

# Collaborative Software Design: the Benefits of Customer and Vendor Working Together

**Dr. Dana Thomas**

*Orion Health*

*10 Post Office Square, Suite 750, Boston, MA 02109*

*dana.thomas@orionhealth.com*

## Abstract

*Over the years, software design methodology has progressed through three generations. With each transition, the importance of gathering accurate and comprehensive requirements has been highlighted. Vendors have come to realize that it is not enough to document what the customer wants; we have to understand why they want it. The emphasis on the user's goals has resulted in the adoption of interactive and goal-directed design methodologies. These methodologies are grounded in a clear understanding of the user. In order to obtain a deeper knowledge, vendors may choose to engage in collaborative design with their customers.*

*In this paper I will discuss my experience of collaborative software design over the last 5 years in my role as a Clinical Consultant at Orion Health. I will describe what a typical project looks like, and list the essential elements for securing a successful outcome. To finish, I will list the potential benefits, to both customer and vendor and present a summary of two collaborative design projects Orion Health has recently completed*

## 1. Three generations of Software design

The Waterfall approach typifies the first generation of software design in which software engineers are concerned with implementing the specifications formulated in response to a set of requirements.

This model suits development efforts in well-understood application domains [1]. It does not work so well for domains where users don't know what they want until they see it, or where software engineers cannot reference their own experiences as a potential user [1]. The biggest problem with this method is that if the 'up-stream' requirements analysis fails to truly understand the problem the customer wants to solve, all 'down-stream' activities will hold little value.

Second generation design methods are participatory [1]. These methods attempt to improve communication between technical specialists and users by bringing users into the world of the developer.

Participatory design focuses on the user as the designer so can fail to utilize the skill of experienced vendor resources [1]. In addition there is a risk of miscommunication because the solution is often described without the technical specifications that developers require to translate it accurately into software [1].

Third generation methods are skill-based [1]. In these methods software development is seen as a cooperative task based on mutual learning. The developer is taken into the user's world. The underlying principle is that to design and build useful and useable systems, software engineers must understand not only what the user does, but also why they do it, whilst the user must understand available technical options [2]. This understanding can be achieved through customer-vendor collaboration.

Collaboration can be defined as a recursive process where two or more people or organizations work together toward an intersection of common goals — for example, an intellectual endeavour that is creative in nature—by sharing knowledge, learning and building consensus [3]. Collaboration is particularly useful when there is a significant cultural difference between two groups, for example IT and Healthcare.

Collaborative software design is greatly assisted by the inclusion of an interaction designer.

## 2. What is Software Interaction Design?

Interaction designers strive to create useful and usable software. The practice of interaction design is grounded in an understanding of real users—their goals, tasks, experiences, needs, and wants [4]. Approaching design from a user-centered perspective, while endeavouring to balance users' needs with business goals and technological capabilities enables interaction designers to provide solutions to complex design challenges [5].

## 3. Healthcare is special: The role of the Subject Matter Expert

Healthcare abounds with complex interactions and workflows that a non-clinical person will struggle to understand. Any layperson who has ever sat in a room of clinicians will agree that healthcare has a language all of its own. As a result a significant challenge for the collaborative method is translating the way users express knowledge into a language that software engineers understand. We can enable interaction design in the healthcare setting with the inclusion of a subject matter expert (SME). A SME is a person who is knowledgeable about the domains being represented. In this case it would be a person with a clinical background who has an appreciation of technology and the software life cycle. The SME acts as a 'translator', facilitating the communication of the users' goals through the interaction designer and into development.

## 4. What does it look like?

The reality of the collaboration will vary from project to project. Factors like the pervasiveness of the problem will determine the scale of the effort. In general, it reflects the following steps:

1. Determine the participants: these will be the business stakeholders and should include those responsible for implementing, administrating, managing and evaluating the design. In particular, representatives from each group of end users should be engaged as a Clinical Steering Committee. On the vendor side you should expect project management, SME, design and technical resources.
2. Establish and agree upon the goal of the design project. This is important because it is the measure by which you know the design is complete. The goal defines the scope.
3. Establish and agree upon the terms of engagement: Who is responsible for what, where and how often will you meet? How will you document and communicate design decisions? What are your success criteria?
4. Understand and document the current process and the vision for the future process. Look at the people, the flow, the tasks and the outcomes. Focus on the goals.
5. Form the list of requirements and prioritize them. To provide context, create a scenario that describes the business process for each requirement. The MoSCoW technique [6] is a simple approach to prioritizing requirements in a time-boxed environment. MoSCoW categorizes requirements as Must have, Should have, Could have and would like to have but Won't have [6]. The technique is distinguished from others because it enables a customer to consider 'what they don't want' which provides a different perspective on the problems and helps define a more focused requirements list. Formal sign-off on the requirements and therefore scope, is obtained at this point.
6. Design iteration-feedback loops. Typically the vendor will come up with one or more low-fidelity designs. An effective feedback loop will provide all participants with the opportunity to respond to the proposals. It is important that all feedback is acknowledged, and that the designers communicate if and how they incorporated it into the design. Design decisions should be documented for future reference, and the scenarios should be revised if appropriate.
7. Design is refined. The prototypes move from low-fidelity to high fidelity.
8. The completed design is presented to the customer. At this point you need to obtain agreement that the design satisfies the goal of the project and addresses all the requirements that were listed in step five.
9. Post-design review: with the benefit of hindsight we can learn how "to do it better next time"

## 5. How it works: Essential elements of effective collaboration

Collaborative software design depends on some critical success factors.

1. Common goal: all parties need to be going in the same direction, to the same place. There may be points within a design where one stakeholder group has to concede a preferred design option for one more beneficial to the organization as a whole.
2. Acknowledge expertise: Customers know a lot about the problem, vendors know a lot about ways of solving it
3. Acknowledge reality: brainstorming rarely equals consensus-decision making. If there is a project owner who will make the final call, the group should know so that there are no misunderstandings
4. Avoid getting caught up in the narrative of “we do it this way”. Asking “what problem will that solve for you?” will keep you focused on the goal
5. Scope: having boundaries keeps the design on track. Issues that are out-of-scope can be logged in a register, revisited in the post-design review and bundled into a subsequent design phase should that be appropriate.
6. Inclusive meetings: mean everyone feels involved and has the opportunity to provide feedback with each iteration of the design
7. Facilitate teamwork: balance dominant personalities and be careful that any effort to maintain harmony does not quash critical thinking
8. Checkpoints: deadlines help us to be focused and productive. Checking in the design progress by presenting to the larger group keeps the design honest and helps the team consider issues in context and keep any issue and its relative importance in perspective.
9. “There is more than one way to peel a banana” and more than one way to solve a problem. Sticking with well-known design concepts may come at the cost of iterating an ever better solution. Be prepared to be flexible about how the problem is solved as long as the goals are met.

## 6. Why engage in collaborative design?

Collaborative design helps you solve the right problem: User experience research can help to highlight which are the right problems to solve. The customer benefits because the end design solves their problems and the software delivered matches their expectations. The Vendor benefits because the design reveals all the potential problems in the path to achieving the customer’s goals.

Collaborative design improves software usability: If we design software in such a way that the people who use it achieve their goals, they will complete their tasks faster, with more ease, and hopefully they will enjoy doing it! Customers that implement this software, benefit from this by having more productive, and less frustrated, clinicians. Vendors that strive to design products with great user experience, benefit from strong customer loyalty and enthusiastic reference sites.

Collaborative design results in a streamlined development life cycle: It is easier, faster and therefore cheaper, to make changes to a design before the coding begins. Customers benefit because they get their software sooner. Vendors benefit because the design is well thought through so the numbers of changes that are made over an existing code base are lessened. This lowers the overall development cost.

Collaborative design enables a smooth implementation: If someone who understands the customer workflow has designed the software, the change management impact will be reduced. The customer benefits as any necessary workflow change will have been established early on, giving them enough time to plan appropriately. There is also less chance of having to raise change requests for go-live issues. The vendor benefits because their products are easier to implement so their pricing for services can be reduced.

## 7. Real world examples

Over the last decade, Orion Health has engaged in several collaborative design projects, the most recent of which are summarized below.

### 7.1. Medications Reconciliation

In 2008, Orion Health worked with their customer, healthAlliance to define and build a Medications Reconciliation solution to complement their Electronic Discharge Solution. This project involved working with healthAlliance,

Counties Manukau DHB and Waitemata DHB to define the business problem and develop and gain agreement between the two DHBs on the scenarios and use cases that represent the definition of the problems and the scope of the subsequent solution.

The customer wanted to address medication errors, with a particular focus on those that occur when care is transferred from one healthcare setting or provider to another. A major goal of the solution was to provide a patient's GP with the information they need to reconcile a list of a patient's medications on discharge from hospital against their own record of the patient's regular medications.

The existing paper-based process was multi-disciplinary and complex. To ensure it was well understood, the tasks, workflows and outcomes that the solution had to satisfy were documented using user scenarios. The SME played a key role in the creation of this document. These scenarios described the scope of the design and needed to be clinically relevant to provide context for the design and distinguish commonly performed tasks from edge-cases.

As the customer was located close to Orion Health's Auckland office we were able to have face-to-face meetings as required. The design was initially communicated via low fidelity and high fidelity screen mock-ups, which were later used as the basis for an interactive HTML prototype that was made available in a narrated video. The mock-ups and the video followed the narrative of the user scenarios.

The outcome of this project was a design for an intuitive medication reconciliation tool that streamlines workflow and can be used to reduce medication errors at critical points of patient transfer.

During this project we learnt that it isn't enough to understand the current process. A vision for the future process with anticipated efficiencies should also be described. This second set of user scenarios can be used to guide the associated change management process.

## **7.2. Laboratory and Radiology Orders**

At the beginning of 2009, Orion Health began a collaborative design project with Greene Memorial Hospital in Ohio, USA. This customer wanted an electronic ordering system that would provide a more convenient way for community-based physicians to place orders for laboratory and radiology services at the hospital or one of its facilities, and accelerate the delivery of tests results to help physicians provide a higher level of patient care. The business goal was to increase revenue with the provision of these services.

The physical distance between the customer and the design team was a major challenge in this project. The agreed approach was to hold an intensive, customer-focused workshop before the design commenced. Orion Health put a SME resource on site for a week with the expectation that this person would gain an understanding of the problem in detail and build relationships with the customer team so that any subsequent queries could be addressed in an appropriate manner. The SME effectively became the customer proxy for the iterative design process that subsequently took place.

Prior to this project, no one at Orion Health had sufficient knowledge of healthcare delivery in the United States, specifically the insurance billing requirements, to be able to appreciate how these factors impact on the ordering process. The customer was able to elaborate on these challenges in enough detail that the requirements could be translated back to the design team by the SME.

At the conclusion of the workshops, the SME created a series of workflow diagrams that demonstrated the current ordering process. A series of order-to-result user scenarios were written to provide context for the flow diagrams and highlight the goals of the actors. These goals and the user's flow through the solution were illustrated with lo-fidelity mock-ups that were easy to change as the design was iterated. Once the design was established, hi-fidelity mock-ups were created and assembled in a PowerPoint presentation that provided click-through visuals of the user scenarios. The SME returned to Greene Memorial Hospital to have the design validated, face-to-face, by each business stakeholder.

The outcome of this project was a design that provided a clinician-centric ordering interface and enabled an efficient and customized ordering experience. The post-ordering workflow was streamlined and all the information required for accurate billing and appropriate results reporting was available as required. This process was regarded as successful because the solution went live without any user-facing functionality being changed.

After the solution has completed a period of extended and varied use the SME will perform an on-site "health-check" to review usability and change management related issues. We anticipate that day-to-day use will uncover elements of the design and workflow that can be improved. A small buffer of development effort has been reserved to take advantage of this real-world feedback and enhance the product offering.

As these examples illustrate, each collaborative effort was unique in its detail and organization but yielded beneficial outcomes for both the customer and Orion Health.

In summary, this paper has described how software design has evolved over three generations to the point where collaborative design has emerged as a real and effective method of delivering benefits to both customer and vendor. I have discussed my experience of collaborative interaction design as a SME at Orion Health, describing the steps in a typical project and highlighting the essential elements for an effective outcome. I have provided some reasons why you may want to consider solving a business problem with collaborative software design and summarized two successful collaborative design projects that Orion Health has recently completed.

## 8. References

- [1] Ostwald J. Knowledge Construction in Software development: the Evolving Artifact Approach. 2006. Available from: <http://l3d.cs.colorado.edu/~ostwald/thesis/home.html> .
- [2] Muller M. PICTIVE: Democratizing the Dynamics of the Design Session. In: Schuler D, Namioka A, eds. Participatory Design: Principles and Practices. Hillsdale, NJ: Lawrence Erlbaum Associates 1993:211-37
- [3] Simpson JA, Weiner ESC, eds. Oxford English Dictionary, Second Edition. Oxford, UK: Oxford University Press 1989
- [4] Cooper A. About Face: Principles of Interaction Design 3: Wiley Publishing, Inc 2007
- [5] Interaction Design Association. About Interaction Design. 2009 [cited; Available from: [http://www.ixda.org/about\\_interaction.php](http://www.ixda.org/about_interaction.php).
- [6] Clegg D, Barker R. Case Method Fast-Track: A RAD Approach: Addison-Wesley 2004.