**Agility Guide**

To get the most out of this chapter, it is necessary to understand the difference between a requirement and a solution. It is also necessary to understand that while we are describing how to write requirements, the most important thing is to understand and communicate them in the way that works for your project.

Rabbit projects have short durations between releases, and as much as possible, they avoid writing the requirements before starting to build a product to meet them. The advice in this chapter that applies to rabbit projects is simple: Rushing to a solution before understanding the requirementnote that we did not say "writing the requirement"generally wastes time. Without knowing the underlying reason for the solution (in other words, the essential requirement), the solution is likely to solve the wrong problem. But, armed with an understanding of the requirements, rabbit projects for the most part bypass writing the atomic requirements and use their scenarios to communicate the functional requirements.

Horse projects usually have a need to write their requirements. Compared to rabbits, horses have longer release cycles **and geographically scattered stakeholders**. This wider distribution of project participants puts greater emphasis on communicating requirements in a more precise and consistent form. It is vital that team members have a solid understanding of what a functional requirement is and what the functional requirements do for the eventual product. That said, horse projects should maximize the roles that their scenarios and a business class model play in communicating the functional requirements.

Elephant projects need a complete and correct requirements specification. All of the information in this chapter is relevant to them, but the discussion on level of detail is particularly pertinent.

Hướng dẫn linh động

Để có ngoài nhất của chương này, cần thiết hiểu sự khác nhau giữa một yêu cầu và một giải pháp. Cũng cần thiết biết rằng trong khi chúng tôi đang mô tả làm sao để viết những yêu cầu, thứ quan trọng nhất sẽ hiểu và truyền đạt cho họ trong cách mà điều đó làm việc cho dự án của các bạn.

Những dự án **Thỏ** có những khoảng thời gian ngắn giữa những phiên bản, và càng nhiều càng tốt, họ tránh viết những yêu cầu trước khi bắt đầu xây dựng một sản phẩm để gặp họ. Lời khuyên trong chương này mà ứng dụng vào những dự án **THỎ** đơn giản: Xô đẩy tới một giải pháp trước khi hiểu requirementnote mà chúng tôi không nói " viết requirement " nói chung tiêu phí thời gian. Không có việc biết lý do nằm bên dưới cho giải pháp (nói cách khác, yêu cầu quan trọng), giải pháp có khả năng giải quyết vấn đề sai . Nhưng, được vũ trang với một sự hiểu biết của những yêu cầu, những dự án **THỎ** phần lớn đi vòng viết những yêu cầu nguyên tử và sự sử dụng những kịch bản của họ để truyền đạt những yêu cầu chức năng. Sống còn rằng những thành viên đội cho phép một chất rắn hiểu của một yêu cầu chức năng nào và những yêu cầu chức năng nào làm cho sản phẩm sau cùng. Những dự án được nói, **ngựa**đó cần phải làm lớn những vai trò của những kịch bản của họ và một mô hình lớp doanh nghiệp trong việc truyền thông những yêu cầu chức năng.

Những dự án **Ngựa** thông thường có một nhu cầu để viết những yêu cầu của họ. So sánh tới **thỏ**, **ngựa** có những chu trình phiên bản dài hơn và …………….. Phân phối rộng hơn này của những người tham gia dự án mang nhấn mạnh greater vào truyền thông những yêu cầu trong một mẫu dạng chính xác hơn và chắc chắn.

Những dự án **Voi** cần một thuyết minh những yêu cầu đầy đủ và đúng . Tất cả thông tin trong chương này liên quan đối với họ, nhưng thảo luận trên mức của chi tiết đặc biệt thích hợp