Enterprise Software Experience Design: Journey and Lessons

Bhuvana Sekar[✉]

Cisco Systems, Bangalore, India
`bhsekar@cisco.com`

Abstract. This paper is a case study of a user experience based project in an enterprise setting at Cisco systems, Bangalore to improve an agent desktop experience through HCD methodologies in an Agile software development setting. The core experience revolved around agents' interactions with customers to solve their queries and resolve them. The product, already existing in the market was dated and needed a user experience refresh. The product, already existing in the market was facing a crisis because customers were not using them actively. The design team was brought in to help design the upcoming features in a User-centered way. Intensive user-driven research using the principles of systems-oriented and interaction-design were applied and the team identified core personas, made task flows, collected customer feedback, conducted user interviews, customer validations and made design concepts, recommendations and approaches. Pursuing this path, the methodologies led to some changes - design, technical and operational. This case study explores the journey and underlying issues of what challenges were faced and lessons learned by the user experience design team in an enterprise setting.

Keywords: Usability and software · Enterprise user experience · Agile · Case study · User centered design · Enterprise software

1   Introduction

As a part of a company wide initiative to introduce design thinking, a deployment based software product was decided to be developed in a user centered design fashion, but within the existing legacy back end and system constraints. The design team created personas, investigated usage scenarios and established user pain-points. In conjunction with the stakeholders, task-flows were carved out, design requirements were groomed and several design concepts were iterated to align with the users' goals and customer demands. The task-flows and personas had a humongous impact, being used not just in design discussions to understand the user flows but also to raise questions about the current system constraints and envision a future experience, thereby creating further technical and design requirements. Personas print-outs were put up in various scrum rooms in the workplace to drive conversations keeping the end users in mind.

The product customers majorly relied upon the partner community and 3rd party developers to configure the product solution for their contact centers. They were intimately aware of the customer needs and were familiar with the competitive offerings by

rivals as well and could configure gadgets and tweak them exactly as per customer goals, needs and demands. Therefore, both these stakeholders were a critical piece of the business and often had a huge say in how the products were designed – although they were not the end-users for the same who were agents and supervisors in contact centers.

2 Background and Context

The agent desktop framework housed multiple gadgets/widgets catering to various customer requirements that could be added/modified by partners/3rd party developers. The product, an in-house gadget within the framework, enabled agents to solve consumer issues through web chats and emails. The User-Interface (UI) had an extremely busy and dense interface, with multiple ongoing simultaneous interactions. Since the UI was also designed in an incremental approach, the feature designs contributed to a design debt resulting in a complicated and confusing UI and an overused phrase – "… Our users are used to this…". Secondly, the product catered to email and chat based query resolutions in contact centers and just had primitive features to do so but, there was a tendency to look at popular email and chat clients and simply emulate features from them. With this mindset, comparisons were drawn to look and behave like Gmail, Outlook etc. which just added to an increasing 'design debt' as the product had an extremely different Information Architecture.

The Project was done in a recursive cycle of 2 sprints over 11 months. We set off by identifying all pain-points observed for different personas, their goals, created task flows and identified areas of opportunity. We observed quite a few low-hanging fruits (such as excessive or poor labelling, too many simultaneous actions, improper hierarchal order of UI elements and usability) but decided to go the route of user feedback to create the awareness and need for bigger changes not just limited to just refreshing the UI but also the backend technological changes, to support user goals and in turn, positive customer benefits.

3 Methodologies

3.1 Mind-Mapping of the Gadget System and IA

The approach for this project began with a mind-map of system understanding and Information Architecture interdependencies. Personas and task flow paths were also identified and marked. Certain overlapping items and tasks emerged that were critical points of interactions that needed to be consistent across different personas.

This exercise led to certain insights – One, the IA for the gadget was driven by various stages of evolution of product, complicating the UI and leaning away from 'goal/task' based experiences. Second, due to the same reason, contextual actions that occur during a specific task became a necessity. For example: An Agent will need the most recent customer interactions and journey history next to the active chat/email interaction. But all customer history was on another tab, thereby - forcing the agent to switch views during the interaction, missing the context and completely breaking the experience.

3.2   User Interview Sessions

Post requirement grooming, concept visioning was done for a required feature. Later a qualitative-research plan was developed to understand the usage context, goals, pain-points, and any other peripherals/tools used to accomplish tasks. We conducted 1.5-hour interviews with 7 agent desktop users (agents and supervisors) - 5 remote and 2 in person. We used a semi-structured script and conducted the interviews via WebEx screen-sharing sessions. Participants were asked to walkthrough the screens to perform a given task and were encouraged to think-aloud during the entire process. They were given a mouse/keyboard and were asked to perform a task using 'In-vision' based mockups to understand interaction modalities and validate the flow and designs. We invited the product owners, managers, and team members to observe the session. Briefing sessions took place before the interviews to illustrate the importance of observation, not leading the users, noting down observations and insights etc. At the end of the sessions, a debriefing would take place where the stakeholders were invited to discuss their learning, feedback and interview highlights.

3.3   Artifacts and Types

Design awareness artifacts such as Oral storytelling, 'Vision design-discussions' and 'Stakeholder Readouts', were facilitated along with artifacts like flows, maps, and mockups. Backlog tracking experiments were visualized using Kanban boards and 'Rally' was used to manage, track and plan the design work ahead of sprints.

3.4   Design Delivery, Customer Demos, Guerilla Testing

The final signed-off designs were supplied to scrum teams ahead of at least one sprint. The teams often would participate in the earlier stages during the 50% and 80% scrum ceremonies where the design is discussed and critiqued. The design specifications and style guides were delivered via 'Berlinux' in HTML format. At the end of every sprint upon implementation, the designs were demoed to customers. Different product teams role-played various personas and conducted Guerilla testing to identify bugs and defects.

3.5   Problems and Solutions

Without any formal design pattern in place, it took a lot of time to make designs and often led to inconsistencies. This was solved by identifying and making a local design library. Another problem being the changing requirements and ad-hoc requests that came up to feed the scrum teams, was more or less tamed by tracking the design work and artefacts. Third, for validating proposed design decisions, user interviews and IA-gadget mind-mapping helped clear and drive this fact home. A big achievement here was, that a need for a better overall experience and visual design was deemed necessary and is being worked upon for the next upcoming product release.

4   Insights, Challenges and Learnings

1. **Customer goals are not the same as user goals:** It became another imperative goal for us to differentiate that our customers' goals might not be the same as our user goals, and therefore the requirements given did not provide the information to improve the user experience. Therefore, we often had discussions to reach a middle ground, where we would recommend designs but give the option of modifying it at discretion, to not kill the business model.
2. **Agile design debt:** In an incremental agile approach, products were built in terms of feature additions and their experiences designed in slices. Therefore things were force-fitted in the UI instead of being looked at holistically, which led to less than optimal end-user experiences.
3. **Designer to developer ratio:** Since the design work ranged from doing design visioning to designing, iterating, implementation support and doing demos, a 1:16 designer to developer ratio led to stress and a myriad of chocked deadlines.
4. **Fragmented end experiences:** Due to a top-down approach for delivering features, organizational and product boundaries start emerging and kill the experience and the overall level macro-level view was muddled and experience fragmented.
5. **Relevant hierarchical awareness:** From their point of view, for e.g.: to engineers – the importance of UI consistency across not just 'their' component but the entire solution by thinking in terms of an agent. To the Customer Representatives about asking the right questions and uncovering the hidden need rather than taking a customer demand at face-value. To the Executives- why a user-centered and consistent experience might be better to stay ahead of the competition pack by comparing with similar organizations who had also made positive changes to their products using design.

References

1. Pandey, S., Swati, S.: Data Driven enterprise UX: a case study of enterprise management systems. In: HCI (2014)
2. Gajendar, U.: Why I Design Enterprise UX, and You Should Too!: Medium, June 2015. https://medium.com/@udanium/why-i-design-enterprise-ux-fa74e9f12671
3. Cleveland, B.: Call Center Management on Fast Forward: Succeeding in the New Era of Customer Relationships. ICMI Press (2012)