

## **FFT 1 – How Software Companies Die?**



---

### **About the Food For Thought (FFT) post series**

---

Starting from this post I will try to establish a new series of posts which will be completely different than the “normal” posts that are published in this blog. The name of this series is called “Food For Thought”, since the posts will deal with some thought-provoking subjects, hoping to intrigue your mind and trying to motivate you think deeper about the presented subject. One major difference in this series is that the main body of each post will come from an internet source – usually an expert at his/her field. So, the idea behind this post series follows the sequence: an interesting article on the web -> my personal thoughts and opinions -> your thoughts and opinions about the article. At the end, I hope that all of us will gain something...

---

### **Thought-provoker**

---

The first FFT post comes from [Orson Scott Card](#), an American author/writer/novelist, who have written several books, but is primarily known for his science fiction novels Ender's Game (1985) and its sequel Speaker for the Dead (1986), which both won Hugo and Nebula Awards. The article below was first published on Windows Sources (page 208), on March of 1995. 20 years have almost passed since its publication; however, I think that it is still valid and up-to date.

---

## Orson Scott Card – How Software Companies Die?

---

*The environment that nurtures creative programmers kills management and marketing types – and vice versa. Programming is the Great Game. It consumes you, body and soul. When you're caught up in it, nothing else matters. When you emerge into daylight, you might well discover that you're a hundred pounds overweight, your underwear is older than the average first grader, and judging from the number of pizza boxes lying around, it must be spring already. But you don't care, because your program runs, and the code is fast and clever and tight. You won. You're aware that some people think you're a nerd. So what? They're not players. They've never jousted with Windows or gone hand to hand with DOS. To them C++ is a decent grade, almost a B - not a language. They barely exist. Like soldiers or artists, you don't care about the opinions of civilians. You're building something intricate and fine. They'll never understand it.*

### *Beekeeping*

*Here's the secret that every successful software company is based on: You can domesticate programmers the way beekeepers tame bees. You can't exactly communicate with them, but you can get them to swarm in one place and when they're not looking, you can carry off the honey. You keep these bees from stinging by paying them money. More money than they know what to do with. But that's less than you might think. You see, all these programmers keep hearing their fathers' voices in their heads saying "When are you going to join the real world?" All you have to pay them is enough money that they can answer (also in their heads) "Geez, Dad, I'm making more than you." On average, this is cheap. And you get them to stay in the hive by giving them other coders to swarm with. The only person whose praise matters is another programmer. Less-talented programmers will idolize them; evenly matched ones will challenge and goad one another; and if you want to get a good swarm, you make sure that you have at least one certified genius coder that they can all look up to, even if he glances at other people's code only long enough to sneer at it. He's a Player, thinks the junior programmer. He looked at my code. That is enough. If a software company provides such a hive, the coders will give up sleep, love, health, and clean laundry, while the company keeps the bulk of the money.*

### *Out of control*

*Here's the problem that ends up killing company after company. All successful software companies had, as their dominant personality, a leader who nurtured programmers. But no company can keep such a leader forever. Either he cashes out, or he brings in management types who end up driving him out, or he changes and becomes a management type himself. One way or another, marketers get control. But...control of what? Instead of finding assembly lines of productive workers, they quickly discover that their product is produced by utterly unpredictable, uncooperative, disobedient, and worst of all, unattractive people who resist all attempts at management. Put them on a time clock, dress them in suits, and they become sullen and start sabotaging the product. Worst of all, you can sense that they are making fun of you with every word they say.*

## *Smoked out*

*The shock is greater for the coder, though. He suddenly finds that alien creatures control his life. Meetings, Schedules, Reports. And now someone demands that he PLAN all his programming and then stick to the plan, never improving, never tweaking, and never, never touching some other team's code. The lousy young programmer who once worshiped him is now his tyrannical boss, a position he got because he played golf with some sphincter in a suit. The hive has been ruined. The best coders leave. And the marketers, comfortable now because they're surrounded by power neckties and they have things under control, are baffled that each new iteration of their software loses market share as the code bloats and the bugs proliferate. Got to get some better packaging. Yeah, that's it.*

*Note: the above article is written by [Orson Scott Card](#) and is reposted due to its valuable content. If the author or anyone related to this article thinks that there is any copyright issue, please send me an email and I will remove it immediately.*

---

## Personal thoughts and opinions

---



The above article exists on the web for almost 2 decades. Despite the humorous and ironic style of Mr. Card – at least this is how I perceive his writing style – I think that the article deals with many interesting topics. First of all, from personal experience, I confirm that software development sometimes “consumes” every second of programmer’s life. In those cases the code becomes an issue of significant importance and everything else (food, sleep etc.) is considered as trivial. You might go to sleep and still thinking how to improve your code in order to become faster, error-free and more efficient. Quite often you reach to a dead end and all your ideas are drying out; then you find yourself to a mental state that you want to bang your head against the wall, or smash your computer with a hammer! Does this sound crazy”? Well, believe it or not, it’s 100% truth!

## Beekeeping



One step further, this “craziness” makes the programmer susceptible to the “beekeeping” method which has two main aspects: the programmer’s necessity to gain enough money from his/her work and the increase of motivation/inspiration by competing to other co-programmers. Simple yet effective method. A “healthy competition” among the programmers would benefit the company, since they will work harder in order to “overcome” their colleagues, or to gain the attention of the “master Player”. Money is equally important; programmers want their achievements to be recognized by the company, and this can be accomplished by increasing their bank accounts. If the company satisfies these two aspects, programmers will be willing to work really hard; a dedicated programmer will return multiple benefits in the company in terms of money. You might think that this is a win – win situation, but in reality the company’s profit is several times higher than the increased salary of the programmer.

## Out of control



Humans are not bees, so inevitably this “beekeeping” method – sooner or later – will fail. The turning point seems to be related – on way or another – with the “master Player”. I am sure that now you might wonder how a large software company might be affected by the departure of only

one person. I have two large companies in my mind, Microsoft and Apple; both software giants were negatively affected by the “retirement” of Bill Gates and the death of Steve Jobs respectively. Their successors didn’t manage to fill their shoes.

### Smoked out



The commercial success of a (software) product is strongly dependent on marketing. However, marketing should not be overestimated. Without a robust, error-free and efficient code, as well as an attractive and user-friendly interface, a software product is probably doomed to fail. Even if the company has the best marketers in charge, the failure sometimes seems unavoidable. The dipole programmer – marketer is the blessing and the curse of a software company. If one of the two feels that the other overshadows him/her, their cooperation becomes difficult and in extreme cases one of the two is possible to leave from the company (usually the programmer).

---

### Epilogue

---

All the above text was related with a software company. However, don’t you think that there are a lot of similarities with any “typical” company? If you think a little bit the question you will find many similarities. People motivated by money and the possibility of expanding their knowledge/skills (remember the role of master Player) can be found almost in every company. Problematic cooperation between two different specialists is also a common phenomenon in many companies. What do you think about?

By [Christos Samaras](#)  
[My Engineering World](#)