## How to get out of crunch mode

Written by Tim Black
Thursday, 04 August 2016 22:11 - Last Updated Friday, 28 June 2019 17:06

I did some serious reading and thinking about how I might be able to help a startup stay out of crunch mode, and came up with the following recommendations.

Getting out of crunch mode is a challenging transformation, because it requires changing a company's **culture**, **business plan**, and **software development process**, and these things are hard to change.

It is a matter of changing the company's **culture**. This has to start with the founders. You have to value a work-life balance.[                                                                    1 ] One pointed way I saw this put was: you have to value your employees' bodies more than your profit. Benefits
: 40 hour weeks improve employees' happiness and productivity[
2
], and enable the company to hire from a broader pool, including hiring not ONLY junior employees who are smart, productive, ambitious, and perhaps naive about the "churn and burn" process at some startups, but ALSO more senior employees who are domain experts (think Ph.D.s), stable, reliable professionals, and seek a better work-life balance out of wisdom and not merely necessity. You plan to expand your development team, and a broader candidate pool would make it easier to find candidates willing to relocate to [your non-Silicon Valley location].

It is a matter of changing your **business plan**. The release deadline pressure comes from selling what you don't have. You can transition to selling the features you have, and selling development services. This can be done gradually, and to a greater or lesser extent. Benefits
: This reduces deadline pressure. Selling existing features to new customers also increases and diversifies your customer base, and you can leverage this portion of profit to fund new features. You can still contract to create new features for clients, but rather than guaranteeing
*specific features*
as the primary deliverables, guarantee your
*development services*
for a period of time as the primary deliverable, delivering new features frequently in an agile manner. This is a paradigm shift which could seem impossible to implement because some clients require feature deadlines, but for some clients faster iteration and constant feedback loops actually give them more needed features faster than they would get under feature deadlines.

It is a matter of changing the **software development process** and **feature delivery architecture** you use to implement your business plan. You can use branch-by-abstraction, put unfinished features (or alternative/optional/sales-tiered finished features) behind feature flags, implement continuous integration and continuous delivery[

3

], and so move to more frequent releases and a rolling release cycle. Continuous delivery and frequent releases are harder with software as complex as operating systems[

4

], but still possible.
Benefits
: Faster development velocity, shorter time-to-market for new features, so quicker response to competition, and less pressure on release day, because every day is a release day in that stable release candidates are built every day[3]. Some companies say they could not compete in today's market without using continuous delivery.

Footnotes:

1. This is the shortest and most blunt article I read which could motivate a founder to change his priorities:   http://chadfowler.com/2014/01/22/the-crunch-mode-antipattern.html.

2. Surprisingly, Henry Ford found reducing from 6 to 5 10-hour days actually made his workers produce more per week, and a further reduction to 5 8-hour days brought a *further* increase in production. This was with assembly line workers; arguably knowledge workers' productivity increases similarly when they are not tired. This and much other related research is mentioned in the paper responding to crunch mode at Electronic Arts at
http://cs.stanford.edu/people/eroberts/cs181/projects/2004-05/crunchmode/index.html.  This paper is the best collection of material I found which makes the point that it's best to avoid crunch mode.
http://www.igda.org/?page=crunchsixlessons
is a good presentation, too.

3. How one company transitioned to continuous delivery:  https://www.infoq.com/articles/cd-ben efits-challenges                                                  . "The engineers commented that they don't feel the same level of stress on the release day that they did previously. That day becomes just another normal day."

# How to get out of crunch mode

Written by Tim Black

4. Startup advisor Jocelyn Goldfein wrote that operating systems require a more regular and probably longer release cycle at    [http://firstround.com/review/the-right-way-to-ship-software/](http://firstround.com/review/the-right-way-to-ship-software/).