

LUCID Stage 1: Envision

The purpose of LUCID Stage 1 is to create a clear, shared vision of the product. This vision is described in the UI Roadmap—a comprehensive high-level document that communicates the design vision, manages expectations, and serves as the basis for evaluating progress throughout the project.

It is difficult to overstate the importance of developing the product concept at the beginning of the project. Although a remarkable number of development projects proceed without a clear product concept, this inevitably leads to serious problems later in development. In addition to communicating the product concept to interested parties, the UI Roadmap is used to guide decision making and evaluate progress throughout the design project.

Goals

- ☐ create a concise description of the product
- ☐ identify the business and design objectives that will shape the design
- ☐ identify constraints on the product
- ☐ identify the target user population
- ☐ identify the main functionality of the product
- ☐ create screen sketches that demonstrate the product concept

Prerequisites

- ☐ A need has been identified.
- ☐ The design group has been assembled.
- ☐ There is corporate support for the project.
- ☐ The basic technical environment has been identified.

Decisions

1. How will the design group solicit user input? For example, you may bring in users, utilize ‘surrogate users’ (typically someone within your organization who once was a user), or perform field surveys.
2. During Stage 1, you may decide that the design group is missing some desirable skill or type of knowledge. Make sure the design group is appropriate for the project, and reconfigure the group if necessary.
3. Based on the UI Roadmap created at the end of Stage 1, you may need to rethink the project. Possibilities include expanding, shifting, or reducing the scope based on time, budget, or technical constraints.

Tasks	Task 1.1 Create a High Concept Statement for the product
	Task 1.2 Identify the business objectives the product will serve
	Task 1.3 Identify the user population
	Task 1.4 Identify usability goals
	Task 1.5 Identify high-level constraints
	Task 1.6 Identify high-level functionality
	Task 1.7 Produce screen sketches
	Task 1.8 Create the project plan
	Task 1.9 Create the UI Roadmap
Deliverables	The primary deliverable of LUCID Stage 1 is the UI Roadmap, a comprehensive high-level description of the product.

The brainstorming activities of Stage 1, and their formalization in the UI Roadmap serve to bring the team together and create a starting point for the work before them. This process can clarify the thinking behind a decision to move forward with the project, expose areas of potential risk, and unite the team behind a common vision.

Many of the elements in the UI Roadmap draw on the work of other groups in a project team. The business objectives, for example, may have been defined before the interface design team is assembled. The purpose of the UI Roadmap is not to duplicate this work, but to examine it in light of its impact on the interface.

Brainstorming

The tasks in Stage 1 are performed by the entire design group, and several of them recommend the use of brainstorming to generate lists of attributes.

Because one of the goals is to solicit as many ideas as possible, it is important to discourage any sort of critiquing or editing during brainstorming. Ideas should be recorded exactly as they are stated. Some groups use index cards or sticky notes to collect and record ideas; these facilitate the eventual arrangement of ideas into groups.

Task 1.1: Create a High Concept Statement for the Product

In Task 1.1, a high concept statement—a brief descriptive summary, typically 25-50 words long—is created for the project.

Prerequisites	See Stage 1 prerequisites.
Deliverables	The primary deliverable of Task 1.1 is the high concept statement. The high concept statement will be incorporated in the UI Roadmap in Task 1.9.
Overview	A list of questions is used to generate the information used to produce the high concept statement.

A high-concept statement is analogous to the mission statement of a corporation. It serves to focus attention and effort. It allows the project team to succinctly describe its development goal.

At a minimum, the high concept statement should include the product name; information about expected users; a description of what the product will do; and a statement telling what problem the product will solve.

Producing a good high concept statement can be surprisingly difficult. Creating a short, concise description of the product demands quality conceptualization and articulation, as well as a willingness to write and rewrite.

Procedure

Step 1

Answer the following questions *in writing*. Make sure your answers are clear and accurate. Be as specific as possible.

1. What is the name of the product?
2. Who will use the product?
3. What will the product do?
4. What problem will the product solve?

Step 2

Use your written answers to draft the high concept statement. Rewrite and edit as needed to produce a 25-50 word statement that summarizes the project.

Examples

Here is an example of a poorly written high concept statement:

Our purpose is to develop a better inventory control system. It will improve sales performance and customer satisfaction. It will improve communication with the warehouse.

This statement is too general. It does not state why the new inventory control system will be better, or how sales performance, customer satisfaction and warehouse communication will be improved.

Here is a better version of the same high concept statement:

The TALLY inventory control system will enable retail sales staff to check the availability of all products in our Vapid VapoursÔ line of perfumes. It will reduce customer waiting time by determining if the desired scent is in stock, avoiding the need for the salesperson to telephone the warehouse.

Here is another example of a well-constructed high concept statement:

The BARFLY system will be used by warehouse staff. It will use bar-coding technology to identify all glassware delivered to and shipped from the warehouse. It will allow warehouse purchasing agents to place re-orders on a timely basis to ensure availability of stock so that customer orders are filled promptly.

Task 1.2: Identify the Business Objectives the Product will Serve

In Task 1.2, the business objectives for the project are identified and clearly stated. The business objectives may be accompanied by a business case and project history.

Prerequisites	See Stage 1 prerequisites.
Deliverables	The key deliverable of this task is a list of specific business objectives. There may be many business objectives, or only one or two. The business objectives are an important component of the UI Roadmap created in Task 1.9.
Overview	A list of general business objectives is used to identify and develop project-specific business objectives.

Every project funded by a business organization is initiated with the expectation of some sort of return on the investment. It is important that specific business objectives be clearly stated. Explicit business objectives are used to shape design decisions and evaluate proposed modifications to the project's scope.

Procedure

Step 1

Document the business considerations and history which have led to the current project. Consider the following list as a starting point:

- What is the opportunity that has led to the decision to develop a new product?
- Is this the first version of the product, or is it a replacement for an older product?

If this is the first version:

- ❑ What is the rationale for undertaking the product now? Possible factors include market forces, a technical opportunity, a new need, an old need that is inadequately met.

If this product is a replacement for an older product:

- ❑ What is the basic reason for replacing the older product? Possibilities include obsolete technology, inadequate functionality in the old system, and competitive pressure.

- ❑ What are the older product's strengths and shortcomings?
- ❑ How, in general terms, will the new product differ from the older product?
- ❑ Will the new product be a revision or a complete re-working?

Step 2

Use the following list of typical reasons for initiating a development project as a starting point to identify and define the business objectives for your project. This list is not exhaustive, and your project may have business objectives that are not listed here. A product can (and often does) serve more than one business objective.

- Revenue generation
- Cost reduction
- Increased efficiency of business operations
- Improved service quality and reduced errors
- Strategic advantage against competitors
- New organization capabilities
- Improved security
- Legal compliance

Step 3

Each business objective you identify should be carefully developed into one or two sentences that state the objective, describe how you plan to achieve your goal, and tell why you believe this approach will work. These objectives are documented, and will be incorporated in the UI Roadmap at the end of Stage 1.

Examples

The following are examples of project-specific business objectives:

<i>The new version will improve security of financial information sent over the Internet by replacing the simple encryption used in the previous version with the SET (Secure Electronic Transaction) technology. This will provide a greater sense of comfort to our users.</i>
--

The new system will reduce operational costs by eliminating the manual process currently required to move data from the current (legacy) patient registration system to the legacy clinical database system. At present these systems are housed on incompatible mainframes.

The new system will allow librarians to access all major on-line research system including: Lexus, Nexus and Dow Jones. This will eliminate the need to sub-contract these searches to an outside vendor and provide more rapid response to requests for information.

The new system will provide us with a strategic advantage by offering functionality not available in competitor's systems: the ability to verify transactions across the Internet, reducing processing time and allowing quicker revenue collection.

Task 1.3: Identify the User Population

The goal of Task 1.3 is to identify all possible users of the product. An initial arrangement of users into groups aids this process and serves as a starting point for tasks performed in Stage 2.

Prerequisites	See Stage 1 prerequisites.
Deliverables	This task produces a list of groups of users. This information will be incorporated in the UI Roadmap in Task 1.9.
Overview	Brainstorming is used to identify the target users and different ways in which they can be grouped for usability and task analysis.

Understanding the user population is central to usability engineering because different users have different needs, work processes, and skill sets.

There are often several different types of users who will be affected by the system. Different types of users may be grouped by job title, job function, or department, or by such factors as computer competence, job experience, and professional training. The specific grouping strategies and number of groups are important during LUCID Stage 2, when users are placed in homogeneous groups called user segments. However, during LUCID Stage 1, the prime concern is identifying all possible users.

In addition to the direct users, it may be useful to identify potential product champions and individuals who may oppose the project. Because new systems may cross organizational lines, they may alter established business practices and threaten some individuals. By identifying interested parties early in the process, you can develop strategies to facilitate acceptance later in the development.

It is sometimes difficult to obtain end-user representation on the design team. This is especially true when the end-users are customers who are not part of the development organization. Design team members who are familiar with the end-user or customer base and can represent the customers' interests are particularly valuable during this task.

Procedure

Step 1

Use brainstorming to create a list of different types of users for the product. Users might be categorized in many ways: experience level, reasons for using the product, job title or tasks – do not worry about mixing different categorizations at this point.

Step 2

Combining types as necessary, organize the lists of users into a chart identifying primary and secondary user groups who will be served by the product.

Task 1.4: Identify Usability Goals

In Task 1.4, the usability goals for the project are identified and clearly stated.

Prerequisites	See Stage 1 prerequisites.
Deliverables	The key deliverable for Task 1.4 is a list of specific usability goals. These goals will be incorporated in the UI Roadmap in Task 1.9.
Overview	A list of general usability goals is used to identify and develop project-specific usability goals.

Specific usability goals should be defined, documented, and included in the UI Roadmap. Their inclusion in the UI Roadmap assures that usability goals will be included in the design process.

Procedure

Step 1

Use the following list of general usability concerns as a starting point to identify and define the specific usability goals for your project. This list is not exhaustive, and your project may have usability goals that are not listed here.

- **Errors.** How critical is it to avoid errors? What are the possible consequences of errors? What strategies will be used to avoid or minimize the impact of errors?
- **Types of users.** Is the product designed for novice users, intermediate users, or expert users? Are there intermittent users (who will use the product on a sporadic basis) that may require re-familiarization? What level of technical skills will users bring to the process? What level of content (domain) skill will be required? Are the users captive (required to use the product, e.g. a corporate system) or discretionary (e.g. a web site)?
- **Training and support.** How much training will be required to use the product? What type of training will be made available? What type of on-line support and help will be used?
- **Integration.** To what extent should the product be expected to integrate with other software? What types of data should the product be able to exchange? How will interaction among software products be facilitated?
- **Workgroups.** Does the product allow and facilitate cooperative activity among multiple users? If so, how are users identified? What types of information will they exchange?

- **Platform issues.** Are special plug-ins or other technology required to use the product? If the product is delivered over the web, what bandwidth is assumed? Can the product set its own colors and fonts, or must standard colors and fonts be used? Is the color depth of the user's computer an issue?

Step 2

Each usability goal you identify should be carefully developed into one or two sentences that state the objective and describe how you plan to achieve your goal. These usability goals are documented, and will be incorporated in the UI Roadmap at the end of Stage 1.

Examples

The following design objectives are taken from a medical software system:

The product will examine all clinical values entered by the user and validate them to make certain that they are reasonable. If any value is outside the reasonable range, the user will be shown the questionable value and offered the opportunity to correct it.

All drugs ordered by a physician using this system will be checked for possible interactions with other drugs. The system will also ascertain that the dose prescribed is within the recommended range. Suspect orders will be flagged, and will require correction or confirmation by the physician.

The following design objectives relate to security:

The software will maintain a user profile which indicates the types of transactions authorized for each user. Transactions which are not authorized for a specific user will not be allowed.

Each user will be assigned a financial authorization level. Transactions which exceed the financial authorization of a user will be automatically routed to a supervisor for approval.

The following objective relates to data entry integrity:

Wherever possible, users will be presented with a "pick list" of data elements rather than being asked to type in a data field.

The following objectives relate to user support:

The on-line help system will contain a tutorial that will enable new users to learn the system basics in a single session.

The user interface will provide a structured “wizard-like” capability to walk the novice user through the process, and will also support an advanced “power” interface for experienced users.

Task 1.5: Identify High-Level Constraints

In Task 1.5, the high-level constraints that affect the design project are identified and documented.

Prerequisites	See Stage 1 prerequisites.
Deliverables	A list of high-level constraints is assembled in this task. These constraints will be incorporated in the UI Roadmap in Task 1.9.
Overview	A list of general constraints is used to identify and develop project-specific high-level constraints.

Most systems must be designed around constraints. These may involve hardware (for example custom-built hardware or special input-output devices, such as audio), software (such as specific operating systems or databases), environmental considerations (high noise, public access), bandwidth issues, legal issues, or special user needs.

High-level (general) constraints are identified at this stage so they can be factored into the design process. More specific constraints may emerge during LUCID Stage 2 during the analysis of user needs and workflow.

Procedure

Step 1

Use the following list of high-level constraints as a starting point to identify and define specific high-level constraints for your project. This list is not exhaustive, and your project may have constraints that are not listed here.

- **Hardware.** Will the product be used with customized hardware? Will it be used on many different hardware configurations? Will monochrome monitors be used? Are specialized input-output devices contemplated?
- **Environment.** Will the product be used in private or public locations? Will the surroundings be noisy? Are there security or vandalism concerns? Is lighting an issue? Are there extreme environmental factors that should be considered (e.g., installed in a vehicle or on a shop floor)?
- **Performance characteristics.** Are there bandwidth considerations? Are there response time considerations?
- **Cost.** What is the target cost? Does this impose restrictions on the scope of the project?

- **Development time.** What is the development schedule? Does this restrict the scope of the project?
- **Size and/or weight.** Will the product be used on portable or mobile equipment?
- **Looks/commercialization.** Does the product need to be very graphic? Should it use multimedia? Are there specialized installation requirements? Does the product need to conform to the look and feel of another software product?
- **Regulatory and/or legal considerations.** Are there copyright issues? Are there liability issues? Are there privacy issues? In the United States, does the product need to conform to ADA (Americans with Disability Act) requirements, EU requirements, or other legal requirements?
- **Need for collaboration.** Does this product require that multiple users collaborate? How will this be done? How will individual users be identified? Are there any special legal, privacy or copyright issues which arise as a result of collaboration? Is communication among users in real time? Is email used as a communications medium? If so, are there compatibility or formatting issues?
- **Need to integrate with other systems or processes.** Does the product need to inter-operate with other software products? Is all data compatible? How will the user know if data exchange has been successful? Are there version compatibility issues?
- **Security.** What security considerations are there for such issues as access to the product, access to specific functions in the product, user profiles, and transaction limits (financial or other)?
- **Reliability and disaster recovery.** What will happen if the product crashes? Are specialized solutions (e.g. redundant systems) required? Can the user “pick up” from the point of error? Can the product clearly display its status and the status of all transactions on recovery? Are there backup facilities? Is backup automatic or manual? How is recovery from disaster enabled?

Step 2

Each high-level constraint you identify should be carefully developed into one or two sentences that describe the constraint and possible design work-arounds. The high-level constraints will be incorporated in the UI Roadmap at the end of Stage 1.

Examples

Here are examples of high-level constraint statements:

The product will be used on factory floors with high dust levels. A special enclosure for the computer and monitor and dust-resistant keyboards will be used.

The product will be unattended in public areas and vulnerable to vandalism. A vandal-resistant kiosk will be used.

The product will be used in a theatre during performances. All “beeps” and other audio cues will be disabled.

The product will be used to publish materials supplied by users. Users will be warned about copyright violations, and all submissions will be tagged with the name of the originating user.

The product will be used in a medical intensive-care unit. If the software fails, an alarm will be transmitted to the nurses station.

Task 1.6: Identify High-Level Functionality

The goal of Task 1.6 is the development of a list of statements that describe the proposed product's functionality.

Prerequisites	See Stage 1 prerequisites.
Deliverables	The goal of this task is the production of a description of the high-level functionality of the product. This information will be used in Task 1.9.
Overview	Brainstorming is used to produce a list of functionality statements, which are then tested against the high concept statement and refined.

Functionality is a concern throughout the design process. In LUCID Stage 1, functionality is addressed at a high level: simple statements are used to describe what the system should do for the user. The goal is not to perform a complete requirements analysis (which will be done in Stage 2), but to define the scope of the development effort.

Detailed statements about functionality should not be included at this point. They will be addressed later in the design process.

It is important that statements about functionality be expressed in the user's terms rather than technical terms. See the examples at the end of this section.

Procedure

Step 1

Use brainstorming to produce a list of functionality statements. Each statement should start with "The user will be able to . . .". Do not edit or critique individual items during this process.

Step 2

Review the list of statements generated in Step 1. Check each statement for consistency with the high concept statement generated in Task 1.1, and discard any that do not fit. Make sure each statement is user-oriented, not technology-oriented. If there is functionality which does not fit the high concept statement but seems essential, consider revising the high concept statement to include it.

Step 3

After you have reviewed and edited the statements, check the statements for appropriate level of detail. Statements that are too detailed may be merged with other statements. A statement that is too general may be used as the basis for several more detailed statements.

Step 4

When you are comfortable with the level of detail, count the number of statements that remain. If you have only two or three, your statements may be too general. If you have twenty or more, your statements may be too detailed. (In the latter case, it is also possible that the project is too big and should be broken down into several smaller projects.)

Examples

The following high-level functionality statements have inappropriate levels of detail:

The user will be able to maintain the database.

The user will be able to specify whether titles are underlined with single or double lines.

These high-level functionality statements are inappropriate because they are technology-oriented:

The user will be able to update the inventory database.

The user will be able to transfer records from the transaction file to the exception file.

The system will be able to validate data fields by edit range.

Here are the same statements expressed in user-oriented terms:

The user will be able to keep the inventory up-to-date by entering items as they are shipped and received.

The user will be able to identify requests that cannot be automatically processed and flag these requests for manual processing.

The user will be informed when an input value is outside the normal range, and will be asked to confirm or re-enter the value.

Task 1.7: Produce Screen Sketches

In Task 1.7, several basic screen sketches are produced to help individuals visualize how the product might work.

Prerequisites	See Stage 1 prerequisites. You will also need a white board or similar tool to create sketches as a group.
Deliverables	The deliverable for this task is a series of screen sketches. These sketches will be incorporated in the UI Roadmap in Task 1.9.
Overview	Screen sketches are produced by the group, discussed, and refined. When agreement is reached, the sketches are transferred to paper for use in the UI Roadmap.

Many individuals have difficulty visualizing how an interactive system might look and operate. A few basic sketches that illustrate the system concept—without showing navigation features or realistic data content—are a useful addition to the UI Roadmap.

Because these sketches are made early in the design process, they will not reflect the look and feel of the final design. It is very important to remind anyone who views these sketches that the final product may resemble them in little more than the basic functionality.

While actual screens can be generated using any one of several prototyping tools, there are distinct advantages to using sketches on paper. The use of a hand-drawn sketch on paper supports the concept that the screen sketches are preliminary. Transferring even a simple sketch to a computer screen gives it a permanence and authority that work against its use as something meant only to convey a concept.

Procedure

Step 1

Assign a group member to sketch ideas at a white board (or some other tool that allows the entire group to view and discuss the sketches). Start by selecting a primary screen and sketching its main features. Keep the screen at a conceptual level—do not include navigation features or real data. Do the same for several other screens.

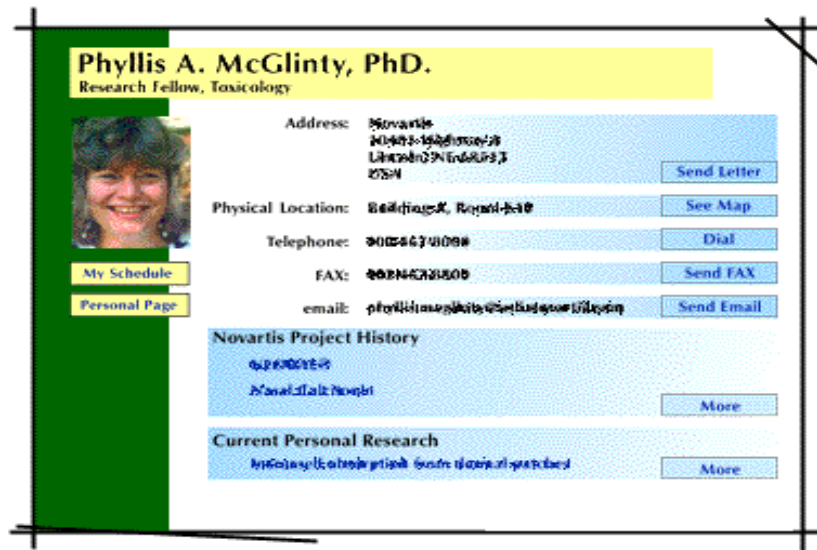
Step 2

After generating sketches of several screens, discuss and refine them as necessary (but do not add detail). When the group is satisfied with the screen sketches, transfer them to paper for

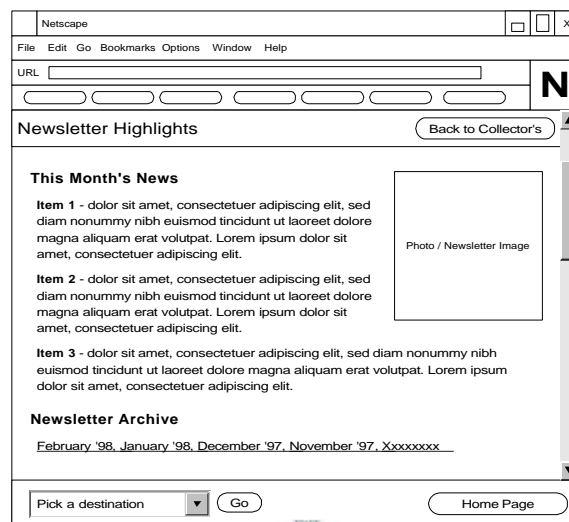
inclusion in the UI Roadmap. They may be 'tidied up' using a graphics program, or may be used as hand sketches, but in most cases should not be used to produce actual screens.

Examples

The following screen is from a corporate personnel directory. The screen illustrates that the each employee's information is displayed along with their picture. Basic contact information is available, along with current projects and other biographical information. Notice how simple the screen is. Because it is meant to show the proposed functionality on a conceptual level, typical screen elements such as a title bar, title bar buttons, and a cancel button have not been included, but that there are roughly indicated buttons showing key functionality.



The next screen is intended for a web site. This page will contain an electronic version of the company's newsletter. The screen sketch does not show any of the newsletter's details (and the actual layout will probably be quite different) but does show that the newsletter can contain photographs and will probably be a scrolling page. By its schematic nature, this screen sketch eliminates the debate about format that would most likely accompany a more detailed presentation.



Task 1.8: Create the Project Plan

In Task 1.8, a project plan is created for the design process. This plan will typically be a subset of the overall software development project plan, and should show points of intersection with other aspects of the software development and testing process.

Prerequisites	See Stage 1 prerequisites.
Deliverables	The deliverable for this task is the project plan, which consists of a project schedule and a budget.
Overview	A task analysis is performed. Key milestones and decision points are identified.

Step 1

The major activities in each stage of the LUCID Design Framework are summarized below. (Detailed task information can be found in the chapter dedicated to each Stage.)

- Create the UI Roadmap (Envision, Stage 1)
- Perform user interviews and prepare requirements analysis (Analyze, Stage 2)
- Create the key screen prototype (Design, Stage 3)
- Review, evaluate and refine the design (Refine, Stage 4)
- Support late stage change (Implement, Stage 5)
- Develop user support materials (Implement, Stage 5)
- Provide release support (Support, Stage 6)

Step 2

For each activity identified in Step 1:

- Identify the individual(s) who will be involved
- Identify milestones and decision points, especially points where information is passed to the 'back end' development group
- Estimate the time required in calendar days for completion

- Estimate the time required in “person days” for completion, and use this to calculate costs
- Add additional expenses (e.g. travel)

Step 3

Use the information generated in Step 2 to produce a task analysis, project schedule, and budget.

Examples

Here is an example of a task analysis, project schedule, and budget for a small project.

Task Analysis

Develop a UI Roadmap

We plan to begin the project with a “kick-off” meeting at which the design team and software engineers will meet with representatives of Marketing (representing the users). At this meeting we will make final decisions about details of the project concept and scope. (This corresponds to LUCID Stage 1). The meeting is planned for two days in length.

Interview Customers to Understand Needs and Workflow

Task 2 is a research project to review meet with customers and learn about their work process and specific needs. We will also evaluate the platforms they are using to access the Web. Although the design team will take the lead on this task, we expect to work closely with the technical organization.

Create concept prototype and presentation for review by managers and customers

Based on the information gathered in tasks 1 and 2, the design team will develop a key screen prototype. The prototype will be a non-working set of screens that illustrate the basic functions of the system. The prototype will incorporate the overall “look and feel,” as well as the navigational flow of the proposed product. The prototype will be used for usability testing and review by selected customers.

Conduct usability test of filter input/management sub-system.

In Task 4, the design team will meet with the selected sample of current customers who have agreed to provide input to the project. We will show the customers the concept prototype and explain the proposed preliminary navigational and interface design. We will conduct usability testing to identify any design problems. These results will provide the information necessary to complete the final design refinement.

Create final design and specification

Based on the results of the usability test and customer interviews, we will create a final design and document it as a formal specification.

Budget and Schedule

Create UI Roadmap	2 days
Determine basic concept and initial functionality	
Identify constraints	
Develop concept sketch	
Study User Workflow/Develop Requirements	5 days
Conduct customer focus groups	
Develop use scenarios	
Produce final functional specifications	
Develop Key Screen Prototype	6 days
Design screens	
Program prototype	
Usability test prototype	
Selection of final algorithm	
Usability Testing	2 days
Customer review/usability test prototype	
Post review meeting/implement changes	
Final Specifications	2 days
Review usability results	
Develop final specs	
Create final graphics	
Totals	17 days

Task 1.9: Create the UI Roadmap

In Task 1.9, the information generated in the previous Stage 1 tasks is used to create the UI Roadmap. This document will be used to communicate the product concept to interested parties, and as a tool throughout the design process.

Prerequisites	Tasks 1.1 through 1.8 must be completed.
Deliverables	The UI Roadmap is the deliverable for Task 1.9, as well as the primary deliverable for LUCID Stage 1.
Overview	The material developed in Tasks 1.1 – 1.8 is assembled into a report.

The format of the report should incorporate all the headings for tasks 1.1 – 1.8 in the order specified.

Procedure

Write up the report and distribute.

Examples

An abridged example of a LUCID UI Roadmap follows.

JJSYS UI Roadmap

(Abridged Example)

Date of Original Version: August 1, 1997

Date of Current Revision: December 1, 1997

Current Revision Level: 1.1

Design Team: Scott Gilkeson (Cognetics Corporation)
Jim Kauffman (Cognetics Corporation)
Charles B. Kreitzberg (Cognetics Corporation)
Anne Rose (University of Maryland)

High Concept Statement

JJSYS (Juvenile Justice System) will be an integrated software system designed to support all aspects of juvenile case tracking for the Department of Juvenile Justice (DJJ) operations including: Field Services, Residential Services, and Administrative Program Services.

Business Objectives

The decision to undertake development of the JJSYS system was based on an analysis of problems relating to the existing system. In particular, the following major problems with the existing software were identified:

- ❑ ISYS operates on obsolete character-based terminals. These terminals are incapable of utilizing a Graphical User Interface.
- ❑ There are an inadequate number of terminals available to service staff needs.
- ❑ The current software is confusing and difficult to use. As a result it is not truly assisting staff members in performing their job, nor is it facilitating them in their primary mission of serving the state's youth,
- ❑ The current software does not adequately support the needs of the Residential Services division of DJJ and is inadequate to the needs of Field Services.
- ❑ The current software has inadequate reporting capabilities.
- ❑ The current software is based on a database system which is inconsistent with the information processing standards used at the central data center.
- ❑ The current software lacks certain key functionality needed to interface with external systems.
- ❑ The current software does not support the most efficient models (best practice) of processing information important to the Department's business.

Based on these observations, the following business objectives have been established for the JJSYS system:

- ❑ Increase the availability of staff to youths and their families by reducing the time that front-line workers must spend on administrative tasks.
- ❑ Prepare DJJ to cope with an anticipated increase in workload which will increase the demand on individual workers.
- ❑ Facilitate the interchange of information between JJSYS and external systems.
- ❑ Improve the validity of data.
- ❑ Increase system availability to workers through an increased number of terminals and improved systems reliability.
- ❑ Provide the ability to produce reports and other documents needed for case processing.

User Population

The following user segments have been identified:

- ❑ intake workers
- ❑ case managers (including related activities such as probation)
- ❑ interstate compact unit staff
- ❑ clerical staff
- ❑ restitution staff
- ❑ field supervisors
- ❑ juvenile counselors
- ❑ group life staff
- ❑ education staff
- ❑ health staff
- ❑ substance abuse staff
- ❑ residential unit supervisors
- ❑ research unit
- ❑ administrative staff
- ❑ some private providers

High-Level Constraints**Privacy**

The system must support privacy at the field level. A profile of users will be maintained which indicates authorization to view information. For example, only medical and psychological personnel may review mental assessments.

Physical Environment

- ❑ Increase availability of access to system through increased numbers of terminals and printers.
- ❑ Identify “secure” space within residential facilities so that all intake can be entered in real time.
- ❑ Install terminals in intake hearing rooms so data can be reviewed and revised as the interview proceeds.

Technical Environment

- ❑ Terminals to be IBM-compatible PCs, meeting departmental standards.
- ❑ Terminals to operate as either LAN nodes or in stand-alone mode.
- ❑ Dial-in access (for laptops) to be supported with appropriate security.
- ❑ Terminals to be supported at 60 sites.
- ❑ Database to be stored on the State of Maryland mainframe in Annapolis. Possibility of DJJ server as client to mainframe to be investigated.
- ❑ Need to increase technical support to sites for both software and hardware problems. Down-time is unacceptable. Back-up facilities must be provided in case of hardware failure.
- ❑ Need to investigate technologies to improve individual identification such as real-time fingerprint analysis to determine if they are cost-effective.
- ❑ Need to investigate technologies to improve input speed and accuracy such as bar-coding, scanning and OCR to determine if they are cost-effective.

High-Level Functionality

- ❑ Maintain a database of youths who are referred to DJJ. This database must contain (a) complete information about the individual, (b) all associated situations (cases and other incidents which led to contact with DJJ), (c) all events and actions associated with a situation.
- ❑ Maintain a database of programs and facilities, linked to the database of individuals.

- ❑ Maintain a database of victims, linked to the situations in the database of youths.
- ❑ Maintain a schedule of key events such as court appearances and hearings linked to the situations in the database of youths.
- ❑ Provide a single source for information management, using standardized procedures covering all aspects of Field Services and Residential Services operations. Provide data verification wherever possible to ensure the integrity of the database.
- ❑ Provide an output facility for producing forms and letters using customized templates, automatically filled-in where appropriate from the databases. Provide the ability to create and modify templates. Provide an ability to deliver these forms and letters in either electronic or hard-copy form where appropriate.
- ❑ Provide workflow management services to workers and supervisors including: tickler files, “Rolodex files”, lists of assignments, and a query and reporting facility to assist supervisors in managing workload.
- ❑ Provide secure email both within DJJ and to outside agencies.
- ❑ Provide an analytic management reporting capability including both pre-defined reports and ad-hoc queries.
- ❑ Provide on-line access to policies and procedures.
- ❑ Provide a means of data interchange with external systems including the courts, law enforcement systems, and local Boards of Education.