Behavioral Indicators That Agility Is Required

Many companies hear the question asked by frustrated employees, “Why can’t we seem to anything done around here?” Operational organizations seek better processes and approaches by embracing Lean, Six Sigma and benchmarked best practices. Marketing and sales can shift messaging and emphasis to respond to change in the market. When a development organization cannot seem to finish projects or deliver products, there is often merely a renewed commitment to expend greater effort but not really solve core problems in how software development is approached. The lack of change can lead to dysfunctional behavior that grows increasingly destructive.

Dysfunctional behavior indicates major problems in organizations and nowhere is this more evident than in the inter-organizational seam between the business and technology development. This paper will identify some factors that indicate that a significant change in the face-off between the business and development is warranted. Key agile principles that are applicable in producing dramatic and demonstrable improvement in development are then briefly listed. Finally, some anonymous anecdotes provide often humorous context. With apologies to Jeff Foxworthy, “if you see any of the following, you might need agile”.

Growing Dissatisfaction with IT

In 2008, 30% of users were dissatisfied with the rate of IT change in the enterprise, but that figure is expected to climb to 50% in five years

Feature Game

The business may demand an unrealistic set of features knowing that development will only deliver some subset. The strategy is used to drive development but the result is poor quality as quantity is strived for due to lack of guidance from the stakeholders. If development does focus on a narrower set of features, it is by shear luck that they are aligned to the priorities of the stakeholders.

The fundamental mistrust in the relationship tends to be compounded as development subsequently drops features, misses dates or delivers poor quality. Under any of these scenarios, the business will be displeased and less trusting in the capabilities of development moving forward. However, the governance model and expectations are not realistic and establish an inevitable pattern of failure.

DIY

The business may grow so frustrated with development that they take matters into their own hands. Proponents of controlling destiny directly advocate solutions built and owned by the business. A highly ineffective or dysfunctional development organization may indeed be to blame, but there may be broader circumstances that lead to slow progress.

Outside organizations may be better or solely aligned to the interests of the business and can thus focus and deliver more effectively. When development has to balance the needs of a number of constituents, outside organizations can indeed be more effective. Therefore, if development were able to focus with a clear budget and expectations, they might be able to execute as effectively.

Standards and other corporate constraints may be bypassed by the business but these explicit directives impede development. Moreover, the IT organization may then be forced to integrate and support something that diverges radically from their core abilities. Vendor packages may also show tremendous promise to solve part of the overall problem, but the most difficult work is often at the process and subsequent technology interfaces. Savvy leaders understand this but still often fall into the trap of seeking a panacea in a packaged application.

Once again, fundamental mistrust of the ability of development to deliver drives this behavior. While not the ideal path, the business will attempt to "do it themselves" out of frustration and eventual desperation.

Vendor vs. Partner

If development is being treated as a vendor and not a partner, there is no perceived added value to being in-house. The commoditization of the development function is a specific business acknowledgement that technology provides no competitive advantage. In some industries, this is the case but in most growing market segments just the opposite is true with the added pressure that competitive advantage based on technology is more fleeting than ever.

There are a myriad of factors leading to the business adopting a neutral posture to development. First and foremost is lack of alignment. If the business must drive development, cajole teams into delivering, educate the staff and independently monitor results then development is not acting like an engaged partner motivated by the success of the business.

Another key factor is rigidity and the means of interaction. If the posture of development is to obtain a virtual contractual obligation of what it needs to provide without governance mechanisms to adapt to changes in business climate, it is acting no differently than an outside service provider. In this case, the business is actually worse off. There is not a diverse pool of talent shared across multiple clients. Any inefficiency is borne by the business and excess capacity cannot be used elsewhere. Being captive means services levels can stagnate or fall unless they are measured and driven upward, which is seldom the case with development.

Track Record of Failure

Between 1994 and 2006, project success rates doubled. However this was a relatively minor improvement as the rate of failure was 72% as measured by business objectives: late, over budget or never delivered as planned.

The deterioration of the business / development relationship to the point of indifference will seldom result in an overt statement that "development is just another vendor". Rather, the business will explore alternatives and start to shift behavior towards a more arms-length relationship. Signs of this shift include the business looking at packages without consultation, actively seeking acquisitions of technology rather than a client/customer base and partnering to shift entire processes (and the attendant technology) outside the organization.

Solutions

Certainly a number of agile principles could be applied to ameliorate the situations above. While the list below is not exhaustive, it serves to identify better practices to repair dysfunctional relationships that evolve over time. Moreover, these practices, when coupled with a large agile program, can create sustainable positive change and also yield the ability to continuously improve. It should be noted that an individual practice, while helpful, is far less impactful if not combined into a full program driven as part of organizational change management.

Collaborate

Failure to collaborate is a root cause that produces a ripple effect and requires a number of remediation points. Collaboration is required at a macro level above the individual features and functions in addition to the more tactical coupling of the business to development.

Alignment: The business and development must clearly agree on priorities, business drivers and expectations. This includes a large number of items such as the following.

- Milestones and key business objectives plus the consequences if dates are missed (e.g., regulatory violation, competitive impact, client attrition, revenue loss, cost escalation, customer issues)
- Budget expectations
- What features are required
- When these features are required
- The maturity or robustness required before the business accepts the features
- Stakeholders and their commitment to help drive the overall project

Governance: Problems always arise before major milestones and require remediation. If development if left solely to their own devices, they will make the decisions that make the most sense to them. Ad-hoc decisions may not align to stakeholder needs, either explicit or tacit. Use of iteration or sprint planning meetings provides more frequent touchpoints between the business and development. An explicit product owner is important but must truly be the voice of the customer to maintain alignment

Planning: Intra-release planning is a fundamental tenet of agile approaches and creates the linkage between specific business needs and the activities of development. More importantly, the detailed planning process creates buy-in and forces the business to put political capital at stake.

Iterate

Incrementally demonstrating value is critical in a dynamic business environment. When delivery has such long cycles that late change becomes exceptionally costly or disruptive, the underlying model is not suited to the business. More adaptive approaches are required to enable iterations to facilitate course corrections as opposed to major strategic shifts.

The ability for the business to "exercise the option" to use software as-is, is highly valuable. If a project is 80% complete but cannot be used, it only has future value. However, working software with only partial functionality may be deployed and wrapped with manual processes and workarounds. While not ideal, such a system may be exactly what the business needs based on a new, higher business priority.

Measure

Businesses appreciate measurements, which development tends to be disinclined to produce. When development is delivering everything the business wants, measures may not be as important. However, when the business feels that delivery is not adequate and it cannot measure progress or improvement, it can feel that development does not have the situation under control.

The following are some key metrics that have proven successful in bolstering confidence in development.

Velocity: Measuring and reporting on actual delivery rate is a very powerful tool in demonstrating that development is producing. Naturally, without proper alignment an increasing level of efficiency may be wasted by producing the wrong results. However, this basic measure of performance is an important first step.

Quality: Quality is best measured in the eyes of customer. In addition to raw defects, other measures of quality, sometimes indirect, can be useful and enlightening if surprising results are uncovered. The author favors quarterly surveys of directional indicators such as "is development improving, remaining the same or growing worse and why?"

Delivery vs. committed features: While development can produce more easily measured items such as velocity and defects, the business has a different view on delivery. Velocity and defects are indicators, but the essential concern of the business is implementation of usable features.

Schedule variance: Agile purists may shudder at the thought, but focus on long term schedules and meaningful milestones are critical. Pragmatically, application of agile principles should improve the ability of development to keep pace with the business, not hinder it. The fact that the business will measure performance is a given; development needs to ensure that it is being measured appropriately and should therefore report on this key (non-agile) metric.

By adopting measures that are meaningful to the business, development can create and environment where it is difficult to argue that development is "working". The focus of the dialog with the business may revert to alignment or the throughput to cost ratio which is a much better discussion to be engaged in.

Benchmarks

Given the importance of technology in modern society, is it somewhat troubling that there are very few accurate measures compared to other industries. However, there are some well respected and thoroughly researched data sources that provide insight into the state of affairs.

Quality

Since computers and technology found their way to the mass market, the supporting software has grown in features, robustness, usability but also complexity. Tools for software developers have made them more productive and enabled them to deliver more sophisticated applications. However, quality has not improved. The defects are spread over a larger code base: the number of bugs in newly written code has remained constant at approximately 5 per function point with 15% of defects still found in production[1].

Timeliness

Meeting the twin business objectives of on-time delivery and in-budget cost has improved dramatically in the past decade. However, the nominal numbers make most industries look like models of efficiency. In 1994, only 16% of projects met these objectives; by 2006 this number rose to 35%[2]. This is consistent with reported project failure rates of 72%[3] as measured by one or more factors: late, over budget or never delivered as planned. Outright failure declined in the same period from 31% but stood at 19% or almost one-fifth of all projects[4].

Agile Success

Agile development techniques have yielded a good deal of success across a variety of key measures that are important to the customer. Business stakeholder satisfaction when agile development is used rather than traditional methods is higher in 78% of organizations while 77% of companies reported higher quality[5]. In

2006, companies reported agile project success rates of 75% or more, which is more than double the 2006 traditional average[6].

Why is this critical? In 2008, 30% of users were dissatisfied with the rate of IT change in the enterprise, but that figure is expected to climb to 50% in five years[7]. Reversing this trend must be a focal point for pure business reasons. Adoption of agile has an impact on the bottom line. Costs dropped in 72% of firms using agile while productivity increased in 82% of the cases[8].

Conclusion

Empirical and anecdotal evidence both indicate that traditional software development organizations are not delivering consistent quality on a timely basis within budget. When agile approaches are adopted, the success rate jumps dramatically. If your business unit or firm exhibits dysfunctional behaviors such as those outlined in this paper, it would be wise to examine how agile might be beneficial in attaining realignment and improving performance as measured by the business.

Anecdotes

The author has witnessed the following destructive behaviors across a number of industries. Should any of the situations resemble your organization, it is coincidental but still just as appalling.

Death march without a destination

Corporate IT was building a major new platform to enable one division to "get to the next level". The business leaders continually added extras features to releases to attempt to exert pressure on a development team that averaged 70-80 hours per week in the trailing year. The project was years late as defect remediation and rework consumed an increasingly large percentage of development work. Development productivity dropped as free pizza at night and complimentary movie tickets for families whose parent spent the weekend working failed to motivate. The business leader and development head were both fired after their strategic initiative never materialized.

Comply or die

Regulatory changes required a major system change by a specific date. The business and development belabored the approach and by the time a written specification was produced, development estimated that there was not enough time to complete the work. Despite their estimates, they attempted to complete the work but it failed to pass testing the weekend before the deadline. The business resorted to workarounds and manual processing. The product manager and development manager were both laid off a few months later.

Garbage in, garbage out

A software company was building a trading system based on past experience. Their CTO believed that development should be locked down for a year and based on detailed specifications while the CEO claimed that the company was agile because it was very concerned about the production defects that were impacting clients. Automated testing was composed of a large staff of untrained junior associates running a static set of regression test scripts and reporting errors to developers. Quality issues caused the top three clients to deploy internal solutions and others to evaluate alternatives. The company is growing increasingly unprofitable and is desperately attempting to repurpose the system for other markets including waste management.

Are multiple full circles an endless loop?

A major financial institution uses an internally developed platform to trade complex products. The system has volume constraints, compliance issues and functional gaps. After development failed to deliver a working system, three consulting firms were engaged in succession to gather requirements, each producing virtually the same results. After a number of leaders were terminated, development was given another opportunity to develop without requirements. Both the business representatives and development leaders are no longer with the company and the project was handed back to the largest consulting shop for another try, angering much of the development staff and leading to attrition despite a tighter market in financial services.

Conclusion

Some fairly pronounced dysfunctional behaviors were discussed in this paper and these were drawn from numerous examples in different industries. There are certainly other indicators that the business is not engaged in a constructive relationship with development. Technology and business leaders should ignore neither the signs nor the behaviors but instead act to make development a highly effective contributor to the value chain in the organization. Development teams that do not create value most likely destroy value by sapping resources that could be used on other vital initiatives. Embracing agile principles and instilling the underlying foundation coupled with a strong linkage to the business can ensure that development is a valuable contributor to a dynamic and flourishing organization.

[1] Capers Jones of Software Productivity Research. *The Economist*, 8 March 2008.

[2] The Standish Group. Biennial "CHAOS Report". 2006.

[3] Weiss, Deborah (META Group). *Computer World Today*. 15 October 2004.

[4] The Standish Group. Biennial "CHAOS Report". 2006.

[5] Ambler, Scott. "Agile Adoption Rate Survey", *Dr. Dobb's Journal*, February 2008.

[6] Ambler, Scott. "Agile Adoption Rate Survey", *www.ambysoft.com*.

[7] Gartner Group. *KM World*. September 2008.

[8] Ambler, Scott. "Agile Adoption Rate Survey", *Dr. Dobb's Journal*, February 2008.