

# 1

## Introduction

*'Software entities are more complex for their size than perhaps any other human construct because no two parts are alike'*

*No Silver Bullet, Brooks, 1986*

Building software is hard: successfully bringing a new software product from conception to market is harder. Building a successful software company that develops and markets multiple software products is harder still.

The days of 'build it and they will come' are over. Simply creating a great piece of software and waiting for the customers to knock on your door no longer works – if it ever did.

Nobody should ever think that building software is easy: some small application may indeed be easy to build, but meaningful software applications and platforms are extremely complicated things to create. However, building software is at least a reasonably well-defined task which isn't true of many of the other things that need to be done. Indeed it isn't even clear what needs doing.

Creating a successful software product, and building a successful software company, involve a myriad of other activities that are far less easy to define and put boundaries around. The first of these is just deciding what it is that should be built. But perhaps the most difficult task of all is bringing all these activities – well-defined, poorly defined and never defined alike – into alignment so that they move towards a common goal. This is akin to an exercise in formation flying in which the individual pilots have their own opinion on where they should be going. The bigger the company, the more complex the environment, the more difficult the task.

## 2 Chapter 1 Introduction

Focusing on the customer helps: after all, we all agree the customer comes first (don't we?) But while it might be clear what an individual customer says they want, what they are willing and able to pay for might be something different. When individual customers are aggregated into a market – be it millions, thousands or just dozens – it can be extremely difficult truly to know what the customer wants.

Customers are not homogenous: they want different things, but if you intend to build a product to sell to many of them, you need to determine common needs. Great leaders help, but great leaders are not a panacea. Leadership can give direction and common goals, but if a company can only survive with a truly remarkable leader, then the leader has failed. Leaders lead, but they also need to put in place mechanisms to prevent single points of failure.

Given all these difficulties, and more, is it a any surprise that many choose to focus on something right in front of them? Something here and now? We might call it 'goal displacement'. Building a company is an engineering task to parallel the construction of the software itself.

The patterns in this book describe many of the recurring problems software companies face and details the solutions found by existing companies. The patterns follow the well-known pattern format: a problem statement, a set of forces that make the problem difficult, a solution, with details of how to build it, the consequences of the solution, and examples and related work.

## The Whole

*'No plan survives contact with the enemy.'*

*General Helmuth von Moltke*

Each pattern in this book contains one part of a solution. It may be used in isolation – a component if you like, dropped in to fix an issue. But each pattern also links to other patterns, and together they build towards a complete thing, a whole.

The whole is more than the sum of its parts. The whole is the thing you see when everyone is flying in formation. The whole happens when all the different parts of the organisation – engineering, sales, marketing – are coordinated, aligned, heading in the same direction. In business, making this happen is *strategy*.

Strategy sets organisational goals, objectives and direction. It is embodied in structures and organisation. It is planned, but it is also dynamic. It is changing in response to events and learning. Strategy looks forward and looks backwards to make sense of events. Strategy may be equated with a plan, something that is

determined by a few – top management, consultants, planners – and executed by the many. But the devil is in the detail: things are just too complicated for a comprehensive plan.

While there are elements of forward-looking planning, strategy is equally the result of what happens; it emerges from events and decisions: what happens in the market, what engineers learn as they build the product, what competitors do. Strategy is backwards-looking; it makes sense of what has happened in order to direct the future.

Patterns are ideal for describing strategy, because they share these characteristics. Patterns are descriptions of what happens – like strategy, they emerge from looking backwards, adding to understanding and making sense. They are also forward-looking: they describe what to do.

Used alone, an individual pattern might be strategic, or it might just be something done to fix a problem – that is, tactical. When brought together, when used in a sequence to form a whole, they work strategically. Patterns direct and inform, but they are not a straight-jacket; you can, and should, vary a pattern as you see fit. This is how new patterns come into being.

Patterns describe the problems and the solution, and importantly, they name these things. Perhaps the most important thing a pattern adds to a discussion is a name. Patterns bring a shared vocabulary, which raises the level of the discussion.

Rather than talking about ‘that thing you do when you involve the customer in creating the product but don’t create the product only for that customer’ you can say CUSTOMER CO-CREATED PRODUCT. This raises the level of the discussion and the abstraction level, it moves the conversation along more quickly and gives everyone, from CEO to lowly software engineer, a common language.

## Patterns

*‘According to leading management thinkers, the manufacturing, service, and information sectors will be based on knowledge in the coming age, and business organisations will evolve into knowledge creators in many ways.*

*According to [Peter Drucker] we are entering ‘the knowledge society,’ in which ‘the basic resource’ is no longer capital, or natural resources, or labour, but ‘is and will be knowledge...’*

*(Nonaka & Takeuchi, 1995)*

Patterns are all around us; it is simply a question of whether we see them and whether anyone has documented them. Patterns are recurring events, situations,