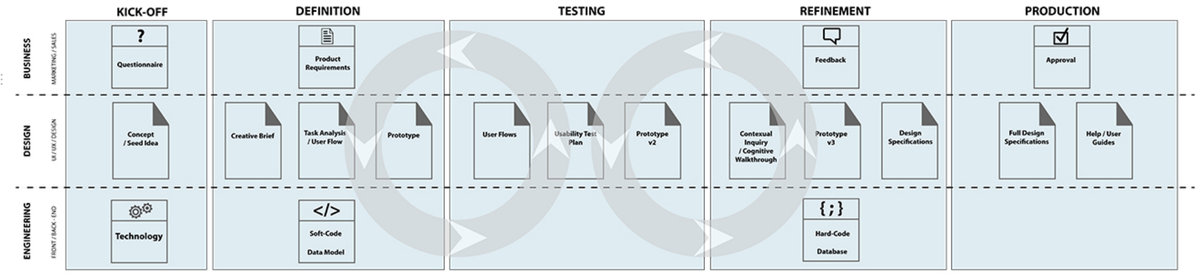
R • O • A • D

RESPONSIVE OPEN ANALYSIS & DESIGN



R.O.A.D. = Responsive Open Analysis & Design - Part 1: Defined

Based on my past experience with projects, I developed the **ROAD** design process to address the dynamic and unplanned nature many projects tend to follow. Due to how the technology industry operates (projects and product ideas in various states), employees that are new to the company or new to a project can utilize **ROAD** to assess how best to proceed from any given point, or **respond** to any *work product* thats handed to them.

The above approach follows a hierarchical and linear flow by default. Ideally, this is the most desirable route to maturing a concept, harnessing each *work product* in sequence. ***Work Products*** are resulting artifacts that come out of a process (i.e., creative brief, prototype, UI specification, etc.)

Though presented linear, the true power comes from the fact that this process is ***Work Product*** centered, meaning that any resulting artifact can serve as the point at which to tailor a process to build or refine a product. Also, because this process is design driven, the *work products* from **design** are at the core, with influencing factors coming from **engineering** (technology/code) and **business** (market research/survey). All can happen concurrently.

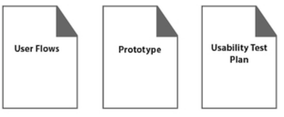
The five modules house *work products* specific or related to that part of the design. Modules have general activity names rather than one defining name since the focus is on the design pattern nature of the *work products* themselves. *Design Iteration* and *Refinement* are iterative, with *Production Build* being the point of hand-off for manufacturing. Though it says sign-off, this could also be a starting point of a project (product improvement or maintenance perhaps). In Part 2 we will provide examples of how **ROAD** works, which can also be demonstrated in person using 8.5 x 11 paper (modules) and 3 x 5 cards (work products).

R.O.A.D. = Responsive Open Analysis & Design - Part 2: Usage

To illustrate how this approach can be applied in diverse situations, three examples are presented below. They are based on my past (and current) experiences on projects and showcase attempts to navigate projects that had very little process or direction. The titles for each pattern (*Design Specification*, *Prototype*, etc.) are more category names than specific techniques, as each has many different forms or approaches they take. For example a Prototype can be either paper, HTML, soft-coded or interactive or static mock-up. Creative Brief could be a requirements specification, PRD business plan or proposal.

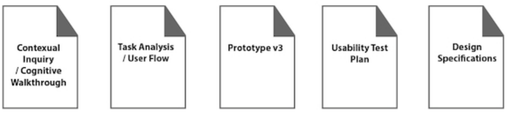
Case One: Experimental

Some projects have been in development well before design is considered, user research conducted, or sometimes a project changes course to address a new business objective. Determining what the *User Flows* are can quickly give insight into what UI elements to employ. Those elements can then be used to build a *Prototype,* which itself can be used in *Usability Testing.* Because the technology was largely built already but not well defined, the design was made to validate an experiment, so finding out what the user will do, building a functional model and testing it was the most efficient path.



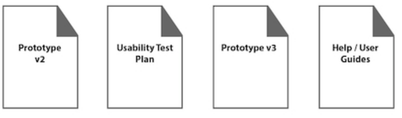
Case Two: Incremental Redesign

There are products that have been totally engineered and released with minimal input from the user or without any design consideration. When the various mistakes are apparent, a redesign and development is called into order and it is this particular situation that analysis can be applied to further uncover the problem areas and plan design solutions for them. *Contextual Inquiry* and *Task Analysis* are two very effective techniques for not only learning what a system or product does (or suppose to do), but where the problem areas are that users face. Once assessed, a prototype can be built from this information and tested. Once validated specifications can be drafted for the development team to make the necessary changes to the system.



Case Three: Nimble

A prototype can be the first thing a designer is given when a project starts out. This prototype can be in the form of a hard-coded demo with hooks to UI elements, a hacked / modified version of a similar product or converting an existing product from one platform to another. As such, the user flows and functionality has largely been defined, so it becomes a matter of testing those features and refining them for a beta release. Depending on the skill set of the designer, size of the team, time constraints and other factors, going right into scripting a better UI / Front-End may make sense or be the only option instead of utilizing all the patterns in this process.



Above are just some examples of how analyzing a project that may come your way can be navigated with the assistance of **ROAD**, using methods that best fit the situation. Timelines, budgets, resources, and other factors of course will impact how anyone of these are selected or carried out. The important thing is to ***Respond*** with a method most effective for the project on hand, stay ***Open*** to those factors, and continue to ***Analyze*** the situation as you ***Design***.

Work Product Output Types

Below list some of the outputs from each category that the designer can produce, analyze and design / build from. This list serves as only an example, and more outputs can be added at any time.

Concept	Creative Brief	Task Analysis	Prototype	User Flows	Usability Test	Contextual Inquiry	Design Specifications	User Guides
Brainstorm	PRD	Hierarchal Task Analysis	Paper prototype	Storyboarding	Eye-Tracking	Cognitive Walkthrough	UI Specification	FAQ
Affinity Cluster / Diagram	Proposal	GOMS Analysis	HTML Mock-Ups	Use Cases	Keystroke-Level Modeling	Heuristic Evaluation	Style Guides	Help Guides
Mind Maps	Requirements Gathering	Cognitive Task Analysis	Wireframes	User Scenarios	Survey / Questionnaire	Context of Use Analysis		Tutorials
	Competitive Analysis							

Design Process Pattern Structure

Below is a detailed example of the pattern description. It list the problem it addresses, when to use the method, how to employ it, and rationale for its use.

The pattern structure is defined for the *work product* category for efficiency and to prevent redundancy.

Task Analysis Models Pattern

What problem does the pattern solve?

- Defines the goals and objectives of a product / application
- Which tasks, skills, or goals ought to be the focus, that is, how to select features that are appropriate for completing goals / objectives
- Understand the sequence in which tasks are performed and / or should be learned / done.

When to use it?

- Use when it maybe unclear what the purpose of the product is
- When a product has been already created, but the performance needs to be improved (from complex to simple)
- When everything specified in the earlier stages of a project ought to be evaluated before the product is released

How to use it?

- Decide purpose of the product
- Get agreement between stakeholders on the purpose of the product
- Identify main goals / objectives
- Decompose into sub-goals
- Validate that task make sense (Test)

Why to use it?

- Part of the skill of the task analysis is in recognizing how data conflict and being able to see, for example, that the official account of how something works (e.g., according to training systems or senior management) is at odds with actual practice.

Suggested Time

- 2 Days