

Product Development – getting it right the first time

- Vijayan Ganapathy



1) Introduction

Developing a product is more about foresight and intuition as much as not going down the wrong alley. According to Steve Jobs "... there's just a tremendous amount of craftsmanship between a great idea and a great product...Designing a product is keeping 5,000 things in your brain, these concepts, and fitting them all together in kind of continuing to push to fit them together in new and different ways to get what you want," he said. "And every day you discover something new that is a new problem or a new opportunity, to fit these things together a little differently." [2] A great product requires as much detail orientation as dotting the i's and crossing the t's. Most product development efforts fail while turning the concept into a product by falling into one or more of the fundamental traps which can be avoidable, if we can see the forest for the trees.

The Fundamental Principle of product development is to **"Uncover the surprises as soon as possible"** [1]. Surprises lead to change, which is always better at the beginning of the development cycle than it is anytime later. The cost of making a change rises exponentially as time passes.

Let us look into the future when we might have an Internet of Things (IoT, hereafter) robot trained using machine learning employed in performing surgeries. The software which drives the robot will have algorithms that determine the correct ways to drive the actuators (electromechanical devices) based on the procedure it is performing. However, there may be edge cases when algorithms and actuators might fail. For example, it might be possible that the algorithm determines the incorrect angle for a scalpel to move, or the actuator might malfunction and not move in the expected path. This calls for a safety monitoring system for the IoT robot. The earlier we realize the need, the less it will ultimately cost to implement. If this need is realized only during failures at field trials, it can make things exponentially worse. With this situation in context, let us explore the specific pitfalls(as depicted in Figure 1), that undermine product development, and what we can do to mitigate them.

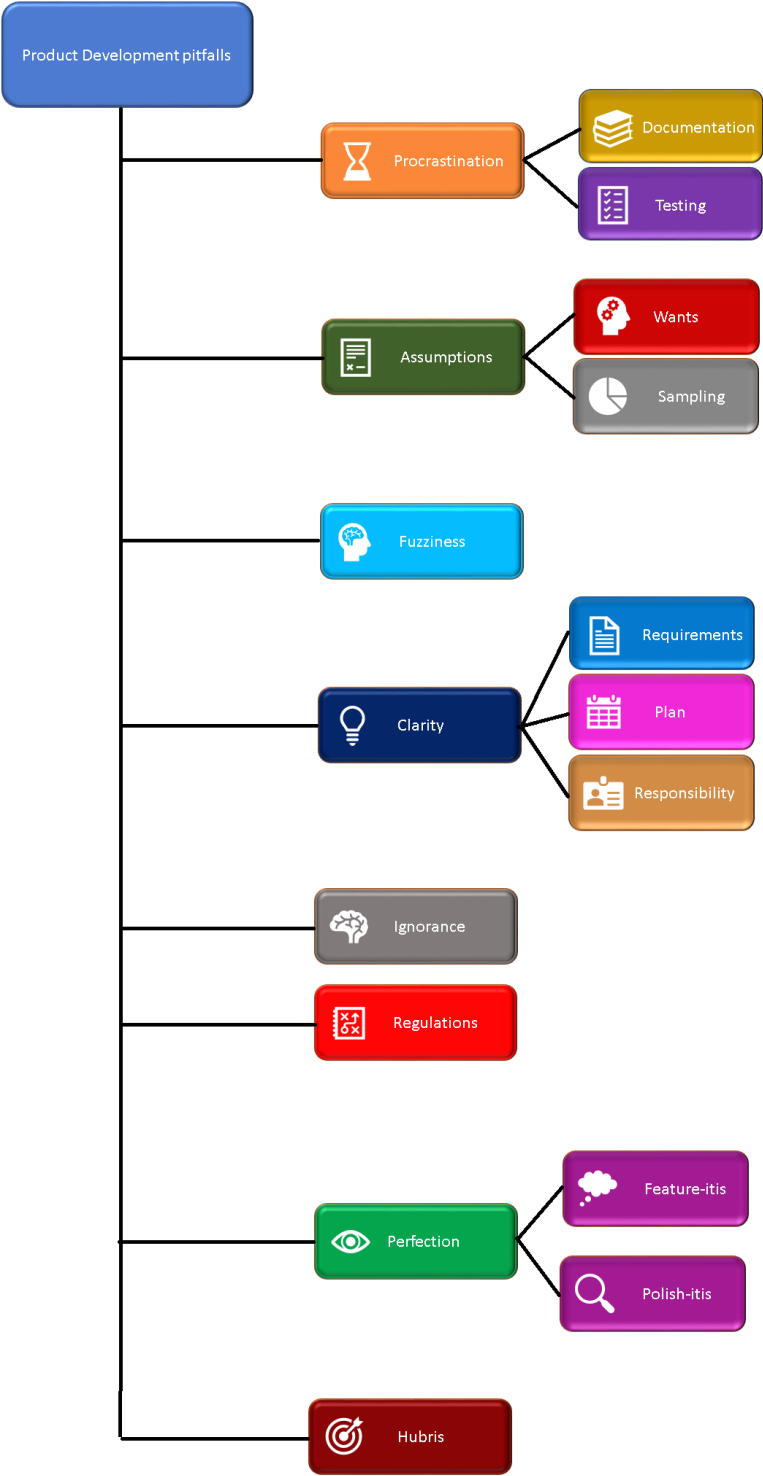


Figure 1: Product Development Pitfalls

This whitepaper stresses on establishing the groundwork for product development in a principled manner. This is done in order to enable the use of any applied software development process and methodology in the implementation stage.

2) Procrastination

2.1) Testing

Testing is an activity that can unearth a lot of surprises. Testing requires a context or a point of reference that describes the behavior of the product at every layer of its architecture. Usability and viability if tested early, can save several days of delays which could occur at a later stage.

2.2) Documentation

The need for spending quality time to document every aspect of the product in detail, prior to commencing active development, will leave little or no room for surprises.

3) Assumptions

3.1) Wants

The majority of humanity are not technophiles. As a Product developer, one prefers things that have more features, the ability to customize and more options to tinker with. Customers are mostly interested in products that are effective and attractive.

3.2) Sampling

It is entirely possible to deliver what customers have asked for without satisfying their needs. This calls for letting them try out the product as early as possible instead of waiting until the end and praying that our assumption (and not theirs) were correct.

4) Fuzziness

Fuzziness is the lack of specificity in planning a product and its roadmap. This is a major source of project failure.

The causes of fuzziness are:

- a. Not having detailed requirements
- b. Not having a detailed plan
- c. Not knowing who is responsible for accomplishing what, during development

The two big challenges introduced by fuzziness are:

- a. Mismatch of stakeholder expectations
- b. Inability to estimate resources and time

5) Clarity

5.1) Requirements

Product requirements are how we communicate our understanding of what the product will be. They ensure all stakeholders are in accord with regards to the salient (or fundamental) attributes and functionality of the product. When

creating product requirements we must capture everything that is important to us and our customers, failing which we may end up with a mismatch of expectations. Lack of comprehensive requirements guarantees feature creep and re-engineering of inter-dependent systems to accommodate the newly discovered needs.

5.2) Plan

Creating a plan is as difficult as filling a tax return and as enjoyable as getting a root canal treatment. Almost anything and everything can go wrong with even a well-defined plan. Some of the possible reasons include but are not limited to design inaccuracy, and incompleteness, incorrect assumptions, lack of resources, improper design, tight deadlines and so forth. Creating a detailed plan forces us to think through issues that are easy to miss when we are planning by taking a rough stab. It helps us remember important details we would otherwise forget about and understand dependencies.

5.3) Responsibility

Even the best laid out plan can fail if there is confusion around who gets what part done and how multiple parts come together. Adding an owner to each task greatly increases the odds of success and eliminates the confusion about who will ensure the completion of the task. The task owner is not necessarily the person who accomplishes the task, but rather is responsible for the timeline, budget, communications and making sure things get done. The task owner keeps an eye on the pre-requisites and ensures that everything falls in place for things to happen according to schedule and budget. Any issues that require an update of the plan must be communicated by the owner.

6) Ignorance

Ignorance covers those things we have no inkling of. It is a state of blissful unawareness until we smack into a problem late in the game, sometimes even after a product release, that can require product changes. Relying on people with experience in technical and non-technical areas that the product touches, either as employees or as advisers will help one identify where the potholes are located and the pitfalls to avoid from people who have walked the walk.

7) Regulations

Most products must address the customer (e.g., functionality, size, color) and business (e.g., profitability, design, consistency) needs, as part of the standard requirements. Apart from these, they must address the requirements imposed by external parties like governments, government departments, regions, insurers and regulatory bodies, which enforce adherence to regulations, standards and certifications pertaining to security, safety, measures and performance. This is an area that is overlooked during product development. It will require a significant investment of time, research, money and sought expertise. The end goal is to ensure we do not trivialize unknown aspects which can result in legal ramifications if ignored.

8) Perfection

Any product development effort aims for perfection since the end product will be judged by many people, perhaps even millions in the case of successful consumer products. It is an opportunity of glory and wealth for a successful

product as much as embarrassment and loss of money for a disappointing product. This tendency towards perfection manifests itself in two ways:

8.1) Feature-itis

The tendency to add impressive or desirable new features, encouraging scope creep and deviating from a locked down plan. New features can:

1. Inadvertently break other features
2. Be the cause of major architectural changes and patches
3. Cause an exponential increase in testing effort owing to the number of features the product supports and their inter-dependencies
4. Cause feature creep, unexpected delays and cost over-runs

Start small, be skeptical about adding new features during development and focus on making the few committed features work well. The aim should be to go to market faster and cheaper, where customers are delighted that the limited features we provide gets their job done.

8.2) Polish-itis

The longer we polish a stone, the smoother it will become. However, we need to take a call to declare when the stone is smooth enough and move on.

This is important because over-polishing with little value addition can delay the go-to-market of a product, result in depleted budgets, missed revenue projections, give space to the competition and lead to a loss of market share. The idea is to find the time for "good enough to ship" early enough without damaging the reputation or brand in the process.

9) Hubris

The blind belief that things will go according to plan and things can never go wrong is common to most product development projects. In reality, most projects fail too soon, fail often and the success or failure of the project is determined by how the failures are dealt with.

9.1) Plan to fail

Even the most perfectly crafted plan must accommodate for failures by padding a buffer time of 20% to 30% to account for the inevitable bad surprises.

If the plan is being devised by not very experienced folks or involves new technologies, it is easy for a project to end up with 100% or more over budget. One good practice that must accompany any project plan is a risk and mitigation plan which lists all the identified risks and concerns, their mitigation plan, some remarks pertaining to how much additional time they could consume if they apply. This could be used as a justification for any buffer time that is estimated and accounted for in the project plan.

10) Ego

Products are all about what customers want. It is not about what we personally want or what the team wants. Ultimately, it is a tradeoff where “what we enjoy” competes with “what is best for the product”.

10.1) Technology vs. Product

The lure for new technology causes people to focus on developing the technology as opposed to developing the product.

If a product can be developed quicker/cheaper/better by integrating existing technologies, then that is the way to go.

IBM, Apple and Microsoft have, to varying degrees largely adopted this approach, which is ubiquitous across most successful companies that develop new products.

11) Conclusion

Successful product development is all about getting things right the first time while planning well for failure.

References

[1] A. Cohen, Prototype to Product: A Practical Guide for Getting to Market, O'Reilly Media, 1 edition (August 11, 2015).

[2] Steve Jobs - The Lost Interview, 36:00 – 37:10,
<https://www.youtube.com/watch?v=HNMAXCfP6K4>

General References

[3] E. Ries, The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses, Portfolio Penguin (6 October 2011).

[4] F. Brooks, The Mythical Man-Month: Essays on Software Engineering, Addison-Wesley Professional; 2 edition (2 August 1995).

[5] J. Gall, General Systemantics: How Systems Work and Especially How They Fail, Quadrangle/The New York Times Book Co., third printing, August 1977, copyright 1975.

About Sogeti

Sogeti is a leading provider of technology and engineering services. Sogeti delivers solutions that enable digital transformation and offers cutting-edge expertise in Cloud, Cybersecurity, Digital Manufacturing, Digital Assurance & Testing, and emerging technologies. Sogeti combines agility and speed of implementation with strong technology supplier partnerships, world class methodologies and its global delivery model, Rightshore®. Sogeti brings together more than 25,000 professionals in 15 countries, based in over 100 locations in Europe, USA and India. Sogeti is a wholly-owned subsidiary of Capgemini SE, listed on the Paris Stock Exchange.

Learn more about us at
www.sogeti.com