



## **Great Team, Great Software** **Understanding the “People” Factor**

What pops into your head when you hear the word “software”? Many business leaders would reply “computers” or “technology”, which is ironic because software is all about people. Software is conceived by people, built by people, and intended for people to use. Naturally then, the quality of a software application depends on the people who create it—their ingenuity, talent, and ability to work together—and the environment in which they work. This white paper discusses the “people” aspect of software development.

## Introduction

Great software doesn't just happen. It can't be summoned with the push of a button. It can't be made from a recipe and churned out by an assembly line. Great software is the intellectual response of an interdisciplinary team of thoughtful, committed people to a specific design challenge.

In software development, every challenge is different and requires creative yet disciplined thinking to overcome. A highly effective development team can take a client's abstract vision—a unique set of business requirements—and turn it into a working software application that solves the end-users' problem. An ineffective team just makes a mess.

Before choosing a development firm, business leaders should understand that software is the final result of countless decisions made by individual members of a team. Software quality, or the effectiveness of these decisions, ultimately reflects the ability of the team members and the quality of their environment. When you trace great software back to its source, inevitably you find a talented development team that has really “jelled” within a respectful, supportive company culture.

In recognition of this vital human factor, software development is said to involve three primary elements: software, hardware, and *peopleware*. Unfortunately, many IT managers view all three in a similar way. When hiring, they tend to look solely at resumes and costs, treating people as a commodity and failing to properly consider the importance of personality, talent, and individual productivity. It's no wonder, then, that so many software projects fail for people reasons rather than technological reasons. The simple truth that technology is developed *by* people *for* people is often overlooked.

In the words of one expert:

Software, unlike other artifacts of engineering, is exclusively a product of the human intellect; our raw materials are the gray matter inside the human skull. As the size and complexity of software grows and as the demand for higher quality software and shorter development cycles increase, the ability of individual software engineers to work as members of teams, and the ability of team leaders to direct the efforts of team members become more critical to success.<sup>2</sup>

This white paper describes the “people” aspect of software teams and explains why it is the key to building great software.

## Singular Importance of People

It takes people to understand the needs of other people. It also takes people to address those needs with appropriate solutions. More precisely, it takes people who are engaged in their work, who revel in the challenge of transforming ideas into tangible artefacts, to listen to their client and then *will* the right software into existence. Sounds deep, doesn't it? Well it is.

*Software is people intensive. It involves creating ideas where there were none and detailing ideas until you have source code. Software organizations that do well tend to have bright people and provide an environment that lets them do their best.*

- Dwayne Phillips  
Software Project  
Manager's Handbook<sup>1</sup>

*...we now know with certainty that peopleware issues have more impact on software productivity and software quality than any other factor. Since the late 1960s, study after study has found that the productivity of individual programmers with similar levels of experience does indeed vary by a factor of at least 10 to 1...*

- Steve McConnell  
Rapid Development<sup>3</sup>



*The primary assets of a software organization are the skills and abilities of the project managers, the software developers, and other software personnel...Because people are corporate assets, they should be managed to maximize return on investment and not to minimize cost... Organizations that regard people as assets systematically determine the skill levels needed, recruit the best candidates available, ensure that the workers have the processes, procedures, methods, and tools they need, and invest in training and career development activities for the workers.*

- Richard Fairley  
Managing and Leading  
Software Projects<sup>4</sup>

*In knowledge work, success comes entirely from people and the system within which they work. Results are not the point. Developing the people and the system so that together they are capable of achieving successful results is the point.*

- Mary & Tom Poppendieck  
Leading Lean Software  
Development<sup>5</sup>

Members of a software team—engineers, architects, developers, testers, and project managers, to name a few—can do profoundly great things when they are treated with respect. Many organizations pay lip service to this basic business tenet without really understanding it, and their lack of authenticity is apparent to the smart people who (briefly) work for them.

Respect is about treating people in a fundamentally moral, just way. It's about being truthful to them, caring for them, understanding their perspective, and valuing them. It's about the big picture. With team performance as the basic yardstick, individuals are rewarded for pitching in and helping rather than penalized for it when their valuable contributions aren't captured in personal performance evaluations.

It's all for one and one for all. It has to be. Why? Because therein lies great software: the pride and joy of all developers and the reasonable expectation of all clients.

People in the software industry, frequently referred to as *knowledge workers*, should not be viewed as grains of sand—distinct yet interchangeable—but rather as jewels to be valued for their individual character. And great knowledge workers are valuable, indeed.

What makes a knowledge worker great? Four things, actually. They are: attitude, aptitude, skills, and environment.

A software development firm has the power to shape and develop these characteristics, which can make a good worker even better—or much worse. This is why management, company culture, work environment, team dynamics, and policies for career development and continuous learning play a crucial role in determining the quality of a finished software product.

If you want to know how well a development firm does software, look at how it does peopleware.

## Management

The role of management in a software development firm is to establish and maintain conditions that allow people to excel in their work, which is, after all, intellectual and creative in nature. This means hiring the best people available, organizing them into effective teams, and motivating them to build great software. Whenever possible, management should enable team members to do self-directed work: then managers should either get out of the way or pitch in, as needed.

To facilitate inspired solutions, managers should use people-centric methodologies that enable their teams to focus, get in synch, and execute projects efficiently. One such framework is Agile development, which employs helpful tools like daily stand-ups, collaborative sprint planning, and retrospectives to ensure team cohesion. Agile recognizes that:

- People build great solutions. Technology is merely a tool.
- Change is inevitable and should be welcomed.



*The only way to design a compelling solution is to really understand the user.*

*It's almost impossible to design something compelling for someone you don't respect or wish to understand.*

- Roger Martin  
The Design of Business<sup>6</sup>

*Software development is a process of continuous discovery and invention. The atmosphere most supportive of that process is one that is relaxed and contemplative.*

*Effective software development requires that developers achieve a level of concentration similar to that of a mathematician or physicist. Can you imagine*

*Albert Einstein sitting at his desk while his manager berates him, "Albert, we need that theory of relativity now! Hurry up!"*

- Steve McConnell  
Software Project Survival Guide<sup>8</sup>

- Inspired people are the most qualified to deal with change. Specifications, project management, and process are no substitute for smart people.

Managers must structure their organization so that information flows to whoever needs it. They should make information sharing the rule—no secrets, hoarding, or silos.

Last but not least, managers should not only solve problems but, more importantly, provide mentoring that develops problem-solving skills in team members.

## Culture

Great software is people-focused. Therefore, to build software that truly works, a development firm must always have end-users firmly in mind. This “people first” attitude should permeate the organization to the extent that respect for people underpins every interaction with colleagues, clients, and end-users of the software.

As an extension of the respect principle, team members should feel empowered to exercise their own judgment, inject their own ideas, and reach objectives in their own way. They should be encouraged to ask for what they need to get the job done—tools, assistance, time, whatever it takes. They will test early and fail fast because they want to solve problems, not live with them or sweep them under the rug.

Team members should operate under the assumption that they will have the resources and support they need to build software they can be proud of. After all, pride of workmanship is a key motivator for people who are doing what they love to do. The end result is inspired software and for this reason, team members should be encouraged to follow their interests.

Essentially, a development firm should have a *culture of possibility* that people can breathe in like fresh air.

## Work Environment

A development firm should strive for a balance of team interaction and personal space for individual creativity. Some aspects of software development are highly collaborative—for example, developing requirements and architecture—while other aspects, like coding and writing documentation, require individual effort and concentration. Open communication and information sharing must be balanced with respect for colleagues' time and workload.

Accordingly, workplace features should include quiet work spaces, open meeting rooms, whiteboards, and software tools like wikis and text messaging. Ideally, the workplace is centrally-located in a city that's appealing to the “creative class” of people who develop software.<sup>7</sup> Working remotely from home should also be available as a secondary option when circumstances require it.



## Teams

Diversity is the name of the game. Team members need to have the right combination of technical and people skills, and a balanced view of the business and technology aspects of their work. They must understand how the details fit into the big picture. Specialists like architects, software engineers, developers, testers, and project managers should also have expertise in other areas of interest, allowing them to pitch in where needed.

Whenever possible, a team should have people with a variety of experience levels so that junior workers can learn from more experienced colleagues and perhaps bring a fresh perspective to familiar topics. Teams with too many junior members can spend too much time reinventing the wheel.

Small development teams work best, and should include just enough people with the skill and experience needed to successfully complete a project. The team must be able to embrace change, adapt to new situations, and learn new methods.

Guided by team strategies that harness individual knowledge and minimize “group think” tendencies, team members are able to create the kind of software that a lone developer can only dream of building. They collaborate, backing each other up and doing what it takes to get the job done. Individual success is directly linked to team success.

## Career Development

A development firm should assist its team members in mapping out a career path and then commit to helping them follow that path. They should be encouraged to develop skills outside their specialty so they can help out in these areas, as needed.

People should be able to slide into roles that align with their interests and work on different projects to gain more diverse experience. They should not be pressured to remain in a specific position just because they have relevant experience and obvious talent for the job. People are most productive and inspired when they are doing work they are passionate about.

## Learning

Treating employees as assets rather than costs means investing in their development. Since the human brain is the “raw material” from which software is built, it’s vitally important for a development firm to promote continuous learning.

Training programs can include skills development, mentoring, and career coaching. Whenever possible, continuous learning activities should be built into ongoing work schedules. Smarter brains make better software.

*A good team is composed of members who ignore specialties, seniorities, and job titles and work together as one to meet the team’s commitments.*

- Kurt Bittner & Ian Spence  
Managing Iterative Software Development Projects<sup>9</sup>

*Software development requires creativity, intelligence, initiative, persistence, and a great degree of internal motivation.*

- Steve McConnell  
Software Project Survival Guide<sup>10</sup>

*Encourage education and self-improvement for everyone. An educated workforce and management is the key to the future.*

- Mary & Tom Poppendieck  
Implementing Lean Software Development<sup>11</sup>



*With people, the things that matter are skill, pride, expertise, confidence, and cooperation.*

- Mary & Tom Poppendieck  
Implementing Lean  
Software Development<sup>12</sup>

*...any organization that's serious about improving productivity should look first to the peopleware issues of motivation, teamwork, staff selection, and training...If you are serious about rapid development, you have to be serious about peopleware issues.*

- Steve McConnell  
Rapid Development<sup>13</sup>

*Trust, teams, and technology in that order. Culture comes first, and then technology. If you reverse them, you're in trouble.*

- Carol Anne Ogdin  
Founder  
Deep Woods Technology<sup>14</sup>

## Consequences of Peopleware Done Poorly

A development firm that treats team members as a commodity will produce second-rate software. Guaranteed. Here are some of the consequences of ignoring the “people” aspect of software teams:

- **Low quality, uninspired software**  
If team members aren't inspired, why should the software be any different?
- **Missed deadlines**  
Managers have difficulty putting it all together when there's poor team cohesion.
- **Lack of project continuity**  
Unhappy workers migrate to competitors who value their contributions, causing a high rate of turnover.
- **Low productivity**  
Even the best people function poorly when they don't enjoy their work.
- **More bugs**  
Workers who are not valued have no incentive to take pride in what they do.

## Benefits

It's not hard to guess what happens when a software development firm has a real respect for people and empowers its development team to do the best work possible. Here are some of the concrete ways that clients benefit from this development philosophy:

- **Great software!**  
Engaged team members do what it takes to faithfully transform an abstract set of business requirements into a great software application. They learn what end-users really need and then build software that works.
- **Deadlines met**  
Team members pitch in where needed to complete the project on schedule.
- **Faster development**  
The best people working in a great environment get more done in less time.
- **Project continuity**  
Happy people have no reason to find work elsewhere, therefore turnover is low.
- **Higher productivity**  
People who are passionate about what they do are more productive.
- **Fewer bugs**  
When they are given the resources they need and the respect they deserve, team members take pride in doing great work.
- **Peace of mind**  
Clients can rest easy, knowing that their software is being developed by a talented team of enthusiastic professionals who love what they do.



These benefits are self-reinforcing. People are happier when they are building great software, which in turn motivates them to try harder and do better work—a virtuous circle.

Remember: *People make software for people.*

Choose a development firm that truly understands and lives by this simple principle.

## References

1. Phillips, Dwayne. *Software Project Manager's Handbook: Principles that Work at Work*, (Hoboken, New Jersey: John Wiley & Sons, 2004), 14.
2. Fairley, Richard E. *Managing and Leading Software Projects*, (Hoboken, New Jersey: John Wiley & Sons, 2009), 449.
3. McConnell, Steve. *Rapid Development: Taming Wild Software Schedules*, (Redmond, Washington: Microsoft Press, 1996), 12.
4. Fairley, *Managing and Leading Software Projects*, 443–44.
5. Poppendieck, Mary and Tom. *Leading Lean Software Development: Results are Not the Point*, (Upper Saddle River, New Jersey: Addison-Wesley, 2010), 198.
6. Martin, Roger. *The Design of Business: Why Design Thinking is the Next Competitive Advantage*, (Boston: Harvard Business Press, 2009), 169.
7. Florida, Richard. *Cities and the Creative Class*, (New York: Routledge, 2005), 35.
8. McConnell, Steve. *Software Project Survival Guide*, (Redmond, Washington: Microsoft Press, 1998), 26.
9. Bittner, Kurt, and Ian Spence. *Managing Iterative Software Development Projects*, (Upper Saddle River, New Jersey: Addison-Wesley, 2007), 132.
10. McConnell, *Software Project Survival Guide*, 25.
11. Poppendieck, Mary and Tom. *Implementing Lean Software Development: From Concept to Cash*, (Upper Saddle River, New Jersey: Addison-Wesley, 2007), 76.
12. Poppendieck, *Implementing Lean Software Development*, 91.
13. McConnell, *Rapid Development*, 13.
14. Neuhauser, Peg, Ray Bender, and Kirk Stromberg. *Culture.com: Building Corporate Culture in the Connected Workplace*, (Toronto: John Wiley & Sons Canada, 2000), 196.





**Architech Solutions** is a Toronto-based technology consulting and software development firm. We design and build powerful, user-centred systems that work. We're agile, disciplined, and passionate about delivering for our clients.

**Architech Solutions**  
3 Church Street  
Suite 602  
Toronto, Ontario  
M5E 1M2

**Phone:** (416) 607-5618  
**Fax:** (416) 352-1768

**To book a free on-site Discovery Workshop led by our team of consultants, e-mail [info@architech.ca](mailto:info@architech.ca) or visit us at [www.architech.ca](http://www.architech.ca)**

