1. Introduction

Hardware, software and system product companies routinely use partners for part or all of their new product development, and yet only a few companies are truly competent at managing those partners. The challenge of working with partners is multiplied if we want to take an Agile approach to product development. Using Agile methodologies such as Scrum, Kanban, Dynamic Systems Development Method (DSDM), or Feature Driven Development (FDD) requires an extremely closely knit and empowered team, so we cannot afford an “arm’s length” relationship with any partner working within the team. Before attempting a partnership for employing the principles of Agile development, a product development organization should have achieved excellence in partner relationship management basics.

In this whitepaper, we discuss the Partner Relationship Lifecycle. In order to get the most out of a partner in developing innovative and profitable products, we first must recognize that a firm’s relationship with their partner follows a lifecycle of phases, similar to a product lifecycle. Several different people in any one organization deal with the partner, and the right person or role to lead the partner relationship changes, depending upon the phase of the lifecycle. For a successful relationship, people need to be trained for these roles – competency does not happen by osmosis.

Next, we show how to decide on the roles of professionals in leading the partner relationship through these phases in order to optimize the collaboration and maximize the return on the investment in the partner. We present a tool called the Circle-Dot diagram to clarify which roles in the organization should lead at each phase. In any phase of the relationship lifecycle, there is a high risk of crossed communications leading to unclear expectations or scope creep due to multiple people in both organizations connecting. So we also present a tool called the Partner Communication Map to avoid this confusion.

We then present the best ways to find the right potential partners, evaluate their strengths, weaknesses and match to the prime contractor’s needs, and then to qualify at least one of them. A good tool for evaluation and comparing potential partners is presented as the Partner Qualification or Performance Scoring chart. This tool can and should also be used for conducting periodic reviews of the partner’s overall performance.

Many companies would like to speed-up their product development process and make it more agile in order to meet changing customer expectations and unforeseen competitive threats. Therefore, we advocate that the best way to streamline the process and allow for changing requirements is to use Agile development principles for both software and many kinds of hardware or system products (see John Carter’s blogs, for examples). These principles place even more demands on a tight, collaborative relationship with the partner, and we describe ways to achieve that kind of relationship.

Finally, we observe that this kind of collaboration usually requires a different approach to writing agreements or contracts with the partner, and we have some suggestions about how to formulate those agreements. We also point to some reference cases and summarize our findings and conclusions.
2. Lifecycles of Partner Management

The process of bringing on and managing a partner involves several lifecycle stages, including the search for potential partners, qualification, official engagement, achieving a symbiotic and optimum relationship, leveraging the partnership through the project and across the company, and then possible disenagement, as shown in Figure 1.

![Partnership Lifecycle Phases](image)

The pre-phase for any partner engagement is to **Search** for the best possible partner for the project. Modern internet and communication tools allow fruitful collaboration with partners that might be located anywhere in the world; so searches might be quite involved and extensive, and they are usually going on continuously among the functional engineering and procurement groups.

In **Qualification**, a potential partner is selected and then a process is executed to determine if that partner has the competencies, capabilities, capacity and match to the prime company for a specific job, or possibly for more general purposes. This phase might include a small development project to see how well the partner team works with the prime company’s team.

In the **Engagement** phase, contracts and/or service agreements are written, usually by someone from a procurement or contract management function with checks by the legal departments of both companies. People or teams are selected from both companies to interact with each other, and at least one Statement of Work (SOW) is generated for a project. Communication protocols are established, and the work is begun.

When the first engagement is well underway, the relationship enters the **Symbiosis** phase, in which the teams on both sides attempt to optimize the working relationship for maximum effectiveness and
efficiency. This phase might be very fast if the people are chosen correctly, or may take some time and possibly changes of roles or personnel to get things working well.

Finding, qualifying, and achieving a symbiotic and productive relationship with a partner company is a big investment. The best companies follow-on with a Leverage phase to find additional areas in which they can use the same partner for other valued contributions or other projects. Note in the figure (dotted line) that the usage of a partner may be waxing and waning as projects start, end, or enter different phases. This variation is normal and healthy, providing the prime company with valuable flexibility.

Finally, there may be a Disengagement phase that can occur for a variety of reasons: Perhaps the next set of projects does not require the expertise of the partner, or the prime company decides to bring the capability in-house, or there is a business change or dispute, etc. In most cases, there is some kind of a wind-down period; although in a few cases it may be a very sharp cut-off. Or it may be that there is never a disengagement phase if it is truly a long-term symbiotic relationship. An example is the 20-year partnership between Hewlett-Packard and Canon to co-develop LaserJet printers.

Working closely with a partner is seldom handled by just one person in the prime organization. Much more often there are multiple people involved on both sides, leading to ample opportunities for mixed or crossed communications. The following are some typical roles we can define, assuming a project or program that involves internal resources as well as one or more partners:

- Project Manager (PM): Responsible for the main deliverables, budget and schedule of the overall project
- Partner Procurement Specialist (PRO): Responsible for the qualification, contracting and business management of the partner
- Technical Lead (TL): The TL is the senior technical architect of the product under development – in some companies she doubles as the project manager
- Technical Specialty Lead (TSL): Responsible for the development in one engineering specialty such as electrical, mechanical, chemical, software, etc. TSLs often report to the TL on a system project
- Functional Manager (FM): Manages a group of technical specialists, such as electrical engineers, for example
- Executive Partner Manager (EPM): This person is usually a high-level executive manager (Director or VP of Engineering, for example) who has a side-job of acting as an overall executive contact to one or more key partners
- Legal or Contracting (LC): Responsible to draw up and possibly enforce the legal terms of the contract

The roles and responsibilities of these people vary depending upon the phase of the partner relationship. An excellent way to clarify the roles is to use a Circle-Dot diagram, like the one on Figure 2. In this figure, the roles are printed along the vertical axis and the major process responsibilities, corresponding to the partner lifecycle phases, are written along the horizontal axis. Wherever the roles and responsibilities cross on the matrix, one of three types of circles or dots is drawn or it is left uncovered. A solid circle means that a person in that role has the lead for the responsibility indicated. A hashed circle means that the person in that role is heavily involved in the process, but not in the lead.
An open circle means that the person is informed, but not necessarily involved in any decision regarding the responsibility, and no circle means that the person is not generally involved or even informed.

Figure 2. Circle Dot Diagram.

A best practice is to have just one person or role in the lead for each major responsibility; shared lead situations often result in conflicts. The example in the figure is just that – an example. The participants in your company should get together, list your major process responsibilities, and decide which roles are in the lead, heavily involved, merely informed or not at all involved for each one. It is important to have full agreement and buy-in from everyone concerned.

In the example in the figure (typical for many system development companies), the FMs take the lead in scanning and finding potential partner companies. They often have the best combination of technical depth, awareness of the suppliers, and business maturity to develop the short list of contenders. After that, the leadership of the qualification process is handed-off to the PRO.

The PRO takes the lead to train all employees involved in dealing with the partner on their roles and responsibilities, their accountabilities, and the communication protocols. Early training and role understanding by all team members can avoid much confusion and miscommunication during the project. The next thing the PRO does in this example is to run the Qualification process (a tool for such a process is described in the next section), using the FMs, TL and TSLs to examine the technical capacity and capabilities. The PRO himself looks at the business factors, including size, stability and track record. We recommend that at least two partners be qualified for any critical project, so that there is a back-up in case the chosen partner somehow fails to deliver or something goes wrong with the relationship.

Upon successful qualification, a formal engagement process is undertaken – led again by the PRO. This process involves writing a Master Services Agreement (MSA) with the prime company that may cover
multiple projects, and usually at least one Statement of Work (SOW) for a specific project. The SOW calls out specific deliverables for the project, whereas the MSA covers more general items such as professional fees for time and materials, escalation processes, scope change management, post-project support, and contract terms and conditions. Legal or Contracting people are heavily involved of course to make the contracts as binding as desired, and all of the technical people who will interact with the partner are involved as well to establish the working and communication protocols.

One best practice for establishing communication protocols is shown in Figure 3. The figure shows which role in the prime company should take responsibility for communications with which corresponding role or person in the partner company. In this example, TSL1 might be the lead mechanical engineer who is working with the contractor’s mechanical designer, and perhaps TSL2 is the lead software engineer running that subproject with the partner. The “X” shows a communication path that should be avoided, mainly having someone like a Functional Manager talking independently to the partner’s mechanical designer without coordinating with her own mechanical TSL. These crossed communication lines can result in conflicting expectations and other problems. The Technical Lead should coordinate the technical subprojects, reporting progress and issues to the Project Manager. The major job of the Functional Manager is to assure that the competency and capacity of technical resources are in place, either from partners or internal groups. The Procurement Specialist helps to make sure that the partner is delivering on business and compliance issues, and the Executive Partner Manager works with his counterpart to keep the two companies aligned and address any major relationship issues.

![Figure 3. Partner Communication Map.](image-url)
As Figure 3 shows, the day-to-day interactions with the partner are usually managed by the technical specialty leads, or possibly by the overall technical lead who may be treating the partner resources as regular team members. In addition, the Tech Lead should be monitoring the contractor’s technical performance, including the quality and timeliness of deliverables. A best practice is for the Procurement Specialist to hold a formal review of the partner’s performance periodically (quarterly is typical) to discuss objective measures such as actual vs. predicted quality, cost, and timing of deliverables. We present a useful tool for this kind of review in the next section.

3. Partner Qualification and Performance Evaluation

The most robust way in which to qualify potential partners is to use a formal framework that breaks the qualification down into specific areas, and attach a score along with reasoning for each axis. As an example, suppose that your project is looking for a software development partner who can work well with your fast-track system project. You are searching for a company that uses Agile or Scrum methodology to match your fast-paced team and flexible market requirements. The first thing your Procurement Specialist should do is to meet with the project team to determine the most important characteristics they are looking for in a partner. An example of such characteristics for this case might be:

- **Programming Competency**: Demonstrated ability to program and control the code in a language that you specify
- **Agility**: Experience and track record running Scrum development processes
- **Test and Debug Competency**: How bug-free is their code and how fast do they get there?
- **Documentation**: Completeness of selfocumented modules
- **Cost**: Your total cost of ownership of the modules developed, including development, support, and other costs
- **Business Stability**: Will the partner be there for your company in the long term?

There could be other parameters in addition, or quite different ones. We recommend, however, that you chose a minimum of 5 and a maximum of 12. The reason has to do with choosing a good granularity that can de-emotionalize any discussion, and yet simple enough to see what the overall picture is in a glance. A good way to visualize the data is to use a “radar chart” or “spider chart” as shown in Figure 4.

The scoring for each axis is on a 0 to 5 scale, where:

- 0 = Supplier has no concept
- 1 = Supplier tries, but is not competent (or other negative attribute)
- 2 = Supplier has below average competence
- 3 = Supplier is above the average in competence
- 4 = Supplier is excellent at the competence
- 5 = Supplier is a recognized world leader in this area

Furthermore, we make the decision that if any important parameter is below a score of 2, the supplier or partner is unacceptable or disqualified. Conversely if the contending supplier has all scores above the 3.5 range, they may be considered as a preferred supplier.
In this sample chart showing the results for two potential partners, you can see that company A is very competent and agile, but seems to be extremely costly (and thus a low score on “Cost”), whereas company B is much more reasonable cost-wise but has only mediocre programming capabilities. These kinds of analyses allow the group to make the best choices and decisions.

After a partner has been chosen, the same metric system can be used for periodic performance evaluations and reviews. Each axis should be assigned to one of the people managing the partner for scoring. For example, the Programming Competence and Agility axes might be assigned to the Tech Lead (TL) or Tech Specialty Lead (TSL) managing the day-to-day partner work, the Test and Debug and Documentation axes might be owned by a different TSL or perhaps a Quality manager, and the Cost and Business Stability axes may belong to the Procurement Specialist (PRO).

The PRO should call this team together periodically (quarterly is typical) to do the scoring and document reasons for each score. One fun and quick way to do the evaluation is called “Scoring Poker” (similar to “Planning Poker”): Give each person on the evaluation team playing cards with the numbers 1, 2, 3, 4, and 5. Pick an axis to be evaluated and ask each team member to think about what the partner’s score should be; and then have everyone throw his/her card down on the table face up simultaneously. If there are wide discrepancies, ask the people with the highest and lowest number to explain their thinking, and then do another round of poker. Within two or three rounds, all the cards should be within 1 point of each other and you can just take an average. The PRO (or her designate) should document the reasoning for each score.

![Figure 4. Partner Qualification or Performance Scoring](image_url)
and partners. It sets the expectations very clearly, and lets them know where they stand and what they need to do to improve without emotionalizing or going on “feelings” of individuals. Using the same set of criteria consistently over time allows the supplier to demonstrate improvement and avoids introducing new or undefined expectations.

4. Agile Partner Management and Collaboration

Whether you are developing software products, some combination of software and hardware products, or even pure hardware products there are advantages to be gained by using some or all of the Agile Software Development principles in collaboration with your development partner. These principles are different from the traditional “waterfall” project management methods in that they:

- Empower the overall product development team
- Catalyze innovation (even by the partner)
- Embrace changes in the product definition throughout the development cycle
- Encourage frequent hardware and software prototype intervals
- Tighten the relationship between the developers and the end use cases
- Utilize predictive progress metrics
- Encourage mid-project retrospectives and process improvements

The principles involved in agile software development were first published as the Agile Manifesto in February of 2001. However, many of these principles hold well for any kind of electrical or mechanical product development process. In the twelve Agile Principles below, I have replaced the word software with either product or prototype or both (notice that I needed to make only 3 substitutions – underlined below):

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable prototypes and products.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working prototypes frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working prototypes and products are the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
Of course not all hardware projects can take advantage of short and frequent development cycles if long lead-time items are involved, but many of them can. Rapid prototyping techniques are increasingly available for many types of development, including 3D printing for hardware, fast-turn printed circuit boards for electronics, and even gate-arrays for quickly turning application-specific integrated circuits (ASICs). In addition, simulators and emulators for modeling the behavior of electronics and sensors and other hardware components are getting faster and more accurate all the time.

It is beyond the scope of this paper to cover in detail the methods and best practices of agile development processes such as Scrum methodology for software or for hardware/software systems, but we will write about how partners can work within these frameworks. We will not consider every type of partner, but instead focus upon what is perhaps the most common case wherein a partner company with multiple engineers is asked to develop a sub-component or subset of the prime company’s product. A typical example might be a “smart” appliance that connects through Bluetooth or Wi-Fi to the home network or internet. The prime company might develop the appliance and its control system, but contract with a partner to develop the electronic hardware and software needed to connect to the network. This communication subsystem will be an important, integral part of the product’s value proposition and there is a lot of interaction with the appliances control system; so the collaboration between the teams must be seamless.

In this kind of partnership, most companies have traditionally taken an “arm’s length” approach: They call on one or more development subcontracting companies with a set of specifications and say to them: “Here is the subsystem we need. How fast can you develop it, how much will the development cost, and how much will the completed subassembly cost to manufacture?” This kind of request causes the potential subcontracting company to “pad” their bid with significant extra time and money to cover major risks – both technical risk and risks that the specs will change or “creep”, which they know is the usual case. The people managing the partner don’t have the means or visibility to check the estimates upon which the contractor’s bid is based, so they just accept a bid and hope for the best. Communication may be minimal while both teams are doing their development, and by the time the units are brought together for integration, there are typically a lot of problems and bugs to work out – very often slipping the (padded) schedule and going over the (padded) budget significantly.

A much better approach that will lead to a better, lower-cost product in a faster time is to find a partner with whom you can follow the Agile Principles above; starting with principle number 5. Use a contractor who is willing to dedicate excellent, motivated people who can work as a team with their counterparts in the prime organization. Both the overall team and the contractor’s team should be given the environment and the support they need, and boundaries within which they can make their own decisions with little or no management meddling. Within those boundaries, the subcontracted partner team should be encouraged to innovate – adding to the value proposition of the overall product (principle #11).

Ideally these people would be collocated for best daily interaction (principle #6), but that situation is very often impractical. In collocated software Scrum groups, the daily communication is initiated with a short morning “stand-up” meeting wherein the members report their progress from the previous day and their plan for the new day. A good substitute in the case of remote teams is to have a short, daily web conference covering the same subjects. When the teams are separated by multiple time zones, it is usual for them to have these short, daily web conferences at a time when the working hours overlap.
The next point for this agile co-development is to build working prototypes (or at least simulators) in the shortest possible intervals and very frequently (principle #3), integrating these prototype subsystems at regular intervals. In the example above, the prime company might build an early “breadboard” of the appliance with simulators for the final sensors or physical elements, incorporating a similar early prototype of the communication module from the partner. A “breadboard” can be thought of as a bench or table with all of the functional elements laid-out on the surface to emulate how it will work, not yet paying attention to the final form factor. Subsequent prototypes will be closer and closer to the final form factor and functionality and use fewer and fewer simulators.

Frequent prototype cycles allows the teams to connect more closely with the dynamic needs of the end customer, usually by having a full-time product manager closely involved – or in some cases even actual end users (principle #4). The product manager’s job is to represent the real unmet needs of the end users and build the business case for the product. As subsequent prototypes are tested, it is common for development teams to find that the use case (or customer needs and stories) may change; so the improvement or changes can be incorporated in the next prototype build (principle #2). The progress in meeting the customer needs (and changing needs) can be measured with “Burn-Down Charts” and other Agile techniques; but the best metric is the number, quality and growing usefulness of the prototypes themselves (principle #7).

During the product development, the overall team including the subsystems contractor should conduct team self-reflection reviews; not reviews imposed by management, but retrospectives on how the team could work better together (principle #12). They should ask themselves: “What are we doing well? What could we do better?” And they should possibly do some root-cause analysis on any major disruptive events. These retrospectives will help the team grow and improve in principles 9, 10 and 11.

5. Agile Partner Agreements and Summary

Regarding partnership agreements, agile product development as described above calls for much more trust-based contracting than just a fixed fee for a fixed deliverable. Time and materials contracts can work, but many prime companies fear that they represent a “blank check” for the partner. This fear is understandable, but usually not well-founded because the partner would like to be a long-term contractor and so is motivated not to overcharge. Such contracts often include incentives for early delivery and corresponding penalties for lateness or other problems.

We have found that getting the most from a partner is done using some kind or risk-reward sharing contract. The partner is paid a base amount for time and materials (perhaps just covering their costs) and then rewarded with some percentage of the revenues or profits from the product once it is on the market. In this kind of arrangement, the partner is motivated to contribute his best people and highest standards of innovation to help the product succeed and get to market on time.

In summary, to be truly competent in new product development, almost all companies today must have strong core competencies in partner management. Many companies do well at product lifecycle management, and yet very few put much effort into managing the lifecycles of their partner relationships to get the maximum benefit. Asking a partner to “go away and develop a subcomponent” of your product often extends the development schedule, causes overspending and can lead to animosity due to changing requirements and expectations. Furthermore it takes little advantage of your partner’s ability to contribute to the innovation or value proposition of your offering. And in too many cases, a partner is used for just one project and your team starts over with a new partner on the next
project – thereby wasting the big investment in the first three phases of the Partner Relationship Lifecycle.

We have presented a case for managing partners with excellence, plus we have presented some simple graphical tools and best practices to help with various phases of the partner relationship. Excellence in managing partners is not difficult, but it takes training, practice and striving for continuous improvement.

Managing partners to collaborate in Agile methodology development projects is more difficult, but if the basics above are first mastered and the principles of the Agile Manifesto are followed as we have described, these collaborations can be very successful; yielding valuable products in a short time that meet the customer needs precisely.

Here are some case studies of companies who have uses Agile methods for hardware development:

How We Learned to Built [sic] Hardware, the Agile way
By Francisco Javier Zorzano Mier, Feb. 24, 2014
http://www.visionmobile.com/blog/2014/02/learned-built-hardware-agile-way/

Interview: what hardware developers can learn from software developers
By Jon Stevenson, October 28, 2013
http://blog.grabcad.com/blog/2013/10/28/hardware-learns-from-software/

PhD Thesis from Sweden: The Challenges of Becoming Agile
By Nis Oveson, June 2012
Dr. Oveson presents 7 case studies (anonymously)
http://theinventivestep.net/Ovesen-2012_The-Challenges-of-Becoming-Agile_WEB.pdf

Your comments are welcome! Particularly if you have any experience using Agile methods in a hardware or system product development.