Catapult Systems
The Microsoft Consulting Company

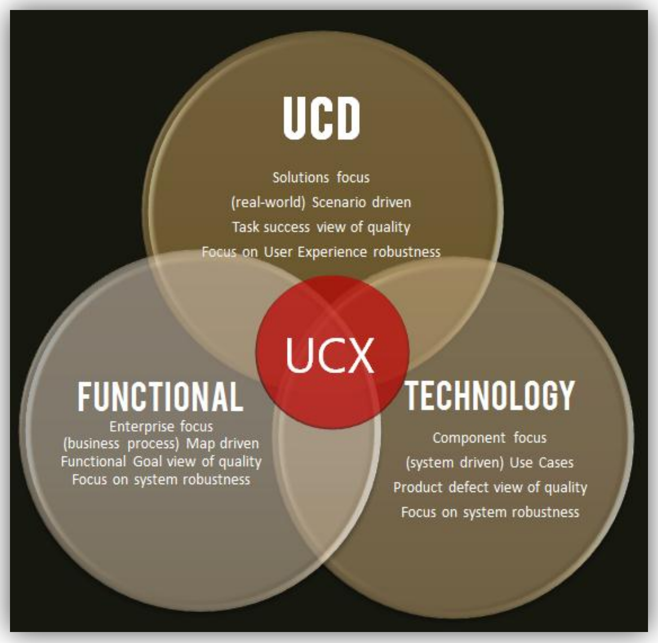
Catapult Whitepaper

User-Centered Experience (UCX)

Catapult’s holistic approach to building a better software product

Revision 1.0
January 1, 2012

By Daniella Matutes


UCD
Solutions focus
(real-world) Scenario driven
Task success view of quality
Focus on User Experience robustness
UCX
FUNCTIONAL
Enterprise focus
(business process) Map driven
Functional Goal view of quality
Focus on system robustness
TECHNOLOGY
Component focus
(system driven) Use Cases
Product defect view of quality
Focus on system robustness

What is User-Centered Design (UCD)?

UCD is a *goal-directed approach* to building systems that puts the user at the center of the design process.

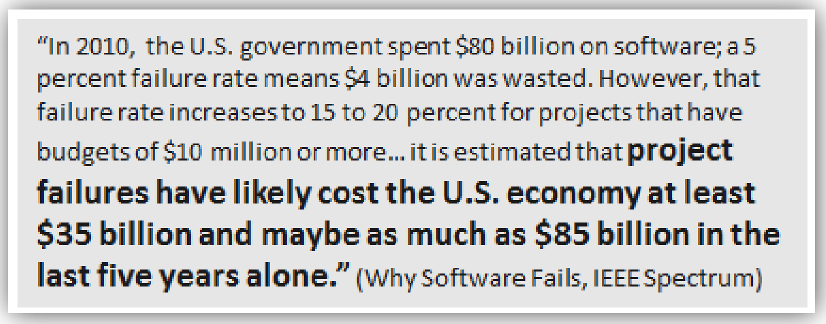
Brilliant, simple, easy-to-use, engaging software doesn't just happen by following a process and hiring savvy engineers. It requires focusing on people (business, stakeholders and users) from the very beginning and during each step in the design, development and deployment of a system.

It is a philosophy that focuses on the user and the organization as opposed to the technology; it is a process that focuses on people and the way they think- perception, memory, learning, problem-solving, as they interact with software systems.

Why should your organization care about the end-user?

Do you want people to use the system your organization just spent x number of dollars and months to create; or, do you want another lonely icon to occupy space on a desktop? User-Centered Design is not a new approach. In fact, companies like Microsoft, IBM, HP, Nike and Apple have created many of their most popular systems and products through User-Centered Design. Countless studies and statistics all point to a very clear conclusion- designing systems with the goals and tasks of the user at the forefront creates software that delights the user and surpasses business goals.

We all know that software projects fail, many of us have "been there and done that". Even when software doesn't ride the fail whale into the sunset, it often finishes late and over budget. To add insult to injury, most of the time software is not used "right" (if it's used at all).



"In 2010, the U.S. government spent $80 billion on software; a 5 percent failure rate means $4 billion was wasted. However, that failure rate increases to 15 to 20 percent for projects that have budgets of $10 million or more… it is estimated that **project failures have likely cost the U.S. economy at least $35 billion and maybe as much as $85 billion in the last five years alone."** (Why Software Fails, IEEE Spectrum)

UCD dramatically increases your ROI in the following four ways:

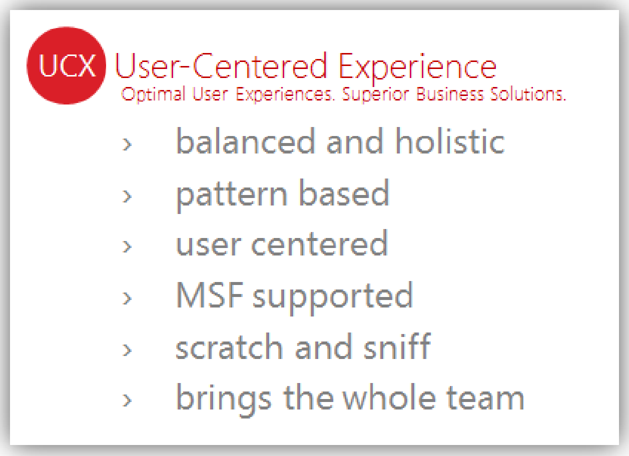
1. Changes the way requirements are gathered and documented- reducing by more than half.
2. User training and help documentation is greatly reduced or completely eliminated.
3. It ends development team churn and rework.
4. Greatly reduces and/or diminishes maintenance costs.

UCD significantly reduces delivery risk in the following four ways:

1. Provides quick, low-fi prototypes: agile iterations of ideas that are socialized with users on day one of the project.
2. Prevents problems instead of "fixing" them after the fact
3. Trumps the "That's not what I thought it would do…" conversation.
4. The goal of a 100% user adoption is intertwined throughout the process, not just added on as a metric at the end.

These are bold claims! Can I see proof?

Sure! There are multiple studies and statistics all over the internet that talk about the effectiveness of the UCD approach. If you are really interested, we have included an Appendix of Platitudes at the end of this document that supports each of the above statements. But for now, please read on as to why Catapult's delivery model of UCD is different than the rest.



UCX User-Centered Experience

Optimal User Experiences. Superior Business Solutions.

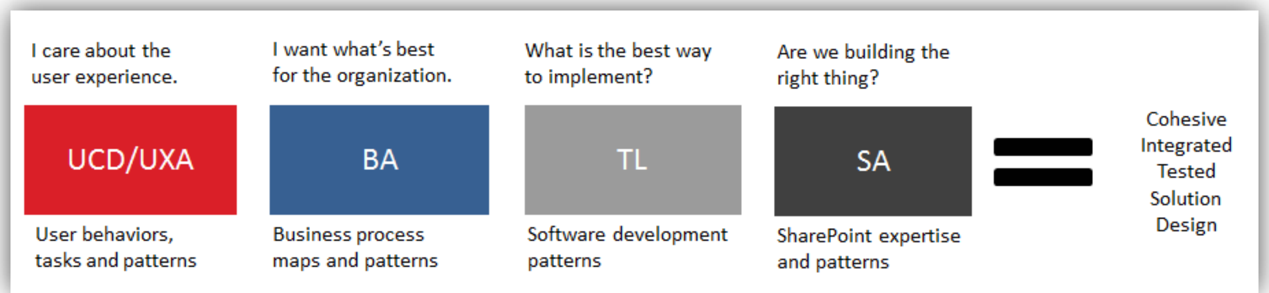
- balanced and holistic
- pattern based
- user centered
- MSF supported
- scratch and sniff
- brings the whole team

You just said this isn't a new process, so what makes Catapult any different?

UCD is an award-winning approach that many best-in-class companies, products and services use to create software systems that delight the user and surpass business goals. So what's different about Catapult and why is our UCX methodology better?

Catapult integrates technology-focused goals, functional requirements and User-Centered Design into one cohesive process, ***a User-Centered Experience***. Our project teams follow a balanced and holistic approach by blending the user community, the business owners and the development teams into a cohesive force that builds upon the experiences of many mental models. The system fits the needs and expectations of the user while solving business problems because there are multiple perspectives on the Catapult Project team. UCX puts people first and uses technology as a tool. Users are the constant; their knowledge and behaviors are the value of the organization. We use technology to maximize the behaviors and access the knowledge for higher productivity within the system.

These modalities all work in parallel- feeding and supporting each other. This integration provides a common vocabulary for our team so that they can work towards the same goal while still coming at it from their own perspective.



I care about the user experience.

UCD/UXA

User behaviors, tasks and patterns

I want what's best for the organization.

BA

Business process maps and patterns

What is the best way to implement?

TL

Software development patterns

Are we building the right thing?

SA

SharePoint expertise and patterns

=

Cohesive Integrated Tested Solution Design

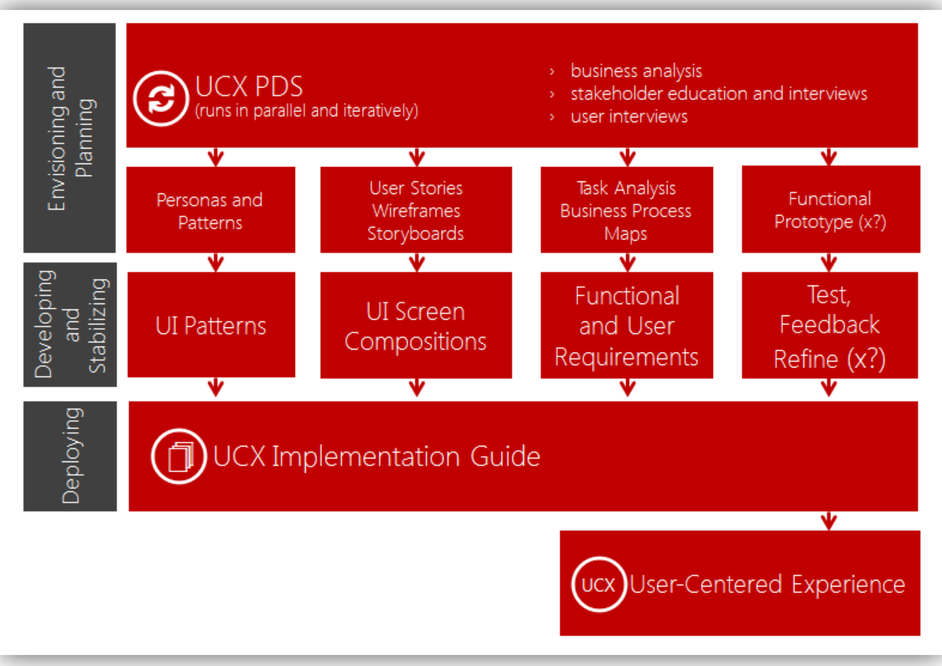
Pattern-based

Our approach is simple, structured and pattern-based. This means that the project benefits from:

- *Repeatability* - UCX builds off of past successes.
- *Problem prevention* -these tested patterns leverage a best practice approach every time.
- *Scalability* - the same (proven) patterns can be used to solve one problem or 1,000 problems.
- *Saving time and effort* - software patterns combine with UX patterns to create simple, elegant solutions in half the time.

User centered

UCX maps the system directly to the way users think and work- intuitive and easy- so people actually use the system with minimal training. Imagine a software system that works with users to help the organization grow, anticipate issues and act proactively.


Envisioning and Planning
Developing and Stabilizing
Deploying
UCX PDS
(runs in parallel and iteratively)
business analysis
stakeholder education and interviews
user interviews
Personas and Patterns
User Stories Wireframes Storyboards
Task Analysis Business Process Maps
Functional Prototype (x?)
UI Patterns
UI Screen Compositions
Functional and User Requirements
Test, Feedback Refine (x?)
UCX Implementation Guide
UCX
User-Centered Experience

MSF supported

Microsoft Solutions Framework is a set of principles, models, disciplines, concepts, and guidelines for delivering Microsoft Solutions. It provides a trusted, repeatable framework that dovetails perfectly into Catapult's UCX solution design. This ensures a predictable, stable, scalable system that integrates killer technology, proven processes, business goals and user needs.

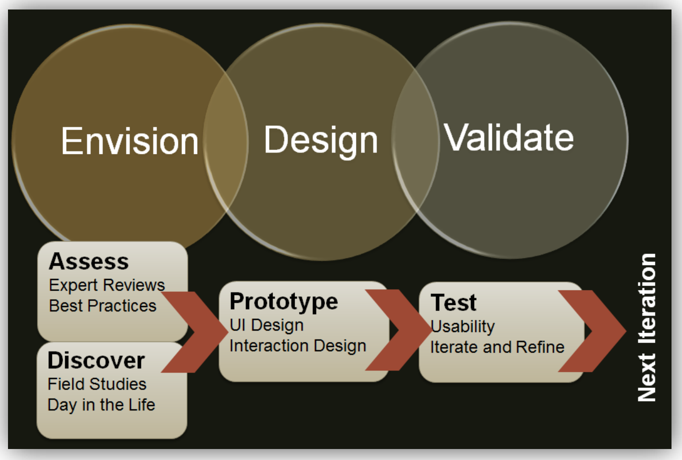
Active involvement

We maintain a constant connection with the project through communication and checkpoints with the organization and the users. UCX creates a project that:

- *Focuses on transparency* - the client is involved in gathering and validating requirements, prototype reviews and testing so that they can see and feel how the system works before build and deployment.
- *Proactively solves problems* -the client is involved in every step of the process- prebuild, build and post build, identifying problems and providing feedback.
- *Ends rework* - eliminates the post-release conversation of, "this is not what I thought I would do."
- *Builds user adoption* - starting with project kick-off all the way until test and deployment, the user is involved.
- *Removes the black box* - the client is involved and validating the system, from the kick-off to the user acceptance testing.

Solving problems, not just building systems

Positive engagement leads to greater pleasure and effectiveness for your users, and negative engagement leads to difficulty, displeasure and wasted time; it's easy to imagine why engagement and good UX are important. Building software systems that help users accomplish their tasks with little or no frustration is especially important. Those tasks are directly connected to important organizational goals. Eliminating any friction experienced in achieving the user's goals will greatly reduce the resistance against accomplishing that important business goal.



We bring the whole team

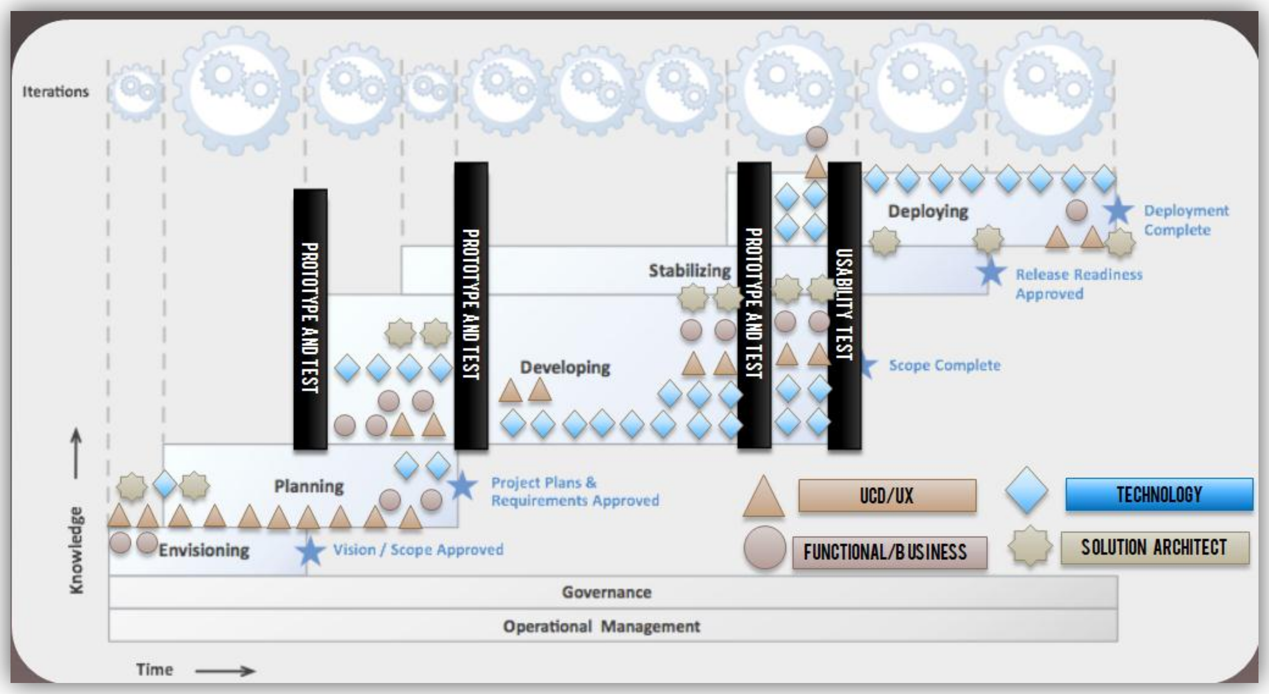
Catapult project teams are blended, working side-by-side, from the very beginning. UCX teams work as a tightly woven net and nothing is "thrown over the fence."

During traditional system design and development process, there is a dreaded "hand-off" between the Design Team and Development Team. The Design Team visits the client, holds meetings, interviews users, and creates interaction diagrams and personas. They record all of this valuable information into a static, linear, hundred-pages-long thesis of how the system should, look, feel and operate.

This manuscript is “thrown over the fence” to the Development team. Upon receipt, a few tense meetings occur and the Development Team goes off to build the system with disdain, doubt and disbelief.

UCX is different due to three main intentions:

- Accountability: the project team is accountable to each other and to the client during the entire project lifecycle. Each professional on the team has to work together to create the solution.
- Depth of experience: it’s not just the project team, we draw on the experiences of over 300 consultants company-wide through our pattern libraries and shared experiences.
- There is no requirements vacuum. The client is involved and validating the system workings during the entire SDLC.



In summary

UCX isn’t just an approach for Catapult and it’s more than a methodology; it’s a belief. It’s the conviction that people really matter. UCX asserts that in order for a solution to be successful, whether it’s an Information Workplace, internet site, business intelligence platform, or a customer relationship management system, it must engage, educate and entertain people while serving the organization’s vision and needs. UCX is a commitment to create software systems for real people - who have flaws, emotions, wants and desires- who work at real organizations.

This unique approach seeks to answer questions about users and their goals and then use these findings to drive system development and design to meet the needs and vision of the organization. Catapult integrates User-Centered Design, a diverse team of project stewards and Microsoft Solutions Framework into one methodology that consistently produces systems that surpass business goals and delight the user.

Appendix of Platitudes

Dramatically increase your ROI.

"Inadequate uses of UCD methods in software development projects have been estimated to cost the US economy about $30 billion per year in lost productivity." (Tom Landauers, The Trouble with Computers)

Saves money and time throughout designing, building, deploying and testing.

"The rule of thumb in many UCD organizations is that the cost-benefit ratio for UCD is $1:$10-$100. Once a system is in development, correcting a problem costs 10 times as much as fixing the same problem in design. If the system has been released, it costs 100 times as much relative to fixing in design." (Gilb)

The requirements gathering time and effort are reduced by more than half.

"When managers were polled regarding the reasons for the inaccurate cost estimates, the top four reasons were issues that could have been addressed by following best practices in UCD. These include frequent requests for changes by users, overlooked tasks, users' lack of understanding of their own requirements, and insufficient communication and understanding between users and analysts." (Barker)

There is a significant reduction in delivery risk.

"Savings from earlier vs. later changes: Changes cost less when made earlier in the development life cycle. Twenty changes in a project, at 32 hours per change and [a minimal] hourly rate of $35, would cost $22,400. Reducing this to 8 hours per change would reduce the cost to $5,600. Savings = $16,800." (Human Factors International)

User training and help documentation is greatly reduced or completely eliminated.

"At one company, end-user training for a usability-engineered internal system was one hour compared to a full week of training for a similar system that had no usability work. Usability engineering allowed another company to eliminate training and save $140,000. As a result of usability improvements at AT&T, the company saved $2,500,000 in training expenses." (Bias & Mayhew)

It ends churn and rework.

"The average amount that engineers spend on rework is 50% of their time." (Why Software Fails, IEEE Spectrum)

UCD trumps the "That's not what I thought it would do…" conversation.

Because UCD puts the user at the center of the design process, it dramatically reduces the chance of unfulfilled expectations and unmet needs. "To build a model intranet, Bay Networks spent $3 million and two years studying the different ways people think about the same thing. The result: all think alike about the $10 million saved each year." (Fabris)

Greatly reduces and/or diminishes maintenance costs.

"One [well-known] study found that 80 percent of software life-cycle costs occur during the maintenance phase. Most maintenance costs are associated with "unmet or unforeseen" user requirements and other usability problems." (Pressman)

Increases productivity and job satisfaction while decreasing customer support needs and documentation.

"Poor user interface design can have a significant effect on user productivity. Consider a very simple transaction, such as filling in an on-line data entry form. Suppose an organization has 20 users, who perform this transaction approximately 80 times a day (quite typical for data entry clerks or other high frequency users). This adds up to 368,000 transactions per year (20 users working 230 days a year, performing 80 transactions per day). If a screen could be redesigned to reduce the transaction time per screen by 10 seconds, a savings of 1022 hours, or 25.5 person weeks could be realized. If improvement on a single screen of the system could increase productivity by 1/2 of a person-year, clearly improvements across the whole system will have a very dramatic effect on productivity." (Deborah Mayhew, Cost-Justifying Usability)