

# SOFTWARE: THE TRUE COMPETITIVE DIFFERENTIATOR

Almost every company uses software these days, but what can we learn from companies using it to their best business advantage? In this Enterprisers Project virtual roundtable, we gathered five IT leaders from a range of industries to discuss new frontiers of software advantage, including how to build a software-centric culture, open new markets with software, and how to use open source as a development accelerator.

## PANELIST PROFILES



**JAY FERRO**

Chief Information Officer  
*American Cancer Society*



**BRYSON KOEHLER**

Chief Information and Technology Officer  
*The Weather Company*



**JOHN MARCANTE**

CIO and Managing Director of  
Vanguard's IT Division  
*Vanguard*



**TOM SODERSTROM**


Chief Technology and Innovation  
Officer, Office of the CIO  
*Jet Propulsion Laboratory*



**CLIFF TAMPLIN**

Consultant and former VP of Technology  
Support and Risk Management  
*Hyatt Hotels Corporation*

**E THE ENTERPRISERS PROJECT (TEP):** Software can be a differentiator at the personal, organizational, or even national level today. At the personal level, the National Public Radio Show Marketplace recently profiled a job opening for maître d' at the Royalton Hotel in Manhattan. The job profile required friendliness, positive presence, the ability to take ownership of a busy room, and Oracle skills. At the organizational level, GE's CMO Linda Boff recently said that in order to market GE as a digital industrial company, they had to build up their software capabilities. This included building a new software center in San Ramon, California, to create, among other things, software-enabled machines that help improve outcomes with customers. **How is your enterprise evolving to be more software centric?**

 **JOHN MARCANTE:** It's funny that you mentioned GE. We were out in California last year and met with Bill Ruh, the CEO of GE Digital, who runs that software group. We've also spent some time with Viv Goldstein-Wiltshire, who runs GE's FastWorks program, which is focused on accelerating new product development, decreasing costs, and increasing customer engagement. Which makes the point, right? Lean principles and software agility is seeping into everything we do.

At Vanguard, we're doing a few things to become more software centric:

**First,** we're dedicating a significant amount of time to educate our leaders about technology. Vanguard wasn't born in Silicon Valley, so our executives aren't trained technology engineers. But anyone driving corporate strategy has to understand what disruptions are coming our way, and technology is the primary disrupter.

“ **TODAY, WE ALL HAVE TO BE AS GOOD AT DEVELOPING SOFTWARE AS TECH COMPANIES.** ”

- JOHN MARCANTE

**Second,** we rotate senior people in and out of technology roles. This helps grow business acumen and technology savviness in our leaders. When you combine those two competencies, you get fertile grounds for innovation. We have also seen a large demand for technology leaders – people who've grown up in the technology world – to lead in different parts of the company.

**Third,** we're driving an organization-wide recognition that our competition is no longer just another financial services company. From a user or client perspective, everything we do gets measured by their consumer experience, whether it's Amazon or Google or Apple. You may have heard that the fastest-growing mutual fund today is at Alibaba in China. They collected \$81 billion in assets from customers in their first nine months, in part because of their technical prowess and reach. Today, we all have to be as good at developing software as tech companies.



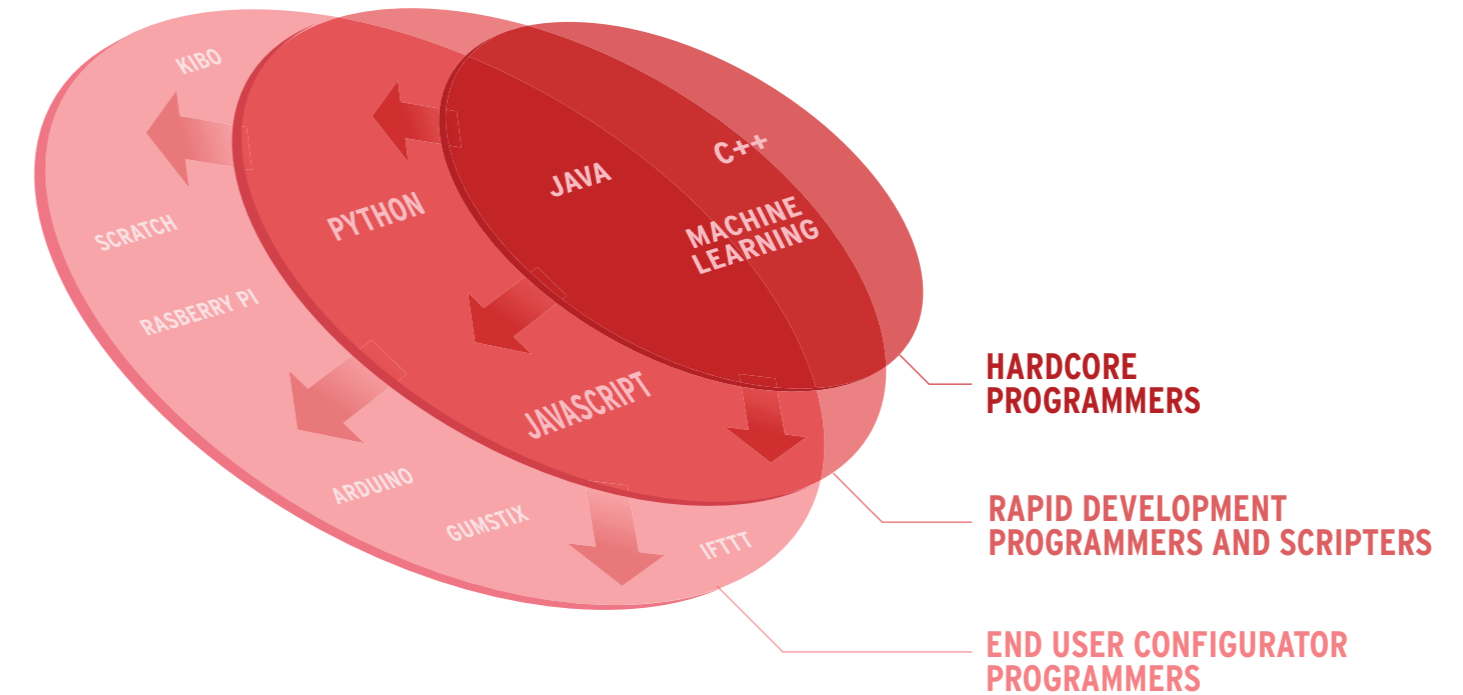
**TOM SODERSTROM:** That's spot on. Software is a revolution and part of that revolution is how programming is changing. Imagine three circles:

- The inner circle is the hardcore programmers. They create the services that everybody else will use. And there are APIs to get into those. So they are using Java, C++, machine learning and so on.
- The second ring is made up of rapid development programmers and scripters. They use the "inner circle" APIs to develop things very quickly. They are the Python programmers, the JavaScripters, the Go people, and so on.
- The outer ring is the end user configurator programmer. And they use things that aren't really languages, things like KIBO programming blocks for ages 4 to 7; Scratch blocks for ages 8 to 16; and then new services like IFTTT, IfThisThenThat, which is really focusing on the Internet of Things market. The idea is that you configure your workplace all the time, just like a programmer would, and you throw in some Raspberry Pis and things like Gumstix, Arduinos – and all of a sudden, you're really getting used to configuring your enterprise. It starts with the rooms, then the devices, then it becomes the enterprise, through things like software-defined networking, and then for us, the software-defined spacecraft.

This is all very unlike the way programming worked even a few years ago. This evolving circle of programming skills coupled with the maker community is leading to a future where everyone will want to customize their own environment...because they can.



**BRYSON KOEHLER:** I think there are multiple ways to look at software-centric evolution. The shift at The Weather Company was asking how we became a platform company. So instead of building all of these custom, one-off bespoke solutions for every idea that came in, how could we get to a place where we were building



See Tom Soderstrom response to left

a common set of capabilities that could be re-used time and again to enable a developer ecosystem for building solutions? Our approach to becoming a software company was to start with the technical foundation of a platform, which in our world was all around data - how you ingest, store, analyze, and distribute data at high volume.

And if you do that, and you build APIs around it, and you make those APIs developer-friendly, and you spend time investing in evangelizing those capabilities to developers, and you create a way for developers to sign up with an access key to get

to the APIs, the theory is that they all would go and build applications. And 40,000 developers have done that to date. So for us, becoming a software company was really around enabling and arming an ecosystem of software developers rather than thinking that we needed to build everything ourselves.



**TEP:** What about this shift has been a challenge?



**BRYSON KOEHLER:** I think that mind-shift change was really hard culturally. One challenge was getting engineers to think agnostically enough to build a platform, when they're used to building siloed spot solutions. You can't build a platform and just have it work for one use case. Amazon doesn't work if the only thing it can support is a healthcare application. Amazon works because you can also run a mobile application, an inventory system, a retail shop – whatever. So pushing the team to constantly change their thinking so that they could become broad enough in the application of their technology for the platform was a real challenge.

The second challenge was recognizing the value in doing that, because building applications, in many cases, is really sexy and cool. You can say, "Here's my mobile app. I put it in the store. You can see it. You can touch it. Isn't that great?" Building back-end applications that can scale and deal with data and APIs – that's engineering with a capital E. And while computer scientists really get excited about that, other people had a hard time recognizing whether that was really adding value. Let's face it: an API is not as sexy as an iOS app.



**JOHN MARCANTE:** I would say that 70 percent of the shift is the change in mindset and culture. We've got to figure out how to test and learn and create minimal viable

“ **LET'S FACE IT: AN API IS NOT AS SEXY AS AN IOS APP.** ”

- BRYSON KOEHLER

products, and to create teams that are empowered to actually solve a business problem versus deliver a feature. That's vitally important. We've begun to rollout the concept of the "Lean Enterprise" in Vanguard for just this purpose.



**JAY FERRO:** I think that as CIOs and as IT organizations are called upon to be more nimble, to respond faster, it changes the conversation about the type of ecosystem we need to respond to those needs. To Bryson's point, we want things on the shelf and ready to go. As an example, when we created Service Match, which uses volunteer-matching logic, our first use case was matching drivers to cancer patients in need. But we knew that the notion of taking a volunteer and matching them to a need is not limited to a ride. It could be a counseling session, or a phone call, or a support group. It can be a lot of different things. Because Service Match is based on one set of APIs, it allows us to start sunseting all our point solutions and begin to put together an ecosystem made up of parts and rearrange them on the fly as needed.



**TEP:** Cliff, have some of your clients faced the issue that it's hard to get people to think agnostically enough at that platform level versus sexy, individual apps?



**CLIFF TAMPLIN:** Yes, there's always an outbreak of that. Though most of the people I deal with now are in travel and hospitality spaces, and they are at a stage that's very much around the guest experience, which is all about that API development where a lot of the back end is done elsewhere. The development of platforms, or ecosystems, as you describe them, is moving apace so that more of the functions are readily accessible to the developers who are creating that frontend experience. And getting the right level of engineers to develop software platforms is increasing as a need as the shortages of highly-skilled computer engineers continues to grow as a problem for us all.

### Opening new markets with software



**TEP:** Can any of you talk about areas where you feel like you've successfully set yourself apart using software, or maybe opened or addressed new markets because of software capability?



**JOHN MARCANTE:** One example is in our institutional business, where we have a history of innovation. Years ago, we were the first company to develop a web-enabled desktop for retirement plan sponsors and company decision-makers. What has really revolutionized this service today is that it's backed by big data solutions that allow plan sponsors to easily understand patterns of investing and saving inside their retirement plan, both in aggregate and across individual employees.



**TEP:** Tom, what has software innovation enabled you to do with recent space missions? Apparently there were thousands of different software-driven events involved in getting the Curiosity rover to land correctly on Mars.



**TOM SODERSTROM:** The software that runs in space is very constrained. It also has to run in radiation-hardened memory and CPUs. That means you have very little space to work in. When we send a spacecraft like Curiosity, there are really three phases to it. There is the launch and cruise, which means the ship has software on it that lets it cruise for about 10 months or so to Mars. Then right before landing, we upload new software, because by the time we send a spacecraft out, the software isn't finished. Once the ship is done cruising, it uses that memory and puts in the landing software. And once it's landed, it replaces the landing software with the driving software. Based on the space limitations, there's no other way.



**TEP:** It's difficult to think of a better way to prioritize an agile development culture than letting people know that if they don't finish and upload the software, the spacecraft is going to crash.



**JOHN MARCANTE:** And I just want to go on record to say that Tom has the coolest job on the planet.




**TOM SODERSTROM:** I certainly get to work with cool people. But I think you're hitting on something big with the agile comment. In our environment, you iterate yourself to perfection as opposed to thinking perfection up front. We're big on agile methodology, and we use it on all software projects, but it has spread beyond software in terms of an overall discussion where IT is permeating the whole enterprise. Now we do all of our projects like an agile project, especially projects where you need to have much bigger risk discussions.




**CLIFF TAMPLIN:** I think the use of the big data to get closer to the customers will be the Holy Grail for a long time in terms of opening up advantage. If you can

understand what your customer wants before they know they want it, that will give any company an edge. And that's what I think everybody's striving for.

**E** **TEP:** Jay, you mentioned Service Match. Does that open a new market for you? Or are there other examples you can name?

 **JAY FERRO:** It certainly has opened up corporate and cause-branding partnership discussions that we didn't know existed when we first came up with the concept. I think where we've made bigger strides with software is in creating a more immersive experience for our donors and our volunteers. The vision is to create one unifying experience through our mobile platforms, all of our online digital platforms, and our call center, so any interaction you have with the American Cancer Society is based on one digital identity that permeates all of it. We deal with physicians, hospitals, donors, patients, caregivers, survivors, and volunteers, and we want to know what's important to you and why you're interacting with us across multiple platforms. That's critical because the days of just the ACS and the United Way as your not-for-profit options are over. There are hundreds of other games in town. Today any new product we come up with, any new research direction we go in, has software technology at its center. Even if that's something as simple as geo-fencing at our events so that we know you're there and can thank you.

**E** **TEP:** Bryson, obviously you've made some fundamental shifts at your company recently with the arrival of IBM. How much has software been a contributor to opening up that relationship and setting yourself apart?

 **BRYSON KOEHLER:** When we started as a media company three and a half years

## **7 WAYS** TO HELP YOUR COMPANY BECOME MORE SOFTWARE FOCUSED

- 1** Take the time to educate your leaders about coming technology disruptions, especially software-based ones.
- 2** Challenge engineers to think agnostically enough to build a software platform, as opposed to building one-off bespoke solutions.
- 3** Develop software in a more agile, test-and-learn way, and create minimal viable products.
- 4** Create teams empowered to solve business problems versus delivering features.
- 5** Iterate to perfection as opposed to thinking about perfection up front.
- 6** Look for ways to securely expose your data sets so partners can build solutions themselves.
- 7** Identify opportunities to outsource commodity software services such as mail so you can address higher classes of problems.

ago we were The Weather Channel and most people knew us for the TV channel. We did have some mobile presence, but it was still really the TV channel that was driving things. And then three and a half years later, IBM bought our company because they saw us as being a big data, technology, and software company. So what we achieved was a complete, 100 percent transition of the business. Even more importantly, in terms of the ongoing growth of our business, it has been important to develop an ecosystem of partners around us that could go and build their own solutions.

What I mean by that is that weather impacts everything, and for us to think that we could go and solve every problem on the planet, and build an application or a solution to go solve that, really isn't possible. If you think about keeping people safe, and allowing for governments to alert their citizens and help prepare and protect them for events that are occurring, I can't stand in the way of taking the data we have, and the abilities that we have, to actually know that there's going to be a problem and that we need to alert people. I don't want to be a bottleneck. So we want to always look for ways to expose the data sets that we have so that others can build solutions themselves. That whole self-service model has been really important to the growth of our business.

### **Software differentiators vs. outsource-able commodities**



**TEP:** John, based on what you said about your true competitors these days being consumers and FinTech companies, how do you determine where software is going to be a source of differentiation, or where it's something you can just outsource? For instance, are there areas where you've stopped building or running software for operations?



**JOHN MARCANTE:** I think these decision points are interesting because they force you to decide what's core to your business. So think about all the algorithms we use for trading and advice. These are not things we would give away. Projects that need tight collaboration are also hard to have someone else deliver for you. Today roughly three quarters of our work is done by IT employees. We are making a commitment to those employees that we will remain a state-of-the-art technology organization and that we value them and the work they do.



**CLIFF TAMPLIN:** The thought about getting rid of commodity services so that you can free your scarce resources for things that will add value to your business is key. Running an email system tends not to be adding a lot of value to a company. But if you can take people that were working on that and refocus them, re-task them to things that will enable the company to move forward, that's absolutely a very good strategy.



**BRYSON KOEHLER:** I don't have anything where I've just done a "lift and shift," taken something that I have running and given it to some outsource provider so they can now run it for me. So for instance, I didn't take my Microsoft Exchange platform and give it to somebody. I said, "Let's go to Google Apps and just let Google run the whole thing for me out of the gate." So am I relying on others for lots of core things that help our business go? Absolutely. Did I take them and give them to somebody to run in some sort of outsourced manner? No. That doesn't mean that relying on others to run a complete solution as a cloud-based service isn't good. Those are great ideas because they enable us to stay focused on what our core businesses are.



**JAY FERRO:** I agree with Cliff and John. I don't know that you ever can just throw something over a wall at a provider and not really look at it, and hope that it's running, but I do think you want to move to a higher class of problem and a higher class of challenge, and offload as much of the commodity stuff as you can. My team wants to solve bigger problems than just the day-to-day care and feeding of hundreds of servers across the country.

“ YOU WANT TO MOVE TO A HIGHER CLASS OF PROBLEM AND A HIGHER CLASS OF CHALLENGE, AND OFFLOAD AS MUCH OF THE COMMODITY STUFF AS YOU CAN. ”  
- JAY FERRO



**JOHN MARCANTE:** I think no matter what you do, you can never put yourself in a place where you're outsourcing your accountability when it comes to IP, data governance, security and privacy. It's also true that by thinking through what you're passionate about and doing an honest assessment of what you can do better than anyone else should help determine the competitive advantage you should own. The only thing I would say on both of these models is that on the Software-as-a-Service side, don't customize. Try to work with what you've got. This gets IT organizations and businesses in trouble.



**TOM SODERSTROM:** That's right. I agree.



**JOHN MARCANTE:** Also don't forget that there are a lot of plug-and-play capabilities that exist in the world of API communities that can accelerate your own custom development.

### Open source as software development accelerator



**TEP:** A few of you have mentioned open source tools as an advantage in driving software-centric enterprises. Can anyone comment on how open has been a benefit?



**BRYSON KOEHLER:** Open source to me is the critical foundational component of what we do. You have to have really strong architectural governance and control when you use open source, but if you have that and you use the open source technologies appropriately, it can radically accelerate your time to market. We stay very agnostic to the open source tools that we select and the modules and libraries we put into our code, so that we can easily remove them and replace them. Because open source projects come and go, things that you think are going to be great end up dying off as soon as the next great thing comes along, so it kind of forces you to be agile because the open source community is changing so rapidly. But that's okay, because I have found that it is changing at the same course and speed that my needs are changing. So we're not being forced to switch stuff out; we can always stay with what we have. But as our needs keep changing, the community continues to change along with us, and it forces us to have an architecture that never allows us to get too locked into any one solution. And that's really critical to how we've been successful. So I wouldn't say we only look at open source these days, but I don't recall the last



commercial software package that somebody really investigated to put into the code to solve a core problem.



**CLIFF TAMPLIN:** A question: How much do you have to throw away of your architecture when you change? When are you at the stage when you've built the platform sufficiently flexible that you can keep on developing and building on what you've got?



**BRYSON KOEHLER:** We were fortunate in that we started a complete rebuild three-and-a-half years ago. And I think we have been pretty smart in how we have been able to stay agnostic to a lot of the solutions we have put in. Some of that was caused by my desire to make sure that we stay cloud-agnostic, so as we were building our applications to deploy we knew that we might deploy them today on Amazon, but maybe next quarter we would deploy them on Microsoft Azure or we might go over to Google Compute or IBM SoftLayer or whatever. So making sure that we never chose a product that didn't enable cloud agnosticism was one of our core architecture tenets. That has the trickle-down impact with the teams of making sure that we were never picking something that would lock us into one approach. I never really intended it to have the agile impact that it did. We were doing agile for other reasons, but what I've seen because we've done that and because the open source community is changing and innovating so aggressively, is that it does have this knock-on impact that when something new comes out. So it's easy for us to look at it and move to it because we are constantly pushing the envelope. We'll change our entire ingestion process on 100TB of data a day in a couple of weeks because we think we've found a better way. And it's amazing to me that we're able to see that level of agility. I think a lot of that is due to the fact that we knew we wanted to rely on the open source community for our solutions.



**TEP:** Anything to add on that, Jay?



**JAY FERRO:** We certainly take a look at open source for just about everything that we do. Like Bryson, we create our solutions in a platform-agnostic way so that we do have that flexibility. We've recognized that one day we might want to be on one provider, but then on another. In any case, we don't want to be locked into anything.



**TOM SODERSTROM:** We use open source for many projects. Cloud computing (where our data and computing can be securely acted on) coupled with the latest open source tools creates a powerful recipe, which we use over and over to cook up powerful applications faster than ever. It has also led to a culture of openness where the developers freely and easily share their code. Why do the developers share their code with all the other JPLers? Because they feel it's the right thing to do, it's easy, and JPL benefits greatly as this has generated a many-fold increase in overall productivity.

### Software and mobile: the final frontier?



**TEP:** Final topic for today's discussion. Mobile is obviously a huge issue for everyone as it becomes the final consumption platform, for consumers at least. As people are interacting more with mobile, how do you need to rethink the way software works as well?



**JOHN MARCANTE:** Clearly it's a mobile-first world, at least among the whole millennial generation, which by the way will be the recipient of \$30 trillion in wealth over the next few decades. I think mobile allows you to design applications in an easy,

user-friendly way. A web capability that can run on mobile devices but isn't explicitly designed for them may not be good enough for the younger generation.

Take my daughter as an example. I was driving her back to New York recently, and she was moving some money around and setting up a bank account. Every transaction she made was texted to her phone. Clearly, interacting has become multichannel and heavily focused on the mobile device.



**TOM SODERSTROM:** The way we look at it is that all our employees are actually on travel as soon as they walk away from their desks. So that means that everyone is always mobile, and the difference between inside and outside of our firewall is how you interact with it. Do you have to have a VPN, for example? If you add too many complications, then the user experience disappears. We created something we call an Innovation Experience Center. The idea is to prototype and share different ways you can interact with computing, but it's really all from mobile.



**JAY FERRO:** Mobile four years ago was an afterthought. Now it's absolutely paramount and first with everything. And we're playing catch-up in some of our legacy technologies, but anything new that we roll out is mobile-enabled from the get-go. With our constituents, our customers, our donors – it's just expected. And if you're just now getting that, you're already five years behind. The desktop stuff, the corporate stuff follows easily when you get mobile right.



**BRYSON KOEHLER:** I would say that it's becoming easy now to figure out how you do mobile really well in the United States. But it is still incredibly hard to get mobile right internationally. Because so much of the world is still running 2G, 3G

connectivity, you've got latency concerns. You can't build something for a 4G LTE network and just expect that it's going to work great anywhere in the world. To get mobile done well on a global basis requires a rethink of how your application is built. What partners and third parties you use? Or do you not use any at all as you build your mobile applications? It becomes a really important bit-by-bit, line-by-line conversation.



**CLIFF TAMPLIN:** I completely agree. I would only add that we have to code and build anything in such a way that it can cope with that elasticity of service, both in speed and also in latency. And more often than not, it's actually latency that kills all our good software intentions.

# THE **ENTERPRISE** PROJECT

Learn more about what leading IT executives are saying about  
the future of business and IT on [EnterprisersProject.com](http://EnterprisersProject.com)