

[blog.gardeviance.org](https://blog.gardeviance.org)

# Wardley's Doctrine

*swardley*

9-12 minutes

---

Whenever examining an industry then there is the landscape, the common economic patterns and the context specific forms of gameplay that need to be used to determine strategy. However, these are context specific and vary from industry to industry and player to player. This area of strategic play is the most fun part of leadership and where your true Machiavellian spirit can let rip. There are however some basic principles i.e. doctrine that are applicable to all industries regardless of context. These are universal.

I keep on getting asked by former employees and friends when I'm going to build a company again, so that they can come and work for me. First of all, that's an incredibly kind thing to say and I'm very touched by this. Second, I have no interest in leading again - I'm a reluctant leader only doing so when necessary. Lastly, I think some of those who worked for me have forgotten just how harsh I can be and are looking through rose tinted glasses. For example, I have commandments of operations regardless of what you are doing and from which I do not tolerate any deviation from.

However, in the hope that it might help others, I thought I'd scribble out my current list of commandments - yes, these refine and

change with time and experience. NB, these have nothing to do with strategy nor even leadership, they are simply the universal doctrine that I apply everywhere.

### Wardley's Doctrine.

In accordance with the wishes of the High Priest Thought Lord, the following commandments will be followed in all circumstances on pain of a most terrible punishment normally involving a good talking to (e.g. hair blower with choice expletives) and then a slice of cake or two in order to recover from the aforementioned terrible punishment.

#### **Thou shalt ...**

**Focus on user needs.** Any value we create is through meeting the needs of others. A mantra of *"not sucking as much as the competitors"* is not acceptable. We must be the best we can be.

**Use a common language.** Instead of using multiple different ways of explaining the same thing between different functions of the business, we use one - a map. If you can't map what you're doing then don't do it. Situational awareness is not optional.

**Be transparent.** We don't hide our maps, we share them and allow others to challenge and question our assumptions. The act of sharing is essential because it helps us to learn. Transparency also requires us to remove all the noise, the pointless gibberish that gets in the way of learning. Anyone not willing to learn, will forgo the slice of cake and be helped to find a new job with a competitor.

**Remove bias and duplication.** We not only share maps, we collate them in an effort to remove duplication (i.e. rebuilding the

same thing) and bias (i.e. custom building that which is a commodity). This is not optional and no, your accounting system is not unique because of the colour of your invoices nor do you have a unique way of purchasing advertising space. We all have a duty to remove duplication and bias in the organisation.

**Challenge assumptions.** It's a duty for everyone in the company to challenge assumptions. Where possible we use data collated from maps as the purpose of the maps is to expose the thinking. I don't care if it was my pet project, you will openly and honestly tell me why you think I'm wrong. Challenge requires transparency and trust. Any form of retribution or bias against someone for challenging is a deadly sin that will not be forgiven and you will be carted off to work for a competitor. For reference, as the CEO of Fotango, I made my CFO the XO back in 2004. One of his duties was to challenge me and agree / disagree with my choices.

**Think fast, inexpensive, simple and tiny.** Ok, Dan Ward might say FIRE (and do go buy his book) but I'll stick with his original terms. You will move quickly with a bias towards action, you will use inexpensive components and be frugal wherever possible, you will simplify the problem as much as possible and you will build in small components ... otherwise you WILL not build.

**Use appropriate methods.** NB, anyone suggesting we should be all agile, all lean or all six sigma will get a right good talking to without the cake. Same with others suggesting one size fits all purchasing methods.

**Use standard components where appropriate.** NB anyone suggesting we should build our own cloud service where a defacto service exists runs the risk of getting hauled up in front of company

as the person trying to spend the entire cake budget on a vanity project. Unless you can clearly show you can out industrialise and make more of a commodity whatever exists, then use it.

**Optimise flow.** Whether it is with what we build or how, we are required to remove bottlenecks and improve throughput where possible. However, care must be taken not to make the ineffective more efficient rather than making the ineffective more effective - hence the importance of situational awareness. Anything which gets in the way of meeting user needs, being fast, transparency on relevant information will be treated ruthlessly and for the guillotine e.g. corporate expense processes. **Hint**, it's usually way more effective to just to give everyone company credit cards, say "*spend in the interest of the company*" and get accounts to sort through the credit card bills rather than having staff fill out expense forms.

**Use small teams.** Everything must be broken down into small teams. As a guide, when exploring the uncharted space a team of 3-5 seems appropriate. When building a product / service capability then two pizza (i.e. 12) seems more apt. When providing an industrialised component then a larger team can be argued for. Under no circumstances will that team size approach anything close to the Dunbar number. Anything larger than 40 should be considered as highly dubious and an immediate candidate for dividing into smaller teams based upon user needs.

**Think aptitude and attitude.** Each team may contain discrete skills (e.g. networks, marketing, engineering, finance) known as aptitudes. But each aptitude has an attitude i.e. the culture, methods and techniques for agile development of an entirely novel act are not the same as those for building a highly industrialised component. When determining composition of team, it is a duty to

consider not only the aptitude but the attitude. The combination of both is what we call capability.

**Provide purpose, autonomy and mastery.** Each team shall be autonomous within the confines of what it is supposed to do (described by a fitness function). Each team will therefore own what it does. Each team shall be able to see how they fit into the whole (hence maps) and develop mastery in both aptitude and attitude.

**Design for constant evolution.** Whilst a team might become a semi permanent structure, the work it undertakes will evolve. It is therefore a duty to ensure that work evolves through the organisation e.g. a pioneering team discovers an uncharted space, a team of settlers take the work and productise it hence forcing the pioneers to move on. A team of town planners then industrialise the product when appropriate forcing the settlers to move on.

**Manage inertia.** We all have it. It's caused by past success. You will realise that you have it. There are about sixteen different forms and you will learn how to recognise this. When you find yourself saying *"but this is how we do it"* or *"but this has always worked in the past"* or *"don't fiddle with the machine as it ain't broke"* then you will question why you are saying this. If someone is challenging what is being done then you will LISTEN and you will ask why you are responding in such a way.

**Learn by playing the game.** Common economic patterns and context specific forms of gameplay are discovered by playing the game. Hence strategic choices must be made by those who play the game and strategy developed internally and not externally. We must also share what we've learned (hence again, maps and the

purpose of collating them). Hiring strategy consultants to write documents telling us what to do will get a another good yelling at and absolutely no cake whatsoever. Certainly use outsiders to learn context specific forms of gameplay but we're the ones playing the game.

**Understand that strategy is iterative not linear.** This is for anyone thinking of writing long winded strategy documents, target operating models and step by step plans of how the future will be. Immediately book an appointment with HR. You are a valuable asset for the company particularly if we can deploy you within a competitor. HR will help you find a strategy position in a competitor and you will be given glowing references especially about how sad we are to lose you. We might even try to put up a "*fight*" to encourage the competitor to think you're an attractive acquisition. We will even pay you to join them. Instead of long winded plans, we will have direction towards a position and adapt as needed. We will "*cross the river by feeling the stones*".

### **A final note**

The above is my list of universally applicable doctrine. Note, there are many context specific patterns that I use e.g. open source, open data that are not universal but suitable in specific contexts. They are hence not included as they are part of strategic play. For example, I'm not a fan of Open by default as a universal doctrine but rather Open by thinking i.e. the use of open in specific contexts.

These are also not "leadership". This is doctrine which is universal and hence operational. Leadership is more the act of setting the

direction (i.e. where we are going) and using context specific play to change the environment in our favour.