainly you can be a model worker, a caring person, and maybe even invite a coworker to a church drama. But I urge you to look beyond the small hard bricks of Christian duty and see the grand temple that God is building, a people for His glory. Let my story be an example of how God has always worked in people's lives. I was seeking God's will, wondering how to use these gifts He entrusted to me, and He revealed a life direction that matched my gifts and desires. That is God's call. You should never wonder if God has called you--all Christians are called to bare the name of Christ to those who need Him. You should simply say "here I am, send me" and go wherever He guides, one willing step at a time. Perhaps your calling is to work in industry as an unashamed follower of Christ who cares more about His reputation than your own. Or perhaps God will call you to leave upper-middle class America and translate His Word for a tribe in Africa. Whatever the case, expect the call to require sacrifice, obedience and trust. You see how God provided an opportunity for me to learn first-hand how my skills can help get God's Word to the nations. Since then, I have been walking slowly, sometimes uncertainly, and willingly in that direction. Of course, I would rather instantaneously be a missionary, but God knows I will need such skills as patience and trust in a job like that. I still don't know my destination, but I am obeying the Lord one step at a time.

What is your mission statement? What is it you seek? What motivates everything you do? Let me leave you with some advice, or if I may be so bold, a command. Hear these words from Matthew chapter 6, verses 31 to 33:

"So don't worry about these things, saying, 'What will we eat? What will we drink? What will we wear?' These things dominate the thoughts of unbelievers, but your heavenly Father already knows all your needs. Seek the Kingdom of God above all else, and live righteously, and he will give you everything you need." (NLT)

Remington Steed *is a 2006 graduate of Calvin with a BA in Computer Science and a minor in Spanish. While he is currently employed by Calvin as the web administrator for The Christian Classics Ethereal Library (www.ccel.org), he has experience in a variety of IT related fields and enjoys doing a little of everything. He feels the Lord calling him to use these skills as an IT support missionary someday, and also hopes his future will include some form of participation in a choir.*

Examining Christian Faith and Software Development

(Responses I can share with students)

by Patrick M. Bailey, MS

What is your response to this question: What is the relationship between software development and Christian faith?

On one exam a student replied "why do you always have to keep asking this question? It has absolutely nothing to do with my faith. I wish professors would stop trying to push it so much."

That response deflated me a bit. That was probably because my faith caught on fire during the past seven years when I realized what it meant to have a relationship with Christ. Coupling that middle-aged enthusiasm with the thrill of gaining an appointment at a Christian college, I was taken aback by that very blunt and very honest reply.

In our curriculum, we certainly emphasize that technology is part of God's creation, and in the web-book for our freshman information technology course, we discuss *the cultural mandate* based on Genesis 1:28[1]. However, the message in that student's comment was clear: we need to do more.

Now I had to ask myself that question: Where is the connection between software development and our faith? Through prayer, I was inspired to approach working professionals involved with software development. This article describes how that was done and presents the many comments gleaned from a questionnaire and phone interviews.

1. The online text is available at http://www.calvin.edu/academic/rit/webBook/

Why we care.

The distinction of a Christian college is an education built on the foundation of Jesus Christ as our Savior. That includes working with students to develop a framework to understand His world. In addition to scripture, there are several academic sources available to nurture that understanding. An example is a collection of papers prepared by faculty at the Calvin Computer Science Department's website[2].

In addition to those resources, I reflected on the substantial literature from secular sources about the software creation process. Much of the professional literature explicitly discusses the critical need for individuals and organizations to have the right attitude to produce quality software. I wanted to take the concept of "attitude" one step further and investigate if faith has an influence on software development. To do this, I enlisted the help of professed Christians who are software developers "working in the field" and invited them to a web-based questionnaire. That was followed by phone interviews.

The Questionnaire

The web-based questionnaire was intended for a Christian audience only[3]. Again, we wanted to hear from Christians willing to share their views with others as the first step in a dialogue between generations. The sidebar "Where They Came From" explains the source of the 80 plus respondents[4].

The survey collected descriptive information about each respondent. This included the number of years they have been in the work force, their work title (software engineer, analyst, manager or other), and the type of organization they were associated with. They were also asked to identify some of their technical skills[5].

Two questions asked the respondent about the relationship between their work and their faith. The first offered four choic-

2. http://www.cs.calvin.edu/p/christian_scholarship

3. The original questionnaire is available at http://www.cs.calvin.edu/sestudy/survey.htm

4. Access control (e.g. use of an authorization number or password) was not built into the webpage. Participants did have to provide some identifying information. One use of that information was to verify if the person was among the groups invited. Even among the invited groups, though, some explicitly said they had a change of heart about being Christian. We sincerely offer our prayers for them, but we did not use their responses. Again, this intention was not to judge, but to share information within a group of people who claim the same faith.

5. Some people responding to the survey misinterpreted the intent to gathering skill set information and wondered (we think tongue in cheek) if the intention was to determine which

Where They Came From

Participants in the questionnaire came from four primary sources. First, Calvin Alumni who either graduated with a degree in computer science or whose record with the Alumni office suggested involvement with software development were solicited. Second, the North American Christian Reformed Church distributed a request to churches to include in Sunday announcements an open invitation to software developers to participate. A third source was the membership of the International Conference on Computing and Mission (ICCM), which is an annual meeting of people with a common interest in computers and mission work. Finally, there were also direct invitations sent to people known by organizers of the study and other participants who had heard of the effort through word of mouth or had stumbled onto the website.

es, and for brevity sake it will be referred to as the faith integration question for the remainder of this article. That question and the possible responses were:

> *What best describes your view of integrating faith with the practice of software engineering?*
>
> 1. *It really does not apply to software engineering directly.*
> 2. *I have adopted a Christian philosophy about design, but coding is impacted very little by Christian issues.*
> 3. *I believe all aspects of software development are influenced by my faith.*
> 4. *Other*

Those who responded with "Other" were provided an entry field to provide a brief description.

Individuals were also requested to *"briefly describe any specific area where your faith has influenced how you develop software."* This required a written response from the participant, and that item will be referred to as the influence comment in this article.

We were also curious about a respondent's motivation to participate. That question and its options were:

> *What best describes the reason for your participation in this study?*
>
> 1. *I believe I have discovered how my faith is part of what I do, and I want to share that.*
> 2. *I have not found a direct connection between my faith and how I develop software, but I am curious if others have.*
> 3. *I am doing this mostly because I was encouraged by others.*
> 4. *Other*

Summary of Questionnaire Response

While the purpose of this article is mostly to present the written and oral comments offered by professionals, a brief look at some numbers from the questionaire is provided.

First, Figure 1 reflects the breakdown of the responses to the *faith integration question*. In order of the responses above, the descriptive labels are 1- Does not apply, 2-Influences design, 3- Applies to all aspects and 4- Other. When consolidating responses 2 and 3, almost two-thirds believe there is some relationship between their faith and the craft of software engineering. Table 1 below lists the specific comments made by those who felt there was a connection but chose "Other" to describe how they saw the relationship.

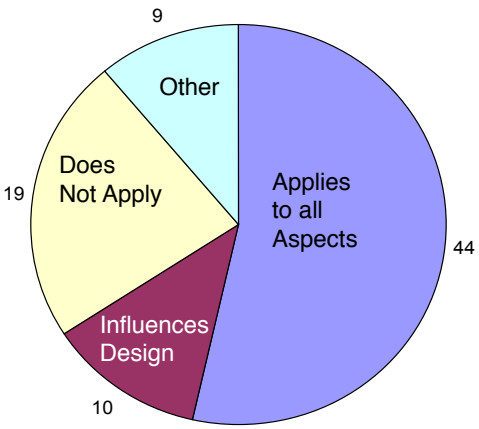


Figure 1
Breakdown of All Responses to Faith Integration Question

Granted, there is a bit of hair splitting among all the options in the faith integration question and much of it hinges on the term "directly." Those who were adamant about faith not having any thing to do with their work usually made an explicit comment when selecting the "other" option.

Further investigation of the numbers begins to hint at some interesting stories. One example is examining the responses to the faith integration question in the context of how people categorized their organizations. When consolidating responses from people who implied their organizations had a Christian base, only 2 out of 27 believed there was no direct applicability between their faith and software development. That proportion grew considerably when the remaining groups were aggregated. There, 16 out of 57 (28%) could not see the link between their faith and software engineering.

Analysis such as this raises more questions. For instance, does a person's environment and associations impact their views? If so, how significant is it, and what lessons are available from that? More analysis on the "numbers" is planned for the future. But for now, this article's focus is on the written and oral comments made by participants.

The replies to the influence comment entry field include some that are humorous, some that provide advice, and others that express a sincere commitment to stewardship and service to others. Some unabashedly stated they do see a direct connection between their faith and creating software. I was personally inspired by many of the thoughts shared.

Table 2 lists all of the responses submitted through the questionaire. In addition to the comments, Table 2 includes some of the profiling information associated with the respondents. The comments are first grouped according to how respondents addressed the faith integration question. Those who said their faith influences all aspects of their work are listed first, followed by those who believe they see the connection through mostly a philosophical level in design. The next group consisted of those participants who believed there was no applicability of

programming languages were more aligned with Christian values. Obviously, it was not. Second, many felt we were implicitly endorsing only open source technology since we did not explicitly mention Microsoft based technology such as .Net. We apologize to Microsoft, but again, our intention was only to determine how much direct involvement a person had in working with software development tools and methods.

faith in creating software. This concludes with comments from those who responded with "Other," a group where some still pointed out the importance of faith in their lives and in the work place.

Within each of these groups, responses were further categorized according to types of organizations respondents belonged to and their role in the software development process. Table 2 also shows their years of experience: less than five years (<5), 5 to 10 years, 11 to 20 years and more than 20 years (>20). Grouping the comments within this hierarchy of categories was done so readers can compare and come to their own conclusion about the implications.

Phone Interviews

Twenty three questionnaire respondents volunteered to participate in a telephone interview. The interviews were designed and carried out by me and three Calvin students: Jared Staal (Information Systems), Michael Bailey (Religion) and Jessica Holtrop (Communication Arts and Sciences).

The approach to interviewing was to ask general questions about the participants' careers and their work places. Our hope was that issues of faith would naturally evolve out of those questions. We started with inquiries about what attracted the individual to software development, what attracted them to their current positions and what considerations they have for future changes. For most, we asked where they saw their work compatible or incompatible with their faith. We were also curious about their views on open source, and finally we asked them to provide general advice to graduating students. Specific notable comments made in the phone interviews are listed in table 3 within the three categories just mentioned: Faith and Career, Open Source and General Advice to Graduates.

When it came to their careers, we found several common threads in the interviews. For instance, we started each interview with a question of how the participant got into the field of information technology. Not one of those interviewed said their primary motivation was income. The implication was a sense of finding their calling. For most in the commercial sector, money became a future planning factor only after they were well into their career.

As we had hoped, the issues of faith were often brought up during the discussion of careers. Several of our callers explained how the integration of their faith and technical skills was directly related to building relationships, or as one explained "it's my form of tent making." Generally, they believe that honing their skills brings them closer to other technicians because of the mutual professional interest. That human perspective is also supported in secular references such as the preface to *Software Craftsmanship: The New Imperative* by Pete McBreen: "Software craftsmanship is important because it … makes us pay attention to the people who do software development."

The trials of office politics was often brought up as well. An interviewee identified lessons from the Old Testament that helped him deal with office politics as he commented:

> I look at it as to how David survived. The royal court was a dangerous place with all sorts of intrigue…God will take care of you. He took care of David and that was when people were using real knives when stabbing you in the back.

The majority of those who worked in the commercial sector were quick to point out the natural conflict with their faith that occurs through common practices in software development. Often, they felt the pressure to promise more than they could deliver because someone in another role had already made an overly optimistic commitment. In other words, there is a nervous silence as developers continue to give external assurances while internally they know the truth. On a more extreme note, more than one developer noted the willingness of companies to encourage practices of padding billable hours to reports, and one reported that he eventually was "let go" because he had made his concerns known.[6]

As we began to look at more technical considerations, I and the students thought that a Christian group might have a strong leaning towards using open source because of its nature, which one respondent described as having the quality of a sharing community. Overall, though, most of those interviewed did not lean in one particular direction. Most said the decisions of what resources to use were based on the needs of the organization and the project.

We concluded each interview with the following request: "If you had an opportunity to give any advice to a graduating student, what would it be?" Again, the final section of Table 3 contains those responses.

Concluding Thoughts

I confess there were moments when I was less than confident about pursuing this project. My incredulity about the project's validity increased somewhat due to comments I received from a secular online discussion group[7] of software professionals I participate in. Some feedback from

6. This was a particularly remarkable story of faith, love and courage. Through prayer, his family supported his decision. Eventually, it turned out that he received a much better paying position.

7. The online group is through Yahoo at groups.yahoo.com. The discussion group is specifically SW-IMPROVE. Several well respected members

that group was harsh based on the premise that religion of any kind has absolutely nothing to do with software development. It was hurtful to feel this sense of rejection from many peers I respect. Also, up to that point, I had not exactly received a flood of volunteers to participate in the project's online questionnaire either.

I turned to prayer at that moment to ask God for guidance, and I believe God answered. Shortly after the inflammatory remarks began on the group discussion board, other voices joined in. This included two regular members of the group who had replied to the online questionnaire in earnest. Then, this group thread was followed by a discussion about faith being part of a person's makeup. Even the harshest commentator agreed, in general, faith could be a consideration for some.

Looking back, that whole incident with the discussion group added more significance to the online webpage questionnaire's question about the motivation for participating. Granted, some commented that they were doing it because they were asked (again, see the side bar "Where They Came From"), but the majority indicated they either wanted to share their experiences or were interested in hearing from others or were interested in sharing and learning. In essence, many Christians in our field want to engage in a dialogue about the role of faith in their work life. I believe this even more after looking at the number of thoughtful comments made through the online questionnaire. This is reinforced when I reflect on the energy and enthusiasm in the voices of the majority of the phone interviews we conducted.

Still, what is unique, though, about this in regards to software development? As one of members of the advisory group to Calvin's Computer Science Department remarked at a presentation that summarized the responses, the "faith link" for technologists is similar to those in other fields. Scripture reminds us of this in several verses such as Colossians 3:17:

> And whatever you do, whether in word or deed, do it all in the name of the Lord Jesus, giving thanks to God the Father through him.

Further, John Edmiston, the CEO and Chairman for Antioch Internet Bible International pointed out in his keynote address at this year's International Conference on Computing and Mission (ICCM) numerous examples of God's presence and influence in everything we do with technology such as the example from Ezekial 1:20:

> Wherever the spirit would go, they would go, and the wheels would rise along with them, because the spirit of the living creatures was in the wheels.

However, I still came back to the question "What about software development?" As I reflected on this, I believe we find something rather significant when considering the agile development movement taking place in software development. Agile development uses metaphors to promote understanding of concepts. The metaphor of craftsman is being applied to developers to discuss career progression. Several leading books on software development and software testing include "craft" or "craftsmanship" in their titles or introductions[8]. Interestingly, the terms "craftsman" or "craftsmen" appear at least 41 times in the New International Version of the Bible.

Given that last metric, the insights from the project's participants and that in my professional career I also have always viewed software development more as a craft, I believe God is telling me I should respond to *the faith* integration question with *"I believe all aspects of software development are influenced by my faith."*

Table 1
Other View Category and Comments to Link

The enterprise wide "system of systems" work I do, on the business to software system interface, is very heavily influenced by my Christian belief system
Trying to make sure I'm honest/ethical in my design, coding, and testing.
Integrating faith into software engineering is mainly manifested in my own personal viewpoint of work ethics. Writing code and design very rarely challenges my faith & ethics. The business environment, however, challenges my faith much more often.
Christian philosophy guides my ethics which guides my work.
As the Director of SW Engineering I promote values of honesty and integrity throughout the department.
Do everything to the glory of God. Attitude toward my job makes a significant difference. Respect of others makes a big difference in a person's ability to relate and get the information to code and design in a successful manner. Living my faith.
Christian morality impacts the "what & why" aspects of the software, but the how seems more guided by professionalism and personal standards of quality than by Christianity.

of our industry participate in that group. I would encourage you to visit this discussion group if you are interested in exchanging ideas on practices for improving software development.

8. Some of the commercial publications using the craft metaphor include Software Testing: A Craftsman's Approach by Paul C. Jorgensen; Code Craft: The Practice of Writing Excellent Code by Pete Goodliffe; Craft of Software Testing: Subsystems Testing Including Object-Based and Object-Oriented Testing by Brian Marick; Software Craftsmanship: The New Imperative by Pete McBreen.

Table 2
Specific Comments about the Link

Influence	Organization Type	Role	Yrs Working	Comment on Link Seen Between Faith and Work in Software Development
All Aspects	Christian Mission Statement	Analyst	>20	As a believer, stewardship of time, money, and other resources is of major concern. As a result, I lean towards a best of breed approach with a strong lean towards the Open Source development models and tools.
All Aspects	Christian Mission Statement	Analyst	11-20	Whether therefore ye eat or drink, or WHATSOEVER ye do, do ALL to the Glory of God! - This means that we need to ensure that we give a quality product, and charge a fair price for work done!
All Aspects	Christian Mission Statement	Developer	>20	Putting the help of the client above all else. In my opinion, our organization does not value profits or sales above this (of course there is a need to meet expenses)
All Aspects	Christian Mission Statement	Developer	>20	The software is built to actually help people, not just to satisfy a requirements document
All Aspects	Christian Mission Statement	Developer	>20	The systems I help develop are used in famine early warning systems, drought prediction, and environmental change studies. My faith directs me to these study areas; once there, it necessitates my getting it right to aid suffering people and creation.
All Aspects	Christian Mission Statement	Developer	>20	I talk to God while I develop software, because He is smarter than I am and has a better plan.
All Aspects	Christian Mission Statement	Developer	11-20	unit testing: if we live in a fallen world, we need to do all we can to fight fallenness
All Aspects	Christian Mission Statement	Developer	<5	My Christian guidance helps me successfully debug complex multi-threaded applications.
All Aspects	Christian Mission Statement	Developer	<5	Ethics in development, and regarding relationships with clients, coworkers, and those I supervise.
All Aspects	Christian Mission Statement	Manager	5-10	We develop web apps for Christians, so there's a very explicit connection between my faith and the software we develop. But my faith also informs how I manage.
All Aspects	Christian Mission Statement	Other Role	11-20	Do all to the glory of God. God has given every talents you have and they should be used to their fullest and for His glory and praise. It is also a witness to those around you.
All Aspects	Ethical Practices	Developer	>20	I believe the abilities and talent that God gave me to write code must be reflect in the quality of the code I write.
All Aspects	Ethical Practices	Developer	11-20	Basically, I apply the "love one another" principal... the quality that I produce directly reflects that which I would like to receive.
All Aspects	Ethical Practices	Developer	5-10	My faith influences how I live my entire life, including my career. I cannot separate my faith from parts of my life. My faith is reflected in my work ethic. I design and develop software to the best of my ability as a witness of the one who owns my life, God.
All Aspects	Ethical Practices	Developer	<5	I believe a Christian approach to software development includes an attitude of servant hood toward all those whom the software will impact, including end-users, testers, maintainers, and fellow developers
All Aspects	Ethical Practices	Developer	<5	Everything that I do at work is motivated by my gratitude for my God-given talents; I want to use them to glorify Him through my work.
All Aspects	Ethical Practices	Other Role	11-20	Striving for quality, looking to help others and go the extra mile to do so, looking for specific ways to witness with the talents given to me by God
All Aspects	Other	Analyst	11-20	Doing a good quality job, for example not releasing software with many known bugs in order to generate additional business fixing the bugs.

Table 2
Specific Comments about the Link

Influence	Organization Type	Role	Yrs Working	Comment on Link Seen Between Faith and Work in Software Development
All Aspects	Other	Developer	>20	Software engineering is a journey, and as there are many stories of divine lessons learned on journeys throughout the Bible, those lessons can be applied to code
All Aspects	Other	Developer	>20	There are many aspects of network communication, network services, security, privacy, etc that are affected by ones worldview
All Aspects	Other	Manager	11-20	It affects how I view my work to ensure it is of the highest quality to bring glory to God. My faith leads me to be more patient with my team and our testing department. It has given me courage to admit to mistakes and aim for correction.
All Aspects	Other	Other Role	<5	I prayed for my hacks to work, and they did!
All Aspects	Principled Secular	Analyst	11-20	I believe that our time is a very valuable gift. I try to make sure that all the software I develop is user friendly and helps the end user to do their job more efficiently. I am also concerned about protecting the privacy of my users' data.
All Aspects	Principled Secular	Analyst	5-10	I spend more time looking up the meanings of words and reviewing the ordering of data and parts.
All Aspects	Principled Secular	Developer	>20	team dynamics, technology trends, legacy systems, development vs. maintenance, software development life cycle
All Aspects	Principled Secular	Other Role	>20	Software development is a *human* activity. Two important ways my faith guides my work is 1. the desire to glorify God in my specific calling, and 2. how I treat others, given each of us is made in His image. These seem pretty obvious to me, however. I suppose a third way my faith influences my work in software is that I limit the reading I do that is overly positive (or even utopian) about the technology and tend to read and listen to contrarian opinions about the perceived benefits of software (Neil Postman, Steve Talbot and _NetFuture_, the Mars Hill Audio Journal). Such reading gives balance to an overly optimistic discipline.
All Aspects	Supports Christian Values	Developer	>20	It's mostly the attitude of serving others - coworkers, customers; keeping the software "honest" (tax compliance software) and convenient; not being an arrogant programmer - recognizing gifts in everyone
All Aspects	Supports Christian Values	Developer	5-10	Inspiration
All Aspects	Supports Christian Values	Developer	5-10	It has influenced my choice to mostly work with community-focused organizations
All Aspects	Supports Christian Values	Developer	5-10	Open source development and the idea of contributing back to a community. Creating software that builds others up, rather than frustrate/ discourage them.
Christian Philosophy in Design	Christian Mission Statement	Developer	<5	The overall design of features, testing, etc, is influenced by my faith since that is where I interact with people and I define how the software will interact with people. In a purely software context (the architectural design and implementation), the influence is much less. There, I see only the desire to do things "right" (i.e. as opposed to cutting corners that will hurt others/myself later)
Christian Philosophy in Design	Principled Secular	Developer	5-10	Willingness to help others and to be content with my position, as opposed to adopting a dog-eat-dog mentality.
No Direct Applicability	Christian Mission Statement	Developer	11-20	Loving your neighbor means providing them a valuable service, not finding ways to get their money.

Table 2
Specific Comments about the Link

Influence	Organization Type	Role	Yrs Working	Comment on Link Seen Between Faith and Work in Software Development
No Direct Applicability	Other	Developer	5-10	My work ethic has probably been influenced by my faith, but that's about it
No Direct Applicability	Other	Other Role	>20	Certainly my moral upbringing underpins everything I do. I don't quite see the relationship between faith and software, though I can see a relationship between faith and how one lives one's life.
No Direct Applicability	Principled Secular	Developer	11-20	Basically, I try to use the talents God has given me to His glory. This includes work ethic, quality of results, appropriate communication to peers and management, and building up those around me, Christian or otherwise.
No Direct Applicability	Principled Secular	Developer	11-20	Frankly, I don't see any connection between one's faith and software development
No Direct Applicability	Principled Secular	Developer	<5	I put into practice the Golden Rule and other similar commands by writing code that is readable and sufficiently documented for the sake of engineers who will do future maintenance. Since humility is a biblical virtue, I am willing to admit when I am wrong on technical matters or in practical matter in my day-to-day work.
No Direct Applicability	Principled Secular	Developer	<5	It hasn't really impacted how I develop software, but more the way I interact with clients in the consulting world.
Other	Ethical Practices	Developer	5-10	I would refer you to the question about integrating faith & practice of software engineering and I will continue to comment here to prove a point. The business environment is a challenging place; speaking now mainly about the political environment of a for-profit business. I am 32 yrs old and part of "Generation X" in the workplace. This leads to many scenarios where I value Faith & Family above corporate politics and issues (a view not shared by my superiors). In many ways, corporations today run under out-dated management styles and these styles challenge my day-to-day attitude towards my job and thus, I often feel like I am angry/ungrateful to God for the employment he has given me. Often I need to "shake off" that attitude, do my job to the best of my ability, and cling to the God/faith that brought me to where I am today.
Other	Principled Secular	Developer	11-20	ethics and coding...
Other	Principled Secular	Manager	>20	I think this can be summarized by saying I emphasize the need to do what is right. This includes taking ownership for issues that come up and being honest about defects when they are discovered. It also includes creating a working environment for employees that balances the needs of business schedules and objectives with the individuals needs for family and other activities outside of work.
Other	Principled Secular	Other Role	>20	Attitude, relationship with co-workers, respect of ALL employees
Other	Principled Secular	Other Role	>20	My Christian faith engenders a set of ethical principles which I rely on in making design decisions about what software systems should/shouldn't do.
Other	Supports Christian Values	Developer	5-10	Areas of Privacy (email addresses/Credit cards, etc.) extra work is put it to see to it that that information is not (and can not be) exploited, even when such uses would be legal and potentially profitable.

Table 3
Selected Comments from those Interviewed

Comments about integrating Faith with Career
To be honest there are a couple of things where my faith has helped me to navigate. From a strictly software engineering aspect I learned at Calvin College…the primary goal is really the end user. SW is a tool used by many people to make their lives easier…I learned that from Joel Adams. God made us that we are to be rulers over God's earth. SW is not made to control people, and we need to realize that people become scared of computers.
Where it comes into play is how to treat the customer. What algorithm to choose, it doesn't influence. As far as day to day, slinging code, writing software, I don't see it a whole lot…where I do see it in my job is in those more nebulous areas.
There is the creativity link. I see creativity being part of joy. It is certainly an attribute of God. Realizing how explicit you must be, gives you some appreciation for the wisdom and sovereignty of God
I can't say that it has on a daily basis. There have been times where there were two ways of doing something and we took the more difficult because it was more beneficial. IT was a call to the greater good.
For the most, not directly. I think what it does is change your attitude towards how you treat your clients. A lot of times, it's a business where time is money all the time. My faith has impacted how I see the client. I'm a lot quicker to be more accommodating.
I'd say that it hasn't affected it very much, but it has at least influenced me to be fair and honest. I could take the easy way out on some things that would end up causing other people more work in the long run and I don't choose to do that.
Faith says look at the client level. What do they need. Faith says write the best code we can. Coding is more [about you] are crafting something.
I'd like to think that it does. I'm thinking that the amount of time that my faith goes through my head is nowhere where it should be. Hopefully, it influences my personality. In terms of the product, what we make is a service for the user. Hopefully [we have] the service attitude
Faith is more apparent on the personal side. I feel the Lord is really working at patience for me. As far as technology, I believe our faith requires us to apply quality to the work we do. When you want to be the light to the world, it applies to the work you do…you express yourself professionally through your work.
My faith impacts me on the job only indirectly such as when I make a mistake and owning up to it. Not being deceitful.
One of the things is how I strive to treat people. I don't know if you'll find this when you get out in the world. We use to talk about peer pressure. That doesn't go away. Some of that pressure is good, but sometimes it may mean that someone may get hurt when it's not their fault. I strive to remember that people are made in the image of God. Jesus put such a stamp on what it means to be human. My faith asks me to forgive. The workplace can be full of gossip. I think about what does it mean to do my work and do it in excellent ways. I'm called to excellence and helping teams to do that. I continue to wrestle how my faith ought to clearly shape the way I do technology. The thing I concluded that a believer in the world of technology has to come with a contrarian spirit. A lot of people think technology will save everything. Technology is just a tool. There's a utopian spirit out there. Our experience as humans is very checkered.
I have been a believer for a good number of years. I was a believer since I was 11. I haven't had a lot of opportunities in the secular world where I really had to go on my faith. Never asked to directly have to do anything compromising. Certainly my faith has helped me all along.
Had to switch jobs because of integrity issues.
I was let go from a previous position. The official reason was that there was no real work. Someone was running a scheme to pad charges to the customer…there was dead time and that was spread across the customers. The responsibility was put on the individual analyst was through the recording of time and to get rid of the administrative accounts … charged extra. Got to the point where I would charge only the true hours I worked and the manager would "lecture" me. My faith carried me through. Each day is for God. I went in one day and thought "you're going to be fired" today…just so happened his wife had fallen down that day. His wife and mother in law were there and supportive. I realized God was there for me. After two months of praying I got another job…interestingly to avoid wrongful dismissal suits they gave a very good severance package.

Table 3
Selected Comments from those Interviewed

Comments on Open Source
I basically work in a Microsoft shop. It's almost a religious debate. The problem with open source is that the products are as viable and useable as the current community supporting. When certain technologies fall out of favor, the support of those seem to dwindle.
The real issues are more legal in nature. For instance, what is the liability? Will someone sue us? Seems to be the communism socialist thing…. sounds good but doesn't work in the real world. People like to point the finger at someone or someone to call.
A lot of the software tools we use are open source. I like the idea of it. It's not an alternative, but it's a good option.
We use it and don't hesitate to use an OS project that can help our business. We won't use anything that requires us to share our license. The success of the business is paramount.
I'm a big supporter. Worked on the KDE project. Everyday I use open source tools. Why? I think the biggest one is freedom of choice. It's always good to have competition and choice available. People who build proprietary system go a long way to lock you into a system so they push the big sell.
We are using a lot of open source. In the sense that you are more able to help your fellow developer it feels more Christian. If we recall that Jesus was not concerned with money and said to give back to Caesar what is Caesar's (referring to coins was the likeness of Caesar on them), then we could argue that open source supports his views. However, Jesus was not AGAINST people getting paid fairly for their work so I can see the non-open source point of view as well.
I think it's a good idea. I keep my hands in the pot. It will have a hard time long term. I don't see the products as fine tuned and polished and marketable. There are certain niches it will stick long term.
MY answer has changed a bit. Was very strong advocate. Lately, advocate where possible. More interested in open data. Make commitment to application, they can pull their data out. Interesting thing on WEB with match-up on WEB….brings up interesting discussion with storage standards. Good experiences with. Community is very compelling and helpful.
We certainly take advantage of some open source. We use it a lot. For example, we are looking at using hibernate. Generally it is better in terms of documentation. We don't do much in terms of contributing anything.
It harbors community, sense of belonging and contributes as a whole to the field. Everyone has a sense of accomplishment and being, and there are emotional and sociological non-benefits.
When I needed a package and couldn't afford Excel, I thought it's great to have open source. The question I have is "can open source really be used for mission critical applications?" I'm kind of conservative, but that's the way I am.
We don't use open source very much.
Pros and cons. I have not made any decisions at all regarding OS because I'm in a semi-retired position. I'm now director of special projects. I think it's good. You can draw upon the talents and the experience of others. I would say in general I would shy away because you are highly dependent on the technician
I look on it from my background with UNIX/XENIX. Couldn't depend on standard API's. I see some of the similar problems with open source. That's why I went from UNIX to Windows with standard API's and common objects I could depend on and deal with….I'm mainly a database programmer and backend programmer.
"God bless Bill Gates for what he's doing," BUT why pay $500 for an office suite. Ties in well for the missionary needs. Corporate environment may get…There is a stewardship issue.

Table 3
Selected Comments from those Interviewed

General Advice to Graduates
have a mindset of wanting to learn and adapt to new situations. The IT world changes so fast and what they (users) demand changes too. Try to cultivate your skills…there's nothing like experience.
You need technical skills , but in this day and age you need someone who can learn and work well on a team. I can teach the technology. I can't teach someone to be on a good team.
When designing an application, you should not first focus on deadlines and money. Focus on stakeholders and think of how the user would be using it. Mostly the servant attitude.
When working for either a nonprofit or for profit, it's easy to get caught up in day to day duties and loose the sense of the greater purpose. You can loose site of the impact of what you do on people. Always keep that big picture of what you are doing. Keep the big picture up front.
ON the general side of things a love for learning is important to maintain, not just at the outset when everything is going to be new. Spend time to live with new technology. Try to stay on top of it. It will help if you.
Do stuff outside of class. What really gave me a competitive advantage was getting involved in abstraction and CSX. All of that stuff. You find out that employers (at least the once
Make sure that you're yourself in the interview. If you can, choose the job you're going to be happy in.
Pay attention to software engineering fundamentals and study design patterns. My [Perspectives class at Calvin] was a key class. Ask what good and harm technology has caused.
Do everything in your power to use your talents for the kingdom of God. There are tons of opportunities. Donate your time. When you're at a secular business, be an example and remember you're always working for God.
I would encourage them to continue to read the scriptures and conform to them and ask God
I don't want to scare you, but it is a real grind out there. God knows where your next paycheck. You don't always see that hand in front of you, but when you look back you will see it. You've just got to rest in your assurance that God is there with you.
Seek the Lord's guidance. He puts ideas into our heads, thoughts…I used to keep a tablet by my bedside. Hardware, software, and the jobs are important, but your being the right person is more important than the technical skills you.
Don't let the world compromise your ethical standards. The pressure of the world will make you compromise. Look at coding as a craft - like a woodworker. At first you might think of a picnic table – how about a Chippendale? It's beautiful.
First of all, it's got to be the Lord opening up opportunities. IT right now…it's more difficult to open doors, but God will. You will be dealing with people from all over the world. You will be with people who never heard of Jesus. God has opened up an opportunity to interrelate with them. Because of out source, God can use this as an opportunity.

Faith at Work and the Environment

by Jared Staal

Jared Staal participated in the phone interviews to support the study on vocation issues concerning integrating faith and software development. Here, he comments on his observation of the interlinked effects between working environments and an individual's faith.

Personally, I believe it is difficult to classify how faith impacts a software engineer's craft or any other type of information technology (IT) career. During the phone interviews I participated in for the faith and vocation integration study, I noticed some who claimed they found a strong tie between their faith and vocation as software developers were hard pressed to describe that link. However, many of the interviews often converged on the most salient aspect of faith in the work place: the work place itself.

To start, I'm sure many can argue (and many do) that there is really no place for religion in the work place. Acceptance, non-acceptance, restrictions or indifference of an individual's faith is part of the work environment, and it is worth exploring. We were talking to fellow Christians, each at a different stage of life and in their walk with God. Among those we interviewed, a recognizable pattern formed. It was the convergence of an employee's faith and the organization's environment to include business policies, other employees and the general culture.

During the interviews, some individuals described environments where they felt very defensive about their faith. Typically, these involved a secular environment where Christianity was not the prominent faith. In one case it seemed as though the person was starved for fellowship with other believers and needed an outlet to discuss his faith on a regular basis. On the flip side, I encountered many who were happy in both environments. Some preferred a Christian environment if it was available, it wasn't extremely important.

During one of the more memorable calls, we talked with someone who experienced a job threatening dilemma which was to comply with a policy he deemed unethical or lose his job. That individual chose to leave his place of work. However, most choices are rarely that simple. Many probably do struggle with wanting to do the right thing, but don't have the luxury to simply leave their job when confronted with an ethical dilemma. In those situations, how does a Christian try to change questionable practices? If your choices are limited, how do you exist in a place that opposes some of your morals? Is that an opportunity God is placing before us to take a stand and make an impact? Those questions emerged, and as a result of that I have a greater appreciation for the importance of the work environment and its influence on faith.

Further, I am now facing some of the challenges illustrated in those calls. As a recent graduate entering the working world, I am looking at all the loans I have to pay off and how to get them paid off quickly. I'm willing to "sacrifice" paying off my loans quickly in exchange for a place where I'm allowed to flourish as a person who is a Christian, but I'm also more aware that the best environment God has in mind for me may not be predominately Christian. There may be opportunities God puts before me that I would certainly rather choose not to accept, but as we see in countless stories in the Bible, the voice of God calling human kind to His will seldom leads us on a completely cheery path with no bumps on the way or trials to face. Our faith is truly shaped, and we show our true colors as Christians in the decisions we make at the most difficult time.

Jared Staal

I also think it is important to realize your place and calling in any environment. There are always two ways of looking at a situation: It can be either a positive or a negative opportunity. Respect and understanding go a long way as well. This includes respect for other people's lifestyles and beliefs in addition to trying to understand them on some level. I think this is very important in this day and age especially when the world is becoming smaller all the time.

Jared Staal *majored in anthropology at Calvin College and after graduation returned to complete an additional degree in information systems. He is currently employed by Spectrum Hospitals.*