



DEVELOPMENT ANTI-PATTERNS

A**HOLE DRIVEN DEVELOPMENT (ADD)

Any team where the biggest jacka** makes all the big decisions is asshole driven development. All wisdom, logic or process goes out the door when Mr. A is in the room, doing whatever idiotic, selfish thing he thinks is best. There may be rules and processes, but Mr. A breaks them and people follow anyway.

COGNITIVE DISSONANCE DEVELOPMENT (CDD)

In any organization where there are two or more divergent beliefs on how software should be made. The tension between those beliefs, as it's fought out in various meetings and individual decisions by players on both sides, defines the project more than any individual belief itself.

PING PONG DEVELOPMENT (PPD)

Two stakeholders with mutually exclusive requirements and visions, both of which constantly insist on their choice of features. Also related are [Eternal PPD \(EPPD\)](#), where the ping pong game is never ending due to the fact the opposing stakeholders will never agree on any feature that the software should have, but will not admit defeat and cancel the project either.

DECAPITATED CHICKEN PROCESS (DCP)

A time honored micromanagement technique where each day managers identify a drastic emergency and require developers drop what they are doing (and whatever process they are using) and immediately attend to the latest conflagration. Since this does the double duty of creating new bugs and making other tasks fall behind, fires become easier and easier for managers to spot and freak out about. Also referred to as [Everything is High Priority \(EHP\)](#), [What's Today's Emergency? \(WTE\)](#), [Put Out the Fire \(POF\)](#) and [Seagull Management \(SM\)](#) "The seagull manager flies in, makes a lot of noise, craps on everything then flies off again leaving a big mess behind."

COVER YOUR ASS ENGINEERING (CYA)

The driving force behind most individual efforts is to make sure that when the shit hits the fan, they are not to blame. All complex, complicated, expensive, or otherwise troublesome decisions/features/issues are pushed off to someone else. Also referred to as [Not My Problem \(NMP\)](#), [Hot Potato Development \(HPD\)](#), and [Musical Chairs Development \(MCD\)](#).

DEVELOPMENT BY DENIAL (DBD)

Everybody pretends there is a method for what's being done, and that things are going OK, when in reality, things are a mess and the process is on the floor. The worse things get, the more people depend on their denial of what's really happening, or their isolation in their own small part of the project, to survive.

GET ME PROMOTED METHODOLOGY (GMP)

Developers write code and create functionality to increase their visibility, satisfy their boss's whims, and accelerate their path to a raise and/or the corner office -- no matter how far outside of the stated goals their efforts go. This includes allowing disasters to happen so people can be heroes, writing hacks that look great in the short term but crumble after the individual has moved on, and focusing more on the surface of work than its value.

WORRY ABOUT IT LATER (WAIL)

We all have done this at some time!

TEMPORARILY UNDER REPAIR DEVELOPMENT (TURD)

What the development engineers do when the site is down, and the sorry page is up.

ALL-BUT-VACUUM MANAGEMENT (ABVM)

Team members operate with no lead/ managerial feedback or processes - just rumor handed down - until project stagnation and butting-of-heads gives management enough one-sided whining to do something pointlessly drastic. This avoids having to deal with team members as humans, actually evaluate anyone's performance, and define anybody's role.

SHOVEL-DRIVEN DEVELOPMENT (SDD)

Get it out the door as quickly as possible, cut-n-paste from anything that you find that works. Maintainability and standards be damned. Closely related to [Duct-tape Driven Design \(DDD\)](#).

IDIOT MBA DRIVEN DEVELOPMENT (IMBADD)

Development driven by someone who thinks they know everything, just because they just graduated with an MBA.

TOO MANY CHIEFS, NOT ENOUGH INDIANS (TMC)

When you have multiple bosses, each trying to take the project in different directions.

BUDGET DRIVEN DEVELOPMENT (BDD)

The time that a project will take is dictated by how much the client will pay, instead of how long it will take to develop the application, generally leading to massively over-budget projects.

OVER-ENGINEERED ÜBER-SPECIFIED DEVELOPMENT (OÜD)

The majority of the development time is spent defining architecture, service interfaces, requirements, and other items. Copious amounts of inaccurate, verbose, self-contradicting and unnecessary documentation are prepared and maintained as if they somehow embody everything that needs to be done in development. A developer must implement the documentation down to typos and pixels in the user interface. Code that deviates from the specification is incorrect, even if it is how the product is intended to behave. No changes to code are allowed unless the document is revised, reviewed, submitted, approved, published and re-distributed. Also referred to as [Document Driven Development \(DDD\)](#).

BLAME THE LAST GUY (BTLG)

Whatever the problem, whether it was your's or not, blame it on the employees who used to work on the project, preferably one whom no longer works for the organization.

IT'S WHAT THEY WANTED DEVELOPMENT (IWTWD)

Absolving oneself of all accountability by inventing a group of people known as "they" and blaming them for one's own inability to design and develop a usable system.

VISIBILITY DRIVEN DEVELOPMENT (VDD)

We're selling the company, so the more times the not-really-ever-going-to-be-product squeaks, whistles, spins, churns, flashes, or wobbles, the better. If it "just works," it makes us look like we kept all that money we said we put into R&D.

NOT KNOWING WHAT YOU WANT UNTIL YOU SEE IT (NKWYWUYSI)

Client or management know they want something, but they're not quite sure what - so there is no design specification. What they want then gets poorly communicated down to the development team, who then produce their version of what they "think" management wants, and then the project enters the [Never Ending Story Development \(NESD\)](#) phase.

THE PROCESS DEVELOPMENT (TPD)

When the process of the project becomes more important than the project itself. No work can be done without a TPS report. See [Sigma Six](#) for further details.

IT'S TUESDAY DEVELOPMENT (ITD)

Feature freeze and requirement specifications? Hell no. Today we are changing the requirements to the degree we are now effectively working on a different project. Why? Because it's Tuesday!

MUST USE SPECIFIC TECHNOLOGY (MUST)

Someone (sometimes a reasonably smart and technically savvy engineer) dictates that the solution **must** use a specific technology "because it's the future". The technology ends up shoe-horned into places where it clearly doesn't fit, or it's too immature to use successfully and eventually everyone looks back and says: "why the hell did we do that?" Also known as [The Square Peg in a Round Hole \(SPRH\)](#) methodology and [Nerd Driven Development \(NDD\)](#).

BUZZWORD DRIVEN DEVELOPMENT (BDD)

Whatever is best for the project does not count; what counts is whatever is "hot" right now. Requires every new and fancy technology to be used within the project, just for the sake of it, whether it actually makes sense or not.

FAITH-BASED DEVELOPMENT (FBD)

Based on the premise that a developer is so good (read: cocky) at what he/she does that he need not test or ensure that anything works with existing components and checks in the code as-is.

FUD DEVELOPMENT (FUDD)

Implementing the feature(s) the right way is huge, scary, time consuming and expensive in every way imaginable, therefore the other way (whatever it is) is the way it will be done.

NO CUSTOMER LEFT BEHIND DEVELOPMENT (NCLBD)

Every feature requested by the client, no matter how unrelated, trivial, esoteric, unusable or out-of-scope, must be included. Resulting in an over-budget, over-deadline, non-specified item the client will never pay for. Also referred to as [Client Wants It anyway \(CWI\)](#) and [Out of Scope Development \(OSD\)](#).

COMPLETELY REDUNDANT APPLICATION PROCESS (CRAP)

This is where you create the same application someone in your company, division, department, or cubicle has created because you had no idea someone else had already done it.

DEAD MAN WALKING DEVELOPMENT (DMWD)

Development teams working on a project with a visible executioner hovering just overhead. Also referred to as [You'll Be Gone, I'll Be Gone \(YBG IBG\)](#), and [Irrelevant Development \(And We Know It\)](#).

ONE BADASS DEVELOPMENT (OBD)

Near deadline time everyone goes to the one badass of the company for help. In the end, the badass finds that everything everyone else has done is crap and rebuilds the entire project by himself in a few days, without sleep. Hire several badasses, they'll often quit when learning that this development process is in place.