IMD-3-1833
08.05.2007

GUIDEWIRE (A): SPRINTING TO SUCCESS

Professor Stuart Read prepared this case as a basis for class discussion rather than to illustrate either effective or ineffective handling of a business situation.

When John Seybold and his partners founded Guidewire (originally Centrica Software) in 2001, he knew little more about the insurance industry than an informed consumer. But as chief architect of the startup company, he also knew that he and the team had to learn quickly if they were going to convince mainstream insurers to scrap their expensive aging mainframes and adopt Java-based solutions from an unknown and unproven provider.

Seybold felt that the flexibility and speed they needed might be realized through current thinking on agile project management and a process called "Scrum." An outcome of the Agile Manifesto, Scrum defined a new way of organization: One that focused on creating products *with* the customer; that cut out middle management; where projects ran on a monthly cycle; and that enabled the team to rethink priorities and processes at the conclusion of each monthly cycle. Seybold and his partners were betting that this process would help them quickly create an opportunity in insurance software.

However, the decision brought with it risk. Scrum had been used on a project basis, but as far as Seybold and his partners knew, never before had it been used to organize a whole company. Would it scale as the organization grew? Might clients reject a startup company with both radical new products and a radical new organization? And would building a new company, a new product and a new organizational form be more than Seybold and his partners could manage? In the hostile post-internet bubble environment where funding was limited and interest in new software from startup companies was even more limited – they would have to be right fast.

The Team

Guidewire was founded by a team of six individuals. John Raguin (CEO, now president), Ken Branson (product management), James Kwak (marketing) and Marcus Ryu (strategy and consulting) had all worked for Ariba, one of the pioneers in electronic procurement software and online marketplaces. John Seybold (chief architect) and Mark Shaw (engineering) worked together at Kana, a firm building software to integrate the customer service interface across web, e-mail and call center.

The Software Environment: 2001

2001 was not an easy year for the software industry. The internet bubble had just burst, slashing the value of the technology rich Nasdaq composite index by more than 80% in just 12 months. Venture capital, so plentiful in the late 1990s, had all but evaporated. And technology infrastructure spending by large corporate institutions, fueled by the internet boom and Y2K upgrades, had dropped precipitously as well. At best, it was a challenging environment for a small software company looking for investment capital or revenue.

In the enterprise software area, even the most consistent performers, such as SAP, PeopleSoft and Oracle, were facing a difficult situation for exactly the same reasons. Finding a good opportunity was a difficult proposition. Seybold recalled:

> There was a sense that "horizontal" enterprise opportunities were finished – that is, there wasn't going to be another cross-industry opportunity like Sales Force Automation (SFA) or Customer Relationship Management (CRM) to exploit. We were going to have to find something else.

The Opportunity in Insurance

Seybold and the team felt that while players like SAP and Siebel had done well serving product firms, the core business operations of services companies had scarcely been addressed with the latest internet, Java and database technologies. Banks, health care organizations and insurers, which had automated, were using cumbersome custom-developed software which required expensive mainframe computers. These firms would eventually have to move to more efficient systems, and there was, at the time, no obvious vendor to offer them a solution. Seybold explained:

> We were looking for something we could do in one or a handful of related verticals; particularly ones that used lots of skilled white collar labor. Insurance was a natural place to start. Nobody had really focused on the needs of rank-and-file decision makers, like insurance adjusters, and how to promote good decision making across thousands of people in the organization.

But where to start? Because the area had been ignored by mainstream software vendors there was no example to follow – no leader to point Seybold and the team to that critical application Guidewire could build a foundation upon. Making matters more complicated, Seybold and the team all came from the software

industry. In their previous business experience, they had served insurance companies, but none of them had direct experience in the industry which might guide them to a specific application opportunity.

Optimize for Flexibility

The founders looked at the problem and made a decision. Above all, they would be flexible. Flexible product. Flexible business model. And flexible organization. They expected to gain insights every day from customers, partners and even competitors and wanted to be able to immediately benefit from everything they learned. Seybold explained:

> During the period when we were founding the company, Mark Shaw and I made a lot of these decisions in a weekend up at Lake Tahoe. We'd seen a lot of software projects between us, and the one constant was that you always wound up doing something different from what you started with. We considered it as an integrated problem: maximize flexibility with the right processes, technology, tools and people. And the last part is important: we looked for cheerful, unflappable people who could deal with chaos.

The Agile Approach

At about the same time, Seybold and his partners were getting ready to start Guidewire, a group of experts, who had been working on developing methodologies for complex product management, particularly in the software industry, issued a document called the "Agile Manifesto" (*refer to **Exhibit 1***). The principles in the Manifesto represented both the culture of the organization the Guidewire founders wanted to create, as well as an approach that might deliver the flexibility they knew Guidewire needed. Further investigation turned up a specific process called "Scrum" (*diagrammed in **Exhibit 2** and detailed in **Exhibit 3***), constructed in line with the Agile Manifesto, and detailed enough so Seybold and the team felt they could put it to work right away. Anxious to get started, they implemented Scrum at the founding of the firm.

> Scrum is different. Work feels different. Management feels different. Under Scrum, work becomes straightforward, relevant and productive.[1]

The Core of Scrum: The Sprint Team

Guidewire organized around small, nimble project teams of not more than nine people. Of course at first, it was just one small nimble project team. Throughout the firm's six years of high growth, a project team's assignment generally only lasted a month. At the start of a month, each "sprint team" picked from the most important tasks to be tackled. The team then selected a leader, a special sort of project manager termed a ScrumMaster, for the month, and devoured the task. At the end of the month, the team wrapped up the project, reflected on its progress,

---

[1] Schwaber, Ken, and Mike Beedle. *Agile Software Development with Scrum*. Upper Saddle River, NJ: Prentice Hall, 2002.

reprioritized, picked a new task and sometimes a new ScrumMaster, and the process started all over.

Take from the Top of the Backlog

The key to keeping the whole organization moving toward success was the "Master Backlog" of projects. As new ideas were generated and new requests came from customers, they were added to the backlog. No changes to priorities were made during a monthly sprint. But at the start of each new month, the organization re-prioritized the entire backlog and assigned only those tasks that topped the list to the following month's sprint teams. Seybold remarked:

> We never wasted time on dead ends. If we were working on something it turned out the customer didn't want, we could kill it at the end of the month by just moving it down to the bottom of the priority list. If we were working on something that was much harder than we initially anticipated and wasn't going to give us return on our effort, we could kill it at the end of the month by just moving it down to the bottom of the priority list. Likewise, if you got a new idea from a customer, you could jump onto it at the start of the next month by pushing it up to the top of the backlog. That gives you flexibility without crushing the organization.

Communicate within the Organization

Though it sounds like mayhem, there was a guide to Scrum that kept teams productive. The ScrumMaster was responsible for, among other things, the ceremonies of Scrum, such as a daily 15-minute morning meeting at the whiteboard to discuss what the team did yesterday and what they hoped to accomplish that day. Individual priorities and performance, as well as team performance, were completely transparent – something which was compelling both for the team and for management (***Exhibit 4*** *illustrates management visibility into the process*). At the end of each month, the teams examined the process to discuss what worked well, what did not work, and what they wanted to change when they reassembled for the next month's sprint. Seybold commented:

> These meetings, which we call GBUs (Good, Bad, Ugly), are the most important aspect of the process, in my view, and they've had a huge impact on Guidewire. They provide a regular structure for improving every part of the operation, but they've also had a big impact on the culture. Part of the reason it's so decentralized is that we delegated responsibility for almost everything to the team. So, if there's a problem, the team figures out how to fix it, instead of looking to a manager. This is why we have an ultra-flat organization – we never planned it that way, it was just a result of this process.

Build for the Customer

In its earliest form, Scrum had been designed to organize a consulting engagement for a particular customer. As Guidewire was using it to build products, and implicitly needed to serve multiple customers, some refinements had been necessary. At Guidewire, the product managers represented the customers in providing input to the sprint teams. This adaptation guaranteed that one of the

core tenets of Scrum, which is that all activity should be focused on creating value for the customer, was implemented in the Guidewire product process. Seybold explained:

> PMs [product managers] are extremely important to Guidewire – it's the hardest job in the company, because they have to have business skills to work with the customers and understand their needs, and the ability to communicate technically with the engineering teams.

Not Just for Development

Scrum was such a significant part of the culture at Guidewire, it was not used only in the R&D organization. Sales, marketing, consulting and even finance organized into monthly Sprint Teams, continually re-prioritizing their backlog and reconfiguring themselves based on their performance and achievements.

The Situation in May 2007

Guidewire had done well in six short years (*refer to **Exhibit 6** for Guidewire's 2006 position in the insurance claims systems market*). It had grown to over 350 employees and counted almost 40 leading international insurers in its expanding customer base. In the fall of 2006 Guidewire held a user group conference, hosting more than 80 attendees. The product had won numerous industry awards and Guidewire had orchestrated very significant implementations for key clients. Seybold said:

> Not knowing everything gave us a fresh perspective. We were able to see things existing players had missed for years, or simply took for granted. And having an organization that could rapidly adapt as we learned gave us the advantage we needed against bigger players.

Strong demand for the product meant the organization would have to continue to grow. And with more than 100 people in the development organization, Seybold faced the forefront of that growth. The development organization was running sprint teams that had, in some cases, bulged to 20 people. The Scrum process, which had transformed an inspiration into one of the most significant players in the insurance software industry, was being pushed to the edge. How much further could this process take Guidewire?

> When we first did this, I thought seeing the huge backlog and the small projects we finished in a month would discourage the team. But exactly the reverse happened. They said wow – we're in control and look at what we did. It was a real morale booster, and it has continued to be. The organization gets tremendous energy from it – Scrum is a central part of our culture.

What Next?

Seybold had to consider how he might best manage the growth he knew was in store in the months and years ahead. As he saw it, there were three options:

1. *Move to a traditional management structure.* This would make the company easier to understand for outside stakeholders, but he would have to hire in or promote a whole layer of middle management that the company had not needed thus far. The revolving ScrumMasters had handled the running of the operation so smoothly it had not been necessary. With this potential new layer of management there would be a loss of authority and autonomy within the teams, and Seybold knew that might not be well received by some within the R&D group. He would be at risk of losing the focus and intensity the company had enjoyed in its initial growth, and would have to come up with a solution for how he would compensate for that.

2. *Redesign Scrum.* Perhaps there were changes that could be made to Scrum that would optimize it for a larger organization. As it stood, Guidewire had largely implemented Scrum as it was described by several of the authors of the "Agile Manifesto." But with six years of experience in the process, shouldn't Guidewire be able to see how to improve it? He felt that they should, but he just couldn't get to how they would actually do it.

3. *Continue working with the existing process.* But perhaps he was just being alarmist and cynical. Why should the process not scale to a company of 500, 1,000 people or beyond? Things were not perfect – but they were not bad either. And to upset a system that had taken the company this far would be a pretty risky – and perhaps irresponsible – decision.

Seybold stalled. He and his partners had already put six years of their lives into this company. There had been many sleepless nights and even more tense moments. And now customers were actually coming to him. Investment bankers were banging on the door wanting to know when they could bring Guidewire to the public equity markets. This was not the time to stumble. No easy choices.

Exhibit 1
Principles behind the Agile Manifesto

Undersigners of the Agile Manifesto follow these principles:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3. Deliver working software 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.

6. Give them the environment and support they need, and trust them to get the job done.

7. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

8. Working software is the primary measure of progress.

9. Agile processes promote sustainable development.

10. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

11. Continuous attention to technical excellence and good design enhances agility.

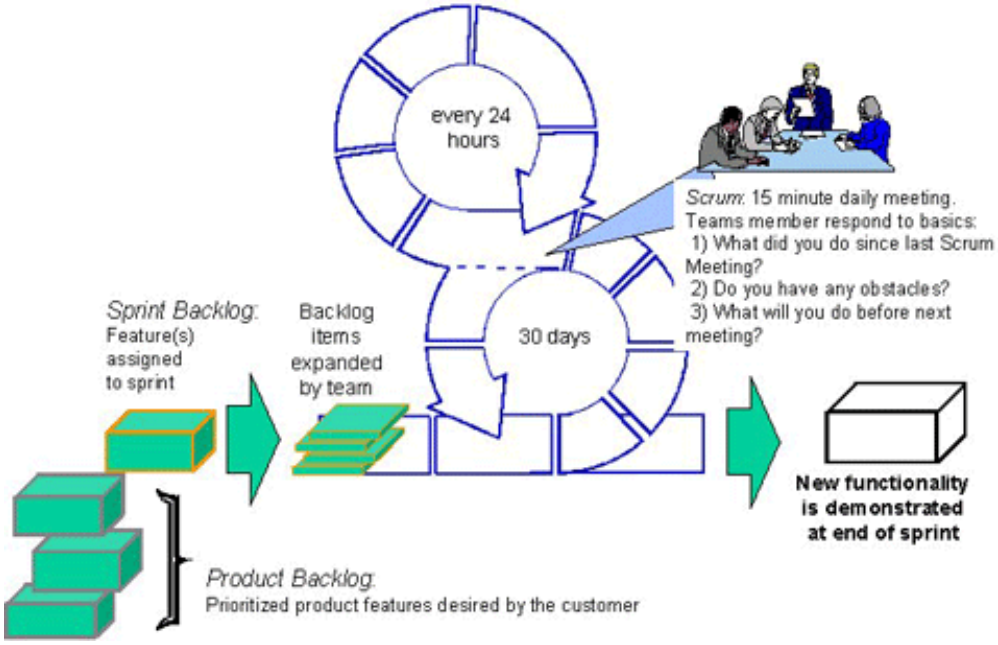
12. Simplicity--the art of maximizing the amount of work not done--is essential.

13. The best architectures, requirements, and designs emerge from self-organizing teams.

14. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Source: http://www.agilemanifesto.org/principles.html

Exhibit 2
Graphical Illustration of the Scrum Process



Source: http://www.controlchaos.com

Exhibit 3
Description of the Scrum Process

1. A Scrum project starts with the customer's vision of the system. The vision may be vague at first, stated in market terms rather than system terms. The vision will become clearer as the project moves forward. A metaphor of the system is also defined to help guide development and to provide a tangible communication model between customers and developers. An initial vision and metaphor can be usually created in several days, if they aren't already part of the customer's business plan.

2. The customer and development team define requirements that will deliver the highest business value from the vision. A prioritized list called a Product Backlog is created that consists of these requirements.

3. The development team works for a short, fixed iteration of thirty days (called Sprints) to create an executable increment that contains the top priority business value. The team selects as many requirements as they can build during the Sprint. They only build the architecture and design needed to deliver this functionality.

4. The customer continues defining requirements that will deliver high business value. They are added to the prioritized Product Backlog. The Product Backlog dynamically changes during the project as the business conditions change and through customer response to the product increments created by the development teams.

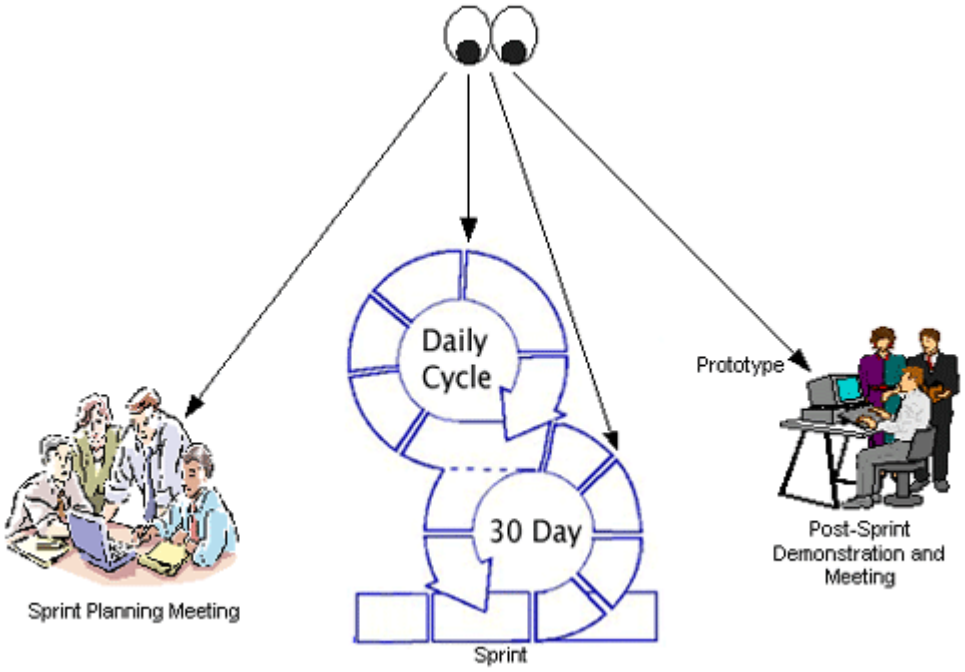
5. At the end of every Sprint, the customer reviews the working system increment with the development teams to see if it delivers the expected business value, and if not, what changes need to be made. These changes are added to the Product Backlog and prioritized with the rest of the requirements.

6. When the customer wants to realize the business value achieved to date, he or she will request that product increments built to date be released. One or more Sprints will be used to polish and implement the system into a releasable product.

The customer steers the cost, date, and business value continuously. By increasing the cost, the customer can cause the delivery of business value sooner. By changing priorities in the product backlog, the customer can change the order in which business value is created. By deferring the date, the customer can slow costs. If the customer is dissatisfied with the competence of the developers to understand the business domain and deliver systems that create business value, the customer can terminate the project at any point. The ability to terminate reduces customer risk. Any work completed prior to termination has produced working functionality that may be able to be used by the customer.

Source: "Agile Processes – Emergence of Essential Systems" (2007) www.controlchaos.com

Exhibit 4
Management Visibility into the Scrum Process



Management can attend and observe the daily Scrum meetings. During these meetings they can observe team spirit, each member's participation, team member interaction, work that is being completed, and impediments to progress.

Management can attend and participate in Post-Sprint Meetings and Sprint Planning Meetings, where – based on progress to date and team capabilities – work is planned.

Scrum provides daily status on team progress, and iterative (every 30 days) reviews of product progress. Everything is visible – what's to be worked on, how work is progressing, and what has been built – supporting management decisions regarding cost, time, quality, and functionality. Plus, management is apprised daily what it can do to help the development teams – what decisions are needed, and what's getting in the way.

Source: http://www.controlchaos.com

Exhibit 5
The Nine Elements of Scrum

Role: Product Owner. Serves as a proxy for the customer and is ultimately responsible for the profit and loss of the outcome of a Scrum process. As such, the product owner prioritizes items on the backlog (rethinking every 30 days), sets target delivery date, and evaluates work results.

Role: Team. Small teams of 5–9 members organize themselves, specify the work results, and have the right to do whatever it takes, within the bounds of the project, to achieve their goal.

Role: ScrumMaster. Owner of the process. The ScrumMaster must be current on project progress, dependencies and requirements, and is responsible for removing barriers for the team. Further, the ScrumMaster facilitates the three Scrum Ceremonies, described next.

Ceremonies: Sprint Planning. This meeting occurs once a month. It is a formal review and prioritization of the Product Backlog. Items on the Backlog are estimated in terms of the effort to accomplish, and items are then assigned to teams for the upcoming 30-day sprint. Maximum meeting time is 4 hours.

Ceremonies: Sprint Review. The first half of the meeting is devoted to a demonstration of the progress the team has made during the sprint. The second half is devoted to assessing the positive aspects of the team to be reinforced, and the negative aspects of the team to be eliminated during the next sprint. Maximum meeting time is 4 hours.

Ceremonies: Daily Scrum Meeting. Daily reporting of each team member on what they accomplished yesterday, what they hope to accomplish today, and what is constraining their progress. Dependency exposure and resolution is handled by the ScrumMaster during these meetings. Maximum 15 minutes.

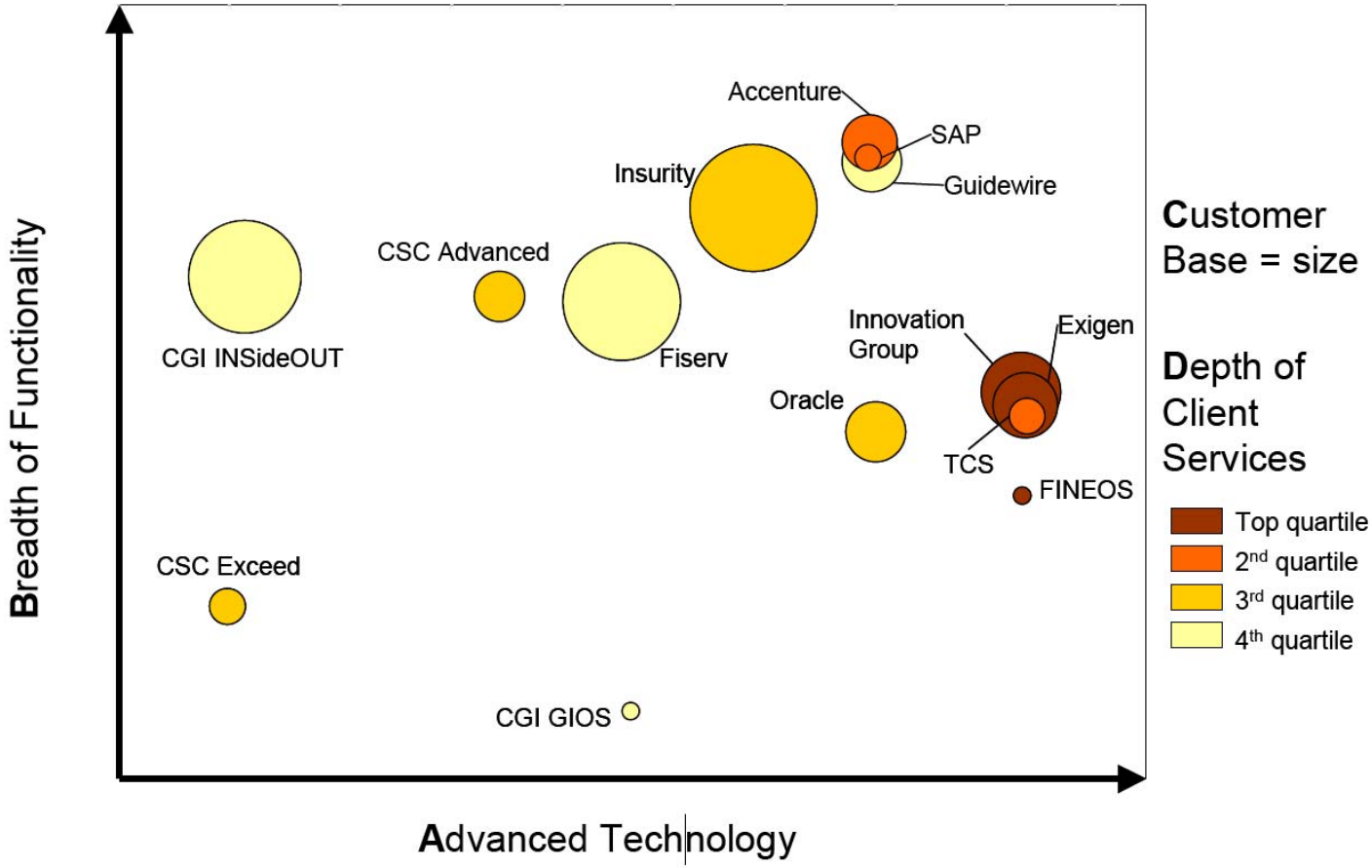
Artifacts: Product Backlog. Product owner prepared list of requirements by business value. This backlog can be reprioritized during a Sprint Planning meeting, but not during a Sprint.

Artifacts: Sprint Backlog. From the Sprint Planning Meeting, a list of tasks for the month's Sprint is generated. This list, the Sprint Backlog, is composed of tasks that should take two or fewer days of a team member's time, and makes up the deliverable for the Sprint.

Artifacts: Burndown Chart. This shows the daily progress of the Sprint Team, enabling the ScrumMaster to reallocate tasks across team members in real time as tasks are completed.

Source: Adapted from http://www.scrumalliance.org

Exhibit 6
Guidewire Industry Position for Insurance Claims Systems 2006

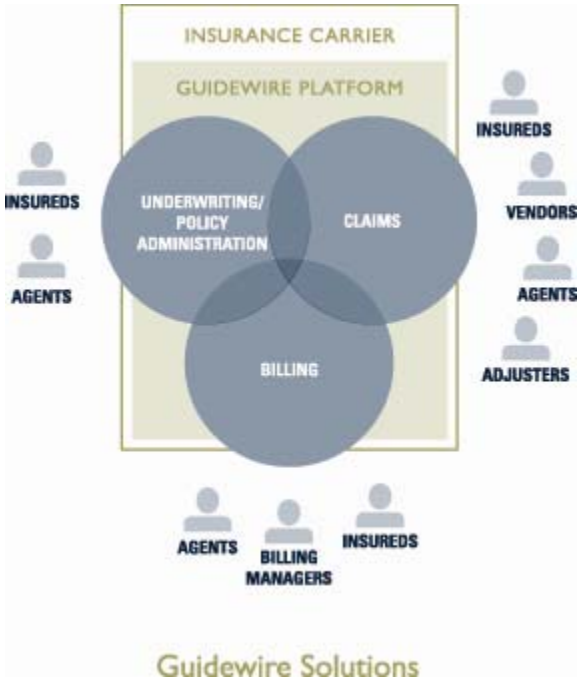


Source: Celent Report (June 2006) "Core Claims Systems Vendors"

Exhibit 7
Guidewire: Our Solutions

The Guidewire Insurance Suite encompasses the mission-critical operations of every personal or commercial lines carrier: underwriting and policy administration, claims, and billing. Unlike the legacy systems they are designed to replace, PolicyCenter, ClaimCenter, and BillingCenter enforce the best practices that ultimately drive lower loss costs and expense ratios. Intuitive role-based desktops support personnel from executives to support staff, while flexible business rules automate high-volume decision-making. Because the Guidewire Suite is entirely web-based, carriers can engage producers, policyholders, and vendors in underwriting and claims, as well as enable a more virtual workforce.

Guidewire systems share a modern technology platform providing workflow between Guidewire applications, integration to complex legacy environments, and upgrade-safe configuration of the user interface and business rules. With proven efficient implementation and scalability to the largest and smallest of carriers, Guidewire has emerged as the leading enabler of the optimized insurance operation.



Source: Guidewire