Data Driven Enterprise UX: A Case Study of Enterprise Management Systems

Sumit Pandey and Swati Srivastava

Clarice Technologies, Baner, Pune, India
{sumit.pandey,swati.srivastava}@claricetechnologies.com

Abstract. This paper describes and makes a case for a data driven user experience design process for Enterprise IT. The method described employs an approach that focuses on defining the key modules (objects) in an enterprise IT software and the data sets used by these modules very early in the design process. We discuss how mapping parent child relationships between key entities in the software and the linked data helps create a holistic view of the product ecosystem which in turn allows the designer to create an uncluttered information architecture and user journey that maps closely to mental construct of the system in the user's mind. We further argue that in the present age of big data, working with well-defined data sets and visible data relationships creates a valuable information repository for the designer to take decisions regarding task optimization and building business intelligence in the system itself. We also discuss the urgent need, advantages and methods of 'consumerizing' the Enterprise UI to increase users productivity and reduce the learning curve. Lastly, these ideas are exemplified through a real life case study for an enterprise server management system.

Keywords: User Experience Design, Consumerization, Design Process, Enterprise IT, User Centered Design, Data Driven Design, Design Patterns, Case Study.

1 Introduction

Enterprise IT refers to hardware and software that helps power business processes which often form the backbone of many consumer-facing services. Traditionally, these IT systems dealt primarily with structured data, (i.e. data stored in an organized fashion in databases or other forms of record storage) and were deployed as standalone data processing systems for record keeping, accounting and data storage [1][2]. These systems have always played a pivotal role in enabling services involving complex interactions between users, applications, services and devices and have evolved to become distributed systems characterized by very high levels of complexity. [3]

With an unprecedented amount of mobile and internet access coupled with cloud computing and social media, there has been an explosion in the amount of unstructured user generated data being created in the form of emails, multimedia, webpages, photos etc. This is in contrast to the structured data that traditional IT systems worked

S. Yamamoto (Ed.): HIMI 2014, Part II, LNCS 8522, pp. 205–216, 2014.

with which was easier to analyze and customize. The abundance of unstructured data [4] presents a wealth of new forms of information, which could lead to newer systems of business intelligence and analytics. Consequently, decision makers at all levels [5] across organizations expect ready access to relevant and actionable information to make better and smarter decisions faster. This can be clearly seen with the increasing demand for big data systems and intelligent analytics across business domains. These advances had a huge impact on the way that large-scale enterprise systems think about data and how the data becomes actionable information. Their role is no longer limited to being data storage/data entry and configuration silos but organizations are increasingly relying on them to be scalable decision making support systems.

There has also been a growing trend of "dual use" devices, networks and services, used both by consumers and businesses like tablet and smartphone devices along with workplace policies like "Bring Your Own Device" [6]. This trend is referred to as "consumerization" [7]. With millennials [8] comprising almost 40% of the enterprise IT workforce, consumerization is a trend the industry cannot afford to ignore [9][10] since the mental models and interface expectations of the present day enterprise workforce stems from the easy to use and refined interfaces of consumer applications rather than legacy CLI interfaces. [11]

Both of these factors have led to a change in the landscape and focus of enterprise IT. The current wave of enterprise IT is based on an industry wide demand for real-time/intelligent analytics systems coupled with improved and adaptive UX [3]. This paper presents a structure for a design process to work with enterprise management systems referred to as the Data Driven Enterprise UX process. This process is outlined using a real life case study of a project completed during our practice as UX consultants at Clarice Technologies. The project was located in the enterprise server and device management space and helps underline the efficacy of the design method in the design of complex enterprise IT software. Further, the paper argues how the process helped us align user goals with the high scalability, feature focused and legacy constraint driven requirements of the enterprise. We then summarize the process and present lessons learned from the case study presented followed by concluding remarks.

2 Enterprise IT and User Experience

Although consumerization of IT systems was initially seen as a trend chiefly linked to dual use devices and a way where consumers brought devices of personal use to their workplace, it has since lead to a fundamental shift in the way enterprise users expect software to behave and perform. On the other hand, rapid technological advances have changed the way that large-scale enterprise systems think about data and how this data becomes actionable [12].

This makes the design of enterprise IT software a challenging and compelling space for user experience designers to work in.

Enterprise IT presents an interesting premise where the designers have to effectively combine the understanding of human behavior, technology and processes to create a system that adheres to workflows and expectations of the target user, works within the technological constraints of legacy systems as well as one which analyses and provides quick and easy access to data that leads to intelligent business decisions. But while technologies like virtualization, cloud computing and storage and software defined networking have been developed rapidly to meet the evolving requirements of enterprise IT, UX has continued to play the role of a retrofitted solution applied on top of fundamentally disconnected feature driven system modules. User insight evaluation and research methods continue to have a micro-level focus on specific modules and tasks rather than taking a top-down, macro-level view of the entire product ecosystem. Hence, current enterprise UX processes tend to overlook opportunities of information flow between system modules along with opportunities for data interoperability and building a consistent product language and may look aesthetically pleasing at the outset but are fundamentally broken from an experience standpoint. As mentioned earlier, in the face of the changing role of IT systems and evolving expectations from the workforce, UX processes need to identify, mold and juxtapose disparate data sets and present them according to user context and role. UX can be a way to increase employee retention and concentration, decrease training time and a way to create a better consumer experience [11].

The process of creating Enterprise IT software can be split into three principal components [3]:

1. Gathering relevant data and information.
2. Transforming data into insightful and targeted performance indicators and operational parameters.
3. Mapping indicators and functions to business objectives and operational constraints leading to the execution of appropriate actions.

Consequently, the requisite goals for an effective design process for enterprise IT can be listed as the following:

1. Restructuring the data sets and modules with an emphasis on highlighting logical relationships, hierarchy and dependencies.
2. Using the uncovered relationships and their priority to optimize tasks and make data more actionable and insightful. Explore possibilities of creating deeper customized analytics and offering insights helping the large scale enterprise evolve, improve and adapt.
3. Create a user interface that maps to the user's mental model and aids the users in performing tasks on the system more efficiently and with fewer errors.

3 Data Driven Enterprise UX

We recognize that the nature of data plays a very important role in crafting the interactions for any enterprise system. To design the user experience of systems, which serve as a reference for important business decisions along with being task-performing tools, it is imperative to view data not just as the content going into different screens and system modules but as a cluster of interlinked information sets. Further, this approach helps in identifying logical object-data links that can be further classified by user context. Interactions, navigation systems and screens designed keeping these information sets in mind result in interfaces which are tailored to the tasks as well as the content being presented. These ideas serve as the building blocks for a data driven model for enterprise UX that we present in this paper. This model is aimed at aiding the creation of a scalable and adaptive strategy suited to the needs of the new generation of IT systems. We outline the steps of the proposed design process in the following sections (Fig. 1).

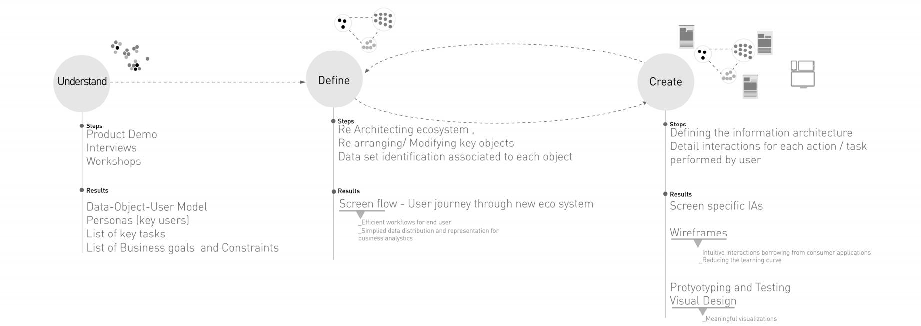

Understand
Steps
Product Demo
Interviews
Workshops
Results
Data-Object-User Model
Personas (key users)
List of key tasks
List of Business goals and Constraints
Define
Steps
Re Architecting ecosystem ,
Rc arranging/ Modifying kcy objccts
Data set identification associated to each object
Results
Screen flow - User journey through new eco system
_Efficient workflows for end user
_Simplied data distribution and representation for business analystics
Create
Steps
Defining the information architecture
Detail interactions for each action / task performed by user
Results
Screen specific IAs
Wireframes
Intuitive interactions borrowing from consumer applications
_Reducing the learning curve
Protyotyping and Testing
Visual Design
_Meaningful visualizations

Fig. 1. The Data Driven Enterprise UX Process

3.1 Understand

The understanding phase of the design process should be driven by subject matter experts and end user (if possible) workshops and interviews. These workshops are aimed at creating a specialized product ecosystem model that we refer to as the 'Data-Object-User' model. The focus of this model is to collaboratively map the various user roles and usage contexts, the primary objects that the product is going to be built around and the detailed information and datasets (and their subsets) that the product deals with. Akin to object oriented programming [13] principles of creating parent-child relationships through inheritance and encapsulating key data sets into objects and classes, this model maps user roles and data access relationships along with data dependencies and relationships at a very early stage. During the course of discussions and participatory workshops, parent child relationships between objects along with logical object-data links and possibilities of data and information flow within and outside the product are also identified.

3.2 Define

Jon Kolko, in his paper on Abductive Thinking and Sensemaking describes the process of synthesis as (a process where) "designers attempt "to organize, manipulate, prune, and filter gathered data into a cohesive structure for information building." [14] During the process of synthesis, we try and identify information clusters and reorganize the dataset discovered during the 'understanding' process into object centric, task centric or task-object hybrid modules. This is done by aligning the objects and actions identified into a consumable hierarchy that maps to the users mental model in the closest possible fashion. For instance, a generic data consumption and visualization system is prone to dealing with a large variety of data-objects and in such a scenario the user's journey typically starts with a task that needs to be completed. In such a scenario, the designer might be inclined to assign the highest priority to the tasks/actions the system exposes and consequently group the objects based on the requirements of the tasks. In contrast, for a system where the objects are limited, the user's journey would typically start by identifying the object he/she is most interested in followed by contextually identifying the tasks that need to be performed. Such a system might be better served by creating a object centric or a hybrid hierarchy. Subsequently, detailed screen flows are created. Each screen in the screen flow outlines high-level information sets that need to be presented on it as well. The identification and placement of information sets is in accordance to the user roles and requirements and it maps the data relationships identified in the Data-Object-User model to screen level information that is presented to the end user. It also allows for examining the relevance of the information presented at a micro, i.e. screen level as well as the flow of information presented at a macro, i.e. task level. While these flows are being fleshed out, possibilities of automation, optimization and anomaly prediction need to be considered and accommodated as well.

3.3 Create

The screen-data flows identified during synthesis should be converted to detailed wireframe mockups built with interaction models focused on surfacing, presenting and manipulating data in a meaningful and easy to consume manner. Fresh interaction and visual paradigms should be developed to present the identified information set. Design has emerged as a clear differentiator in the consumer application space because the availability of multiple, often free, options, users rarely choose an application with a badly conceived experience. Users tend to prefer uncluttered and focused interfaces with fewer but more relevant options. The designer could potentially look at interaction models being used in the consumer product domain [15] to build concepts with increased familiarity and reduced cognitive load for the end user. Specialized views and sub views, popularly referred to as microinteractions [16], based on different combinations and juxtapositions of the same dataset could be created to the specialized needs of a specific user role/context. In the case of mobile contexts, an adaptive/responsive strategy could be developed using a mobile first framework.

4 Case Study 1: Server and Device Management System

This web based management and monitoring console was a part of a broad set of offerings from a large unified service delivery management company providing end-to-end network and application-based solutions. The brief was to redesign the existing user interface considering some legacy constraints and adding some new feature sets. The software was successful in the market but was receiving a lot of usability complaints that made the company consider a ground up redesign exercise. The software consisted of several interconnected modules catering to various customer requirements. Although the software modules were functionally very well thought out, it relied on extremely data rich screens laid out with a feature centric navigation. This resulted in a very complicated system for the end user who had to rely very heavily on their training to navigate through the software and to interpret the data as well, since the data connections were not well represented in the system. Furthermore, as users performed action that affected these data sets directly, the system needed a forgiving and feedback oriented UI, to reduce the user's overall cognitive load. The following sections outline the redesign process using the Data Driven Enterprise UX process.

4.1 Understand

As discussed, the first step of the re-design exercise was gaining an in depth understanding of the service ecosystem. The project was conducted in a four month sprint, which left little time for gaining access to end users. The design team had access to the product and sales managers to conduct interviews and workshops. The two initial sessions involved the managers giving a detailed walkthrough of the software ecosystem. On the basis of these workshops, the design team outlined the as-is ecosystem in the form of a mind map in collaboration with the client's team, validating it as we moved forth. This mind map was a mix of data-object and modules. The next sessions involved exhaustive discussions with the client executives regarding the user types and the actions performed by them along with modules (objects) and data accessed by them to perform these actions. These sessions provided two important insights. First, the relations and interdependence of data as designed in the system; secondly users' actual navigation pattern to perform the required actions. Subsequently, the design team did a comparative study of these two mind maps to identify the following main components:

- Data Sets (e.g.: User lists/Upgrade files/Login time).
- Objects accessing the data sets. Modules like Activity logs accessing data sets like login/log out times from the event log and data sets from the other modules.
- User Roles and logical navigation between the modules to suit the users' workflow, habits and needs. During the understanding phase, logically related data sets and key objects began to surface, even though they were scattered around in the whole system in the original design.

After involved discussions with the client team the design team started re-architecting the system's ecosystem and subsequently re-organizing the granular details within the modules. This step was one of the most intense and important steps.

The design team analyzed the existing product ecosystem and based on the user goals and the data-object relationships uncovered, created the Data-Object-User model. To create this model, the design team re-thought the clustering of the uncovered the data sets and then re-organized the data-object relationships (Fig. 2) to align them to workflows and key user tasks. E.g. the deployment data set in the current system had hardware (infrastructure specifications) as a part of the configuration module. This data set had no immediate relation with the configuration of the system itself. Hence, this data set was pulled out of the configuration module and a separate module created for it. For an end user working in the configuration module the task became a lot more uncluttered and focused and hence he/she could perform it faster.

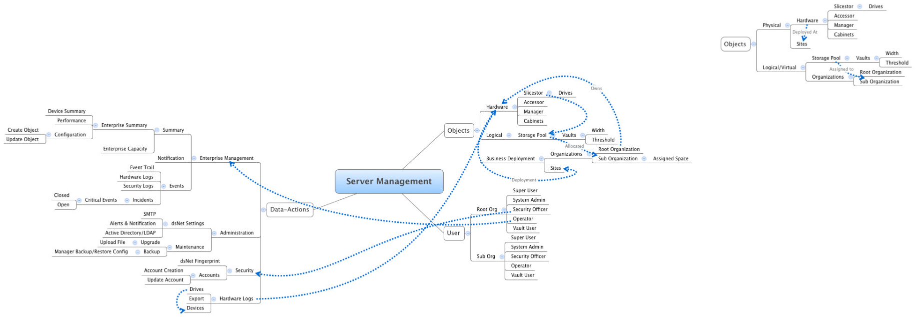


Fig. 2. Data-Object-User Model

4.2 Define

Although design synthesis strictly focuses on the end user needs and habits, in enterprise software, legacy business is an important constraint that cannot be overlooked. The re-arrangement of the components in the eco-system also takes into consideration the business logic. Designers are always end user advocates in this process but there are certain points in legacy systems where a hard bargain has to be struck and an appropriate design has to be created around these legacy constraints. These constraints play an important role when the screen flows and subsequently the information architecture are being designed. Considering these legacy constraints and the finalized Data-Object-User model, the key user tasks were arranged into a descriptive screen flow as described earlier. These flows (Fig. 3) show the user journey through the new eco-system for accomplishing various tasks. These screen flows gave an in-depth representation of the relationship between data sets and objects. For e.g. one of the primary task of monitoring the network was detailed till user viewing an anomaly, doing root cause analysis and taking action to remedy or report the same, is shown in one complete flow. In addition, screen flows also started outlining the high level actions on the data sets. This formed the foundation for next steps of creating detailed wireframes.

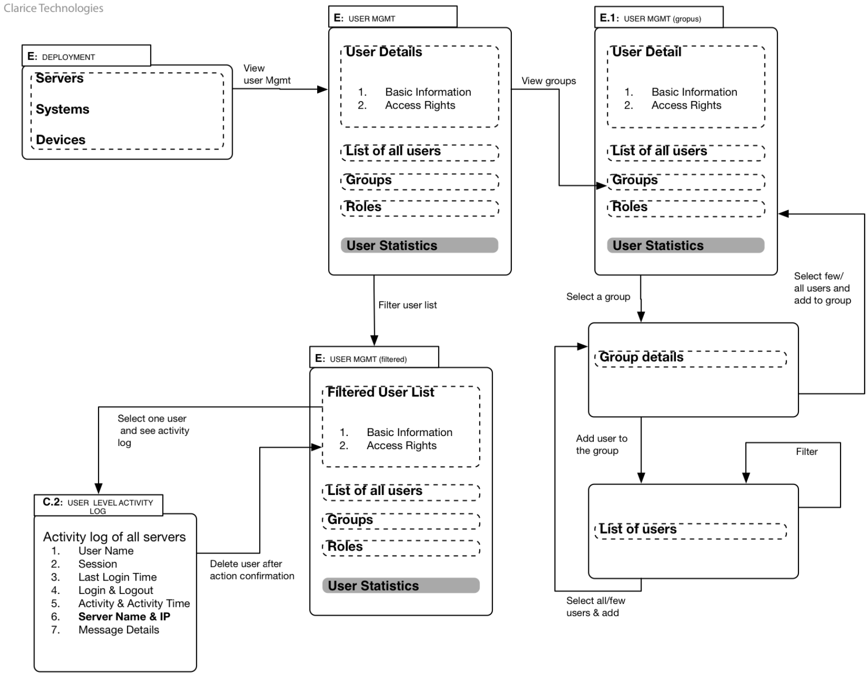


Fig. 3. Information/Screen Flows

4.3 Create

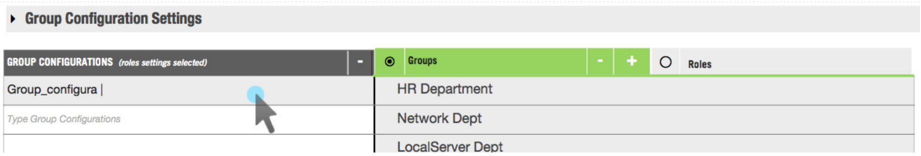
Enterprise UIs rampantly follow a common UI pattern of showing all the data and all the related actions at once at the top panel creating a very busy interface. This UI was no different. The screens included a couple of dashboards along with more detailed data screens. Design team followed a consistent approach of showing summarized high-level information before displaying the detailed data set. Borrowing from the consumer application pattern that makes use of contextual actions to de-clutter the interface, the design team refrained from displaying the whole set at once. Each module had a set of global actions and subsequently based on the selected data set, the contextual actions were displayed.

We realized that we could take a page out of the consumer application design patterns to enhance the overall experience of the server management application as well. Hence, we adopted a layered information dissemination approach by pushing the less frequently accessed information and actions further and deeper into the UI and by indicating the most relevant information bits on the top level overlaid with clear drill-down indicators.

While crafting the detailed design for the system the design team consciously worked towards creating a user interface which has the ease of use and clarity as that of a consumer application. Although enterprise applications encompass far more complex data sets of very high scale, the interactions when broken down on a per use case basis, often have a similar nature to consumer applications. e.g. Add/Delete/Configure actions. Furthermore, it should be understood that in this scenario, human nature and behavior is a common factor. Although the users of enterprise application are generally trained but when they perform similar actions in a more complex manner in the context of enterprise software than while using a consumer application, it results in frustration. Additionally, they are not able to leverage their mental model that evolves from using consumer applications on a day-to-day basis and even in their work, while multitasking on the same system as the enterprise software. A word of caution while following this process: the security and integrity of data and business logic should not be compromised while attempting to create a simpler interface.

We outline the process of designing consumerised microinteractions [16] through some examples below:

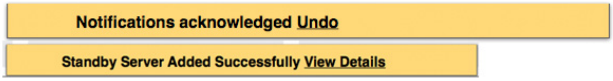
- The existing UI of this system displayed all the user information related data fields at all times in most of the forms even though in more than half the use cases the fields were not even needed till they were actually needed for a specialized use case. To reduce the complexity of the forms which the user was expected to fill, the design team took the "Progressive Disclosure" approach by using radio buttons on the table itself to convey only one column of the table was to be filled to complete the form. The whole UI was designed such that it showed a minimum number of fields when only one user was to be added but UI expanded itself dynamically for batch additions of the user. (Fig. 4)


▸ Group Configuration Settings
GROUP CONFIGURATIONS (roles settings selected)
-
● Groups
-
+
○ Roles

GROUP CONFIGURATIONS	Groups
Group_configura |	HR Department
Type Group Configurations	Network Dept
	LocalServer Dept

Fig. 4. Progressive Disclosure

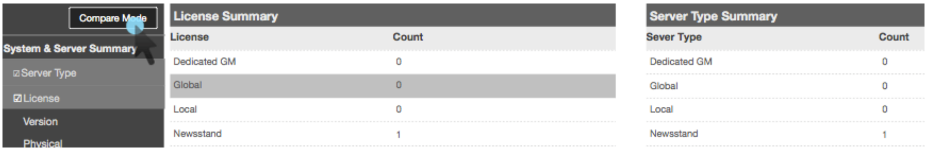
- A popular consumer application – Gmail [17] provides a very interesting microinteraction that lets the user quickly view any sent email right after it has been sent. This caters to the often-practiced habit of users to confirm if they sent the correct content and to the correct people. In order to create an interface that provides ample reassurance to the user performing critical actions in complex enterprise systems, this pattern was borrowed and used in this system. Whenever the user performed an activity such as adding a user, or configuring a device etc., a view details link was provided to the user in case he/she wanted to see the results of his/her activity and wanted to revert or make any further changes (Fig. 5).


Notifications acknowledged Undo

Standby Server Added Successfully View Details

Fig. 5. Contextual links for validating actions

- Not only do clear data-object relationships help in identifying areas for task optimization and creating optimal data clustering, they also potentially help identify possible areas of adding an automation or business intelligence. One such example was the deployment module that was initially used to display static counters and reports but was identified as a module that had the potential to transform into a fully dynamic and customizable reporting tool. For instance, the user could select different types of device specifications in a multitude of different combinations to create a comparative report as required (Fig. 6).



License Summary

License	Count
Dedicated GM	0
Global	0
Local	0
Newsstand	1

Server Type Summary

Sever Type	Count
Dedicated GM	0
Global	0
Local	0
Newsstand	1

Fig. 6. Comparative reporting module

4.4 User Validation

The final UI created as a part of this redesign exercise was converted into HTML prototypes and the screens were tested with actual end users. The users mentioned that they felt a sense of cognitive relief as the system was aiding them with intelligent suggestions and the improved action clustering helped them move almost in a guided path through the system. Additionally, the interface paradigms were similar the ones they usually came across and they were able to decipher the way to use them easily with almost no training needed for the redesign. Lastly, the notification center had become like an inbox where they manage, monitor, archive and take actions.

5 Conclusion

The Data Driven Enterprise UX process is designed as data centric process that targets task optimization as well as putting important data usage patterns collected through system logs and expert interviews, to work as a reference for taking important business decision. This process follows the core tenets of the user centered design process and juxtaposes it with an early and deep understanding of the data models upon which enterprise IT systems are built. Time-tested tools of the design process are put to use to uncover these data relationships and object hierarchies and their correlation to the end user's journey. This helps create possibilities for the creation of interfaces that provide access to data in a contextual relevant manner and helps optimize user tasks as well as exposes opportunities for adding business intelligence. In addition, we tried to explore how consumerised interaction and UI patterns could help in creating a familiar and more efficient experience for the end user. Finally, we presented a case study to demonstrate the application of this approach in a real world scenario.

During the requirement gathering phase, mind maps were created to reflect the existing ecosystem and in the internal relationships of the between the modules and data sets utilized by these data sets. Key archetypes were identified followed by crucial tasks performed by them. With enough domain knowledge and understanding, the existing system was re-architected. As discussed in section 4.1, existing modules (objects) were analyzed to create the Data-Object-User model that helped in an in-depth understanding of data-object relationships. This analysis included exploring the purpose, dependencies, hierarchy and actions present in the modules. As can be seen in in section 4.2, this provided better guidance to the design team and helped create more efficient workflows for the user by optimizing and automating certain aspects of tasks. This also resulted in the identification and reduction of redundant data display and actions in the UI.

Another direct and important contribution of the Data-Object-User model was the vocalization of ideas from the high-level decision makers, relating to further ways of employing the data for gaining better insights. This is clearly demonstrated in the redesign of the server and device management system's deployment module as discussed in section 4.3.

As mentioned in section 3.3, with the advent of cheaper technology, smartphones and tablets, incorporating consumer driven interaction patterns has become imperative in the design of enterprise UI. It also aligns with the goals of a design team to create a user-friendly interface with low learning curve and can potentially lead to great insights and fresh approaches during the 'create' (section 3.3) phase. One of the approaches to accomplish this is to understand the intention of the user's task and compare it to similar tasks in the consumer product domain. However, the design team must always be wary of oversimplifying the UI to the extent that it starts compromising data security and integrity. Experts at the client side must always verify this while the wireframes are being designed and shared.

Acknowledgements. We thank Milind Sonavane for agreeing to be a perpetual sounding board for our ideas and insights and the UCD team at Clarice Technologies for their constant help and support.

References

1. Moore, G.: Systems of Engagement and the Future of Enterprise IT: A Sea Change in Enterprise IT. AIIM White Paper (2010)
2. Lara, Z.: Enterprise Software History, Part 2: Minicomputers to the PC, `http://blog.softwareadvice.com/articles/enterprise/software-history-part-2109062011/`
3. Delic, K.A., Dayal, U.: Adaptation in Large Scale Enterprise Systems. ACM Ubiquity 2004, p. 1 (August 2004)
4. Structure, Models and Meaning: Is "unstructured" data merely unmodeled?, Intelligent Enterprise, `http://www.informationweek.com/software/information-management/structure-models-and-meaning/d/d-id/1030187`

5. Fisher, D., DeLine, R., Czerwinski, M., Drucker, S.: Interactions with Big Data Analytics. ACM Interactions 19(3), 50–59 (2012)
6. Gartner Predicts by 2017, Half of Employers will Require Employees to Supply Their Own Device for Work Purposes, Gartner, http://www.gartner.com/newsroom/id/2466615
7. Moschella, D., Neal, D., Opperman, P., Taylor, J.: The 'Consumerization' of Information Technology. CSC Research White Paper, El Segundo (2004)
8. Generation, Y.: They've arrived at work with a new attitude. USA Today, http://usatoday30.usatoday.com/money/workplace/2005-11-06-gen-y_x.htm
9. The consumerization of IT: The next-generation CIO. PwC Center for Technology and Innovation Report, 3-6 (2011)
10. Gillett, F.: Forrester, Employees Use Multiple Gadgets For Work — And Choose Much Of The Tech Themselves, http://blogs.forrester.com/frank_gillett/12-02-22-employees_use_multiple_gadgets_for_work_and_choose_much_of_the_tech_themselves
11. Mcree, J.: UX for Enterprise: Part I. Universal Mind, http://blog.universalmind.com/ux-for-enterprise-part-i/
12. Krieger, M.: Revolution or Evolution: What is Big Data's impact on enterprise information systems? Enterprise CIO Forum, http://www.enterprisecioforum.com/en/blogs/mrkrieger/revolution-or-evolution-what-big-datas-i
13. Pierce, B.: What is Object-Oriented Programming? Types and Programming Languages. MIT Press (2002)
14. Kolko, J.: Abductive Thinking and Sensemaking: The Drivers of Design Synthesis. MIT's Design Issues 26(1) (Winter 2010)
15. Nazarian, S.: Enterprise UX - How can we transform enterprise user experience to be more mobile, flexible, and responsive? Design Mind, http://designmind.frogdesign.com/articles/enterprise-ux.html
16. Saffer, D.: Designing Microinteractions, Microinteractions. O'Reilly, 2 (2013)
17. Gmail, http://www.gmail.com