36: Unity

Don't split into silos

Too many companies separate design, development, copywriting, support, and marketing into different silos. While specialization has its advantages, it also creates a situation where staffers see just their own little world instead of the entire context of the web app.

As much as possible, integrate your team so there's a healthy back-and-forth dialogue throughout the process. Set up a system of checks and balances. Don't let things get lost in translation. Have copywriters work with designers. Make sure support queries are seen by developers.

Even better, hire people with multiple talents who can wear different hats during development. The end result will be a more harmonious product.

37: Alone Time

People need uninterrupted time to get things done

Basecamp is spread out over four cities and eight time zones. From Provo, Utah to Copenhagen, Denmark, the five of us are eight hours apart. One positive side effect of this eight hour difference is alone time.

There are only about 4-5 hours during the day that we're all up and working together. At other times, the us team is sleeping while David, who's in Denmark, is working. The rest of the time, we're working while David is sleeping. This gives us about half of the day together and the other half alone.

Guess which part of the day we get the most work done? The alone part. It's not that surprising really. Many people prefer to work either early in the morning or late at night — times when they're not being bothered.

When you have a long stretch when you aren't bothered, you can get in the zone. The zone is when you are most productive. It's when you don't have to mindshift between various tasks. It's when you aren't interrupted to answer a question or look up something or send an email or answer an IM. The alone zone is where real progress is made.

Getting in the zone takes time. And that's why interruption is your enemy. It's like rem sleep — you don't just go to rem sleep, you go to sleep first and you make your way to rem. Any interruptions force you to start over. rem is where the real sleep magic happens. The alone time zone is where the real development magic happens.

Set up a rule at work: Make half the day alone time. From 10am-2pm, no one can talk to one another (except during lunch). Or make the first or the last half of the day the alone time period. Just make sure this period is contiguous in order to avoid productivity-killing interruptions.

A successful alone time period means letting go of communication addiction. During alone time, give up instant messaging, phone calls, and meetings. Avoid any email thread that's going to require an immediate response. Just shut up and get to work.

Get Into the Groove

> *We all know that knowledge workers work best by getting into "flow", also known as being "in the zone", where they are fully concentrated on their work and fully tuned out of their environment. They lose track of time and produce great stuff through absolute concentration... trouble is that it's so easy to get knocked out of the zone. Noise, phone calls, going out for lunch,*

> *having to drive 5 minutes to Starbucks for coffee, and interruptions by coworkers — especially interruptions by coworkers — all knock you out of the zone. If you take a 1 minute interruption by a coworker asking you a question, and this knocks out your concentration enough that it takes you half an hour to get productive again, your overall productivity is in serious trouble.*

—Joel Spolsky, software developer, Fog Creek Software (from Where do These People Get Their (Unoriginal) Ideas?)

38: Meetings Are Toxic

Don't have meetings

Do you really need a meeting? Meetings usually arise when a concept isn't clear enough. Instead of resorting to a meeting, try to simplify the concept so you can discuss it quickly via email or im or Campfire. The goal is to avoid meetings. Every minute you avoid spending in a meeting is a minute you can get real work done instead.

There's nothing more toxic to productivity than a meeting. Here's a few reasons why:

* They break your work day into small, incoherent pieces that disrupt your natural workflow
* They're usually about words and abstract concepts, not real things (like a piece of code or some interface design)
* They usually convey an abysmally small amount of information per minute
* They often contain at least one moron that inevitably gets his turn to waste everyone's time with nonsense
* They drift off-subject easier than a Chicago cab in heavy snow
* They frequently have agendas so vague nobody is really sure what they are about
* They require thorough preparation that people rarely do anyway

For those times when you *absolutely must* have a meeting (this should be a rare event), stick to these simple rules:

* Set a 30 minute timer. When it rings, meeting's over. Period.
* Invite as few people as possible.
* Never have a meeting without a clear agenda.

Have fewer meetings

> There are too many meetings. Push back on meetings that do not make sense or are unproductive. Only book a meeting when you have an important business issue to discuss and you want or need input, approval, or agreement. Even then, resist the urge to invite everyone and their brother — don't waste people's time unnecessarily.

—Lisa Haneberg, author (from Don't Let Meetings Rule!)

Break it Down

> As projects grow, adding people has a diminishing return. One of the most interesting reasons is the increased number of communications channels. Two people can only talk to each other;

there's only a single comm path. Three workers have three communications paths; 4 have 6. In fact, the growth of links is exponential...Pretty soon memos and meetings eat up the entire work day.

The solution is clear: break teams into smaller, autonomous and independent units to reduce these communications links.

Similarly, cut programs into smaller units. Since a large part of the problem stems from dependencies (global variables, data passed between functions, shared hardware, etc.), find a way to partition the program to eliminate — or minimize — the dependencies between units.

—The Ganssle Group (from Keep It Small)

39: Seek and Celebrate Small Victories

Release something today

The most important thing in software development is motivation. Motivation is local — if you aren't motivated by what you are working on right now, then chances are it won't be as good as it should be. In fact, it's probably going to suck.

Long, drawn out release cycles are motivation killers. They insert too much time between celebrations. On the other hand, quick wins that you can celebrate are great motivators. If you let lengthy release cycles quash quick wins, you kill the motivation. And that can kill your product.

So, if you're in the middle of a months-long release cycle, dedicate a day a week (or every two weeks) for some small victories. Ask yourself "What can we do and release in 4 hours?" And then do it. It could be...

- A new simple feature
- A small enhancement to an existing feature
- Rewriting some help text to reduce the support burden
- Removing some form fields that you really don't need

When you find those 4-hour quick wins, you'll find celebration. That builds morale, increases motivation, and reaffirms that the team is headed in the right direction.