



**Sapient Corporation**

---

Improving Communication between Customers and Developers

# Improving Communication between Customers and Developers

Andy Takats, Sapien, [atakats@sapien.com](mailto:atakats@sapien.com)  
Nathan Brewer, Sapien, [nbrewer@sapien.com](mailto:nbrewer@sapien.com)

## Abstract

*Reducing waste and building a system based on the right requirements are key benefits of agile methods. Agile delivers on this promise thanks to constant business user involvement with developers, and frequent checkpoints of working software to encourage a tight feedback loop between the developers and the end users.*

*Through exposure to complex business problems that our clients present us, however, we have found situations where these traditional agile approaches have left us unprepared. In many cases, our clients simply cannot continuously work with us because of the nature of their business. In others, the complexity of the business problem is such that producing working code takes significantly longer than we would like. In both cases, we still need to make sure business people communicate well with developers, so that we can build the best system possible for our clients.*

*In the context of an actual case study involving a system developed for the U.S. military, we present a sampling of techniques we have developed to address these tough but not uncommon situations.*

## 1. Introduction

Our clients present us with a wide variety of business situations, each essentially a new problem to be solved, with a unique combination of existing software systems, new systems, organizational issues, budget constraints, technology assumptions, and, of course, people. Agile methods are ideal for this environment, where we (Sapien and our clients) must discover much about both the problem and solution domain, and where the factors just mentioned are constantly changing. Within the boundaries of clear long-range targets -- critical to business plan management for our clients -- we remain flexible with short iterations, maximum client collaboration and transparency.

Collaboration between clients and Sapien -- between customer and developer at the lowest level -- is perhaps the most critical aspect of making our clients successful. Based on numerous projects, we have developed a suite of organizational patterns to apply to nearly all our projects so we can deliver as much value as possible. Many are grounded in common sense but may seem counter-intuitive to those familiar with traditional consulting techniques. We present a sampling of these techniques in this paper.

The patterns we have developed find origins in the JAD (Joint Application Development) work pioneered by IBM in the late 1970's. Developed as a better way to collect requirements and resolve issues as early as possible in a software development lifecycle, they may seem ill-suited to the Agile era, where an emphasis on finding all the requirements up front is not relevant. To the contrary, we have found that these patterns truly complement agile methods by establishing a better baseline of requirements and a better customer-developer working relationship. This relationship is strengthened throughout the project lifecycle by continually applying the patterns.

In this paper we present four patterns, each introduced with a real-life scenario related to a project we have conducted for the U.S. Office of Naval Research (ONR). We draw conclusions from the scenario and then extract the pattern.

## 2. Project details

In response to the changes to naval warfare brought about by the events of September 11<sup>th</sup> and the changes in the way the U.S. Navy engages the enemy as well as supporting ongoing operations on land or at sea, the Office of Naval Research is executing an iterative three-year, four-phase plan to develop an entirely new set of naval logistics command and control capabilities that can be rapidly transitioned to the operating forces. The focus is to integrate and aggregate data from various systems to enable operators to view and act on supply and logistical data in a collaborative manner. The goal is to provide a solution for Naval, Joint, and coalition commanders and decision-makers that integrates a supply and logistical component into the common operating picture, provides immediate operational benefit by addressing critical command and control issues, and supports future operations in a "Sea Based" environment [1].

Sapien began working with ONR on this engagement in June, 2004, to transform nebulous high-level requirements into a functional field-testable system that could eventually be deployed to the afloat and ground components of the U.S. Navy around the globe. To address the evolving requirements, together we crafted a nine-release agile program approach with full testable functionality to be available to operational users within three releases. The team has recently completed the

# Improving Communication between Customers and Developers

second phase of the program and released its first prototype software release to ONR in early June, 2005. The first release was delivered on-time and on-budget with no known defects.

In addition to traditional agile patterns (iterations, releases, and a project backlog revisited each iteration), we applied some in house patterns. The principal pattern is Group Solve – a collaborative approach to working as one team. This pattern has a number of subordinate patterns. In this paper we describe Group Solve and three of the subordinate patterns as shown below.

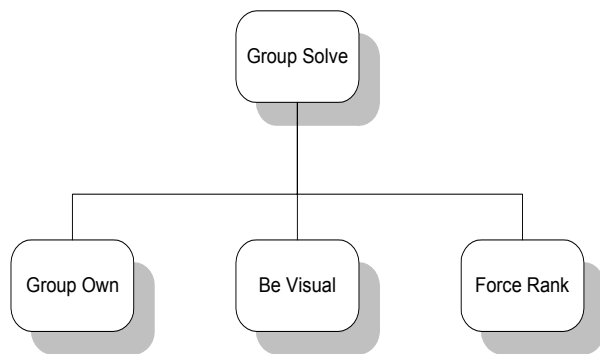


Figure 1. Pattern Hierarchy

### 3. Pattern format

We introduce each pattern with a story to illustrate the forces at play and how the team addressed them in the project for ONR. To make our point about the solution, we first show a more “typical” response to the forces at play that would **not** have had the same effect we wanted to achieve. We summarize each pattern in the format below.

Table 1. Pattern format

Name	A snappy name and a tag-line
Summary	<i>If a certain condition exists, then apply the pattern</i>
Context	The overall context of the situation
Forces at play	The particular forces at play that indicate use of this pattern
Solution	Key features of this pattern to be applied
Don't use this when...	Counter-indications for this pattern (if any exist)

### 4. Pattern: Group Solve

ONR knows that to successfully transition a science and technology application (similar to a research and development effort in the commercial world) into a fully

fielded operational system, they need to get the early and frequent support of the end users. With the project based out of Sapient's Arlington, Virginia (U.S.A.), offices, ONR reached out to various operational, supply, logistics, and systems commands throughout the Navy in order to get a good representation of future potential users of the system and subject matter experts to help define the requirements and to form a team for the three-year program.

The challenge of assembling a diverse group of subject matter experts and users for a given period of time in the Navy is similar to the challenge commercial firms face when embarking on a new system development project: the thought leaders that the development team needs access to are fully engaged in their day-to-day responsibilities and do not have the ability to invest enormous amounts of their time in developing new software systems. A secondary challenge that the team faced was that with the rotational assignment system of the Navy, future users and subject matter experts that would participate in the early phases of the program were unlikely to be in the same role at the end of the three-year development cycle. By using an agile-based approach to conducting a series of workshops with the key thought leaders in the Navy on this effort, the development team was able to ensure that they had just-in-time expertise from the Navy without requiring the workshop participants to devote large amounts of their time to the effort outside of the workshops and user testing checkpoints.

By working with a diverse group of operational users as well as supply and logistics subject matter experts, ONR's executive sponsor greatly increased the chances of the program's success from day one by working with and gaining the buy-in of the various different stakeholders on the project by involving them in every step of development from design to testing.

A typical approach would not have made us successful.

#### 4.1 A typical approach

A typical professional consultant approach to this type of situation is to use a series of interviews. Starting from the project vision supplied by the executive sponsor, the consultant team has the more senior members develop a protocol and then travel (or use telephone) to conduct a series of interviews to better define the high-level requirements. This same approach is used to define the requirements for the initial release. Two-person interview teams divide and conquer the large number of stakeholders, and within a week, perhaps, the initial round

# Improving Communication between Customers and Developers

of conversations is complete. The consultant team then analyzes the findings and defines follow up meetings to resolve issues. During this period, the project materials are documented and made available to the customers via an extranet.

Later, as the project progresses, a steering committee of these stakeholders is formed, and the consultants (including developers) conduct regular meetings with this committee to checkpoint the progress of the development work.

This approach could most likely have “worked” to produce a set of requirements, but it has some real drawbacks: 1) considerable time and money is needed to reconcile stakeholder differences; 2) it becomes very difficult to form a true customer “team” which has many downstream negative impacts; 3) it becomes a major challenge to distribute knowledge to Sapient developers. We examine each in detail below.

At best, resolving conflicts between stakeholders becomes very time-consuming. The supply and logistics communities naturally have very different needs and priorities than the operational community of “ship-drivers” who are commanding carrier strike groups. Overlaps in drivers and requirements will obviously exist, but reconciling the differences requires either dictatorial control (and involvement of the project sponsor in this) or a lengthy two- or three-phase process of identifying conflicts and bringing these parties together (virtually or in person) to resolve the issue. The dictatorial approach is almost always doomed: the customers whose requirements are snubbed will no longer support the initiative. At the worst, resolving conflicts is unsuccessful, and the requirements fail to meet the needs of the users that the system was designed to meet.

By conducting only interviews and effectively isolating stakeholders from each other, it becomes nearly impossible to create a healthy client team. Without a team, differences in opinion are more likely to remain and cause conflict for the project (as the developers must somehow resolve these); communication to the customer group becomes strained, and ongoing decisions must either use a top-down approach or take much longer. Collaboration is strained and Sapient becomes the only real “owner” of the project, encouraging an environment of blame when things go wrong.

A third challenge with this approach is that the Sapient developers have a more difficult time learning the client’s business. Not everyone can attend each interview (and issue resolution session), and hence a fair amount of information learned by the development team would unfortunately be ‘second hand’. However, an alternative

approach does exist.

## 4.2 Our solution

Charting a different approach for the project, Sapient and ONR wove a series of workshops through the three-year program plan. For each workshop, the key client stakeholders would travel to Sapient facilities to work with the Sapient team (leadership and lead developers) in heavily facilitated working sessions designed specifically to produce the most impact for the participants investment of their time. Initial workshops focused on establishing a vision and high-level requirements; follow-on workshops were and will be used to define the detailed requirements prior to beginning work on a release. This approach shifted the ownership for the project to be truly collaborative, and engaged the customers (the U.S. Government) and the developers from the first workshop day in a more productive manner.

Sapient conducted three initial workshops in a six-week period between July and September, 2004. Because the vision that was being developed was so large and the impact was potentially so great on the future of the U.S. Navy, Sapient and ONR felt it was critical to spread work across three workshops to be able to iteratively refine the message and get everyone “bought into” the concept and vision of the overall program as well as to explore the actual functional elements of the system being developed much more thoroughly.

In each of the three workshops, stakeholders from ONR, the Navy’s operating forces, and other organizations, each with valuable expertise and opinions, arrived at Sapient’s Arlington, Virginia, offices. Each workshop consisted of two or three days of intense highly-planned and choreographed sessions. The Sapient workshop team included key project members: the project manager, technical architect, and three developers (who would play track lead roles once development would begin). For the workshops, they took on different roles: lead facilitator, two facilitators, and two note takers. Two Sapient SMEs (in technology and logistics) also participated as part-time facilitators.

Sapient’s facilities are structured to support these workshops. Each office contains one or more “Design Centers” that each contain: a large conference room equipped with projectors connected permanently to two PCs, smaller conference rooms each with a PC for smaller sessions, conference phones as needed, a catering and phone bank area. Nearly all vertical surfaces are whiteboards. Wireless internet access allows guests to check email on workshop breaks. Design Centers share printing and plotting resources.

# Improving Communication between Customers and Developers

Over the course of each workshop, group exercises are balanced with “breakout” sessions where the large group divides and conquers the work to be completed and then regroups to “checkpoint” with the full group. For example, in the second workshop, a key objective was to “develop and prioritize capability sets that fulfill the vision of the NLC2 program.” To reach this objective, the full group first spent time identifying the characteristics of the solution. They then split into two breakout groups to create “capability sets” and then solutions to these sets. In between these parallel working sessions, the groups presented progress being made to each other.

```
>> 0800 - 0830 Breakfast
>> 0830 - 0845 Kickoff
>> 0845 - 0930 Create capability relationship diagram (part II)
>> 0900 - 0945 Break
>> 0945 - 1045 Identify capability sets
>> 1045 - 1100 Break
>> 1100 - 1200 Checkpoint capability sets
>> 1200 - 1300 Lunch
>> 1300 - 1330 FORCEnet brief
>> 1330 - 1400 Break
>> 1400 - 1610
    > Group 1: Develop solutions (Part I)
    > Group 2: Explore key supporting areas
>> 1610 - 1620 Checkpoint and exercise feedback
>> 1620 - 1630 Feedback and close
```

Figure 2. Example agenda – 2<sup>nd</sup> workshop, day 2 of 3

As issues surfaced in discussions, Sapient facilitators encouraged progress in discussions by capturing issues and tabling them until the end of the workshop where all would be reviewed and closed or acted upon. Note takers captured all output from the sessions, allowing facilitators to make rapid progress, and allowing participants to focus on discussions and not on their own capture of the outcomes. As the client team presented the results to each other, they developed a strong sense of ownership of the solution that began to take shape.

The collective project team (Sapient and clients) produced numerous deliverables throughout the workshop. The following deliverables are a sampling from the initial three workshops:

- Program vision – a concise 1-page description of the accomplishments of the program, with specific time horizons.
- Logical Architecture – a 1-page summary visual depiction of the new system and its relationship to existing operational systems
- Executive briefing – (for each workshop) a summary presentation for a wide range of senior and peer audiences within the U.S. military
- Operational Concept Diagram – a diagram showing the interplay between operational

entities (e.g., battleships) and the new system in the context of a military operation

- Capability timeline – a diagram showing the capabilities for the system, prioritized by time, over three years

## 4.3 Team Building

A key benefit of the Group Solve pattern is the formation of strong, committed teams: the customer team, the Sapient team, and the combined, collaborative project team. Trustful relationships between customers and developers set the stage for communication and progress through the course of the entire project, especially important when face-to-face communication would not be possible continuously.

Three factors contributed to the strong sense of team that was created: 1) client solution ownership; 2) cross-functional groups, and 3) a shared struggle.

Very deliberately, the Sapient team positioned themselves mainly as facilitators and note takers from a project delivery point of view and sought to leverage the naval expertise that each workshop participant brought to the table. During the workshop, clients would often present back to one another the results of their individual breakout sessions. In addition to this, at the end of the workshops, the executive sponsor for the program along with all workshop participants were given an executive briefing made up of the materials that they had created during the week. This briefing could then be used by each participant to present back to their home commands about the overall effort. By involving all of the workshop participants in the production of the outputs of the workshop, the participants were eager to share what had been developed with others within their various commands about this new effort.

The Sapient team took care to create cross-functional teams whenever possible. By constructing working groups that consisted of a variety of ranks and roles, workshop participants were able to work in a diverse community of naval thought. Two benefits of this approach surfaced throughout the workshop: one, various participants gained a better understanding of the others’ work; and two, with this new knowledge of each others’ work, they were able to define a solution for the future that would best meet the needs of all participants and not one community of interest over another. New relationships were forged in a cooperative, facilitated environment, where organizational learning made the process enjoyable and new thought patterns resulted in breakthrough ideas.

Entering the workshop, few stakeholders knew each

# Improving Communication between Customers and Developers

other, but under the duress of time-boxed workshop format, they worked hard together to resolve issues and define the future. This shared struggle pushed them to successfully deliver the needed vision and functional guidance to the development team.

During the intense days of the actual workshops, the Sapient team's day started before the clients, and ended long after the clients departed. Each night the Sapient team reflected on the day's progress, discussed and shared information learned, and adjusted the schedule for the following day, often preparing other materials based on new information discovered. By adjusting on the fly to what the group was learning throughout the workshop experience, the Sapient team was able to ensure that the group was free to explore new areas of interest that arose while still marching steadily towards the goals and objectives established for the workshop.

## 4.4 Developers Learn

Developers on the project gained first-hand knowledge of the problem domain. By hearing real stories directly from the naval officers, developers built a rich repository of information that would later give them confidence when directing tracks of development work. Throughout the workshop, everyone on the Sapient team was intensively engaged in learning about the Navy and the problem that this proposed system was meant to address either as a facilitator, a note taker, or as a participant in the workshop.

The pressure of being "on" – as a facilitator or note taker – and being expected to absorb the information quickly and help shape and direct discussions, accelerated the learning of the Sapient team members. In later workshops as the Sapient team increased in size, other developers were able to participate, or, at a minimum, to observe Sapient's unique style of facilitation and problem solving with clients.

Another factor that contributed to rapid developer learning was the choice of communication vehicle between customer and developer. The cross-functional customers (from operators to the Navy's research community) and the cross-functional Sapient team (project managers and technologists) shared information most successfully with visual models. Working sessions emphasized diagrams such as a three-year timeline, logical architecture, and operational concept diagram. See the Be Visual pattern for more details.

## 4.5 Speed

One of the benefits of agile methods is to increase the speed of delivery of working valued software to

customers. The Group Solve approach complements this by very efficiently extracting draft requirements, resolving issues, and making knowledge transfer fast and easy.

By rigorously time-boxing all sessions, and positioning key executive checkpoints closely after each workshop, the Sapient facilitators maintained a keen sense of urgency: both ONR and Sapient teams had a huge incentive to make the most of the time available.

Within each workshop, the Sapient team used an iterative approach to maintain momentum and not get bogged down finding the "perfect" solution to a topic. For example, on day one of the first workshop, the group identified the key challenges the program would be facing; later that day these were revisited and prioritised. On day two, the group revisited the highest priority challenges to explore ways to address them.

The Sapient team worked to remove all obstacles to progress, and to maximize the value of the customers' presence, through extensive preparation, efficient facilitation and note-taking, and "after-hours" structuring of the days output. Every day started further ahead than the previous day's end and this pace fed on itself. At end of the workshop feedback sessions, participants regularly commented on the high level of productivity that the workshops were able to achieve in such a short timeframe.

## 4.6 Summary - the Pattern

Group Solve can be used whenever there is urgency, empowered stakeholders, and a problem to be solved. A lasting relationship is forged between the customers and the developers. Customers and Sapient team members experience the model of very efficient, fun, high-energy communication that they will continue to use throughout the project. Their personal relationships will facilitate the resolution of issues that undoubtedly surfaced during the project.

Table 2: Group Solve Pattern

Name	<b>Group Solve</b> - find a better solution to a problem, faster
Summary	<i>If</i> an urgent, complex problem exists where a diverse group of people need to all play a role in the solution <i>then</i> bring the group together and use a highly facilitated, intense, collaborative approach.
Context	A large project will impact many people across an organization. The knowledge needed to solve it resides in multiple

# Improving Communication between Customers and Developers

	people's heads (typically geographically dispersed). There is limited time to solve the problem. There may or may not be a central "owner" for the solution – as it crosses many organizational boundaries.
Forces at play	<ul style="list-style-type: none"> <li>• Complexity of the problem</li> <li>• Distributed knowledge</li> <li>• Tight timelines</li> </ul>
Solution	<ul style="list-style-type: none"> <li>• Assemble everyone, strong facilitation</li> <li>• Give ownership to the group, shared struggle</li> <li>• Prefer visual over non-visual artifacts</li> <li>• Cross-functional sub-groups</li> <li>• Time box and iterate</li> </ul>
Impact/Results	<ul style="list-style-type: none"> <li>• Momentum, excitement</li> <li>• Breakthrough ideas</li> <li>• Connected customers and developers</li> </ul>
Don't use this when...	<ul style="list-style-type: none"> <li>• Group is not empowered to make decisions</li> </ul>

## 5. Pattern: Group Own

As described in Group Solve, we agreed with ONR to conduct a series of two- to three-day workshops over six weeks, bringing ONR staff, operators and other stakeholders together to set the vision and the high-level requirements for the project. What would be the most effective way to work together?

### 5.1 A typical approach

A traditional consulting approach would treat the assembled cross-functional team as a group of SMEs from whom information is extracted. Knowing the needed outputs, the consultant team creates draft versions up front, reviews these with the SMEs and collects feedback, and then spends off-hours to update the documents. Once satisfied, the consultants send the SMEs home and concentrate their efforts on pleasing the project sponsor and preparing presentation to the senior commanders.

A huge opportunity we would have missed with this approach was the creation of a respectful healthy relationship between customer and developer. Furthermore, we would have lost the opportunity to explore and discover the true needs of the operational forces that were not being stated through more tradition communication modes.

### 5.2 Our solution

Instead of treating our clients solely as a source of

information, we approached the effort as a team where everyone could learn from one another. While the Sapien team would ultimately be held accountable for the project by our executive sponsor at ONR, the entire effort would fail if a sense of shared ownership was not developed with all of the stakeholders involved in the project.

We actively listened and did not presume to know where we would end up at the conclusion of the workshop. We did not listen just for confirmation of our ideas; we listened to learn and uncover new ideas and approaches. We developed a schedule for the workshop, but expected to change course as we learned more. We only planned at a detailed level ½ day at a time, and regrouped to shift course constantly. This created a healthy relationship of respect and a safe learning environment where we, as developers, could learn directly from the operational stakeholders. The customers, feeling listened to, were more apt to be open in their communication to developers. As they were listened to, their respect for the developers increased and they then became more apt to listen to developers' ideas.

A "magnetic force" formed between customer and developer -- customers want to push information to help direct the project and developers want to learn more about the business problem. This force would serve the project throughout the first two phases of the program.

We extended this listening approach further by having the customers present to themselves at all times possible in the workshop setting. Although in most cases the Sapien team would create a draft (of a vision document, for example), the purpose of the draft was to accelerate the process as a jumping off point, to spur discussion, and not to assume Sapien knew the answer. In a breakout session the draft would be revisited by the customer team and ultimately be replaced with completely new material by the workshop participants. A member of the breakout group of customers would then present this back to the full workshop team – the Sapien facilitator would be standing by only for support. This culminated with co-presentation of the workshop results to various groups of interest throughout the Navy with the workshop participants taking the lead and the Sapien team playing a supporting role.

The customers leveraged our subject matter expertise but were in control over the direction of the solution. This sense of "supported ownership" early in the project minimized any sense of "us" and "them" in the project. New leaders emerged throughout the workshop experience that would become invaluable throughout the first two phases of the program.

One other technique in this pattern is the use of the

# Improving Communication between Customers and Developers

customer’s language and customs. The Sapient team prepared before the workshops extensively by learning the customer’s acronyms, terminology, jargon, and general organizational structure. Facilitators learned on the fly as well by working hand-in-hand with workshop participants; note takers did not “translate,” but captured material exactly as described by the customers. This approach reinforced the sense of cooperation and mutual respect, and encouraged information to flow freely between customer and developer.

Table 3: Group Own Pattern

Name	<b>Group Own</b> – let the group own the solution
Summary	<i>If a diverse group of people need to all play a role in the solution of an urgent problem in the short- and long-term then shift the ownership of the problem to the group.</i>
Context	A project that has impacts across a company or organization.
Forces at play	<ul style="list-style-type: none"> <li>• Complex problem</li> <li>• Knowledge to solve is spread across many people</li> <li>• Tight timelines</li> </ul>
Solution	<ul style="list-style-type: none"> <li>• Actively listen to learn – do not assume to know the answers. Be open to change</li> <li>• Customers present to each other (with strong “behind-the-scenes” support) and feel empowered</li> <li>• Speak in the language of the customer</li> </ul>
Impact Results	<ul style="list-style-type: none"> <li>• A truly collaborative customer-developer relationship is formed</li> <li>• Engagement and better ideas</li> <li>• New friends</li> </ul>
Don’t use this when...	<ul style="list-style-type: none"> <li>• Some cultures find it difficult to take ownership for a solution</li> </ul>

## 6. Pattern: Be Visual

With more than a dozen naval subject matter experts assembled for just two or three days of sessions, we needed an effective and very efficient way to reach our objectives and create the needed deliverables.

### 6.1 A typical approach

A typical consultant approach is to use largely textual, detailed versions of deliverables as the focus of sessions with the clients. In this approach the consultant team sits the groups down with paper and pen to begin to list out

capabilities, or to list future system interfaces. A combined group merges and pares down the lists and distributes the resulting list to everyone for review, ending with a fully-detailed list of specifications to be passed to the developers. To create a vision, separate groups write out paragraphs, to be wordsmithed collectively into a final vision. The report to the commanding officers is a thick document of detailed deliverables.

For some participants, this approach would probably have felt comfortable: detail-focused, very concrete, and very complete. But for others, it would have been difficult to engage at such a low level (especially if the terminology was not their own), or to absorb new information effectively based on the sheer volume of data.

### 6.2 Our solution

Instead, at all points in the workshop process (and in the project at large), the Sapient team led with visual models and complemented these later with non-visual models. Visual models engaged everyone more quickly, worked better with outside parties (i.e., for communicating the results of the workshops to senior commanders and other stakeholders), and served as “entry gates” into more complex non-visual models.

The team presented the workshop schedule and approach for three days of activities in a very visual format. This engaged the team and created a mental image that could be referred to constantly.

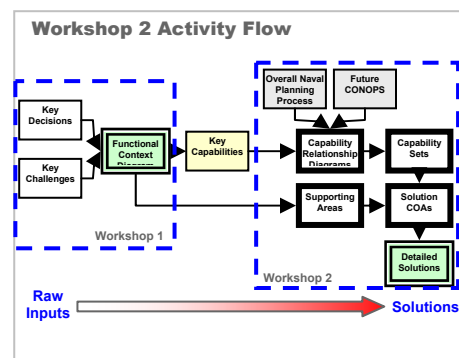


Figure 3. Visual depiction of workshop flow

The team used whiteboards to focus nearly all discussions. By drawing pictures, the team rapidly engaged everyone in discussions in a common language. For example, the team created extensive visual models of capability relationships, as steps in the process of identifying a complete set of capabilities (to be prioritized later in the workshop).

# Improving Communication between Customers and Developers



Figure 4. Visual capture of capability relationships

Another valuable technique in this pattern is the use of a “wall of wonder”: as the workshop progresses, the team regularly posts information on the boards of the workshop. This visible show of progress emphasizes how much the team has accomplished and builds momentum and excitement, which contributes to a happier more productive customer and developer team. This approach can be carried through to the development environment to strengthen the bond between customer and developer.

Visuals makes workshop sessions fun and fast-moving. This energy and momentum carries through the project. Customers and developers can exchange ideas with pictures. A visual “language” can help bridge Customers and Developers as they each learn the others more detailed language – the jargon of military acronyms, or the specifics of a use case specification or a test case.

Table 4: Be Visual Pattern

Name	<b>Be Visual</b> – words alone are boring
Summary	<i>If</i> you need to communicate something that is not 100% known and understood <i>then</i> prefer visual over textual
Context	Varying backgrounds of customer stakeholders and developers.
Forces at play	<ul style="list-style-type: none"> <li>• Different communication and learning styles</li> </ul>
Solution	<ul style="list-style-type: none"> <li>• Use visual artifacts as principal discussion and communication tools: context diagrams, process flows, paper or mocked up prototypes of system flows</li> <li>• Visualize back-end processes such as accounting, to engage stakeholders</li> <li>• Prefer stickies, whiteboards, and projectors over documents on a table</li> <li>• Complement visual artifacts with appropriate textual information (e.g., a system context diagram</li> </ul>

	with a spreadsheet of each system’s details)
Impact/Results	<ul style="list-style-type: none"> <li>• Improved energy, excitement, momentum all around</li> <li>• Better understanding</li> </ul>
Don’t use this when...	<ul style="list-style-type: none"> <li>• There is a lot of low-level detail to communicate since too much visual information can become distracting</li> <li>• Use this approach with caution when working with dispersed teams since using visual patterns can be difficult when everyone is not sitting in one location participating at the same time</li> </ul>

## 7. Pattern: Force Rank a.k.a. “King or Queen for a day”

As with all projects that involve building a system, a major challenge that the team faced was reaching consensus on the prioritization of the functionality—what capability would be developed first over the three year engagement? What capabilities would really provide the most value early, among the wide range of needs and stakeholders? Each of the workshop participants brought a different view of the risk and importance of each feature of the new system. Having completed the series of three visioning workshops, the Sapient team conducted a more detailed workshop in the early Spring of 2005 to articulate the details of the actual system to be developed and to refine the initial three-year roadmap for delivery of functionality and to take it to next level of granularity.

### 7.1 A typical approach

There are many different approaches for this situation. Many consultant teams simply make a recommendation to the client sponsor. Others simply ask the sponsor for a recommendation. Finally there are a variety of voting techniques where key stakeholders (or all stakeholders) are given one or more votes to cast for their favorite feature and the tabulated results drive the prioritization.

### 7.2 Our solution

We followed a hybrid approach that involved the key sponsor, and also engaged the full group to gain buy-in. The Sapient team (including developers) was also actively involved in the process, which gave them a deeper understanding of the drivers behind the functional choices that were being made by the workshop team. This level of traceability would provide the developers guidance later in the project if actual customers were not

# Improving Communication between Customers and Developers

available.

The Force Ranking pattern combines a systematic, numerical analysis of various factors with qualitative assessment and adjustment. The numbers gave the group confidence, but the process left room for common sense. In just two hours, the group reached consensus on the relative priority of nearly twenty major functional areas targeted for development.

The quantitative element of the framework was a simple spreadsheet, with functional areas as the rows, and evaluation criteria – each with definitions for a High, Medium, Low rating – as column headings. The goal was to fill out each cell intersection, and then to use the combined ratings to generate a score.

As a first step, the Sapiient team developed a draft of the evaluation criteria, prior to the actual workshop. Based on the knowledge of the customer situation, the Sapiient team recommended a set of mutually exclusive and collectively exhaustive (MECE) criteria. For each criterion, Sapiient defined explicitly what High, Medium, and Low meant. The criteria ranged from the straightforward “Solves Immediate Need” (where High = “Critical Need, Medium= “Not Critical” and Low = “Future Requirement”), to “Enhances Decision Making Capability” (where High = “Drastic Impact”, Medium = “Impact”, and Low = “Little Impact”). Importantly, the Sapiient team defined this criteria *before* the actual list of functional areas was fully defined, to ensure objectivity. Sapiient worked with the sponsor and key stakeholders to get buy-in to the criteria before the workshop.

The next step of the process was in day one of the workshop. The Sapiient team facilitated parallel tracks of work to explore and define a core workable list of high-level features. This meant diving into details in areas, and creating a set of 10-20 well-articulated high-level features that the entire customer (and Sapiient) team clearly understood. At the end of the day, the team reviewed the current list with the customers. Later that evening would come a crucial step.

The full Sapiient workshop team (seven people including developers and subject matter experts) spent time that evening to “level out” the list of features, discussing each track’s sessions and analyzing the work done during the day to ensure that all areas were consistently represented, at a similar level of detail.

The other very important step the team did that evening was to select customer team members to lead the prioritization effort the following day – the “kings” and “queens” for the day. Each king or queen would take the lead in the first part of the prioritization – for a particular criteria, he or she would assign a gut level rating (high,

medium, or low) for each functional area. The person needed to be knowledgeable, respected among peers, and feel empowered to make decisions.

With all the preparation complete, the second workshop day started with Sapiient presenting the full “matrix” (with empty intersection cells) and slightly updated list of functional areas for buy-in to the group. The three-step process of force ranking could begin.

Table 5: Generic Prioritization Matrix

	Criteria 1	Criteria 2	Criteria 3
Capability 1			
Capability 2			

The Sapiient facilitator braced the customer group for a long and gruelling session, and then kicked off the process, by one at a time calling upon the “King” or “Queen” for each of the columns (criteria). Viewing a spreadsheet projected on a screen, the facilitator encouraged gut-level ratings – the intent being to get a first broad understanding of the relative priorities. Once all the columns were completed with High, Medium, or Low ratings (according to the definitions established for each decision criteria), the group could view the calculated “overall value” and “relative rank” for each functional area.

The next step was to take the output of the rankings, and apply a first round of common sense. By repeatedly reviewing the calculated force ranking of the functional areas with the underlying ratings (done by the kings and queens), the group iteratively adjusted them until the outcome made more sense.

With a quantitatively-based ranking of features, the group then applied the final step of common sense. They reviewed the calculated ranking of capabilities, and applied the “big picture” view: among other changes, they moved an area ranked 12<sup>th</sup> up to third place because they realized that it was less dependent on external factors, and could give a quick win and a needed capability to the operational forces despite its relatively low ranking in the exercise.

By involving the entire group, carefully setting expectations throughout the process, and mixing defensible quantitative underpinnings with common sense, the group defined a robust three-year capability delivery roadmap in less than two hours! The struggle of prioritizing the features was genuinely shared between the customers and the developers. This common kernel of understanding would serve to better connect the customers and developers for the duration of the project.

# Improving Communication between Customers and Developers

Table 6: Force Rank Pattern

Name	<b>Force Rank</b> – prioritize with gusto
Summary	<i>If</i> a group needs to prioritize in the face of real constraints <i>then</i> use an objective and interactive approach to reach the best solution
Context	People with differing drivers all want their needs met first, but time, money, or people constraints almost always exist in the real world to make this difficult.
Forces at play	<ul style="list-style-type: none"> <li>• Different personal drivers</li> <li>• Knowledge to solve is spread across many people</li> <li>• Buy-in crucial</li> <li>• Minimize time defining this so development can begin</li> </ul>
Solution	<ul style="list-style-type: none"> <li>• Carefully selected ranking criteria</li> <li>• Smoothed set of functionality</li> <li>• Strongly facilitated prioritization session, with key stakeholders driving the initial ratings</li> </ul>
Impact/Results	<ul style="list-style-type: none"> <li>• Closure</li> <li>• Buy-in and ownership</li> <li>• Quick but thorough decision – project momentum</li> </ul>
Don't use this when...	<ul style="list-style-type: none"> <li>• Group is not empowered to make decisions</li> </ul>

## 8. Complementing agile methods

By using an agile-based workshop approach, the Sapient team was able to form lasting relationships with the various client stakeholders while determining what were the most vital new capabilities that must developed by this effort to meet the needs of the Navy. Although the program has just completed phase two of a four-phased program plan, all indications are positive on the value the approach taken thus far.

With the project team's varied background, limited time available to work with the developers, and keen desire to progress rapidly, the patterns presented here made and continue to make a real difference in how customers and developers communicate and interact on a regular basis:

- the relationships formed between the executive sponsor, the user community, and the developers has helped to ensure that the program is only building the capabilities that the operational community needs most;

- the collective program team is incredibly focused on getting this solution right from day one and working with one another to develop new, cutting edge operational capabilities that do not exist in today's fleet;
- visual artifacts enable rapid communication and help speed the engagement along without heavy documentation; and
- the relative value of the capabilities are well-understood by both the customers and developers because the prioritization was developed as part of a team effort.

As phase two of the program came to an end in early June, 2005, the results of the agile based workshop approach could be seen as many of the same original workshop participants returned to the Sapient's Arlington, Virginia, facilities to participate in user testing of the first major release of the program. After an intense user testing round, the assembled team of users was able to identify only a handful of defects (exceptional for a first release) and was able to focus on defining the capabilities to develop in future iterations of the program. By the time the first release was delivered to ONR, no known software defects were present in the system, a clear testament to the strength of the customer-developer communication patterns established and followed on the project.

## 9. References

[1] Sapient, *DCC2 Executive Summary Brief*, May, 2005.

## 10. Bibliography

[B1] Coplien, Jim, [Organizational Patterns of Agile Software Development](#), Prentice Hall, 2004.

[B2] Gottesdiener, Ellen, [Requirements by Collaboration](#), Addison-Wesley, 2002.

[B3] Office of Naval Research website:  
<http://www.onr.navy.mil/default.asp>.