

# An overview of Customer Experience Engineering

This paper discusses Customer Experience Engineering (CXE), a new discipline that addresses key shortcomings in the current practices around customer management.

# Why are Customer Success initiatives often such failures?

Our research of businesses across various industries illustrates that Customer Success initiatives lack consistent ROI. We believe that the heart of the problem is a lack of discipline and inclusiveness in the approach many businesses take to achieving Customer Success and retention initiatives. Customer Success initiatives are often conceived as either the sole concern of Marketing or a low-level support team, after development activities are complete. These CS initiatives are often characterized by a layering of tools atop application software to “measure” customer success after-the-fact, or tools to gather sentiment and input from customers after they have “experienced” the application.

Ironically, there is great consensus among those engaged in the CS industry that customer success begins with a focus on customers, outcomes, and metrics, but there appears to be little rigor, in the approaches they adopt. The Chassi team has defined a new approach and market category that addresses this lack of discipline. In the process, we have defined a path that results in reliable, repeatable, and measurable customer success. We call this new discipline Customer Experience Engineering (CXE).

This paper will lay out the basic precepts of Customer Experience Engineering, a new discipline that addresses key shortcomings in the current practices around customer management. Although these practices benefit all organizations, this paper will pay specific attention to software businesses where the problems of distance between business and customer are particularly pronounced.

# Introduction to CXE

Customer Experience Engineering (CXE) is a new discipline that integrates elements of process modeling, business activity measurement, value stream mapping, and user-centered design to create a reliable and proactive approach for planning customer experience, and ultimately, customer success. For years, software engineering has focused on similar disciplines such as Agile, XD, JAD and many others that outline successful software projects. These software development disciplines prescribe precepts that help focus and specify the nature of the software being built, and the method by which it is created, resulting in testable and verifiable acceptance criteria which are used to determine whether the software works as anticipated. CXE takes a similar approach, but it focuses entirely on the individual customer. Specifically, CXE focuses on the value derived by the customer and the key characteristics of the customers' process in order to establish a firm basis for a customer experience. Furthermore, it creates an open repository of customer action and metrics to be used company-wide to evaluate and act on behalf of the customer prior to a problem emerging.

CXE is a mindset and engineering-like approach to create the outcome of customer success in much the same manner that software engineering seeks to create applications. It is characterized by its deliberate and iterative approach to creating points of measurement, engagement, and interaction between the business and its customers – built into the applications and processes that face them. CXE ultimately focuses the entire organization by breaking down the silos that have historically been used to define customer facing functions and by creating a platform and knowledge base to inform the organization as a whole.

There are significant differences between software engineering and customer experience engineering. Software engineering is focused on particular endpoints within software itself— how it should work, how it should look, and what it should do as defined by the

entity creating the software itself. Software engineering is normally internally focused, based on the needs of the enterprise that conceives it and the technology chosen that constrains it. This leads to a very narrow focus. It presupposes that the mindset of a user and that of the software's development staff is well aligned. Despite advances in user centered design, this is often not the case. In fact, users often choose unpredictable approaches to software. As a consequence, their experience is not uniform, and can often be surprising. Furthermore, a user's experience is a combination of many factors including their ability to accomplish some goal using the software, the speed with which they're able to accomplish this goal, and the difficulty with which that goal was attained.

In contrast, CXE focuses externally, on the customer as an individual with their own idiosyncratic needs, biases, and preferences. Customer experience engineering is an attempt to match the actual impact of software with real life, discrete, and measurable events in the life of a user. The discipline of customer experience engineering creates a quantifiable measurement of customer progress as they attempt to reach the goals supported by the software. By beginning with customer journey mapping, CXE creates a framework around which instrumentation can be placed to measure progress by a customer along a particular journey towards a goal. Imagine looking at a city map. There are many ways to get from point A to point B. Depending on your particular goals, one route may be preferable to another, but all routes are equally valid. Which route you choose and your judgment of it as successful depends on your particular goals. The same can be said for software where the precedents of activities are not always clear. What is necessary to be successful at any given step in a process supported by software is based on the user's particular situation, training, even the state of the system itself. Some data may have had to be entered first, or perhaps the user is expected to understand the data to be entered at their current step. In any case, the designers of software make hundreds of little decisions and assumptions throughout their development process that affect the user's ability to navigate and achieve their own goals. Therefore, CXE begins with painting the entire map of possible paths that might be used and all states that implies. That map is the basis for the analysis of the points along the way that provide value to the customer. Having developed a comprehensive vision, measurements can be taken as to where in that map users actually stand at any point in time.

Unlike brick and mortar type interactions, the business of software is plagued by a tyranny of distance. The use of the software almost always occurs physically apart from the business that supports it. This has led customer success organizations to adopt an entirely reactive approach. Problems are reported by users after they are encountered, and the loudest voice usually garners more attention than the others. CXE bridges this distance by reimagining customer experience as a confluence of factors, each measurable and integrated into the software itself. This enables software businesses by exposing the data collected in real time, as customers follow a journey through the software. Now, even quiet users who are encountering difficulties will generate a real-time signal back to the company that developed the software, causing either an automated or human response. For example, a user who is taking a longer than average time to complete a process could be contacted by a chat bot or forwarded to someone in customer support who can intervene. The triggers or alerts may also be a source of data in that the software processes may themselves be suspect if similar conditions persist or trend across users in the same part of their journey. This personalizes the experience for each customer, harmonizes departments within the company to jointly focus on that individual, and maximizes the opportunities for that customer to be successful (and therefore loyal).

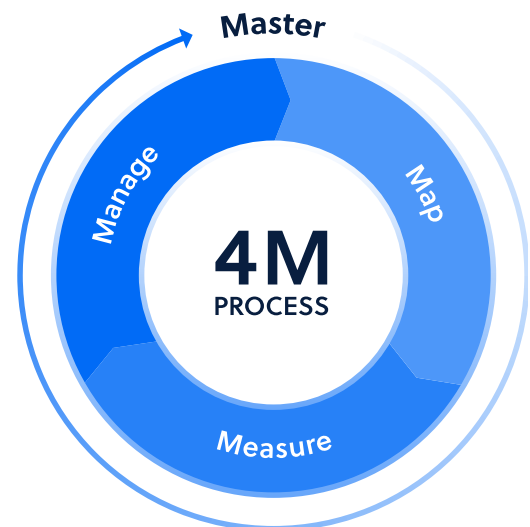
CXE goes far beyond the currently accepted practice of customer experience management which appears to target only the user interface and not the actual customer experience. Many so-called User Experience tools tend to instrument the activity in a user interface (or “front end”) of a system. This shows which buttons were clicked, which menus navigated, the timing of activities taken, etc. These tools appear to give some insight into customer behavior by looking at their activity rather than their outcomes or the impact their activity has made on progress towards their goal. Some proponents of retrospective management even include surveys and other subjective elements in the analysis that they offer as substantive data on customer experience. Subjective data is very difficult to make actionable and is generally unreliable because it is affected by many external factors, including mood, coincident stress, social pressure, etc.

CXE is a revolutionary rejection of these practices and thinking. Experience can and should be deliberately engineered, not just as a response to a software user interface but rather a holistic, ongoing, and rigorous approach to defining the delivery of value and

the measurement of key performance indicators that characterize the process. Activity and sentiment alone are generally poor indicators of success. Furthermore, by applying a lean approach to designing the experience from the outset, we believe that substantial improvements can be made to the reliability and repeatability of customer success. Borrowing from software development methodologies, quality customer experience must be built into each process and all customer touch points throughout an organization in order for it to have substantial impact.

This is perhaps why CXE is only now coming into wide acceptance. It is simply easier to measure activity than it is to proactively and rigorously define customer facing interaction throughout the development process. However, so many customer success projects ultimately fail to make a meaningful difference to customer outcomes, and the typical reaction to that failure is to double down and hire more support staff. Knowing which customers to call and staffing up to do so is not nearly as effective as interceding before customers face difficulties or delays. Adding customer support labor is expensive, reactive, and often ineffective.

At its core, CXE embraces a process: “Map, Measure, Manage, and Master.” This approach was developed by consolidating our experience with customers. It embraces an engineering analysis of the customer journey beyond the states of the user interface, usually starting with the customer expectations and goals and ending with a plan to instrument the journey(s) taken by a customer to provide the necessary measurements.



First, we create a map and analytical framework for the entire customer experience. From there, we collect data within that framework and manage the events that are indicated from that data by orchestrating responses to customer requirements either within the software or using third-party tools. Then we master the process through iteration and ongoing refinement. This is an ongoing, enterprise wide commitment that is vital for success.

So, what are the precepts that must be followed in order to make CXE real within organizations? The rest of this paper will lay them out in very broad detail. Much more remains to be written about each one of these new laws and, for that matter, the techniques that we discuss below. This is the beginning of your own journey. Reducing the distance and increasing the symbiosis between your customers and your software business hinges entirely on your ability to figure out and implement an approach that is sticky by design.

# The Laws of CXE

## 1. Know your customer

In order to fully consider a value stream, you have to know which parts of the customer's company are affected by the software, and specifically the personas and specific goals of the individuals using your software. This is easier said than done. Approach this as a hierarchical exercise, which may start with the customer as a broad definition, but should quickly move to the roles, educational profiles, biases, and prior experience of customers as individuals. At first, this will look pretty chaotic since no two customers are identical. As the analysis progresses you will find similarities, especially as they relate to the reasons why these individuals are using this software to begin with. So many companies seem to miss the opportunity to create synergy with their customers by identifying the core motivations and underlying expectations faced by their users. Beware if your Customer Success processes occur without an acute awareness of the differences, even more than the similarities, within this group. Define and subsequently refine the personas that represent your user base. Go deep and get specific. Revisit this as your market grows.

## 2. Know your own value and where/how it is created

Where is the value created within your process and your software? What is essential to convey this value? What activity supports these essential milestones? Value stream mapping and fish boning are well accepted techniques for business analysis. Apply these disciplines from the customer perspective. Sometimes just a shift in this perspective provides valuable insights to the design team. When applying CXE it is absolutely vital because knowing where in your process value is created informs where measurements should be taken to monitor it as it occurs.



### **3. Comprehensively map the full customer lifecycle**

In order to be useful, the customer lifecycle must be comprehensive, including all possible steps to be taken as a user gets value from the software. An exhaustive approach is necessary so that a customer's progress can always be defined. If you leave out steps or end-states of the lifecycle, you lose the ability to track all customers. Despite the likelihood of any particular state, defining them rigorously up front, and then re-checking their current completeness over time provides a complete framework for analysis of each customer's outcomes. Once drawn, value delivered (positive or negative) can be attached to any given lifecycle step. Likewise, additional information can be laid atop this framework, including expectations of timing, volume, and messaging attached to each step. This framework is the essence of CXE because it lays out the entire experience in an objective and rigorous manner.

Keep in mind that the customer lifecycle isn't just software-based and can also include elements of customer facing activity from other parts of your organization. Expectations set by Sales, technical support calls made to your help desk, and every other customer facing function should be incorporated as they all are considered at the point at which a customer sits down to use your software.

Return to the lifecycle continuously as you learn about the actual usage of your software and keep it updated. As new features are added your lifecycle will change. It is as dynamic as your software, so don't consider a "one and done" approach here.

### **4. Consider analysis at the outset, and always**

The purpose of CXE is to enable analysis based on holistic user data. We will be collecting data as each user navigates the software and attempts to use it. As this occurs, we intend to do an analysis both in real time and retrospectively. We must plan for this analysis so that we can make sure we have all requisite information collected. This requires a lot of forethought. Just as we considered various paths a user might take along a lifecycle, so should we consider the information of importance to complete an analysis and evaluation

of that user's experience. Remember that experience is not the same thing as activity and a frustrated user may look like a very active one. Therefore, how will we know when value is conveyed and how will we compare baseline measurement to ongoing process improvement? The answers are as unique as the software to which CXE is applied. Pay particular attention to the quantifiable aspects of the software itself. For example, if the purpose of the software is to automate a business process, how much time must the user spend compared to what they would otherwise spend outside of the software? The difference represents quantifiable value. Be ready to measure it. Again, this process is iterative and will change as the software matures and as our understanding of the user matures.

## **5. Define the data to be captured**

In addition to the analysis, we should include all of the contextual data (metadata) that gives meaning to the data collected as this software is used. Sometimes this requires identifying prior experience, external events that impinge on customer experience, system performance and other factors. A customer's experience is not a unitary thing — it exists in context, so capture all contextual data that might be useful in analysis to help explain the behavior we are trying to influence.

## **6. Define process controls and indicators**

What process controls are important to your user's experience? Do system outcomes depend upon the volume of concurrent usage? Does concurrent load indicate much about individual user experience? What is the effect of attempting to use this software in any non-ordained sequence? By focusing on controls that add to customer experience, and creating indicators that might show user problems before they become technical support requests, we can anticipate events and consequent reactions so that the user is never stranded or left to wonder, "What next?"

## 7. Define orchestration in keeping with customer preferences

CXE offers an opportunity to create an automatic reaction to various customer experience endpoints and processes. Alerts, triggers, and similar responses to customers who, for example, either take too long, miss steps, or fail to experience the value of the software, can be automated. Human intervention is also possible, depending on the company's preference. CXE embraces orchestration as a key tenet of its purpose. By creating action based on the data we collect, the framework developed at the heart of CXE becomes an enterprise resource, drawing together and focusing the efforts of a myriad of client-affecting resources on the same endpoint – the user. Even offline activity can be brought into alignment with online activity and the entire enterprise can share in the rewards brought about by knowledge of the customer's true progress and the value provided. Imagine the ability to create stickiness within Sales at renewal time that comes from the ability to actually prove the utility provided so far.

## 8. Do no harm

Just as orchestration creates the opportunity for action on behalf of, and in service to the customer, it may also provide distraction or concern if done without customer consent or at inappropriate times. Remember "Clippy?" Clippy was an iconic addition to the Microsoft Office suite some years ago, and it was meant to provide "helpful tips" and interact with the user to make adoption of newer features a bit easier. Many people felt it was intrusive and bothersome and so disabled it entirely. It became something of an embarrassing joke and a reminder that helpfulness is in the eye of those being helped, not some oracle that stands above and unilaterally decides where help is needed.

Engagement and orchestration require a sensitivity to customer preference and context. Helpfulness will be judged by your users, not by you, and the lighter touch may be better for users who are not "stuck" or "in trouble."

## 9. Use your data (a lot) and continuously tweak

CXE was meant to be performed in iterations. It is usually the case that the first implementation may miss the collection of some data, or mis-identify journeys through a lifecycle. For example, do not expect that an early stage, first-release application will get the CXE tuning correct the first time through. Adding measurements is common once data begins to flow and coalesce. The CXE discipline is best served when it is approached as an evolution of understanding, drawing from the data collected and building upon it as inferences can be made. CXE demands that you watch your customers as closely as possible, testing hypotheses until you have a relatively correct model of their behavior. The analytical approach CXE defines provides value from the outset, but this value builds over time as many customers take similar journeys through the lifecycle framework. Regular observation of the data will change the approach you make for adding new features and judging their usefulness. Keep at it.

## 10. Do not permit silos

Perhaps the most important consideration is that CXE demands an enterprise approach. Customer Success is not a department, it is an ideal shared across departments. Perhaps this is the greatest failing of other CS-enabling approaches – they tend to reinforce the thinking that CS is an afterthought, or something solely in the portfolio of Product Development or Product Management, or worse still, Information Technology. CXE is a team sport, and it requires the collaboration of many enterprise functions – Sales, Marketing, Product, Operations, Support, and yes, IT. The more central the adoption of the CXE process, the more likely it will be to succeed.

# Summary

CXE is a new discipline, one that requires careful planning, deeply considered analysis, and the collaboration of departments in order to be successful. It is ironic that so many Customer Success tools fail to move the needle on customer success. Perhaps this is because they appear to be an application of technology after-the-fact rather than a disciplined approach to mapping, measuring and managing of quantifiable information. In any other context, this would be obvious.

Imagine, for example, an application that attempted to implement financial controls by adding software or collecting survey results after the money was already spent. It would be a ridiculous prospect. Customer success is the same – a business outcome that should be assured by the nature of the management of the controls that define it. It is now time to take that potential seriously as we embrace this new capability.

## Where to start?

Once you've decided to modernize your Customer Success cycle and begin to implement CXE within your own business, you should begin to assess your current practices and compare them to the precepts discussed here. It also helps to engage experts like the people at Chassi in order to frame the work to be done and discuss methods to capture and act on the information you will gather, like the Chassi Platform product.

CXE is not difficult or elusive. In fact, because it borrows from existing disciplines of analytical thought and measurement, it is actually quite easy to learn. Putting the pieces together and getting a result that is reliable and predictable does require some training and guidance, so at least the first projects or so can be expected to leverage this kind of expertise.

# About the Authors

The Chassi team is comprised of software industry veterans. Our six-person executive team are all software industry experts with more than 125 years of combined experience building, launching, and scaling successful software businesses. From founding and scaling startups like WebPT and DrFirst, to leading within large enterprises like Siebel Systems, BlueCross BlueShield, and Oracle, the Chassi team is intimately familiar with the challenge of staying ahead of unforgiving customer expectations and market demands.

We have launched successful applications and enterprises that are collectively worth billions and we have the scars to prove it. Since the dawn of the cloud, we've experienced the market focus shift from 'product capabilities' to 'customer experience', and we've been wrestling with this problem for a long time.

We're business operators, product leaders, and technologists who are excited to usher in the next evolutionary leap in customer experience with CXE.



[chassi.com](http://chassi.com)

[info@chassi.com](mailto:info@chassi.com)