The LUCID Framework

An Introduction

Charles B. Kreitzberg, PhD.

Cognetics Corporation
PO Box 386 Princeton Junction, New Jersey 08550
609.799.5005 x235
www.cognetics.com
charlie@cognetics.com

The LUCID Framework

You are developing an interactive product – typically a software application, a web site or an electronic device – and you turn to The LUCID Framework™ because you want to get it right. Getting it right means that:

- the functionality of the product reflects your users' real needs and desires
- the product is simple to learn and use – organized the way your users think and expect
- the user's experience is uniformly positive – creating loyalty to the product and its brand
- the product supports your business goals – generating the desired return on its investment
- the development process is straightforward and without surprises – delivering the envisioned functionality within the allotted time and budget
- the product has design integrity – making sense in both its broad dimensions and its details

You know that getting the product right can be difficult; that, on the whole, only about 15% of software projects come in on time, on budget and with the originally envisioned functionality. The record is probably equally bad for other types of interactive products. And you know that few products are truly usable: easy to learn and use, efficient, engaging, and error tolerant. You want your product to be one of these truly effective products.

Imagine a software, web or interactive product that, from the moment you install it, surpasses all your expectations. When you configure it, it installs easily and flawlessly. Every set up option is clear and relevant. When you first use it, you find it intuitive, flexible and powerful. Like a fine car, it's fast and responsive. The screens are beautifully organized and easy to read. Every prompt and message is clear. You rarely encounter problems, and when you do the solution is at hand. When you occasionally need assistance, you are supported with thoughtful help that really explains. As you become familiar with the product you find shortcuts that make your interactions easier and faster. Over time you learn increasingly sophisticated operations and customize the software to fit your needs and your preferred style of working. And should you stop using the software for a while, support to recapture your past fluency is right at hand.

Are such software and web products common? Sadly, no. Is such software possible? Absolutely! Just as products can be designed to high standards of technical excellence, they can also be designed to create a superb user experience. As with technical quality, creating an outstanding user experience requires planning, analysis and testing. And just as carefully following an engineering framework supports a high quality technical product, following The LUCID Framework™ will help you achieve a highly usable one.

The LUCID Framework™ will help you create the products you envision. And it will save you time, money and frustration. We know this is true because we (and others) have been using and refining The LUCID Framework™ for more than a decade. There is no magic

around The LUCID Framework™. It is not a silver bullet. Using LUCID you will produce a better product in less time and at lower cost because you'll be working smart.

Most software products fail before the first line of code or HTML is written. The killer problems are far more often related to people and managing the process than they are to technology. Choosing the wrong team members, poor communication, failing to recognize and manage risk, dueling agendas, shifting priorities, scope creep, inadequate executive support, poorly thought-out milestones, and sloppy specifications are among the manageable problems that derail projects and degrade products. The LUCID Framework™ will help you avoid all these pitfalls.

Whether you are developing products for internal use, for partners or selling to customers, adopting LUCID can save your organization significant time and money and help avoid costly mistakes and re-work. Followed faithfully, it will guarantee a highly usable and engaging product and a uniformly positive customer experience. And LUCID is easy and satisfying. Teams that use LUCID are amazed at how rapidly they make progress and how readily they cut-through problems. Managers see reduced risk and better performance. Executives see progress on strategic objectives. Finance sees lower development and support costs. Internal customers experience less training and ease of use. Marketing and sales get products that address the real needs of the marketplace. The LUCID Framework™ will transform the way that you develop products and help you get it right the first time.

What is LUCID

LUCID is a framework for conducting the activities that define and shape interactive products. It was originally created to support development of highly usable software and web sites. It has also been used for other types of interactive products. LUCID guides you decide what functionality the product requires. It not only focuses on *what* the product should do but on *how* functions are best designed from a user perspective. This is important because traditionally, many products have been created by constructing lists of functional requirements and then "checking them off". This practice that fails to recognize that good product design is more than simply an inventory of capabilities. Products need to work in an intuitive, integrated way.

A key principle of The LUCID Framework is that products should be designed to offer users the "five E's:" effectiveness, efficiency, engagement, error tolerance and ease of learning.

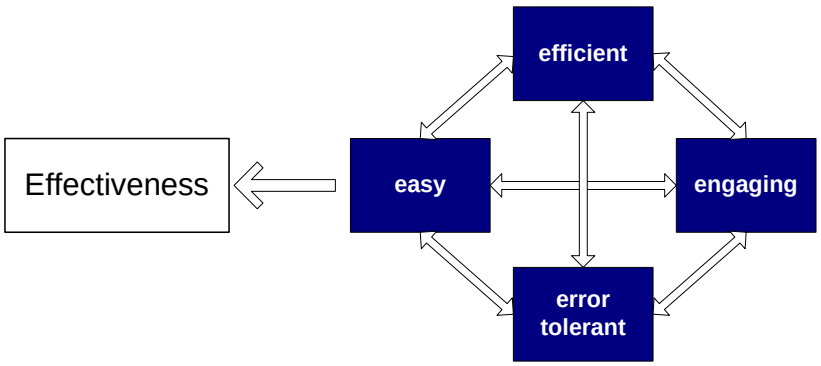


FIGURE 1: THE 5-E'S

There is never any ambiguity about people's desire for usability. Users always want a product to be extremely usable. But conflicting goals often make usability difficult to achieve. Conflicting agendas among the various stakeholders, technical constraints, time and budget pressures, legal considerations and miscommunication, all take their toll on usability.

Even in the simplest environments, there is often a conflict among technical considerations, business goals and user needs. For every project, there is a "sweet spot" that represents the best balance among the conflicting goals.

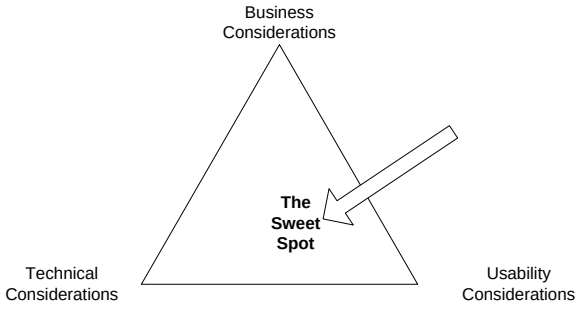


FIGURE 2: THE SWEET SPOT

Typically, the business players are interested in such element as costs, revenue, market share, and time to market. Technology players are concerned with such elements as ease of development or configuration, maintainability, and reliability. Users are interested in extreme usability – products that deliver the functionality they envision in an efficient and easy to use package. The LUCID Framework helps you balance these pressures and preserve usability in what can easily become a chaotic environment. It accomplishes this by:

- defining the processes and tasks that support usability throughout the product development lifecycle.
- providing templates for gathering data and producing outputs.
- defining roles and responsibilities of the LUCID Team

- suggesting when management reviews are appropriate.
- helping with communication and collaboration among all the stakeholders and, when collaboration breaks down, troubleshooting the problem and getting the project back on track.

Because every project is different, LUCID has been designed to be both flexible and adaptable. It suggests what activities need to be performed and their sequence but it does not mandate specific techniques for accomplishing them. It provides decision templates to help structure the activities for a particular project. Over the past two decades many powerful techniques for user-centered design have been created. LUCID provides a framework for selecting and sequencing appropriate techniques but it does not replace them. The goal of LUCID is to leverage the best thinking of the user-centered design community.

LUCID is also scalable; it can be applied to projects of different sizes. Managing a project with three developers is different from managing one with 30 developers and both are different from managing projects with 300 developers. As the size of the project team grows, the complexity of communication increases, so the need for formality and clear task divisions increases as well.

Having a clear, replicable development methodology is critical to creating usable products. From the developers' point of view, technical development *is* the project. But other players, such as business and marketing participants, business process engineers and usability professionals, see technical development as the means to accomplish a business goal. It may not be realistic to demand that all participants in the project see it in the same way. What is necessary is for the various members of the team to understand when their input and decisions are required. Because the technical development is often the most expensive, time-consuming and risky element of the project, chaos is minimized when the development team follows a clear development process. This not only helps the development team structure its work, but makes it possible to identify the points at which input and decisions are needed from other members of the team.

Integrating LUCID with Development Frameworks

The LUCID Framework is not designed to be a complete product development methodology. There are many development frameworks available (the *Capability Maturity Model*[1] and the *Rational Unified Process* [2]are two of the best known). These frameworks can drastically reduce cost and decrease risk and there is no question that an appropriate development framework should be used by all development projects. But these frameworks are inevitably techno-centric; rarely are they built around best practices in user-centered design and user experience engineering. LUCID is designed to integrate with virtually any product development life cycle and enhance it by providing the missing elements.

[1] http://www.sei.cmu.edu/
[2] http://www.rational.com/

The reason that LUCID fits well with so many development frameworks is that it is "front-loaded." This means that most of its activities take place in the early stages before the bulk of the development activities occur. While there are elements of the framework that support the build and product release, the main focus of LUCID in on conceptualizing and designing the product. This allows organizations to integrate The LUCID Framework™ into their product development lifecycles without major changes to existing procedures.

The diagram below shows, in general terms, how LUCID activities fit into the overall development lifecycle.

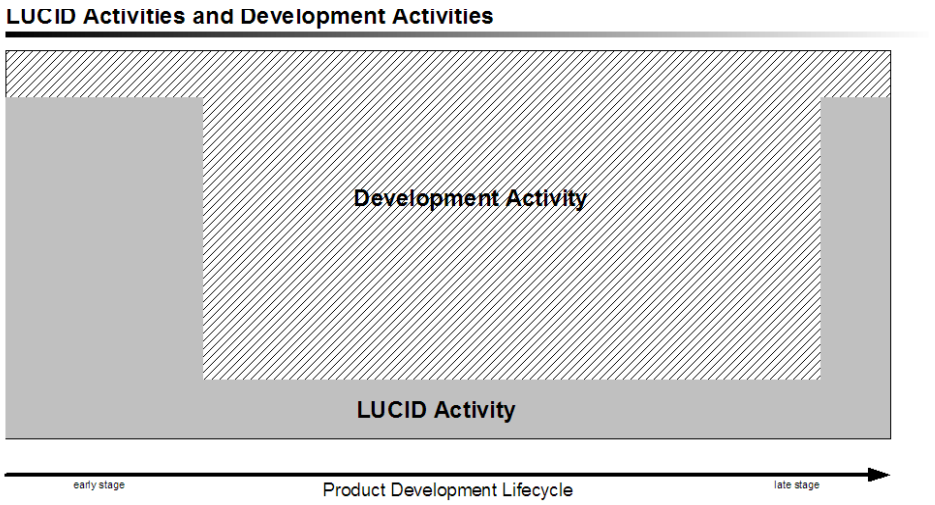


FIGURE 3: HOW LUCID INTEGRATES WITH DEVELOPMENT

In the early stages of the development lifecycle, LUCID activities supporting development of a shared vision, understanding user requirements, defining the product and designing the user interface predominate. With these activities completed, the development activities can take over. In most cases the development activities are significantly larger in scope than the LUCID activities. A key goal of LUCID is to make certain that that the product definition is complete and well documented. This enables the development team to focus on the complex task of building the product. During the build phase the LUCID team supports the development team if unforeseen problems require changes to the design. At this time, the LUCID team also prepares for product release. At the end of the development process, LUCID activity rises again as training and user support activities are initiated in the service of the product's release.

LUCID Stages

LUCID is structured as a series of six stages:

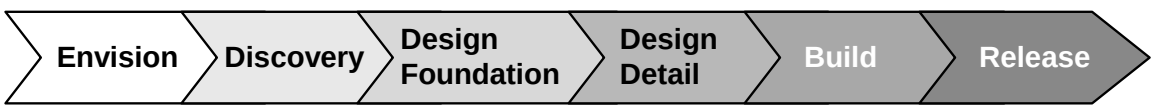


FIGURE 4: THE SIX LUCID STAGES

The LUCID process begins with the Envision Stage, when the project is initiated, and continues until the product release is complete. The first four stages, Envision through Design Detail, focus on product definition and design. The Build Stage provides support to the development team. The Release Stage provides support for fielding the product. Because there is little need for interaction between the LUCID design team and the developers during the Build Stage. The LUCID Framework™ works well in an outsourced environment where the development team expects complete specifications before beginning work.

Templates

At each of the LUCID stages there are specified deliverables, templates and timely feedback through reviews, for components such as:

- *Product definition*: high concept for managers and marketers
- *Business case*: pricing, expected revenues, return on investment, competition
- *Resources*: duration, effort levels, team members, back-up plans
- *Physical environment*: ergonomic design, physical installation, communication lines
- *Technical environment*: hardware and software for development and integration
- *Users*: multiple communities for interviews, user testing, marketing
- *Functionality*: services provided to users
- *Prototype*: early paper prototypes, key screens, running prototypes
- *Usability*: setting measurable goals, conduct tests, refine interface and goals
- *Design guidelines*: modification of existing guidelines, implementation of review process
- Content materials*: identification and acquisition of copyrighted text, audio, and video*
- *Documentation, training and help: specification, development, and testing paper, video, and online versions*

Waterfalls and Rapid Iteration

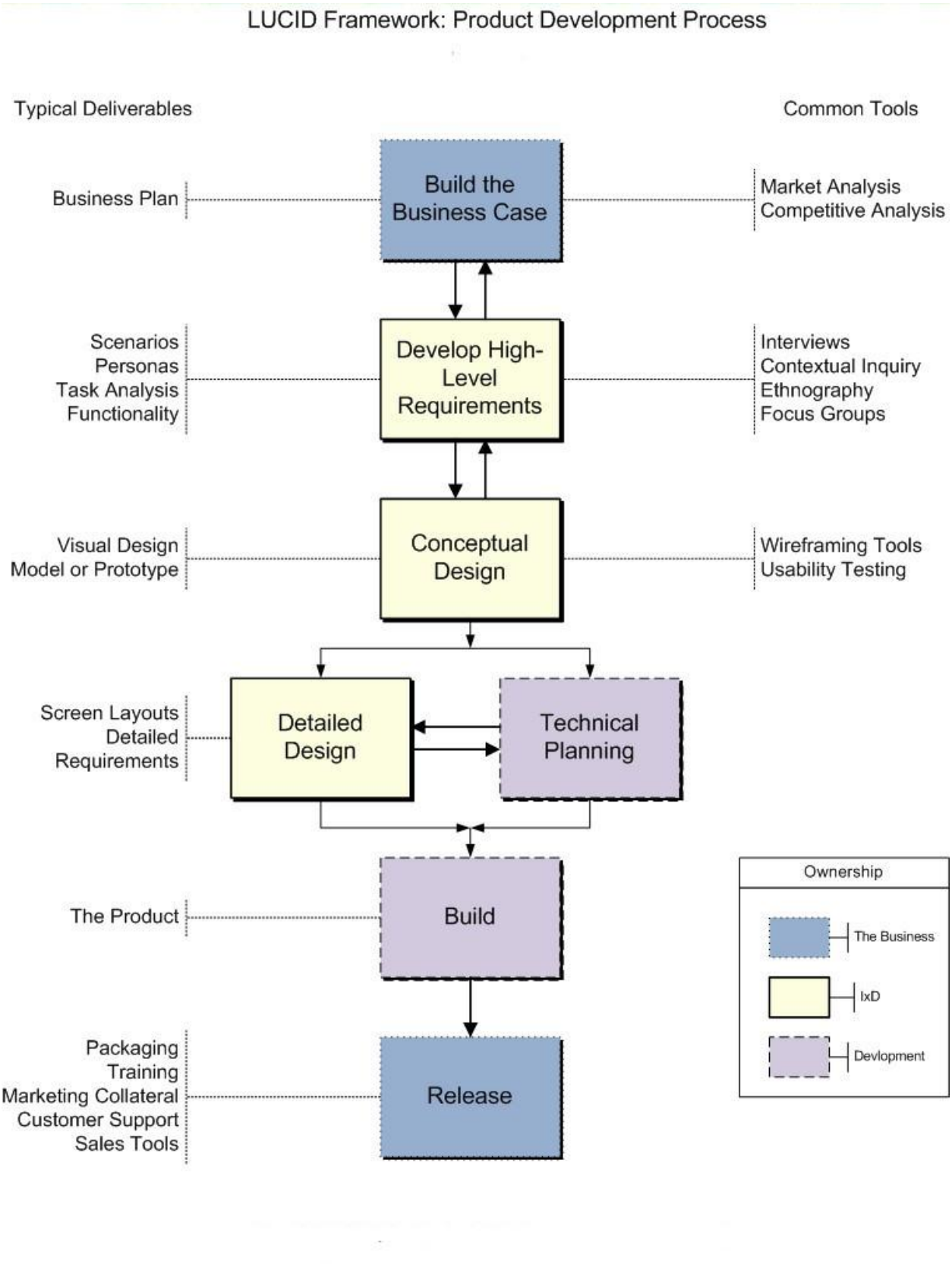
There is a lot of debate in the software engineering community about the best model for managing development. Traditionally, developers used a "waterfall model" in which the development process is structured as a series of steps, each of which is completed before the next step is initiated. Waterfall models have a lot to recommend them but often do not reflect the demands on deliverables and organization that pressure development teams. As a result, developers have turned to "iterative models" in which a functional but incomplete version of the product is quickly developed, refined and elaborated in a series of rapid development cycles.

LUCID is an agile process because design is intrinsically best conducted in an iterative manner. However, The LUCID Framework™ works effectively with either a waterfall or an iterative development model. In the case of a waterfall model, LUCID activities fit comfortably into the various stages of development. If an iterative model is used, it is still important that the development team and the various stakeholders share a clear vision of

the product being produced. Without a clear, shared and comprehensive vision of the goal, the project is at risk for slipping off track or becoming chaotic. An analogy is building a house. Many families cannot afford their ultimate vision at the beginning of the project. They build their home through iterative refinement by adding on new rooms (such as expanding the attic into living space) and elaborating existing rooms (such as transforming a basic kitchen into a gourmet kitchen). However, the foundation and supporting beams of the house should be designed to support the final plan. Otherwise a great deal of expensive reconstruction may be required.

LUCID Stages 1 – 3 provides the stakeholders with the needed clear and comprehensive vision of the intended product. Because these stages can be completed rapidly, if a reasonable amount of lead time is available, developing a comprehensive design concept will not slow down the development process. Stage 4, in which detailed design specifications are developed, can be fully completed or tied to an iterative development cycle as appropriate for a specific project.

The illustration that follows shows how LUCID can be integrated with an agile model.


LUCID Framework: Product Development Process
Typical Deliverables
Common Tools
Business Plan
Build the Business Case
Market Analysis
Competitive Analysis
Scenarios
Personas
Task Analysis
Functionality
Develop High-Level Requirements
Interviews
Contextual Inquiry
Ethnography
Focus Groups
Visual Design
Model or Prototype
Conceptual Design
Wireframing Tools
Usability Testing
Screen Layouts
Detailed Requirements
Detailed Design
Technical Planning
The Product
Build
Packaging
Training
Marketing Collateral
Customer Support
Sales Tools
Release
Ownership
The Business
IxD
Devlopment

LUCID Stage Flow

Each stage of LUCID is executed iteratively. The diagram below shows the typical flow of a stage:

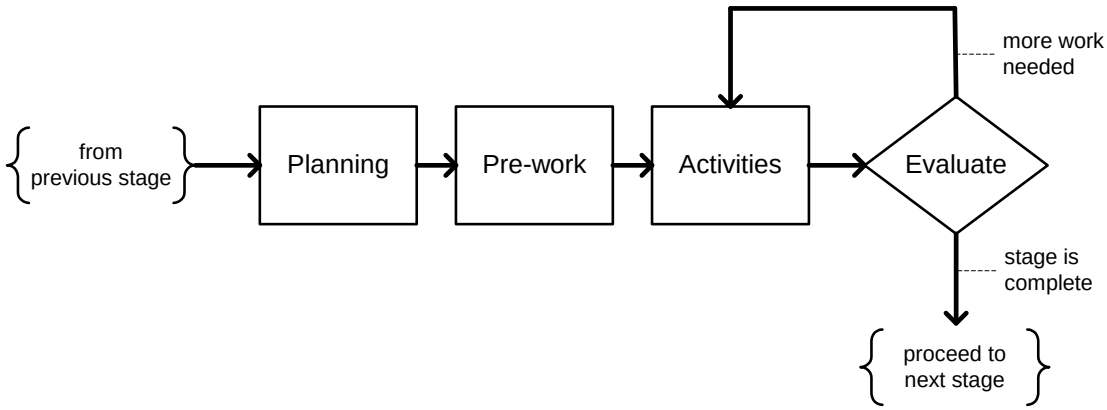


FIGURE 5: HOW STAGES ARE EXECUTED

Each stage begins with planning. The LUCID manager assembles the needed resources, makes any needed changes to the roles, responsibilities or staffing of the LUCID team, establishes schedules, manages communication and expectations of the various stakeholders and performs the other management tasks needed to ensure a successful outcome to the stage.

Next the LUCID team undertakes whatever preliminary work is needed for that stage. This typically involves research, information gathering and developing materials that will be required to complete the tasks in the current stage.

The team then conducts the various activities that are the core of the current LUCID stage.

When the activities are completed, the team completes the deliverables that will document the progress of the stage, performs its own internal review and communicates results to the various stakeholders.

Finally, the team determines if the work of the stage is complete. If so, they advance to the next stage. If not, they cycle back to refine and elaborate the work.

LUCID and Project Management

LUCID activities interact with other project activities. In many organizations there may be a well-established project management capability or a project management office. In such cases, LUCID activities should be integrated into the overall project management strategy. Some LUCID activities such as creating a business case and legal review may also be handled by existing organizational process.

While the availability of existing process and project management can make the LUCID team's work easier, it may be necessary to educate the owners of these processes about the priorities and particular requirements of user-centered design.

For example, an organization may require that a business case be developed before new projects are approved. The templates for developing the business case are frequently geared toward financial analysis and may not provide the LUCID team with key information needed to guide the product design. Frequently, the LUCID team will make use of an existing business case in the Envision Stage but will need to revisit some of the assumptions as it identifies user-oriented issues.

One area where The LUCID Framework™ may not fit with existing business practice is in providing sufficient time at the beginning of the project for product conceptualization and product definition. This is a serious error that costs companies a great deal of money and risks damaging the quality of products. The result is frequently frustrated and disappointed users. The LUCID activities can be completed rapidly and efficiently and should not compete with development activities.

Stage 1: Envision:

The *Envision Stage* is the first stage of the LUCID Framework. Its goal is to get all the stakeholders on the project to agree on a common vision. Elements of this vision include:

- coming up with a concise statement of the product definition (called a *high concept*)
- reviewing business objectives and anticipated sources of ROI and relating them to product design
- agreeing on the roles, responsibilities and staffing of the LUCID design team
- deciding how the team will conduct its activities, make decisions, and manage conflict
- identifying environmental, technical and legal constraints that affect product design
- identifying the various types of users who will interact with the product
- coming up with some sketches so the product concept can be shared with others

Envision Stage Goals

To align the agendas of all stakeholders, balancing the needs to meet business objectives, manage technical constraints and support users' needs for a highly usable product.

To develop a clear, shared product vision among the stakeholders.

To identify and deal with potential problems that could impair the development team's ability to collaborate effectively.

Surprisingly, many product development efforts are launched without a clear concept of the product or without agreement among stakeholders. This can occur because people interpret words differently or because some stakeholders may not be able to visualize the results of the development process. Whatever the reason, a lack of shared vision inevitably harms the quality of the product and can lead to conflict later in the process.

The major deliverable of the Envision Stage is a white paper that presents the product concept comprehensively and from a number of perspectives,. This is called the *Envision Roadmap* and is one of the most useful documents

Stage 2: Discovery

In the *Discovery Stage*, the design team develops a plan for studying the users, for analyzing and reporting the information gathered, and for identifying the user requirements and the user interface issues that emerge. Because the design team must understand the setting in which the product will be used, the Discovery Stage typically begins with a review of the environment in which the product will be used. Since the domain and content may be technical or specialized, it is critical that members of the design group work with subject matter experts and become familiar with key concepts and issues. This way interviews and observations of users are as meaningful as possible. An important part of the discovery process is identifying specialized vocabulary and jargon.

Discovery Stage Goals :

To develop a clear understanding of the characteristics of each distinct segment of the product's users.

To understand the tasks users perform, the information they need, the terminology they use, their priorities and their mental models.

To analyze the data gathered and create the product's user requirements.

Next, the user population is segmented into groups with similar needs and a data-gathering plan is developed. Depending upon the situation, the team may use various techniques such as contextual observation, user interviews, participatory explorations, focus groups, and usability testing. Satisfaction surveys or benchmark testing of the current solution or competitive solutions may also be done to provide data for longitudinal comparison. In addition, the team may gather data from secondary sources such as: interviews with user surrogates, review of training materials, and review of other printed materials.

Data is summarized using a number of techniques. *Scenarios* are narratives that are rich in their ability to capture nuances of the task flow and attitudes. *Use cases* describe interactions with the system from the point of view of an "actor" or user. *Personas* (also called *user archetypes*) are descriptions of idealized or representative users that include typical behaviors, attitudes, goals and skills.

Once adequate data has been gathered, it is used to create a comprehensive description of tasks, workflow, information needs and how the software will fit into the overall business process. This workflow information is then used to develop a set of high-level user requirements that specify both *what* the system must do and *how* the processing should flow. In the Discovery Stage, these requirements are developed at a relatively high-level – the goal is to create a complete "big picture" view of user requirements so that scope of the proposed system can be determined and the foundation for a conceptual design created. Because developers will require more detailed specifications, these user requirements will be elaborated into complete specifications in Stage 4.

Stage 3: Design Foundation:

With a clear, shared vision and in-depth understanding of the users, in hand, the design team begins the conceptual design of the product in the *Design Foundation Stage*. The conceptual design includes overall navigation, basic screen layout, visual design and information organization. The conceptual design is not intended to be a complete specification. Rather it is intended to convey a design approach so that business partners and potential users can develop a clear understanding of how the proposed product will look and function.

The process of creating the conceptual design is typically an iterative one. Early design concepts may be produced as paper "wireframes" and tested using informal usability tests. As the design concept becomes better defined, the quality and formality of the design representations evolves. Paper prototypes are replaced by on-screen mockups (often in the form of a "slide show") and then by navigable prototypes. Each iteration of the prototype is usability tested and refined, based on the results of the tests.

Design Foundation Stage Goals

To develop and validate the basic conceptual design of the product.

To develop a visual look for the product.

To present the completed design as a key screen prototype.

The ultimate deliverable of the *Design Foundation* stage is a *key screen prototype*, which is a model of the product that incorporates all the major screens and flows so that stakeholders – developers, marketers and users – can have a clear idea how the product will look and function.

The key screen prototype is an important component of The LUCID Framework™. A key screen prototype need not be complex. It can be as simple as a slide show with "hot spots" for navigation. To a developer, the word "prototype" may convey the idea of a working model supported by a significant amount of technology. Developers use such prototypes to validate their technical designs. A key screen prototype, in contrast, is intended to communicate and support validation of the visible elements of the system.

Traditionally, developers present users with lengthy, text-based systems specifications that, for a large system, can be hundreds of pages in length. While such detailed

specifications are useful for programmers who need to understand such details as field lengths and validation rules, business users find such specification extremely difficult to interpret. As a result, miscommunication and misunderstanding about what is being built often results in mid-project conflict and the need for late-stage course corrections.

By creating a simple navigable prototype or model of the product, non-technical users are better able to understand the product concept and can provide useful input and buy-in. The availability of the navigable prototype also facilitates early-stage usability testing to identify and correct design problems. The key screen prototype can also serve as an early model for marketing people to show to customers.

Stage 4: Design Detail

In the *Design Detail Stage* the team expands the high-level requirements produced earlier into complete design specifications for the product. Depending upon the needs of the particular project, the team may work sequentially, designing each screen and function in turn; or several teams may work in parallel, each handling different portions of the system.

The result of the Design Detail Stage is a complete specification of the user interface which provides a sufficient level of detail so that the developers can code without needing to ask for frequent clarifications. Typically, the specification includes detailed screen layouts and an explanation of how the elements of the screen are to behave. These elements can include navigation ("if the user clicks on this control, perform this action"), data entry rules ("accept up to five digits followed by a letter"), rules for data presentation, animations, sound and whatever other elements are required to fully define the screen and its behaviors.

To simplify the specification, a style guide is generally recommended. The style guide provides general guidelines for such elements as screen titles, button or link placement, colors, fonts, wording of messages and prompts, error handling and similar design elements that are consistently used throughout the product. Often companies develop a general style guide to be applied across multiple products and supplement them with specific style guides developed for particular products. This helps create a common look and feel across a family of products.

Design Detail Goals

To flesh out the high-level design into a complete specification.

To create a style guide

To develop a strategy and plan for users support and assistance

In the Design Detail Stage, a strategy for user support and assistance is also developed. The goal is to make the user interfaces as intuitive as possible and to integrate support elements directly into the interface to the extent possible, rather than relying on external help and training.

Stage 5: Build

With a complete specification in hand, the development team should be able to create the product with a minimum of input from the design team. This is important because the development team needs to be able to manage its internal priorities, schedule and resources. In addition, since a lot of development is outsourced to external companies, it is difficult and wasteful to require constant communications between the development team and the design teams.

However, no matter how carefully the specifications have been designed, there is likely to be a need for some level of mid- and late-stage design changes. If the LUCID process has been carefully followed, there should rarely be changes that result from business participants realizing that the scope, planned functionality or design of the product does not meet their expectations. However, a need for change may occur because technology fails to perform as expected, because of unexpected resource constraints or because the market, business process being addressed, or regulatory environment has shifted.

Build Stage Goals

To support the build process through review and late-stage change management.

To develop user support materials.

To prepare for the product release.

The design team needs to be available to make required changes quickly, so that the developers do not lose unnecessary time or decide to make ad hoc revisions that might compromise the integrity of the design. Because there is little time for formal usability testing, the team may elect to use more informal approaches on late-stage design changes.

The build stage can be lengthy because the development process is complex. The LUCID team may use this time to prepare (or contract for) user support materials including:

- on-line help files
- documentation
- training materials
- job aids

This time is also used to prepare for the product release.

Stage 6: Release

The final stage of The LUCID Framework is supporting the release of the product. As with other stages, the work of this stage varies with the type of product, the relationship between the development team and users, and the organization of the company. For example, the product may have an entire field services team to support a client installation,

or may have little more than a marketing announcement to accompany it into the world. Each of the activities for the *Release Stage* must be coordinated with the other teams who are part of the release efforts. For this reason, Release Stage planning should often begin quite early in the development cycle.

The Release Stage often includes materials to help users become familiar with the product. It may include surveys or other instruments to collect feedback. A project wrap-up to collect and document what was learned during the development cycle is also a valuable process.

Release Stage Goals

To develop a rollout plan to support the new product.

To validate the usability of the "out of the box" or installation experience.

To ensure the quality of such user support functions as help desk and field consulting.

To document lessons learned and prepare for the next version

The LUCID Framework™ suggests that the release strategy be based on a *barrier/incentive model*. This model looks at user acceptance from two complementary perspectives.

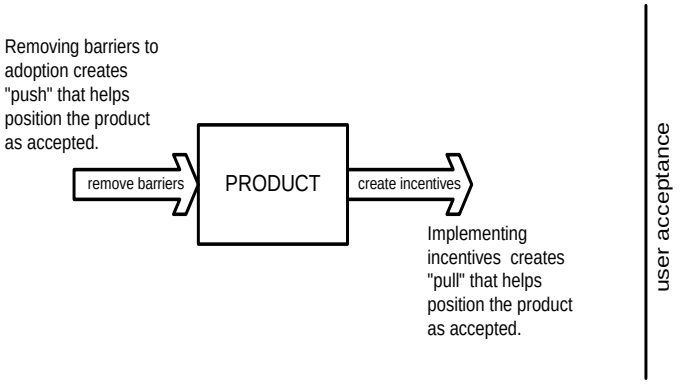


FIGURE 6: THE BARRIER/INCENTIVE MODEL

The *barrier perspective* assumes that users would easily adopt the new technology if only the barriers to adoption were identified and removed. So a key step is to determine what barriers to acceptance might exist and develop ways to eliminate them.

The *incentive perspective* assumes that users will not accept the new product unless they are given incentives to do so. So another key step is to craft incentives that encourage users to work with the new product.

Summary

Many product development teams have found The LUCID Framework™ to be a powerful and through framework for managing the design and usability components of interactive

products. LUCID can help ensure that key tasks are identified and executed in an orderly fashion with iterations within a stage and predictable progress from stage to stage.

The value of The LUCID Framework™ goes beyond usability. To be successful, a product needs to please users but it also needs to meet business goals and technical constraints. The LUCID Framework will help the team balance competing interests and find the "sweet spot" that justify its investment.

The LUCID Framework™ was originally developed by Cognetics Corporation a decade ago. Over the years it has evolved into the current rich and flexible framework presented here. Despite the fact that Cognetics never formally published LUCID, The LUCID Framework™ became widely used by companies world-wide. It is taught at a number of universities and referenced in textbooks. Obviously, LUCID has a lot to offer.

LUCID continues to evolve. We welcome input from everyone who has used LUCID and wants to contribute to it.