CYBER SECURITY SOLUTIONS DIRECTORATE

Cyber Engineering Solutions

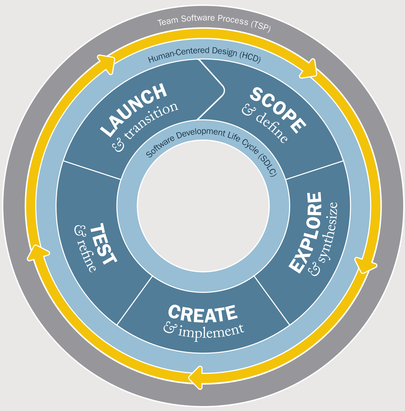
HOW WE CREATE INNOVATIVE SOLUTIONS FOR PEOPLE

process

Our process is a unique combination of three well-established frameworks. The heart of the process is made up of five **Project Stages** that have evolved from the **Human-Centered Design** (HCD) process and the **Software Development Life Cycle** (SDLC). These stages are then governed by the third framework, the **Team Software Process** (TSP).

An additional sub-process of **continuous iteration** occurs to ensure constant improvement and evolution of the solution.

Moving through the individual **Project Stages**, the team executes a faster instance of the entire process to accomplish milestones or to solve problems that come up during each stage. This is called **process iteration**.



expertise

While process and methods are important, they mean nothing without the **skills** to execute them. Our expertise falls into four major categories: **research**, **design**, **development** and **management**.

A successful **project team** has a balanced combination of carefully selected skills and attitudes during the entire **project life cycle** to match the needs of the project. Only then can the solution be the best it can be.

Research

- Abstract Thinking
- Diligence
- Domain Expertise
- Experimentation
- Findings Presentation
- Focus Refinement
- Hypothesis Formulation
- Measuring Results
- Narrative Construction
- Note Taking
- Questioning Mindset
- Resource Identification
- Source Attribution
- Training

Design

- Aesthetic Ability
- Analytical Thinking
- Animation
- Brainstorming
- Branding & Identity
- Color Communication
- Color Theory
- Communication Strategy
- Conceptual Modeling
- Content Strategy
- Creative Ideation
- Critical Thinking
- Data Visualization
- Design Critique
- Design Thinking
- Empathy
- Ethnography
- Graphic Design
- Grid Systems
- Human Factors
- Immersive Research
- Information Architecture
- Information Design
- Interaction Design
- Narrative Argument
- Participatory Design
- Print Design
- Prototyping
- Rhetoric
- Service Design
- Sketching
- Storyboarding
- Storytelling
- Synthesis
- Technical Writing
- Typography
- UI Design
- User Experience
- User Research
- Video Editing
- Visioning

Management

- Adaptability
- Budget Management
- Coaching
- Customer Representation
- Delegation
- Discipline
- Estimation
- Evaluation
- Goal Setting
- Leadership
- Loyalty
- Mentoring
- Motivation
- Persuasion
- Planning
- Process Definition
- Public Speaking
- Risk Assessment
- Risk Mitigation

Development

- .NET Framework
- AJAX
- Algorithms
- Android
- Artificial Intelligence
- ASP.NET MVC
- Automation
- Bash / Shell Scripting
- Bottle.py
- C / C++
- C#
- Code Review
- Continuous Integration
- Database Management
- Design Patterns
- Distributed Systems
- Embedded Programming
- Groovy / Grails
- Hibernate
- HTML / CSS
- HTTP
- iOS
- Java
- JavaScript
- JSON
- Machine Learning
- Microsoft SQL Server
- MongoDB
- MVC
- MVVM
- MySQL
- Networking
- Node.js
- Objective-C
- Oracle
- Perl
- PHP
- PostgreSQL
- Python
- Quality Assurance
- Real-Time Analytics
- Real-Time Applications
- Redis
- Responsive UI Design
- RESTful / HATEOAS API Design
- RESTful Web Applications
- Ruby on Rails
- Service-Oriented Architecture (SOA)
- Socket.io
- Software As a Service (SaaS)
- Spring MVC
- SQL
- Swing
- System Integration
- Task Estimation
- TCP/IP
- Timeseries Analysis
- Unit Testing
- Virtualization
- Windows Desktop Apps
- XAML
- XML

methods

We use the following HCD research and development methods to create **useful**, **usable and desirable solutions** for our clients. They help us effectively obtain, analyze, synthesize and manage information needed to solve the problems our clients face.

The methods are divided into six categories according to the key areas needed to successfully complete a project. These methods can be used during one or more of the five **Project Stages**.

Ask

Research requires engagement with people. The following methods help us get information through a **meaningful conversation**.

- Abstraction Laddering
- Buy a Feature
- Cognitive Walkthrough
- Contextual Inquiry
- Critical Incident Technique
- Critique
- Cultural Probes
- Customer Experience Audit
- Design Charette
- Design Workshops
- Diary Studies
- Directed Storytelling
- Draw the Experience
- Extreme User Interviews
- Five Whys
- Focus Groups
- Graffiti Walls
- Interviews
- KJ Technique
- Photo Studies (Camera Journal)
- Picture Cards
- Predict Next Year's Headlines
- Questionnaires
- Speed Dating
- Surveys
- Think-Aloud Protocol (Narration)
- Trioding
- Unfocus Groups
- Word-Concept Association

Look

Often times, people say and do differently. **Observing people** in their everyday surroundings brings us meaningful insights into their activities.

- A Day in the Life
- Artifact Analysis
- Behavioral Mapping
- Behavioral Archeology
- Design Ethnography
- Exploratory Research
- Fly-on-the-Wall Observation
- Participant Observation
- Personal Inventories
- Rapid Ethnography
- Shadowing
- Social Network Mapping

Understand

Gathered data must be carefully analyzed to identify patterns and determine priorities. Only then does the research translate into actionable **design implications**.

- Activity Theory (Activity Analysis)
- AEIOU
- Affinity Diagramming
- Brainstorm Graphic Organizers
- Bull's Eye Diagramming
- Business Origami
- Card Sorting
- Case Studies
- Cognitive Mapping
- Content Analysis
- Content Inventory
- Critical Incident Technique
- Flow Analysis
- Historical Analysis
- Importance/Difficulty Matrix
- Kano Analysis
- Laddering
- Literature Reviews
- Mind Mapping
- Object-Oriented Analysis
- Personas (Character Profiles)
- Problem Tree Analysis
- Rose, Thorn, Bud
- Secondary Research
- Stakeholder Maps
- Statement Starters
- Task Analysis
- Territory Maps
- Thematic Networks
- Usability Report
- Value Opportunity Analysis
- What's on Your Radar
- Web Analytics

Make

To bring new ideas to life and share them with others, research and design implications must be expressed through imaginative, **visual means**.

- Agile Modeling
- Automated Code Documentation
- Co-Located Development
- Collaborative Coding
- Collage
- Concept Mapping
- Continuous Deployment
- Continuous Integration
- Cover Story Mock-up
- Creative Matrix
- Deployment Automation
- Design for Change
- Elito Method
- Experience Prototyping
- Extensible Architectures
- Feature Driven Development
- Flexible Design
- Frequent Software Delivery
- Generative Research
- Image Boards
- Lean Software Development
- Mental Model Diagrams
- Object-Oriented Design
- Open-Source Software
- Parallel Prototyping
- Pragmatic Programming
- Revision-Controlled Documentation
- Round Robin
- Scalable Systems
- Scenario Description Swimlanes
- Scenarios
- Storyboards
- Sustainable Development
- Test Automation
- Test Driven Development
- Thumbnail Sketching
- Time-boxed Iterations
- Use Cases
- User Journey Maps
- Video Sketch
- Word Clouds

Evaluate

To achieve innovation, new ideas and concepts must be tested and improved upon via **frequent iterations**.

- A/B Testing
- Build Automation
- Cognitive Walkthrough
- Competitive Testing
- Continuous Code Review
- Continuous Requirements Analysis
- Continuous Unit Testing
- Critique
- Customer Experience Audit
- Desirability Testing
- Heuristic Evaluation
- Key Performance Indicators
- Participatory Design
- Quality Through Client Engagement
- Rapid Iterative Testing & Evaluation
- Role Playing
- Scenario Testing
- Stakeholder Acceptance Testing
- Stakeholder Walkthrough
- System Usability Scale
- Think-Aloud Protocol
- Time-Aware Research
- Triangulation
- Usability Testing
- Weighted Matrix

Manage

To ensure sure everything happens as planned, careful and iterative project management and **communication** is a must.

- Adaptive Re-Prioritization
- Agile Culture
- Agile Unified Process
- Cross-Functional Teams
- Daily Standups
- Iteration Kickoffs
- Scrum
- Self-Organized Teams
- Task Point Estimation
- Velocity Tracking

structure

As projects move through the main **Project Stages** , the focus, resources and effort of project team members shift to accommodate the current project needs to accomplish various project milestones.

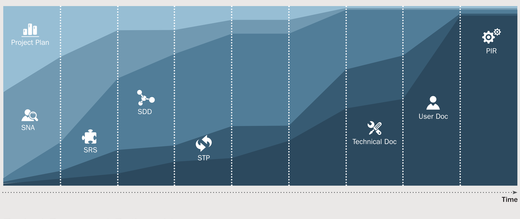
The beginning of a project tends to focus on **scoping** and **defining** the problem area as well as **exploring** the domain area, stakeholders and existing solutions. All research is then **synthesized** to inform the future solution.

Once research is translated into actionable requirements, the project moves into its middle phase and project team efforts shift to **creation** and **implementation** of new ideas. **Testing** and **refinement** efforts are happening in parallel. As the solution becomes more and more refined, resources shift to **launching** the solution, **testing** it within the target environment and **transitioning** it to the client.

Throughout the entire project life cycle, various **documents** are generated to maintain common understanding between the client and the project team.

SCOPE & Define
EXPLORE & Synthesize
CREATE & Implement
TEST & Refine
LAUNCH & Transition

Team Effort Through Project Life



Project Document Deliverables

Project Plan

The Project Plan identifies **project goals** and documents the **estimated resources** and **time** needed to achieve them. The main challenge is to optimize the allocation of necessary inputs and integrate them to meet pre-defined objectives.

In our approach, the project is seen as a series of relatively small tasks conceived and executed in an **adaptive manner**, rather than as a completely pre-planned process. We collaboratively track all project tasks using **infrastructure** that supports multiple users modifying different sections of the plan.

SNA

The System Needs Assessment (SNA) synthesizes the **user research**, **context of use** and the **gap area analysis**, creating the very first high-level system requirements. It also defines the main **stakeholders** and **target audiences**, recording their **needs** and **pain points**.

It outlines the current **use case scenarios** and makes high-level recommendations on how to accomplish these tasks using the solution. The document also defines the **assumptions**, **dependencies** and **constraints** and may include the projected development effort and timeline.

SRS

The System Requirements Specification (SRS) further defines the **scope**, **business context** and **purpose** of the future system. It expands upon the high-level requirements from the SNA to identify all required **system functionality** in terms of business, functional, system, user, hardware, user interface, logical data and information management requirements.

It documents all client-requested, organizational and team-identified **features** and **requirements** of the future solution and should be approved by the client before proceeding with any further development.

SDD

The System Design Document (SDD) translates system requirements into a complete design solution, outlining the **system architecture**, its **components**, **interfaces** and **data**. It documents the **user interface** as well as the **look and feel** of the system.

It is usually written in **two stages**: (1) the preliminary design in which the overall system and data architecture is defined, and (2) the detailed design stage when more detailed data structures and algorithms are developed for the defined architecture.

STP

The Software Test Plan (STP) provides an overview of the **test strategy**, a list of **testing deliverables** as well as the plan for development and evolution of the solution. It prescribes the scope, approach, resources and **schedule** of all testing activities.

The plan must identify the items and features to be tested, the types of testing to be performed the **applicable metrics**, the personnel responsible for testing as well as the risks associated with the plan.

Technical Doc

The Technical Documentation describes **handling**, **functionality** and **architecture** of the system. The intended recipient is the **administrator**, service or maintenance technician.

It translates the highly formalized SDD into more readable prose and provides enough information to understand the inner and outer **dependencies** of the solution. It specifies the **configuration** and **implementation** of the solution, including the build and installation instructions. It should also include the release notes and history.

User Doc

The User Documentation describes how the system is used from the **user's point of view**. It documents each feature of the system and assists the user in accomplishing the **tasks** the system affords. It includes a glossary and a thorough **troubleshooting assistance**.

A common way of writing is the **tutorial**, which guides the user through a step-by-step walkthrough of accomplishing a particular task. A **thematic approach**, where sections concentrate on one particular area of interest or interface screen are useful to a more advanced user.

PIR

The Post-Implementation Review (PIR), or the Post Mortem, determines project success by assessing the project **scope** and whether the required deliverables were produced within the agreed **time frame**. It compares the **expenditure** against **budget**. It is intended to **mitigate future risks** and **promote best practices**.

The PIR is the **last critical step** in the project life cycle as it allows an independent party to **validate the success** of the project. It also gives confidence to the stakeholders that the project has met the objectives it set out to achieve.

gains

Stemming from an ongoing exchange of ideas between the government, academia and law enforcement organizations, our projects generate a variety of important outcomes that are not directly related to client deliverables. We write **white papers**, **best practices, conference papers, blog posts** and other **publications**, increasing our **domain expertise, skillsets** and **reputation** as well as **innovating** the state of practice of digital intelligence and forensics.

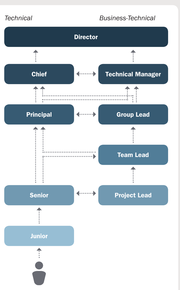
As we author thinking in real world concerns and empower practitioners with new approaches, methodologies, toolkits, services, artifacts and processes, we also create opportunities for **training**, **consulting** and **speaking engagements** and share our knowledge with the clients as well as the public.

Thanks to our focus on improving the lives of our clients and stakeholders, we increase their **trust** in us and build long-lasting relationships. Finally, as we complete projects, we gain the **intellectual capital** to advance our **career paths**.

Project Outcomes

- Best Practices
- Blog Posts
- Conferences
- Consulting
- Domain Expertise
- Increased Skillset
- Innovation
- Intellectual Capital
- Publications
- Reputation
- Speaking Engagements
- Stakeholder Trust
- Training
- White Papers

Career Path



We recognize that people have different interests and career goals. These two tracks accommodate people with both technical, project management and business backgrounds.

The fluidity of both tracks allows people to move across as their interests and career goals change and enables them to grow their careers while still allowing them to focus on the type of work they are best at and enjoy most. However, it also provides people with the opportunity to gain experiences in different career tracks so they can make an informed decision regarding their future career goals.

One of the biggest advantages of having two tracks is the retainment of good technologists who want to grow in their careers, but want to keep doing the type of work they are best at and enjoy doing: technical work.

This double-track setup also avoids brilliant technical people from being "pushed" (by themselves or their supervisors trying to "reward" them) into people-management responsibilities.

Lastly, it also reduces the likelihood of becoming top-management heavy by giving people enough opportunities to advance but still be able to participate as part of a project team.

CERT | Software Engineering Institute, Carnegie Mellon University.

This material has been approved for public release and unlimited distribution.

The content of this poster was collaboratively developed by the members of the **Cyber Engineering Solutions** team of the **Cyber Security Solutions Directorate,** part of the **CERT Division** at the **Carnegie Mellon Software Engineering Institute (SEI).**

SOURCES: IDEO Method Cards | Universal Methods of Design - 100 Ways to Research Complex Problems, Develop Innovative Ideas and Design Effective Solutions | LUMA Institute - Innovating for People | IEEE Standards

DM-0001077

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.