

What's your ProtoTYPE?

Picking the Proper Prototype to Suit the Task

Tony Walt | Lead Creative Technologist, EffectiveUI

Summary

Prototype has become a hot word in recent years, and it has collected a range of connotative meanings. Its primary use comes from the manufacturing world, and it means to make a model of something, to test it out before committing to full-scale production. Its origins extend back to early Greek civilization: *protos* = *first* + *typos* = *impression*. Holding this original definition in mind will provide a framework for how it is used in the world of digital product development, and the writing that follows will analyze and define the many different kinds of prototypes currently in use, each serving a different function according to the objective at hand.

Because the word prototype is so loosely defined, many people (drawing from the manufacturing definition) assume a prototype to be a hi-fidelity, functioning artifact – a working model. However, this represents just one instance in a spectrum of prototype *types*. They range in levels of sophistication, cost, aesthetics and functions, and a thorough understanding of each (when to use, why and to what end) will be an invaluable asset to the team tasked with building a digital product.

See examples of each type of prototype at <http://tinyurl.com/prototype-portfolio>

PROTOTYPE TYPES

While there are many *types* of prototypes, they all should achieve one or more of the following:

1. Facilitate more efficient iteration
2. Discover and address unknowns
3. Assist in the transfer of knowledge

Sometimes, one prototype might be invoked that answers one question which only leads to the discovery of an unknown or another question, the answer to which might come from using a different prototype, and so on, and so on. So, the idea of using multiple prototypes, concurrently as well as sequentially, is an essential feature of the prototyping process. Designers and developers must have thorough knowledge of each prototype *type* in order to know which one to use and when, when to combine them, the benefits and limitations of each, the associated costs, and the intended audience.

THE FOLLOWING IS A LIST OF EXAMPLE PROTOTYPE DEFINITIONS

Paper Prototype

Many people wouldn't consider pencil sketches or white board scratches to even be a prototype because of their simplicity and roughness. But it is precisely these features that make paper prototypes such powerful tools. Time investments at this

point are small, collaboration is high, iterations are rapid, and a collective, rudimentary vision can be struck upon quickly. This is where audience becomes a factor. While paper prototypes might work well among team members representing the different disciplines involved (because they have expert knowledge), they might lose their effectiveness with stakeholders and executives. Nevertheless, paper prototypes can be used with other prototypes as the audience changes during the design process.

Typical Project Phases

Sales	UX	Design	Design-Dev Handoff	Development	
What it Tests	Benefits	Inherent Risks	Visual Fidelity	Relative Cost	Time
User flows	Test flows quickly	Reluctance to redraw template pieces	Low	Low	Short
Design validation	Inexpensive				
	Encourages iteration	Difficult to present to client			
	Easy team collaboration and testing	Hard to demonstrate transitions			
		Sense of sizing difficult to convey			
Sample Tools: pencil, paper, scissors, etc.					

Animatics/Animated Wireframes

Animatics and animated wireframes are the first step in creating the *story* that will contextualize the product under development. While time and cost investments increase under this prototype, a clearer vision of the product comes into view. Even so, the costs at this point are still minimal, and complete changes in approach and direction are still possible – after all, the product is still in its infancy. There are many unknowns yet to be identified. Team members are still discovering which questions to ask and determining which prototype will answer them. Iteration is still very high, and testing with users can begin with these artifacts, powerful tools for something that is still in the low-fidelity arena.

Typical Project Phases

Sales		UX	Design	Design-Dev Handoff	Development	
What it Tests	Benefits	Inherent Risks		Visual Fidelity	Relative Cost	Time
Story or concept	Focus on wireframes over design	Video format less agile if structure is reworked		Low	Low	Medium
User flows						
Timing	Quick to build	Changes after production begins can be expensive				
Sample Tools: pencil, Adobe After Effects, Apple Keynote						

Vision Demos

Vision demos are story driven. They imagine an everyday user and create a scenario in which the key components of a digital product can be highlighted. They show the obstacles and propose solutions in a short video format. The fidelity and expense can be rather high, but this is a prototype model that *sells* a design construct and can do so at scale. The final video is easily shared, tells a consistent story, and takes minimal time to consume.

The beauty of a story-driven prototype is that *story* is a natural construct to our species. We gravitate toward it. Ancient peoples used stories to convey a universal truth, to teach a lesson, and to make a point. It then seems only fitting that this construct should be a valuable option for realizing digital product and convincing others that moving in a particular direction is logical.

When crafting a vision demo, you must make decisions about the production style of the video that supports the story being told. Productions can be focused solely on interface, or they can involve

characters and an ecosystem of applications. They can be silent or involve intricate scores and professional actors. Some are less than a minute in length, while others span several minutes.

These decisions can heavily affect the final cost of a vision demo, and thus should be considered carefully. The biggest cost driver in a vision demo, however, is major changes that occur after production has begun. It is best to utilize animatics to prototype a story construct and timing prior to the start of production.

Typical Project Phases

Sales UX Design Design-Dev Handoff Development					
What it Tests	Benefits	Inherent Risks	Visual Fidelity	Relative Cost	Time
Concepts	Portable	Concepts may not be technically viable	Medium / high	Medium	Medium
Ecosystems	Tells a story that quickly conveys and idea and a message	Often small timeline and budget, so failure puts greater job at risk			
	Can encompass an ecosystem	Changes after production begins can be costly			
	Excellent for attaining funding				
	Extreme longevity				
Sample Tools: Adobe Premiere, Adobe After Effects					

Click-throughs

Think of click-throughs as skeleton websites or apps. They are static artifacts, but the key dynamic features are added so that a feel for user experience can be quantified. Click-through prototypes can either be built by stringing together pre-existing screen designs, or they can be constructed dynamically via code or a dynamic prototyping application.

It is often best to start with the simpler click-throughs made from images strung together because of their minimal investment. The images can be as simple as wireframes void of aesthetic refinements used to inform good UX design, or they can be visual design iterations used to test the look and feel of the application.

Once key structural questions are answered in simpler click-throughs, the prototypes can evolve into the more dynamic click-throughs, or even into functioning prototypes.

Typical Project Phases

Sales UX Design Design-Dev Handoff Development					
What it Tests	Benefits	Inherent Risks	Visual Fidelity	Relative Cost	Time
User flows	Controlled story/flow	Finding hotspots can be difficult	Low / Medium / High	Low / Medium	Medium / Long
Interaction models	Can run anywhere	Visuals can look raw			
Visual design	Easy/Quick to build	Won't resize			
	Extreme iteration				

Sample Tools: Envision, custom-built tools, Adobe Acrobat, Omnigraffle, Apple Keynote, iRise, Axure Keynote

Functioning Prototypes

A functioning prototype is a fully coded mixture of static and functional screens. It might be built once a lower-fidelity (lo-fi) prototype has been accepted. If risk management is the model upon which clients determine vendor viability, then it holds that risk management is the lens through which developers must view prototyping. Functioning prototypes are likely to only be presented once other prototypes have been accepted. They often serve as an ideal means to transfer knowledge to the final development team as to how an experience and its code should manifest.

The beauty of prototyping is that these lo-fi and hi-fi prototypes need not be developed in sequential isolation. It would behoove the developer to make small investments in a hi-fi prototype alongside its lo-fi counterparts as the dialogue between entities becomes entrenched.

The hierarchy of prototype complexity would be something to consider as it becomes increasingly obvious that a client will award a contract. The carrot, though it is small, oftentimes gets the wagon where it needs to be.

Typical Project Phases

Sales UX Design Design-Dev Handoff Development					
What it Tests	Benefits	Inherent Risks	Visual Fidelity	Relative Cost	Time
User flows	Very accurate representation of final product	Sometimes difficult to define (what gets coded to what level)	Medium / High	Medium / High	Medium / Long
Interaction models	Potentially reusable code	Clients may mistake it for version 1			
Developmental feasibility	May have reflow abilities	Structural changes can be difficult			
Design responsiveness					

Sample Tools: Web technologies (HTML, CSS, JavaScript, etc.)

Interaction Vignettes

An interaction vignette focuses on motion and transition or single complex interactions. Producing an interaction vignette allows an animator to easily prototype motion or visualize an interaction. Once complete, the vignettes are used to transfer that knowledge to a developer rather than allow such things to be an after thought.

These prototypes are becoming increasingly important as newer Web technologies are bringing motion back into the forefront of Web interfaces. The vignettes help to inform how and when to utilize motion, which story each movement tells and whether it's necessary.

Typical Project Phases

Sales UX Design Design-Dev Handoff Development					
What it Tests	Benefits	Inherent Risks	Visual Fidelity	Relative Cost	Time
Timing	Accurately convey a complex interaction without the need for expensive coding	Interaction could be too complex	Medium / High	Low	Short
Transitions		Can lose time if assets are not properly prepped			

Sample Tools: Adobe After Effects, Adobe Flash, Apple Keynote

Proof of Concept

Sometimes, a client will need to know that a singular, critical feature can be deployed, and this is where proof of concept becomes vital. Proof-of-concept prototypes identify the challenging features that are core to a digital product and isolates them to determine their feasibility and effectiveness. These are the lynchpins, and if they cannot be reconciled at the incipient phases, then the product cannot move forward.

Proof-of-concept prototypes can be constructed initially in isolation, free from the non-essential constraints and weight of the greater system. This allows for quicker coding, testing and iteration. Once proven, they are integrated into the complete system for additional testing.

Typical Project Phases

Sales	UX	Design	Design-Dev Handoff	Development	
What it Tests	Benefits	Inherent Risks	Visual Fidelity	Relative Cost	Time
Concept feasibility and/or usability	Determine if a concept can be realized and if it will work as intended	Idea may not be viable under constraints of current infrastructure	Low	Low / Medium	Low / Medium
Sample Tools: the technology of the final product					

CHOOSING BETWEEN PROTOTYPES

To determine which prototype best suits a situation, you should consider its goal and match it with the most efficient prototype capable of meeting that goal. Choosing an option that achieves the goal – yet requires little effort to produce – facilitates iteration that encourages the exploration of more solutions and reduces the risk of failures down the line.

One of the most important functions of prototypes is that they help identify which questions to ask. Questions are the guiding force of any exploration. You wouldn't think of going to your doctor without a ready-made list of questions. Questions are not an indicator of what I *don't* know. They are a guide for what I *want to know*. Below, you will find a list of useful questions that can help steer your decisions on which prototypes may be most useful in your current stage of development.

BUSINESS-FOCUSED QUESTIONS

What are the business goals for the prototype?

This is the most important question to get answered, as it should

provide you with enough information to answer many of the following. Examples may include: user testing, stakeholder buy-in, proof of concept, product beta.

What is being tested?

Each of the prototype types can be used for testing different things. So be sure to choose a type that is ideal for testing what you want to learn.

What is the budget?

Some prototypes inherently cost more than others, but all prototypes have cost drivers that can be controlled to determine the final expense.

What is the deadline?

All prototypes should reduce production time and risk of the application they are designed to test. However, some prototypes take longer to produce than others, and therefore may not be the best prototype option when constricted to a short timeline.

Will the prototype be addressing a single device/program or an ecosystem? If the prototype addresses multiple devices or programs, then you may need to build several prototypes that work together. Alternately, if the intent is to sell a concept, then a vision demo will likely be most useful.

How does the prototype relate to other work on the project?

If the prototype needs to integrate with other work being done on the project, it is best to consider how that may happen. For example, it may be useful to supplement a functioning prototype with a style guide or wireframes of pages that are not covered in the prototype itself.

Do you already have a prototyping tool the client wants you to use? If you don't, there may be some time spent acquiring and learning a new tool, and that must be factored into the timeline and budget.

DEVELOPMENT-FOCUSED QUESTIONS

Does the code need to be reusable?

If the code for a prototype is intended to evolve into the final production code, then the structure of the prototype and its code should be of a higher standard. This will increase the development effort, but it should save time later in the project.

It is important to note that coding built this way should not take place until key structural decisions are determined through lower-fidelity prototypes that facilitate more iteration.

How complex of interactions need to be demonstrated?

If complex features like charting, form validation and touch gestures are a concern, it's likely that more functional code needs to be written. This pushes the prototype toward a functioning prototype.

It may be possible to tackle such features in a proof-of-concept prototype, while other decisions are made in simpler click-through prototypes. As the structure solidifies, you may then combine the two into a functioning prototype with several static screens that get coded out as needed.

Which browsers/devices does the prototype need to support?

When a prototype is required to support more browsers and devices, the development effort will increase.

Simpler click-through prototypes typically don't resize and are better-suited for a pre-specified display resolution. They can be made to resize if hand-coded, but that will take extra work.

Functioning prototypes can accommodate various displays, but they must be built with fluid and responsive layouts. If older browsers must be supported, there will also be a need to detect supported features, and code must be written to handle instances when those features aren't supported.

Because prototypes are mainly constructed for speed and efficiency, it's best to minimize support to only what is absolutely essential. High-functioning knowledge transfer prototypes that are intended to evolve into production code are a likely exception to this.

Some technologies for constructing a prototype, like Flash, will not be viable options if the prototype must run on a mobile device.

USER TESTING-FOCUSED QUESTIONS

Is the prototype for user testing?

If the prototype will be employed for user testing, then the following questions will need to be considered.

What will be the testing methodology?

Some prototypes, like click-throughs, are built with limited paths through the application and may require a means to guide the user down that path. A few methods of doing this are click-scripts, a linear navigation button or hotspot indicators that flag the active buttons. Examples may include: in-person guided, in-person self-guided, remote user-guided, and remote self-guided.

At EffectiveUI we have developed a script that highlights image map hotspots in any Web technology-based prototype. The files for this script can be found on GitHub.¹

Will the prototype require analytics?

Prototypes that are self-guided may benefit from analytics that flag events like "undo" or "back" button clicks, as well as metrics like time-on-page or eye tracking. This information may be gathered by integrating third-party software, or via custom-developed tracking code. The custom code may be reusable between prototypes.

DELIVERY-FOCUSED QUESTIONS

Is the prototype considered a deliverable?

Prototypes that are final deliverables typically require a higher fidelity than ones that are used solely for internal testing.

How will the prototype be deployed?

If the prototype will be hosted and tested online, then security concerns may need to be taken into account (especially since prototypes are often created for products still in development). A simple password requirement may be enough to satisfy some security concerns, while others require that the prototype have individualized access, or prevent it from being hosted in the cloud.

Many times a prototype will need to run in a variety of unknown environments that can include unpredictable Internet access and multiple uncontrolled devices. In these instances it's wise to build functioning prototypes without reliance on a server or other files found online. This allows the prototype to be portable and easy for user testers to run on their own devices with no set up.

CHOOSE WISELY

Prototype is a loosely defined term because there are many different kinds, and the name and definition of each type should reflect the primary function of the prototype. Most projects will require several different prototypes implemented at various stages of development. Some will be internal, while others will be able to be transformed into deliverables.

Always choose the simplest prototype that will achieve your goal. This promotes iteration and exploration. If you aren't able to act with agility, you are probably using the wrong prototype or technology to generate it. There is no predefined set and sequence of prototypes that will work every time. The demands of the project will determine which will be used and when. It is also important to be flexible in shifting between prototypes. In the end, choices among prototypes might be the difference between a successful product and a failed product, because prototypes can generate manifold perspectives of a desired outcome. Anything that offers a clear vision of what is intended, what is possible and what is practical will ultimately contribute to the success of the product. Know the prototype options available to you, and choose wisely.

About the Author

Tony Walt is a lead creative technologist at EffectiveUI, where he applies his expertise in user experience, design, media and development for key client projects. He has worked with such diverse clients as Microsoft, Adobe, Navistar/International Truck, TLC, the Office of the Director of National Intelligence (ODNI), National Oilwell Varco (NOV), Fidelity, TIAA-CREF, Vanguard, Wells Fargo, the U.S. Navy Blue Angels, Deloitte, Level 3, Emerson, Kodak, Century Link, Navy Federal Credit Union and Cancer Treatment Centers of America.

With 14 years of experience, Tony is known for combining rich media (like 3D, video and animation) with development of sites or applications. He has a knack for solving complex issues via thorough research and a very prudent process. He is efficient at honing a workflow to remove inefficiencies. He aspires to create experiences that are unique, emotional and memorable, and is always keen to learn new things to expand the wide spectrum of his skills.

As a respected industry thought-leader, Tony has written articles for UX Magazine and a white paper on responsive Web design for EffectiveUI. He has spoken at technology events such as Microsoft MIX and Streaming Media West and HTML5 Video Summit. Tony is a founding member of Creative Mornings and serves on the Professional Advisory Committee at the Art Institute of Colorado, helping determine the course curriculum.

Tony's work has been honored with numerous awards, including the American Business Awards "The Stevies," the Creativity International Awards, the Web Marketing Association's WebAwards, the Webby Awards, the OMMA Awards for Web Site Excellence and Graphic Design USA's American Web Design Awards, among others.

Prior to EffectiveUI, Tony was the lead CQ5 developer at Faction Media, where he architected the CQ5 site for Ping Identity. As senior vice president of retail user experience at Nexus Retail, he directed the CRUX program, fusing physical retail and digital facets of clients' brands. Tony founded Fusion Media Interactive, an award-winning agency that produced 3-D, print, film, video and interactive Web, kiosk and disc interfaces for high-profile clients.

Tony earned his Bachelor's degree in animation and media arts from Art Institute of Colorado. He also studied fine arts with a concentration in electronic media at the University of Cincinnati. He enjoys exploring new technology, 3-D, motion design, psychology, interface design and development, skiing, soccer, brewing beer, painting, rock climbing, rafting and mountain biking.

LinkedIn: <http://www.linkedin.com/in/tonyjwalt>

About EffectiveUI

Navigating the most complex business problems, EffectiveUI designs and builds digital solutions for the Fortune 1000 and ambitious innovators. Driven by the belief that the interface between people and technology makes or breaks the success of every digital product, application and user experience, the company's purview spans anywhere an effective UI is needed, from smart phones to flight decks. In addition to creating award-winning digital products, mission-critical Web applications and integrated mobile apps, EffectiveUI provides its clients with strategic guidance, training and embedded experts to spur internal progress. EffectiveUI is a WPP company (NASDAQ: WPPGY). WPP is the world's largest communications services group.

References

- 1 <http://github.com/effectiveui/area-highlight>