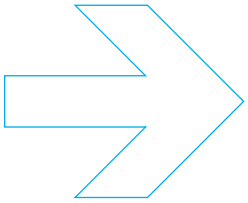
How To Get Amazing Software Out The Door Fast

5 Steps to an Unbeatable Product Creation Process

macadamian



Planning, building and releasing software that is successful in the market is a completely different challenge than it used to be. In the early days of software, success was all about technical wizardry. Who had the longest list of features? Who had the fastest engine?

WITH FACEBOOK, GOOGLE, SALESFORCE.COM AND THE IPAD setting new standards, customers now expect software to be visually impressive and easy to use, with software updates counted in days and weeks rather than months and years.

But, if we're all using the same tools, have all jumped on the Agile bandwagon, and are all hiring the best designers, why are all products not created equal?

We believe the next level of success is about getting the right design, product management, and development competencies seamlessly integrated, working together as one team with a **common goal**—amazing software that gets out the door fast.

Our organization has had the opportunity to work with some of the biggest software companies leading this movement. In this paper we describe five recurring patterns in the way these companies are now structuring **multi-disciplinary software teams** that fuel their current success.

Pattern #1

The Holy Trinity of Software Design

ACROSS ALL INDUSTRIES, COMPANIES ARE WAKING UP to the power of design. They are realizing that a great design can differentiate a product in a field of competitors, reduce development churn, and sell more product.

However, a number of companies are spending top dollar to hire designers and build a User Experience (UX) team, but are making frustratingly little progress.

Who's In Charge of your UI Design?

One of the first questions we ask every new client is simply "Who's in charge of your software design?" The most common answers are instantly revealing:

"Our software architect also designs of UI"

Most major universities now offer 3-4 year bachelor's and graduate-level programs in Human-Computer Interaction (HCI) or Interaction Design—disciplines that have almost no crossover with a Computer Science degree. Unless your software architect has had formal training, it is unlikely he or she will have the formal UX background required to create a usable, intuitive UI.

Moreover, architects are an extremely valuable resource—their time should not be split owning the user interface as well as the system architecture.

"We have a designer who takes care of it."

Most often when we probe more into this answer, we find out that what they really have is a graphic designer, who is in charge of making their product "look good".

Great graphic designers (also known as visual designers) are essential. Good ones know how to present visual information in a way your users immediately "get". They help make the potentially confounding analytics graphs in your product easy to understand. They design icons that help your users instinctively understand what to do. But there is a lot more to creating a design users love than visual look and feel.

It Takes Three

There are, in fact, *three disciplines* involved in creating a great design that are as different from one another as marketing, sales and engineering.

- **Design Research.** Design researchers specialize in uncovering user needs. They train for years to learn how to interview and observe users. Their findings almost always yield incredible insights that can be used to determine the exact point in the workflow where users are abandoning your e-commerce website or even help you uncover the next big innovation in your product line.
- **Interaction Design**. Interaction designers are the masters of intuitive layouts, workflows and content prioritization. They work with product management and design researchers to obtain market and user research and translate it into a draft of what the product will look like, how it will behave, and how it ties back to the user's goals—usually in the form of sketches called "wireframes".
- **Visual Design.** Visual designers are graphical experts that specializing in tools like Photoshop and Illustrator to add the right visual "wow" to software. Good visual designers can provide users with an instant emotional connection to a product even before they start using it.

Each discipline is so different from the other that companies are best served with even part-time help from a specialist in each area rather than one full-time jack of all trades.

Good visual designers can provide users with an instant emotional connection to a product even before they start using it.

Product Managers and Design Researchers—Partners in Strategy

Sometimes a great product strategy might not even make it to market if it is first intercepted by internal executives with their own perceptions and biases.

When a product manager and design researcher form a strong partnership, they can make sure they get the organization's buy-in by focusing on the *facts* rather than opinions and speculation. The "facts" are a combination of **user data** and **market data**.

Market data is crucial information about customer demographics, perceptions, market demand and market opportunities. User data provides specific and actionable information on your particular end user (who may be different from the person actually purchasing the product). This information offers insight into what customers and users actually need—not just what they say they want or what the company thinks they want.

What works best is when a product manager and design researcher co-present the user and market data to all internal stakeholders along with the plan for product design and rollout, using storyboards, narratives, charts, and concept designs.

KEY TAKEAWAYS

- **Great design encompasses**
  - How the product looks aesthetically
  - Whether the product flows naturally and intuitively
  - How the product ties users' goals back to the software organization's business goals
- **A design team is made up of 3 distinct competencies**
  - Design research
  - Interaction design
  - Visual design
- **Product manager and design researcher improve their odds of receiving organizational buy-in on product strategy by co-presenting market and user data.**

FURTHER READING

- "A unified approach to visual and interaction design." (Cooper)
- "User Experience Design–Helping Product Managers Sleep At Night." (Macadamian)

Pattern #2

Product Management,Design, and Software Development Groups Working as ONE Team

POOR PRODUCT DESIGNS ARE OFTEN THE RESULT OF structural and process-oriented problems. Clients will often say, “The product manager presented a great-looking product concept, but the final product just didn’t live up.” or “Our developers are really frustrated with our designers. You guys aren’t going to give us a blue sky design are you?”

Avoid Functional and Geographical Silos

There are a number of underlying root causes for these problems. Here are a few we frequently encounter:

- Product management works with the design team at Global HQ, then ships the design offshore. Everyone crosses their fingers and hopes the finished product comes back as beautiful as the original design (hint: it *won’t!*).
- A single, understaffed design team (or in many cases, a single designer) within the organization serves all of the development teams. Naturally, the lone design team is swamped. Managers and developers lose faith that the design team can deliver and, as a result, begin to exclude designers from projects.
- An external design firm creates a design and “throws it over the wall” to your development team, never to be heard from again once engineering begins.

Leading product firms like Apple, Google and Facebook are much more integrated, because the silo-ed approach inevitably leads to problems. When a design team is pushed to work in a vacuum, the development team often ends up with a bunch of pretty pictures (in their words) and is left to interpret and fill in a lot of blanks. What does clicking on this button in the corner do? What happens when the user navigates back? What do we show when the app loses connectivity? How should controls resize and anchor on the screen when you pinch and zoom?

Certainly there are advantages to working in a functional group or having multi-site teams spread out geographically. But being in a separate function or geography is no excuse to work in a vacuum.

Communication Accelerates Your Success

The same product leaders—Apple, Facebook and Google—could afford to use hundred-page style guides and detailed design documents that dictate the interaction and behavior of each and every control, the pixel-level spacing between every control on the screen, etc. Most organizations, however, don't have the time or budget to write up such a detailed guide.

In our experience, the fastest, most cost-effective and most efficient way to ensure that the final design looks as great as the original mockups is to **communicate as one team**.

Daily communication between all design disciplines allows for a superior design

Daily, detailed communication can only happen when all design disciplines (research, interaction design and visual design) are included as part of the team—in the daily scrum meetings, in the same project intranet, with the same product manager and project manager, and ideally in the same office.

Communicating daily as a team will:

- Ensure *product strategy* and *design intent* are maintained throughout product development.
- Prevent the design team from making a blue sky design that the rest of the team doesn't buy into or can't build in the desired timeframe.
- Allow the design team to give input on the design and the kinds of interactions that may occur.
- Help developers show designers what the technology platform can do.
- Allow designers to quickly explain complex interactions verbally and eliminate the need to spend days drafting diagrams to encapsulate every detail.
- Help designers and developers to better understand each others' goals, motivations and processes.

This daily communication between all design disciplines allows for a superior design because it delivers a complete experience that leverages the available technology—not simply a static design that looks good in Photoshop.

KEY TAKEAWAYS

- Teams may be dispersed geographically or by function, but should be encouraged to work together as one team to ensure product success.
- Frequent (daily) communication between all team members ensures each function's intent is understood and no context is lost.
- Ultimately, the investment in communication ensures less time wasted and a quality product.

FURTHER READING

- "Designers and Engineers Unite!" (Macadamian)
- "Reminder: You Are Not Managing Lemmings." (Thizy)

Pattern #3

Clear Roles and Responsibilities

WHAT IS A PRODUCT MANAGER RESPONSIBLE FOR? The answer to this question changes with every company we speak with. Product managers tend to be jacks of all trades with as many as *40 different responsibilities*, according to the Pragmatic Marketing framework. We've encountered product managers responsible for QA testing, investigating legal liability for their product, and even developing code.

What about a business analyst, software architect, design researcher, or interaction designer? What are they responsible for? Ask and you get a stunning number of different answers.

The fact that every company expects different things from different roles is not, in and of itself, a major barrier to product success. What does affect success is when team members aren't clear on their role within the team, their co-worker's roles, and their responsibilities to one another.

First, Understand Each Other's Goals and Motivations...

"*That jerk!* He just doesn't understand what we need to do here." While this may express a common frustration between functional groups, Steven Covey, author of *7 Habits of Highly Effective People*, famously said: "Seek First to understand, then be understood". In a nutshell, to work as a team, all groups must understand each other's goals, motivations and priorities.

The following generally hold true from company to company:

Product management is responsible for the overall success of the product in the market. This can be tied to product revenue, or simply getting a product out the door that satisfies a particular set of business requirements.

Designers are looking to design the best possible experience for the user. This group is measured on design success and, to some degree, product success in the market.

Developers are measured on speed and quality. They need to deliver on time and on-budget and their code needs to be bug free (which only adds to the on-time problem, since bug-free code takes time).

Understanding these main differences is key. Developers are not pushing back on feature requests because they're stubborn, but because they're responsible for maintaining timeline and quality. A designer doesn't look to change the design "yet again" because he gets a giddy thrill each time he pushes out the timeline—it is because he needs to make sure that users will love the product.

To ensure your teams understand each other, consider investing in basic training around each others' disciplines. There are a growing number of experienced consultancies that offer half or day-long seminars for product teams on this very subject.

Before the project begins, determine who the team will count on for each deliverable so you can avoid arguments down the road

Focus on The Gray Areas

Since product managers are ultimately responsible for the success of a product in the market, we recommend they take the lead on clarifying roles and responsibilities throughout the team, ensuring team members are each put into a position where they are maximizing their strengths.

For example, though product managers often take on the development of personas as well as requirements and workflow definition, these are all specialties of trained design researchers. Effective delegation of these tactics to the design researcher will free up more time for strategic business-level product planning.

What about the design? Who has the final say? In our experience, there needs to be a lead designer who takes feedback from the team and uses his or her experience and judgment to make the best design decision for the good of the product. This doesn't mean the designer "takes over" product management, which is the last thing we'd recommend. It also does not mean the designer should ignore technical considerations. Instead, the lead designer should "own" final design decisions in the same way that your chief architect or lead developer likely has the last say on challenging architecture decisions.

And what if a particular feature will impact the timeline? Who decides if it is in or out of scope? We've often seen this call made by the development team, but ultimately we believe this is a product management call. The product management team should decide whether to ship late with a fuller scope, or earlier with a limited scope.

To ensure all team members are clear on their respective roles, we recommend writing out a list of these responsibilities. Before the project begins, determine who the team will count on for each deliverable so you can avoid arguments down the road and committee decisions that ultimately slow down the process and hurt product success.

It Comes From the Top

Ultimately, what's truly important to the organization is revealed at the top. Organizations serious about delivering acclaimed products in the market often invest in a VP of Product Management who is separate from the Marketing department.

A more recent trend is the addition of a VP of User Experience. This VP ensures emphasis is placed on the user's overall perception of the product during use. The best way of growing business is through referrals from satisfied users.

If you are a GM or CEO reading this paper, *who is on your management team*? Does it reflect the priorities you have for software product success?

KEY TAKEAWAYS

- Product Managers, Designers and Developers are evaluated differently on product success.
- Each function needs to understand the others' motivations through communication and training.
- Agree upfront on roles and responsibilities of each function within the product context.
    - Focusing primarily on the gray areas where there is overlap (product management / design researcher, interaction designer/software architect, etc.).

FURTHER READING

- "Did you bring a knife to a gun fight? Three reasons your UI design investment isn't paying off." (Macadamian)
- "Living in an Agile World—Product Management when Development Goes Agile." (Johnson)

Pattern #4

The Right Amount of Upfront Planning

THE FIRST GENERATION OF SOFTWARE PLANNING PROCESSES were termed “Waterfall” and were severely derided over the past decade for being heavy, time consuming and wasteful. The second generation of processes took on various forms of “Agile”, where many teams abandoned upfront planning and jumped right into short sprints or iterations.

What we’re seeing now is the emergence of a third generation involving just the right amount of upfront planning in something called “Sprint Zero”, a term borrowed from the Scrum methodology. What you focus on and what you accomplish in your **sprint zero** will have a tremendous impact on the success or failure of the overall product.

Requirements Upfront

No matter how agile development is, you’ll never build a successful product if the work being done isn’t aligned to the company strategy and market needs.

What level of product requirement detail is needed upfront and how should those requirements be expressed? This is a big topic and already well-covered by Steve Johnson in “Writing the Marketing Requirements Document”.

We highly agree with his recommendation to use personas, not only because it is a way of communicating context that would otherwise be challenging to express in a bulleted list of functional descriptions (“The software shall perform…”), but also because personas are a key tool used by designers.

The Design Blueprint

It is clear that you need to establish the basic skeleton of your design from the outset, as it will shape the entire product in each subsequent sprint. This requires some upfront planning.

In his paper "Introduction to Agile Usability," Scott Ambler recommends having at a minimum:

- An overall organization for the parts of the UI that fit with the structure of user tasks.
- A common scheme for navigation among all parts.
- A visual and interaction scheme that provides a consistent look-and-feel to support user tasks.

We call these information architecture, primary workflows and basic interaction structure, but it matters less what you call them, and more that you do them.

In sprint zero, the software architect defines the high-level structure of the system so that the software developers have just enough information to start developing in parallel. Similarly, the lead designer needs to define a UI blueprint that includes just enough of the main UI framework and guidelines that multiple team members can work in parallel with confidence that their individual pieces taken together will form a consistent user experience.

The entire team—product management, design, analysts, QA, architects and developers—need to have a firm grasp of the technology that the software product will be based on.

Communication Tools and Artifacts

One of the biggest hurdles that product teams face is understanding what each functional group will be producing, and what they need from each other to do their best work. In a way, designers, developers, and researchers all speak different languages.

While a design concept is good at communicating a sense of what something might be, it's far from being in a language that developers will be able to use to implement. Just think for example of a concept car–"It's the automobile for the new generation, completely integrated with social networks and mobile technologies while being completely simplistic in its operation. The car you feel safe that your teenager can drive and enjoy." Interesting concept; could you build it? At the same time, engineers need to understand how best to communicate technical constraints without snuffing out the design process.

Ambler recommends using modeling tools and artifacts that reflect agile practices and are familiar to designers and the product owner. XP teams, for example, prefer to work with index cards and user stories. AUP teams prefer whiteboard sketches. All of these mediums are used regularly by design professionals.

Everyone Needs to Understand Technology

Contrary to the idea that only developers need to worry about the underlying details of the software technology, we posit that the entire team–product management, design, analysts, QA, architects and developers–need to have a firm grasp of the technology that the software product will be based on.

Consider the Android mobile platform as an example–Android users have become accustomed to the look and feel of popular Android applications developed by Google and its partners. These were developed using Android-specific design constructs like activities, fragments and intents that determine how the application behaves.

Using these constructs is vital in developing an application quickly! Developing custom designs that don't leverage these default Android constructs can take up to ten times longer and still not have the right look and feel at the end of the day.

What's more, understanding these constructs will ensure the whole team is again speaking the same language. For example, Android developers use the term *gravity*, which is a close equivalent of the term control alignment on other platforms.

The Android example aside, we've found this to be true for any of the numerous modern technologies that are proliferating, from iOS to Silverlight to Google Web Toolkit. If the designer knows how to use the native controls, the development will be much faster than if the designer or product owner ask for custom controls, and the team will be speaking the same language. To ensure product success, ensure that the entire team has a thorough understanding of the platform, its feature and its constraints.

KEY TAKEAWAYS

- Plan to have the right information defined and agreed in a short initial project phase ("sprint zero").
- Use personas to explain the intent and context behind product requirements quickly.
- Create a design blueprint with basic information architecture, primary workflows and interaction structure.
- Agree on communication tools like index cards or user stories.
- Ensure all team members have an understanding of the technology platform and its constraints.

FURTHER READING

- "Usage-centered engineering for web applications." (Constantine & Lockwood)
- "Your First Android Release — It Could Go Really Well or Really, Really Badly." (Macadamian)

Pattern #5

Teams That Work In Parallel

Perhaps the most challenging aspect of modern software creation is having multiple team members—designers, developers, architects—working **iteratively** and **in parallel**. Wouldn't it just be easier if Product Managers completed their requirements definition, then handed it off to designers to create the full design, who then handed it off to development to build and test?

Easier? Maybe. Will it yield a more successful product? Definitely not.

There are a couple of reasons functions can't "take turns" sequentially designing and building a product:

1. The steps are not sequential. Designers need to test that a design is working over the course of development. Product Managers need to adapt requirements as they get updated competitive data.
2. Designers need room to be creative, and that means the ability to fail and iterate.
3. Part of success is releasing in time to capitalize on a market opportunity. Working in parallel with daily communication is much faster than sequentially.

Here is a glimpse at how we've seen very successful teams manage this idea of mass parallelization and iteration.

Start With Back-End Development

Many developers are used to the idea of developing the UI first and then implementing back-end functionality progressively. This is normal—business stakeholders are visually inclined and have always put more emphasis on having the visible parts of an app built first for things like demos. How many managers do you know that see a UI with no back-end and assume the product must be almost complete, when in fact the back-end has barely been started?

Developers who are comfortable with the idea of starting on the back-end and making sure to have a robust data model while the designs are being fleshed out typically thrive in modern software environments. They recognize that well-architected software has a clear divide between the front-end and back-end and starting with the back-end actually contributes to a solid architecture.

What this means is that Product Managers need to work closely with the architect and developers early on to establish the data model based on product requirements. Follow the requirements advice we gave earlier, and you should be in good shape.

The other more unfortunate consequence is that from time to time, developers will be asked to develop the UI and then re-develop it when the design changes. A good software team leader will be sure to offer the right explanations and support, as this could otherwise be a morale killer for developers who see it as inefficient re-work.

For the good of the product, it is better to continuously adapt a design than wait for it to be “locked” before starting development.

Continual Estimation

Once the Design and Development groups have developed trust and are communicating daily, they will often need to discuss time vs. design trade-offs.

Many software developers are amazingly creative and can build almost any design—even an astoundingly complex one—but it’s always a question of time and budget. Since most designers don’t have the deep technical knowledge to know upfront which design elements will be costly and which won’t, successful development teams get into the habit of continually estimating the time required to develop the proposed designs and provide that information back to the team as the designs evolve. Here is an example of this workflow:

1. The designer proposes an initial design concept.
2. A software developer estimates the rough time to achieve the design.
3. The developer highlights specific layouts or controls that will take time, especially if they risk threatening the timeline.
4. The developer gets back to design and product management with this information.
5. All three can discuss alternatives. If a particular design element is deemed critical to product success, the team can decide to leave it in scope despite its time and cost impact.

Does this constant negotiation-like process take a lot of time? It can, but it’s worth it. We have found two elements that can dramatically speed the continual estimation process:

1. Communication and trust. If these are present, design/development discussions quickly go from harsh negotiations to mutual support.

2. The developers' ability to estimate quickly, which comes with practice and knowledge of the system. Estimates don't have to be 100% accurate. In fact all that is really required is a ballpark estimate to flag any major complexities in the design.

Several years ago, Joel Spolsky recommended a process called Evidence-Based Scheduling—a way of using his company's software so that developers could progressively learn to estimate quickly and effectively. While we have not used his tool, certainly the theory is one we ascribe to.

Buffer Time For Creative Failure

This is probably the hardest one to achieve with a senior management team that is not bought in to an integrated Agile design process. You need to allow for some buffer for designers to be wrong. Good designers are trying to solve problems much like the scientific process—they come up with a hypothesis, in this case a design, and try to prove it through prototyping and usability testing.

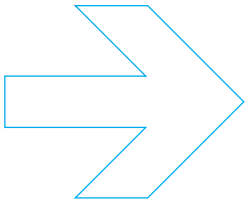
Unless your product strategy is to be first to market at the expense of potential quality trade-offs, the product owner should actively plan for some "failure and re-work" time. Designers will need this to iterate designs. The additional time may not even push out the release schedule much, as developers can use this time to perform activities that are otherwise notorious for holding up a product release at the end—refactoring code and fixing last-minute bugs.

KEY TAKEAWAYS

- Software team members need to effectively work in parallel, not sequentially.
- Developers working from the back-end will allow more time for designers to iterate the design.
- Developers should continually estimate the time impact of design changes and discuss it with the team.
- The product owner should buffer time for re-working the design and architecture, unless time-to-market is the key driver over quality.

FURTHER READING

- "Introduction to Agile Usability." (Ambler)
- "Four Ways to Review an Estimate with Limited Time and Expertise." (Macadamian)



Conclusion

Increasingly, software companies are recognizing that traditional software process not enough to create winning software. Product management alone is not enough to win in the market, and agile alone is not enough to get out the door fast. It is a matter of integrating the right mix of competencies - design, product management, and development - and get them working together towards a common goal of product success. Remember these five patterns:

- Involve design researchers, interaction designers and visual designers *throughout* the product creation process
- Ensure that the product management, design, and software development groups *work as a single team* and remain in *constant communication*.
- Clarify each function's roles and responsibilities within the product team
- Plan the product framework *upfront* in just the right amount of detail
- Implement a *flexible* process that allows your design and development teams to work in parallel and iteratively towards a successful launch

By implementing the recommendations listed throughout this paper, your organization can emulate the most successful product creation companies and boost its ability to deliver a product that is visually impressive, immediately intuitive and, ultimately, a standout innovation in the market.

Contact Us

For questions or comments about this white paper, or for more information on a consultation, please contact:

Didier Thizy
Director, Marketing
819-772-0300 x136
didier@macadamian.com

About Macadamian

Macadamian is a global UI design and software innovation studio that provides a complete range of highest quality usability, design, and software engineering services to industry leaders across North America. Our experience, and proven ability to work seamlessly with product management executives and software teams is why companies turn to Macadamian to develop product strategies, design compelling user experiences, and build quality software.

Whether you're a small start-up or a corporate giant, we can help you transform ideas into market-ready features or products that will stand out from your competition and delight customers.

Additional information can be found at www.macadamian.com.