Building and Managing Competitive Software Teams

by

David Brian Ward

Overview

Almost every company today is in the software business. Companies like Microsoft and Facebook are native to the software space, but software is a component of just about every product and services offered today. Companies that are not native to the software world are challenged to get up to speed rapidly on the discipline of software development.

Companies of all sizes and types often struggle to manage the complexity, talent acquisition and people management issues while still delivering an aggressive backlog of scope on a limited schedule. Scaling software development teams rapidly to quickly exploit new software technologies is a particular challenge.

For both small and large companies, meeting these challenges involves staffing new, right-skilled project teams quickly using all available means:

- Employees
- Contractors
- Professional services firms
- Staff augmentation firms, etc.

An integration architecture to rapidly absorb new software assets is also a key success factor. Companies of all sizes must master the key tradeoffs, such as technology build/buy/reuse, or choices around infrastructure, the web, mobile, big data, etc.

For larger companies, a pre-defined software integration process synchronized with their existing deployment process is also necessary if schedules are to be kept short, and high business quality assured. This is especially critical when engaging with today's fast-moving startups.

The pace of today's software industry is dizzying. An "app culture" characterized by small-footprint packages with short life-cycles drives the need to recruit new software talent at affordable rates. Software natives and companies new to the field are challenged to keep up the pace and the demand for more and more apps, that need to be developed on shorter and shorter timelines. Software development project management is more critical than ever, and yet there remains no silver-bullet, off-the-shelf process that works for all companies, at all times.

While the Agile approach to software development has become popular it needs to be stripped to its most pragmatic essence:

- Have the smallest, co-located team
- With right-skilled engineers
- That deliver the highest quality software
- At the lowest cost
- In the shortest period of time

Companies that master the many challenges of software development also learn that *there is no process priesthood*: coaching and mentoring is better than a top-down approach for improving software management.

Common Problems

- **Companies are challenged to meet the ever increasing software demand from customers and employees.**

  Everything runs on software or soon will. There seems to be no limit to the number of apps that customers and employees want. Companies face a range of questions for example, does the application need to be mobile or delivered on a website or downloaded like traditional PC software once was? In addition, product life cycles continue to decrease which puts huge demands on software talent to deliver the right app on an almost impossible schedule.

  Meeting this insatiable demand is the fundamental challenge that both software natives and newcomers to the field are facing. And the demands are coming from all sides: from paying customers, from employees, from suppliers, from the media, etc. As the Internet matures and grows into the "Internet of Everything" the pace has become dizzying and even the most experienced players are challenged to keep up.

- **Making the key tradeoffs remains a challenge: these include technology build/buy/reuse choices for infrastructure, web, mobile, big data, etc.**

  We can now assemble solutions from tens of thousands of software components, many of them license- free. Each of those solutions has pluses, minuses and costs. We also have

a choice between physical or virtual infrastructure. Wise build/buy or reuse choices can make or break a software development project. The days of easy outsourcing around an application is over, as virtual platforms such as Salesforce and Workday take prominence. To achieve faster schedules, many firms are, once again, seeing the benefit of cultivating in-house talent. At the same time, technology continues to develop, calling for talent in specialized areas that did not exist yesterday.

Companies are challenged just to be aware of the range of choices, much less making intelligent, informed decisions.

- **Building SaaS business units while maintaining engineering excellence is a challenge.**

Every company needs to learn how to be a software company. Your interaction with customers, with your employees, with suppliers and all of your stakeholders is through software. The term "software as a service" usually refers to companies like Salesforce and Lending Club that only deliver their services via software. However, inside almost every company today there are services delivered by IT to business units or from one business unit to another. Even internally, within corporations, we now see business units that are transforming themselves into software-as-a-service entities.

For example, GE has forged new software entities to provide services for storing big event data on behalf of their different business units, whether it's aircraft tension, diesel locomotives, or some other business. Corporations such as GE are now building business units which are in fact software-as-a-service entities. These entities, within companies that are not native to software, must be built, managed and maintained, while adding value to the core business. Companies are challenged to maintain their core capabilities in other areas, while also perfecting their ability to develop new software.

- **Companies continue to struggle to find the right talent at the right price.**

Companies across the globe are waging a war for software talent. In the Bay Area alone there's a large quantity of venture capital and other forms of investment casing software ideas. Those companies are all competing for software talent at the same time that the traditional corporations are fighting over the same pool.

There is an order of magnitude difference between the best software engineers and all the rest. In some cases the best talent is overseas but finding and reaching that talent is difficult. The political situation in some parts of the world inhibits companies from accessing the very best talent. In general, finding and retaining software talent is among the most common problem companies face today with respect to the software component of the business.

- **In a virtual world, evolving software processes is a hurdle many companies face.**

The software business is constantly developing new ideas about the best way to write and deploy software. Every time there is a fundamental change, either in tools, components, or in

the demand for software, the methods used to create quality software also need to evolve. The biggest trend now is Agile. However Agile refers to about a dozen specific practices that very few companies follow completely because not all of the Agile suite of practices is universally applicable.

The reality is that every company creates a software development process that works within its context. Since there is no one-size-fits-all process, companies are challenged to create processes that are unique to their business, their culture, their workforce, and the software universe at large.

Key Trends

- **All companies must become skilled at building and consuming software.**

  It doesn't matter if your business is digging coal out of the ground or building technology -- your business now runs on software. Your interactions with your customers, suppliers and other stakeholders occur via software. Increasingly, we're seeing companies that never thought about software before confronting the need for entirely new skills across the software spectrum. We don't see this trend ending any time soon.

  Companies are finding the need for product managers or business process analysts who have a deep understanding of end to end business processes in their industry and who are able to give concise, accurate specifications to their software development teams. This requires cross-domain knowledge, the ability to transfer knowledge in one area to an entirely different area. It's like asking someone to be an expert at software development *and* also to have a medical degree. And then a finance degree. And then an advanced math degree. The intellectual labor has to be divided up but the trend is that digitization is demanding companies to increase their skill sets and open themselves to new worlds.

- **We now live in an "app culture"; this means fast paced releases with smaller, self-contained deliverables.**

  The product life cycles of software, and many of the devices that deliver it, have decreased dramatically. The days of huge, multi-featured applications downloaded and installed on local desktops is vanishing. These behemoths are being replaced by much smaller, mobile, distributed apps that fulfill a single function. This change has given rise to the term app culture."

  It means creating deliverables much more quickly. It means shorter timelines on products that must deliver their value quickly within a narrow window of opportunity. The app culture entails moving targets that, nonetheless, must be hit quickly and accurately.

- **Companies now more than ever need to merge architecture, development, data and QA resources on co-located teams.**

The Agile approach posited that if you want to develop software rapidly you need to move away from a functional matrix approach. It advocated bringing all of the key resources necessary for the project together in one, collocated team.

It has been shown that it is extremely difficult to manage software projects with virtual, remote resources or with resources that are matrixed across an organization. When resources to not sit together, when they report to different managers, development becomes far less efficient and timelines lengthen. Advocates for collocation point to the advantage of short lines of communication. The value of collocation, however, is not just in the fact that people can walk over and lean over a desk to talk to a colleague. It's value is related to the social dynamics of high-performing teams.

- **There continues to be a scarcity of talent and it remains expensive.**

It is harder than ever to find the right people who can form the right team -- and yet it is more important than ever to do so. Companies need to find and retain talent in the domains of architecture, programming, data modeling, and testing. And all of this talent needs to be collocated which is a challenge.

For example there is some top talent available in Eastern Europe, but to be most effective a US-based company must move these individuals overseas. This raises immigration issues and other complications that figure into the time required to assemble the team. At present, the need for speed far outweighs the need to keep costs at a minimum. Forming small, high performing teams, and forming them quickly, is what it is all about in the software development industry. Whether it's inside a big company or a small startup, the issue of finding the right talent is the most visible trend.

- **Non-software companies need to learn software product management.**

The whole world is going to run on software. If a process or a company is not already software-dependent it soon will be. Every company must become adept at managing software projects. Whether they have the expertise on staff or whether they contract for it, they generally have a great deal to learn about it. Product management is a key concept because many companies are not necessarily in the software product business. In most cases they're not. Product management is about the forward-looking requirements for software over a significant period of time.

If you define product management in this way, then all companies that rely on software have to learn the disciplines that are unique to software. They have to learn what platform software means, what infrastructure software means, what their change dynamics are over time, etc. Then they must think tactically about how to build and deploy applications or acquire applications. We are now beyond the point of just cutting a deal with a third party to develop the software and trusting that company to take care of us. That approach is becoming increasingly untenable in a competitive corporate environment.

Risks, Opportunities & Skills Required

- ## Major risks in the software domain:

    - Competitive Risk: software needs to have features that are distinct from what direct competitors are offering.
    - Lack of in-house software expertise when needed can hamper or impede the development of the right product at the right time.
    - Poor decision-making around outsourcing vs. building in-house across the entire spectrum of software.
    - Cost is a risk factor in the development of any product, but too many companies are taking a CFO-driven approach to their software where they're handing over the reins of their technology to managers who do not understand the best way to create high quality, competitive software.

- ## Major opportunities for well-managed software development:

    - As companies become increasingly digital they discover entirely new ways of identifying prospective customers and bringing them into their company's customer base.
    - Digitization allows companies to make knowledge-based, fact-based decisions in areas where intuition and judgment ruled in the past.
    - As software becomes the medium through which companies interact with customers, business partners, and suppliers, it opens the opportunity to create significant competitive changes in business ecology that can move enterprises ahead much more rapidly than in the past.
    - In particular, the right software creates an opportunity to improve supply chains, which could lead to new price points which were not achievable without the software component.

- ## Skills required for competitive software :

    - Acquiring and retaining software talent at all levels: architecture, development, data management, operations, etc.
    - Understanding the entire software life cycle. for example its cost characteristics, maintenance, the triggers for the next iteration of the software etc.
    - Understanding the effect of cutting corners in terms of quality and functionality: it's common that software that's been developed to a certain level
    - Ability to respond to an app culture: building fast-paced releases of small, functionally focused applications.
    - Level of functionality and quality but in the rush to get it to market it was released earlier than a company might have hoped; high quality software developers know how to manage getting the right product to market against time and cost constraints.
    - Ability to make wise, evidence-based build/buy/reuse choices.

Best Practices

If your company is being driven-to software, the following 7 Best Practices provide some guidelines for making management decisions in our "software century":

1. Strip "agile" to its essence: the smallest, co-located team of engineers that deliver the highest quality software, at the lowest cost as quickly as possible.
2. Data is different: Data-centric projects are different from software application development and require unique skill sets, mind-sets and training.
3. There are no silver bullets: And there is no one perfect software development process that fits all companies and all deliverables.
4. Learn to integrate nimbly: Larger companies especially need software architectures and development processes that allow them to develop software as fast as smaller companies.
5. You don't need a process priesthood: Coaching is better than top-down management for improving software processes.
6. Software talent is global: Multiple resourcing strategies, often global, are essential for finding the right talent at the right price.
7. Product Management is essential: Good software is built from good requirements clearly specified by business process experts. Blueprints matter.

About The Author

David Brian Ward: Mr. Ward's career spans 32 years of experience as an entrepreneur, executive and consultant, leading technology innovation and implementation at Global 1000 firms and start-ups. With Telegraph Hill, he intends to bring his high performing technology consulting online to technical managers worldwide.
email: David.Ward@thpii.com Web: www.thpii.com