

You Are Not a Code Monkey

So Stop Acting Like One

by Gianfranco Berardi



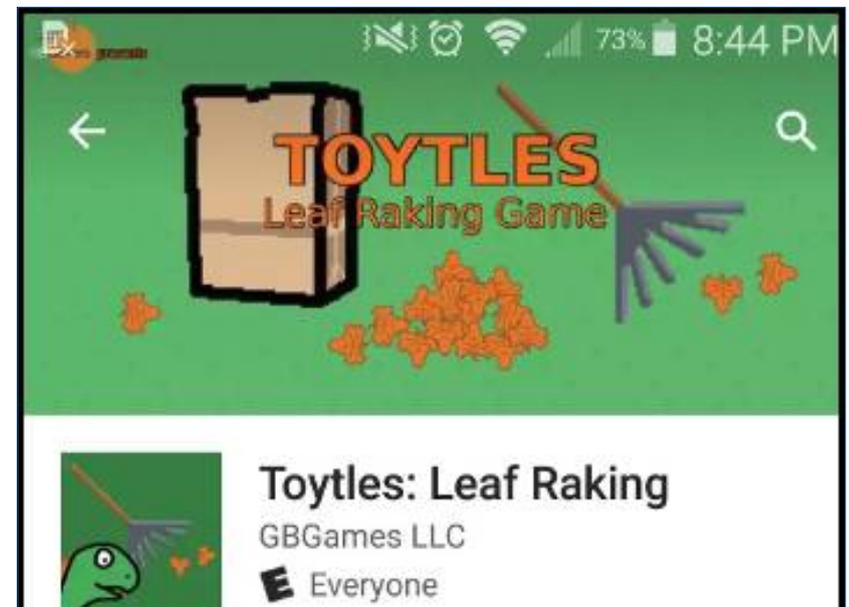
Why are you here?

- **Why?** - Because too many people and organizations are not living up to their potential
- **What?** - You with a more fulfilling attitude about the role of a software developer
- **How?** - Learn why the metaphor of code monkey is awful, no good, very bad *beats chest*

Who Am I?

Quick Bio:

- Gianfranco Berardi
- Full-time software developer since 2004
- Part-time independent game developer
 - Create educational entertainment that encourages exploration and a love of learning
- <https://www.GBGames.com/>
- Pronouns: he/him/his



Who are you calling a code monkey?!

Definitions:

Wikipedia: “A term for a computer programmer, sometimes considered an insulting or derogatory oversimplification of their skill set”

Urban Dictionary: “A programmer who isn't actually involved in any aspect of conceptual or design work, but simply writes code to specifications given.”

Who are you calling a code monkey?!

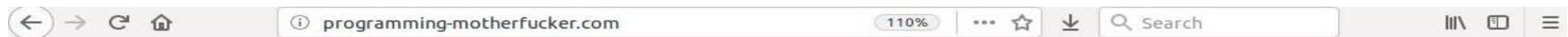
Definitions:

Wikipedia: “A term for a computer programmer, sometimes considered an insulting or derogatory oversimplification of their skill set”

Urban Dictionary: “A programmer who isn't actually involved in any aspect of conceptual or design work, but simply writes code to specifications given.”

Code Monkey != Software Developer

Yet there is Code Monkey pride?!



Programming, Motherfucker Do you speak it?

We are a community of motherfucker programmers who have been humiliated by software development methodologies for years.

We are tired of *XP, Scrum, Kanban, Waterfall, Software Craftsmanship* (aka *XP-Lite*) and anything else getting in the way of...Programming, Motherfucker.

We are tired of being told we're socially awkward idiots who need to be manipulated to work in a Forced Pair Programming chain gang without any time to be creative because none of the 10 managers on the project can do... Programming, Motherfucker.

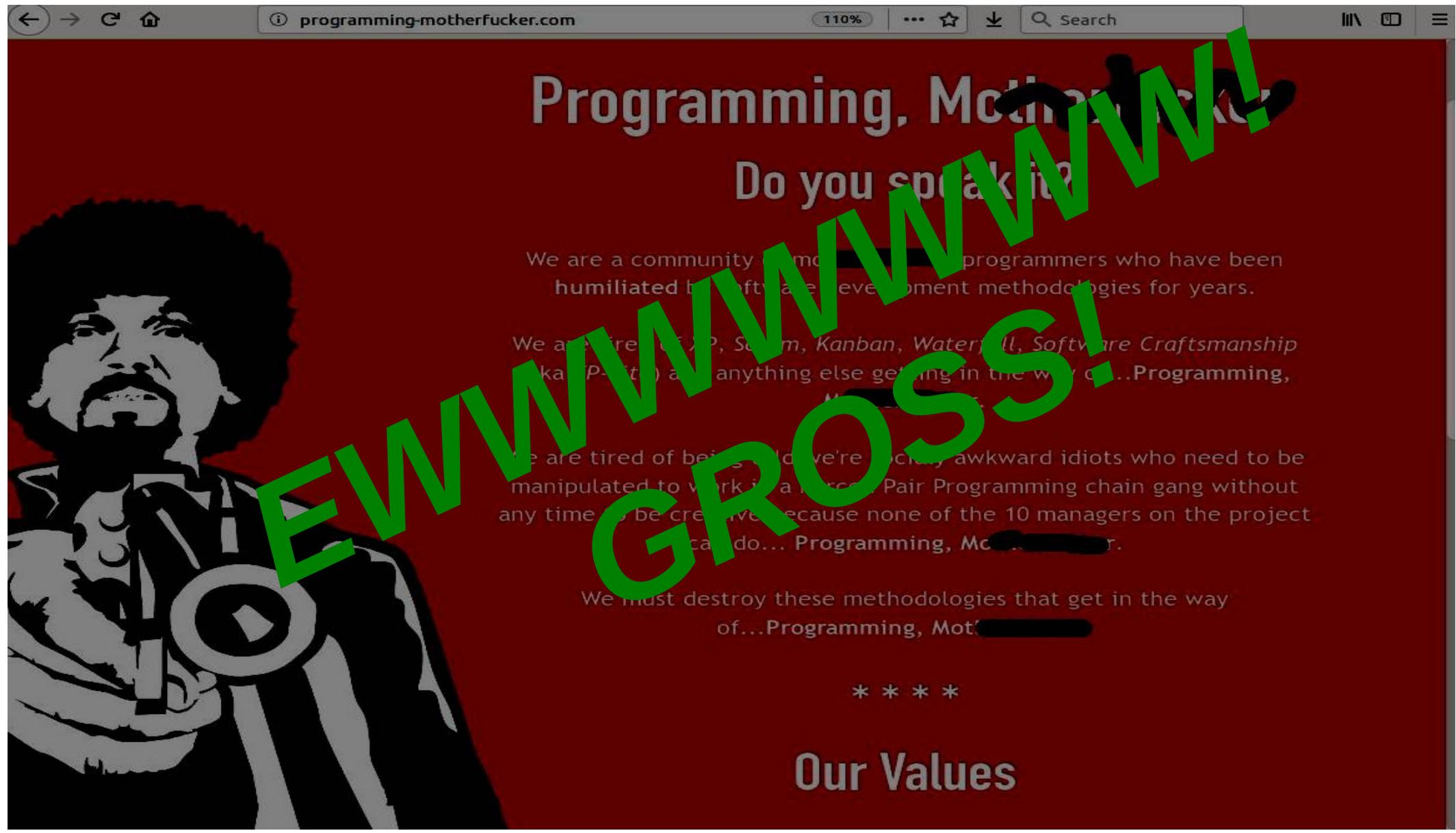
We must destroy these methodologies that get in the way of...Programming, Motherfucker.

* * * *

Our Values



Yet there is Code Monkey pride?!



programming-motherfucker.com 110% Search

Programming, Motherfucker

Do you speak it?

We are a community of **code monkey** programmers who have been humiliated by software development methodologies for years.

We are tired of **XP, Scrum, Kanban, Waterfall, Software Craftsmanship (aka P-It)** and anything else getting in the way of...Programming, Motherfucker.

We are tired of being told we're socially awkward idiots who need to be manipulated to work in a large Pair Programming chain gang without any time to be creative because none of the 10 managers on the project can do... Programming, Motherfucker.

We must destroy these methodologies that get in the way of...Programming, Motherfucker.

Our Values



Speak No Evil, Hear No Evil

- “I can't refactor/pair program. My manager won't let me.”
- “The business has no idea what it is doing.”
- “The developers would waste time on work that has no value if we let them.”
- “I just want to code. I don't want to deal with squishy human stuff.”
- “The deadlines are made up anyway.”



Speak No Evil, Hear No Evil

- “I can't refactor/pair program. My manager won't let me.”
- “The business has no idea what it is doing.”
- “The developers would waste time on work that has no value if we let them.”
- “I just want to code. I don't want to deal with squishy human stuff.”
- “The deadlines are made up anyway.”

This is dysfunctional, by the way.



The Story of the Pot Roast

“Why do you always cut off the ends of the roast?”

“I don't know. My mother **always did it that way**, and I learned it from her.”

We've always done it that way.

But what if we used our brains?

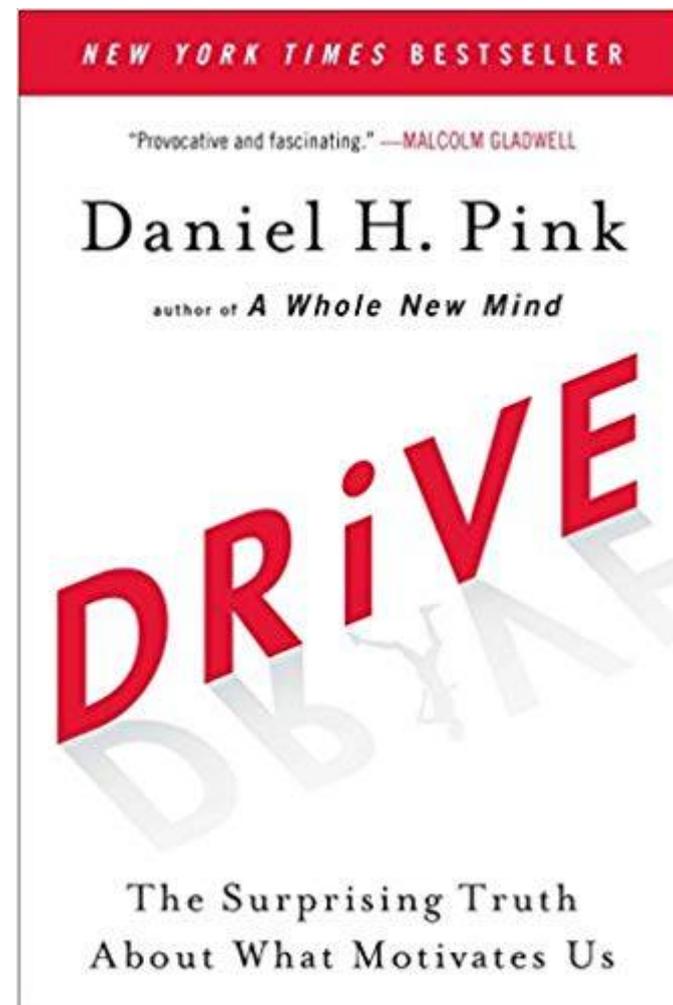
Could we find a better way?

Could we help our organizations find a better way?

Learned Helplessness

Drive: The Surprising Truth About What Motivates Us by Daniel H. Pink

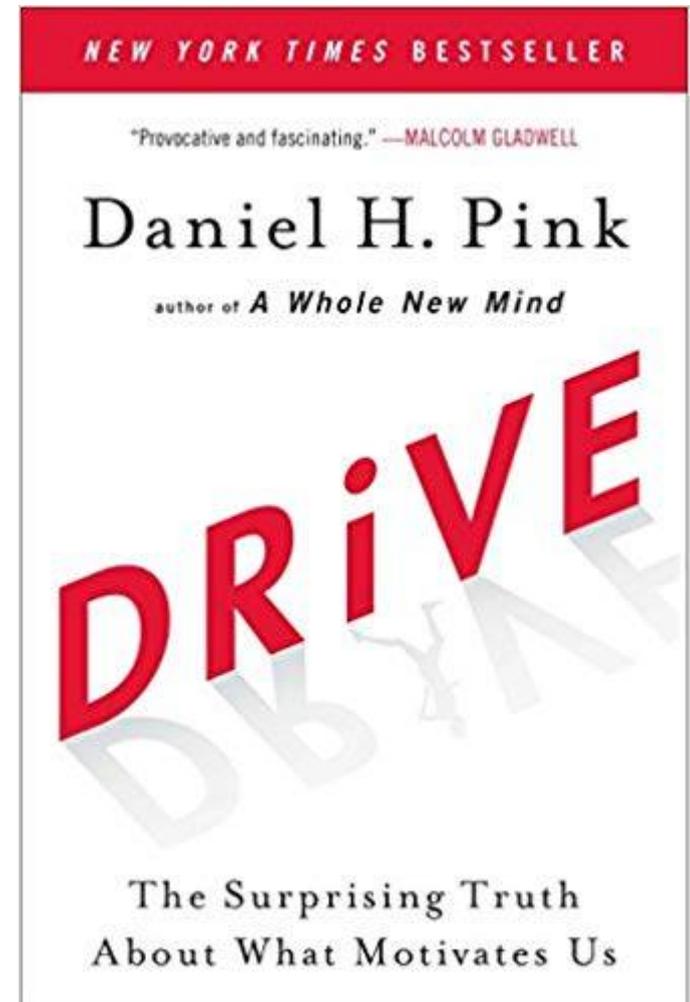
- Autonomy
- Mastery
- Purpose



Learned Helplessness

Drive: The Surprising Truth About What Motivates Us by Daniel H. Pink

- Autonomy
- Mastery
- Purpose



Learned Helplessness

1960s psychology research involving dogs

Can't escape painful stimuli?

- Feel loss of control
- Give up trying
- Even when situation changes!

About control, not competence.

Learned Helplessness

- What organizations do:
 - Micromanage
 - Punish
 - Scoldings
 - Worse?
 - Decide **without developer input**
 - Wonder why results stink
- What developers do:
 - Accept it
 - Self-constrain
 - Stop thinking critically
 - Leave, eventually
 - Or don't, sadly
 - Perform suboptimally

Learned Helplessness

Sound familiar?

- Told to be self-organizing but
 - assigned to a team
 - told what tech stack to use
- Told to be responsible and accountable but
 - given no decision-making ability
 - except to work longer hours
 - told to not work on maintenance (refactoring, cleaning, documenting, etc)
- Told that we're all in this together but
 - ignored when reporting the impossibility of meeting a deadline
 - expected to make up for project management problems with overtime and “commitment”

Step back: Why hire software devs?

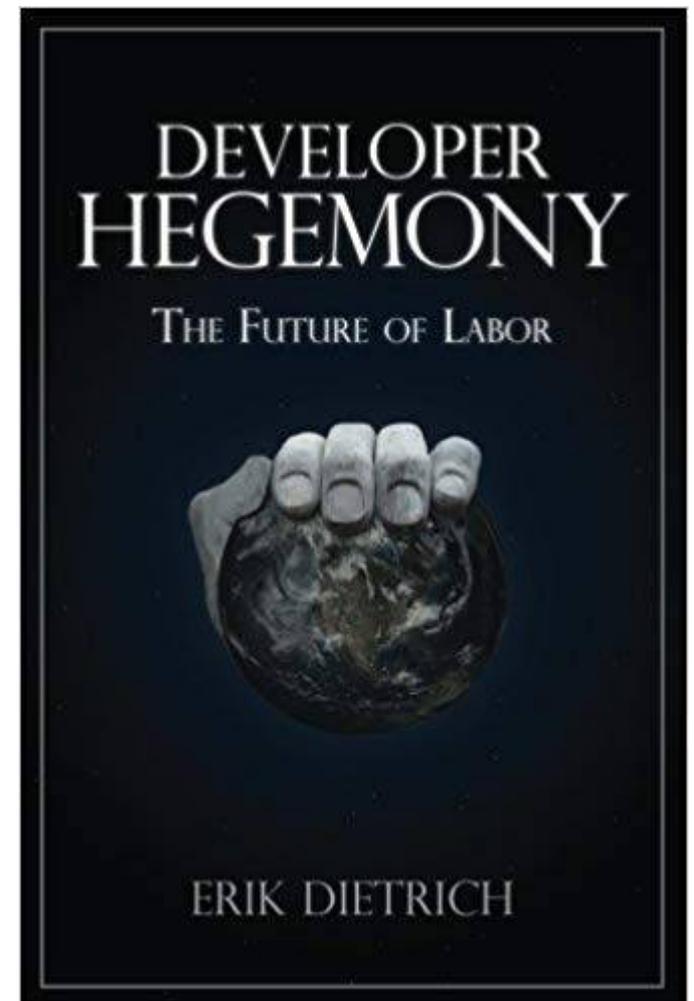
Organizations want to be more efficient:

- Do more in same amount of time
- Do same amount in less time

See Erik Dietrich's

Developer Hegemony:

The Future of Labor



Step back: Why hire software devs?

Organizations don't care about code

- It's a means to an end
- If they could avoid it, they would
- But they need it
- So they need you
- Because they don't know code
- **But they'll still act like they know how to tell you how to deliver it**



The image shows a screenshot of a Twitter thread. At the top, Heidi Araya (@HeidiAraya) posted on Sep 10: "The more practices are imposed, the less self organizing and 'responsible/accountable' & engaged people will feel. it's funny how leadership wants self organization and for people to 'be accountable & self organizing' but pushes practices. It won't work. #agile". Below this is a retweet by Daniel Mezick (@DanielMezick) featuring an image of a beetle and the text: "Freedom is Slavery. Ignorance is Strength. People are Resources. newtechusa.net/aic2/". Eric Smith (@paytonrules) is replying to Heidi Araya and @GBGames, stating: "Management doesn't want people to 'be accountable and self organizing' - management could care less. Management wants people to get things done really fast and heard this is how you do it." The tweet has 7 replies, 10 retweets, and 21 likes. It was posted at 7:33 AM on 10 Sep 2018.

Organizations want value

They invest (expense)

- Computer hardware
- Networking
- Electricity
- Facilities
- Free soda pop
- Pool tables
- You (manager/dev)

They expect a return on investment (ROI)

- Faster underpants procurement
- ???
- Profit!

Organizations want value

They invest (expense)

- Computer hardware
- Networking
- Electricity
- Facilities
- Free soda pop
- Pool tables
- You (manager/dev)

They expect a return on investment (ROI)

- Faster underpants procurement
- ???
- Profit!

The way they try to realize that value can be...clumsy.

Organizations want value

Strategic planning

- Vision (What?)
- Purpose (Why?)
- Mission (How?)

At scale?

- Hierarchy
- Filtered down vision
- Interpreted “Requirements”
- Idealized deadlines
- No ownership

Very aspirational!

Very watered down.

Organizations want value

Surprise! IT doesn't deliver!

And it's IT's fault for not delivering what it didn't decide to deliver by the deadline that IT had no input into in the first place?

That's dysfunctional, and the organization is shooting itself in the foot.

Organizations want value

Missed deadlines? Tighten the reigns on IT!

- No more pool tables!
- No more free soda pop!
- Working lunches!
- No weekends!
- Where's your commitment?

PRO TIP:

Crunch is unhealthy, self-defeating, and demoralizing.

Organizations want value

IT gets managed poorly

- Focus on stories, ignore defects
 - Eventually story work slows down
- Stop work on stories, focus on “quality”
 - No value being created
- No time “allowed” to do job right, focus on “getting things done”
 - Poor quality code gets worse
 - New code takes longer to create due to poor quality code

Ugh, who actually wins in this situation?

Weird Quote Time

- @GBGames

The screenshot shows a Twitter thread. At the top is a tweet from **GBGames** (@GBGames) posted at 7:30 AM on Sep 14, 2018. The tweet contains a quote: "A cat in gloves catches no mice." - Ben Franklin, followed by the text: "What a weird quote. But I'm totally going to incorporate it into my next presentation." The tweet has 1 retweet and 7 likes. Below this is a reply from **Marty Rabens** (@Martoonster) on Sep 14, replying to @GBGames. The reply says: "You should incorporate a weird quote into every presentation. It could be your thing." This reply has 1 retweet, 1 like, and 1 message. At the bottom is a reply from **GBGames** (@GBGames) on Sep 14, saying: "I like it, and I think I might! Thanks for the idea!" This reply has 1 like. The interface includes a text input field for adding another tweet and a Twitter logo at the bottom.

GBGames @GBGames

"A cat in gloves catches no mice." - Ben Franklin

What a weird quote. But I'm totally going to incorporate it into my next presentation.

7:30 AM - 14 Sep 2018

1 Retweet 7 Likes

1 1 7

Add another Tweet

Marty Rabens @Martoonster · Sep 14

Replying to @GBGames

You should incorporate a weird quote into every presentation. It could be your thing.

1 1 1

GBGames @GBGames · Sep 14

I like it, and I think I might! Thanks for the idea!

1

You have a mind. Don't waste it.

Even if your organization doesn't know it,
they want you to bring your expertise to the job.

You have a mind. Don't waste it.

Don't

- Stay quiet
- Ignore “biz stuff”
- Become complacent
- Accept the spec
- Accept status quo
- Accept initial request

Do

- Ask questions
- Learn what org values
- Think about users
- Think about value
- Seek improvement
- Seek better way

Remember Dude's Law?

Value = Why/How

Cynical code monkey
view:

- Biz work is valuable
- Dev work is waste

Healthy software dev
view:

- Biz work is valuable
- How I approach my work has huge impact on value realized

Hypothetical example

- “We can't do it this way because the API requires these arguments.”
- Do we own the API?
Then change it!

Tales from the Bit Mines

Map Legend feature



No 0 Rates



With 0 rates

Tales from the Bit Mines

Map Legend feature



No 0 Rates



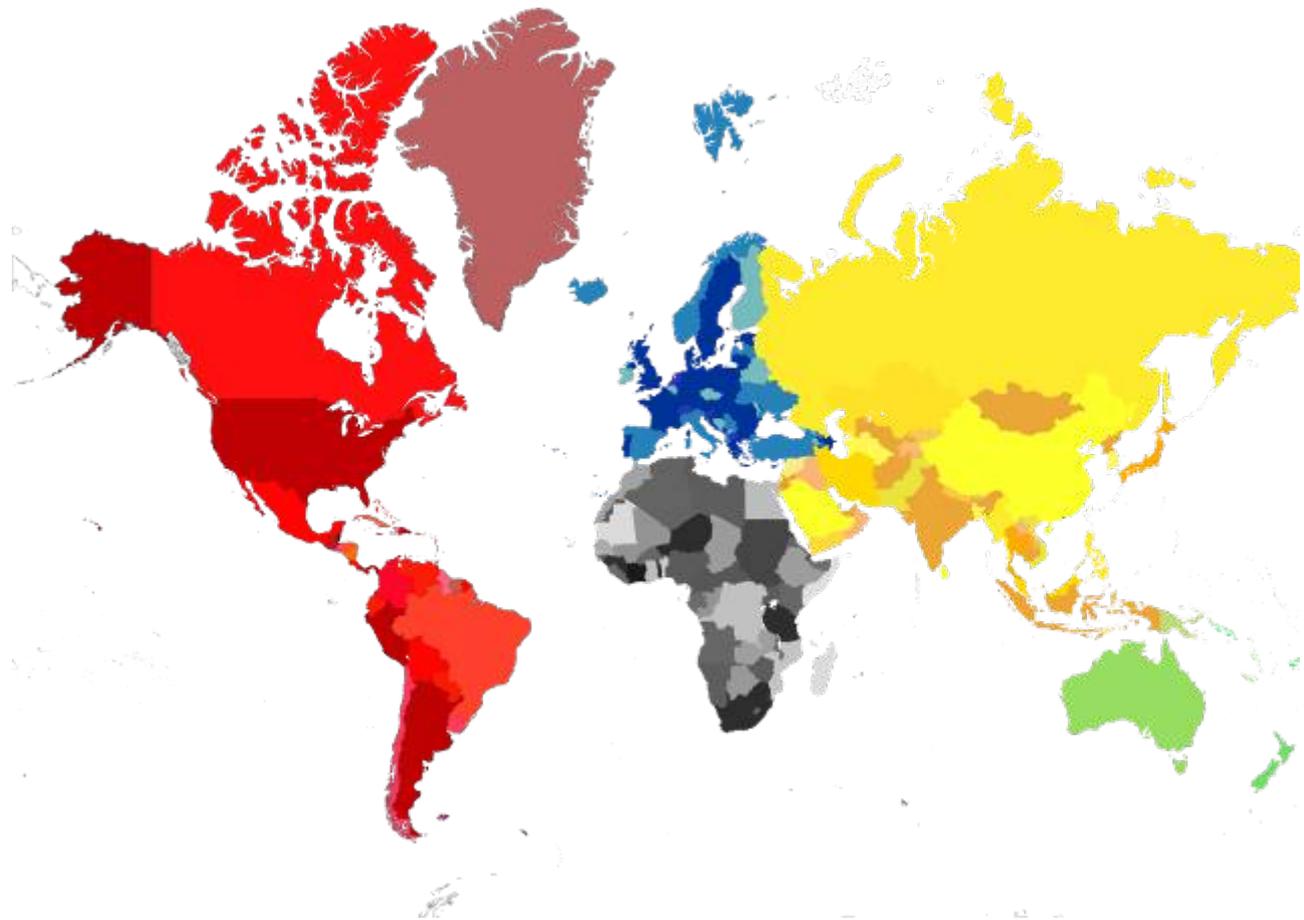
With 0 rates

**What if we just always showed the 0 rates version?
Save time, deliver faster, reduce waste**

Tales from the Bit Mines

Continent colors in geography game

- Asia = yellow?
- Africa = black?
- Americas = red?



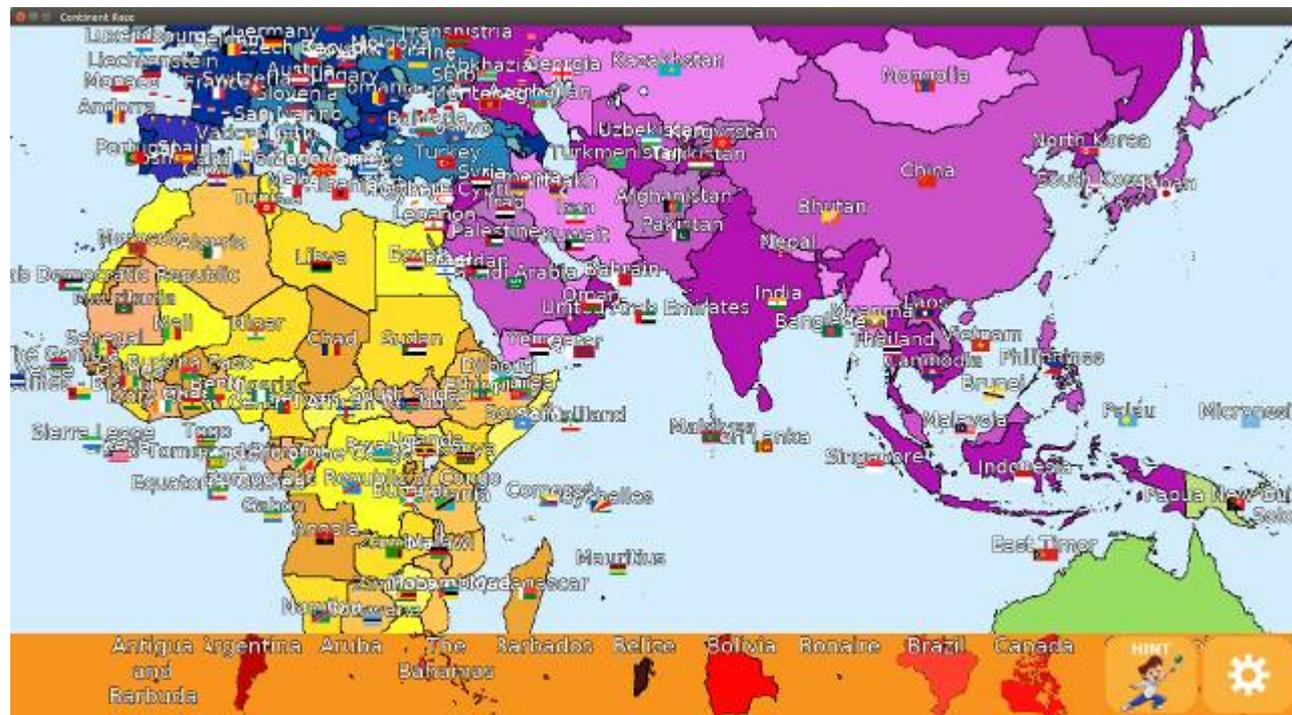
Tales from the Bit Mines

Continent colors in geography game

“We have decided to go ahead and change the colors on Asia and Africa. Since this game consultant is now the seventh person who’s brought it up even though obviously we were aware but ... I know you mentioned it early on. So we are finally taking everyone’s advice and will change Asia to magenta ... and Africa to yellow tones.”

Tales from the Bit Mines

Continent colors in geography game



Less inadvertent racism, more inclusive app

Harry Potter's no Code Monkey

From Chapter 20 of Goblet of Fire

"So. . . got any ideas how you're going to get past your dragon yet?" said Moody.

"No," said Harry.

"Well, I'm not going to tell you," said Moody gruffly. "I don't show favoritism, me. I'm just going to give you some good, general advice. And the first bit is - play to your strengths."

"I haven't got any," said Harry, before he could stop himself. "Excuse me," growled Moody, "you've got strengths if I say you've got them. Think now. What are you best at?"

Harry tried to concentrate. What was he best at? Well, that was easy, really --

"Quidditch," he said dully, "and a fat lot of help -"

"That's right," said Moody, staring at him very hard, his magical eye barely moving at all. "You're a damn good flier from what I've heard."

"Yeah, but.. ." Harry stared at him. "I'm not allowed a broom, I've only got my wand..."

"My second piece of general advice," said Moody loudly, interrupting him, "is to use a nice, simple spell that will enable you to get what you need."

Harry looked at him blankly. What did he need?

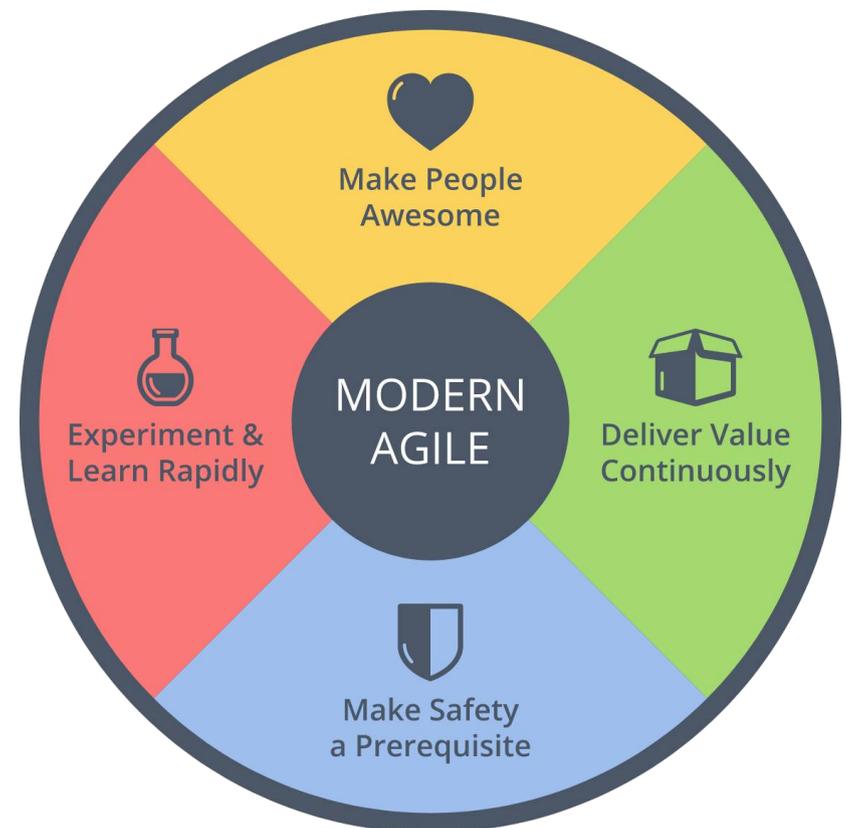
"Come on, boy. . ." whispered Moody. "Put them together... it's not that difficult..."

Safety

- Company culture needs to be there
- Need to be able to experiment
- Failure can't be a death sentence
- Learning orgs vs fear-driven orgs

ModernAgile.org:

- Make Safety a Prerequisite



Safety

- Applying for a job?
 - Ask questions
 - You're interviewing the company, too
 - What kind of company is it?
 - Command & Control monkey cage?
 - Learning environment?

The future of software development

Automation, AI/ML, etc.

Do you have a job in the future?

Not if all you do is “write code” and follow spec.

Not if all you sell is
“experience in programming languages.”

You are not a code monkey.

The future of software development

You solve problems.

You are a first-class citizen in your organization.

You leverage your talents to help
your organization do more in the same time
or do the same in less time.

Start acting like it, and your organization will treat you like it.

Game Over

Questions? Comments?

Gianfranco Berardi

<https://www.GBGames.com/>

gberardi@gbgames.com

Twitter: @GBGames

plus.google.com/+GbgamesLLC

facebook.com/GBGamesBlog

You Are Not a Code Monkey

So Stop Acting Like One

by Gianfranco Berardi



Presented at dsmAgile 2018

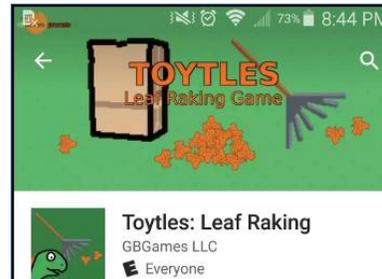
Why are you here?

- **Why?** - Because too many people and organizations are not living up to their potential
- **What?** - You with a more fulfilling attitude about the role of a software developer
- **How?** - Learn why the metaphor of code monkey is awful, no good, very bad *beats chest*

Who Am I?

Quick Bio:

- Gianfranco Berardi
- Full-time software developer since 2004
- Part-time independent game developer
 - Create educational entertainment that encourages exploration and a love of learning
- <https://www.GBGames.com/>
- Pronouns: he/him/his



Who are you calling a code monkey?!

Definitions:

Wikipedia: "A term for a computer programmer, sometimes considered an insulting or derogatory oversimplification of their skill set"

Urban Dictionary: "A programmer who isn't actually involved in any aspect of conceptual or design work, but simply writes code to specifications given."

I don't normally cite Urban Dictionary, but this definition is too good.

Who are you calling a code monkey?!

Definitions:

Wikipedia: "A term for a computer programmer, sometimes considered an insulting or derogatory oversimplification of their skill set"

Urban Dictionary: "A programmer who isn't actually involved in any aspect of conceptual or design work, but simply writes code to specifications given."

Code Monkey != Software Developer

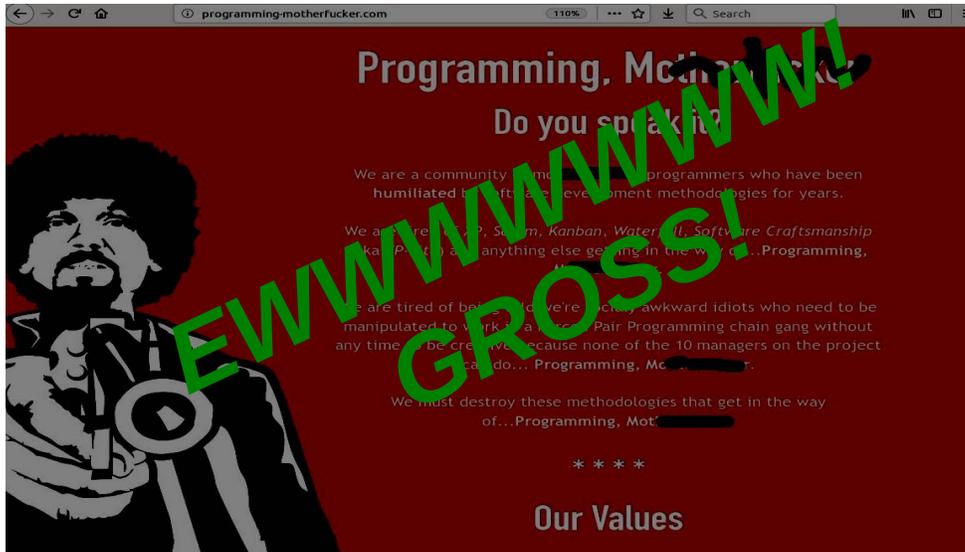
Yet there is Code Monkey pride?!



Sometimes people are proud of being code monkeys.

I hate the sentiment behind this website. I don't understand the idea that “just writing code” is somehow supposed to provide value enough to be paid.

Yet there is Code Monkey pride?!



Speak No Evil, Hear No Evil

- "I can't refactor/pair program. My manager won't let me."
- "The business has no idea what it is doing."
- "The developers would waste time on work that has no value if we let them."
- "I just want to code. I don't want to deal with squishy human stuff."
- "The deadlines are made up anyway."



"Japanexperterna.se"

Do these comments sound familiar?

Speak No Evil, Hear No Evil

- "I can't refactor/pair program. My manager won't let me."
- "The business has no idea what it is doing."
- "The developers would waste time on work that has no value if we let them."
- "I just want to code. I don't want to deal with squishy human stuff."
- "The deadlines are made up anyway."

This is dysfunctional, by the way.



"Japanexperterna.se"

Do these comments sound familiar?

They are symptoms of a dysfunctional relationship between IT and the rest of the business.

The Story of the Pot Roast

“Why do you always cut off the ends of the roast?”

“I don't know. My mother always did it that way, and I learned it from her.”

We've always done it that way.

But what if we used our brains?

Could we find a better way?

Could we help our organizations find a better way?

A woman is learning how to cook from her mother, and she asks why she cuts the ends off the pot roast.

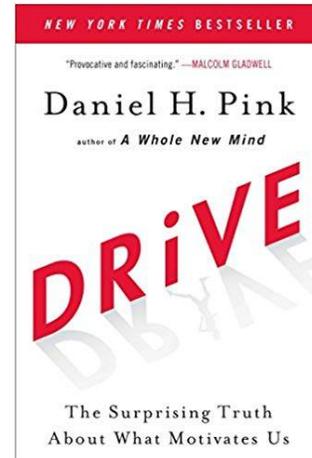
Mom thinks, and says, “I don't know. I learned how to cook from my mother, and that's what she always did.”

Repeat a couple of generations, and eventually great-grandma says, “Oh, the pan was too small for the roast, so we cut the ends to make it fit

Learned Helplessness

Drive: The Surprising Truth About What Motivates Us by Daniel H. Pink

- Autonomy
- Mastery
- Purpose

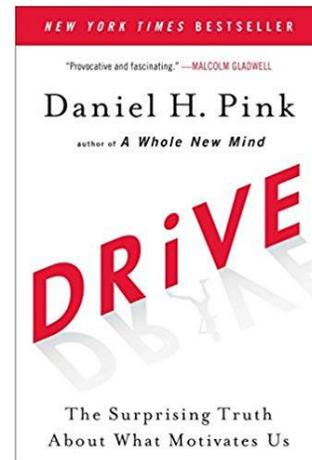


While this book covers a lot more, here's the cheat sheet version.

Learned Helplessness

Drive: The Surprising Truth About What Motivates Us by Daniel H. Pink

- Autonomy
- Mastery
- Purpose



But for our purposes, let's focus on autonomy and how organizations tend to drive it out of their developers and how our developers tend to feel unable to change the situation.

Learned Helplessness

1960s psychology research involving dogs

Can't escape painful stimuli?

- Feel loss of control
- Give up trying
- Even when situation changes!

About control, not competence.

Learned helplessness is insidious.

It can be overcome, but it requires a demonstration.

But dogs and employees can learn that they have no control and don't bother exerting control even when they do get it back.

Learned Helplessness

- What organizations do:
 - Micromanage
 - Punish
 - Scoldings
 - Worse?
 - Decide **without developer input**
 - Wonder why results stink
- What developers do:
 - Accept it
 - Self-constrain
 - Stop thinking critically
 - Leave, eventually
 - Or don't, sadly
 - Perform suboptimally

In dysfunctional organizations, IT is usually told what to do and has expectations placed on them that have no bearing on reality, and IT shrugs and says, “Eh, that's how it goes.”

Learned Helplessness

Sound familiar?

- Told to be self-organizing but
 - assigned to a team
 - told what tech stack to use
- Told to be responsible and accountable but
 - given no decision-making ability
 - except to work longer hours
 - told to not work on maintenance (refactoring, cleaning, documenting, etc)
- Told that we're all in this together but
 - ignored when reporting the impossibility of meeting a deadline
 - expected to make up for project management problems with overtime and "commitment"

IT hears one thing, but experiences another.

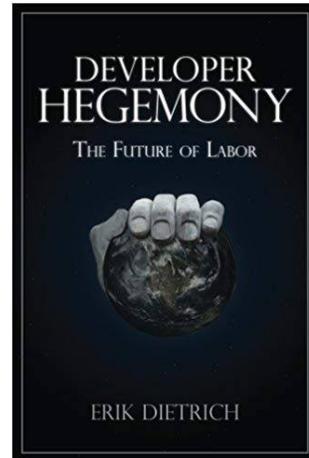
They understand how it really works, and after a few painful experiences, they learn they can't fight it. They don't have the power.

Step back: Why hire software devs?

Organizations want to be more efficient:

- Do more in same amount of time
- Do same amount in less time

See Erik Dietrich's
Developer Hegemony:
The Future of Labor



A fantastic book!

Organizations want efficiency.
Period.

Step back: Why hire software devs?

Organizations don't care about code

- It's a means to an end
- If they could avoid it, they would
- But they need it
- So they need you
- Because they don't know code
- **But they'll still act like they know how to tell you how to deliver it**



They want the result, and right now, the best way they know to get efficiency results is to hire developers to make efficiency happen.

Unfortunately, they act like they know how to tell developers how to do their job best, even when they don't. They're confident, though. Gotta give them that much!

Organizations want value

They invest (expense)

- Computer hardware
- Networking
- Electricity
- Facilities
- Free soda pop
- Pool tables
- You (manager/dev)

They expect a return on investment (ROI)

- Faster underpants procurement
- ???
- Profit!

There tends to be this idea that if you provide a bunch of free soda pop and startup furniture and a “fun” environment, developers will eventually put together some code that does what you want.

Organizations want value

They invest (expense)

- Computer hardware
- Networking
- Electricity
- Facilities
- Free soda pop
- Pool tables
- You (manager/dev)

They expect a return on investment (ROI)

- Faster underpants procurement
- ???
- Profit!

The way they try to realize that value can be...clumsy.

When they apply the mechanics without understanding, it feels disingenuous at best.

It can be cynical at worst.

Organizations want value

Strategic planning

- Vision (What?)
- Purpose (Why?)
- Mission (How?)

At scale?

- Hierarchy
- Filtered down vision
- Interpreted "Requirements"
- Idealized deadlines
- No ownership

Very aspirational!

Very watered down.

Organizations have aspirations. And they have to translate those aspirations into tactics and actual work that gets done.

Large organizations especially tend to dilute the message. By the time a user story is in front of the team, it is often devoid of context. Some manager insists that a feature is absolutely required at their level, even if it only barely relates to the higher level strategic vision.

Organizations want value

Surprise! IT doesn't deliver!

And it's IT's fault for not delivering what it didn't decide to deliver by the deadline that IT had no input into in the first place?

That's dysfunctional, and the organization is shooting itself in the foot.

So developers work, doing what they were told, because “everything is required”, and surprise, it doesn't all get done in time!

But is project management blamed? Perhaps, but more likely it's the developers who have shame brought upon them of not delivering what they never promised they could do in the first place.

Organizations want value

Missed deadlines? Tighten the reigns on IT!

- No more pool tables!
- No more free soda pop!
- Working lunches!
- No weekends!
- Where's your commitment?

PRO TIP:

Crunch is unhealthy, self-defeating, and demoralizing.

Then management tends to drive the developers harder. It's demoralizing, which results in worse performance.

Eventually developers have their commitment questioned. That's always a good one.

Crunch is awful, and there are plenty of stories of developers doing way more than is healthy on behalf of an organization that is more than happy to get more labor for the same

Organizations want value

IT gets managed poorly

- Focus on stories, ignore defects
 - Eventually story work slows down
- Stop work on stories, focus on “quality”
 - No value being created
- No time “allowed” to do job right, focus on “getting things done”
 - Poor quality code gets worse
 - New code takes longer to create due to poor quality code

Ugh, who actually wins in this situation?

There's a yo-yo effect as the project moves along. Work continues until the project is unusable and a reckoning comes, forcing the project to stop producing value as rework occurs to bring the project back to a running state.

And developers feel like they need permission to do their jobs right because of the perception that it “takes time away from delivery” or some similar nonsense. Learned

Weird Quote Time

- @GBGames



I was listening to Benjamin Franklin by Walter Isaacson, and I heard this weird quote. I'm going to put weird quotes in my presentations from now on. You're welcome.

If you forgive the change in metaphors to cats, I think it is an appropriate metaphor.

You have a mind. Don't waste it.

Even if your organization doesn't know it,
they want you to bring your expertise to the job.

Organizations teach developers to put on their gloves and fail at catching mice. Developers get the message that they are to do what they are told, even when their knowledge and expertise should be leveraged to teach the organization a better way to accomplish its goals.

Organizations need developers to do this work, even if organizations act like it.

You have a mind. Don't waste it.

Don't

- Stay quiet
- Ignore “biz stuff”
- Become complacent
- Accept the spec
- Accept status quo
- Accept initial request

Do

- Ask questions
- Learn what org values
- Think about users
- Think about value
- Seek improvement
- Seek better way

Developers need to know what their code is supposed to do. What if code isn't the best solution for the actual problem? What if a different approach would solve the problem in a much less expensive way? What if the approach is right but the way it's being presented to users is off?

Organizations are guessing at what their software should do. Developers should help turn those guesses into real value.

Remember Dude's Law? Value = Why/How

Cynical code monkey
view:

- Biz work is valuable
- Dev work is waste

Healthy software dev
view:

- Biz work is valuable
- How I approach my work has huge impact on value realized

Brandon Carlson's keynote talked about Dude's Law, and I think a cynical approach to it is to treat developer efforts as waste to be eliminated, as a cost center.

Instead, developers should see themselves as playing a large role in what kind of value gets realized. A bad approach at a solution reduces value, and developers should keep that value in mind when they make their recommendations to their

Hypothetical example

- “We can't do it this way because the API requires these arguments.”
- Do we own the API?
Then change it!

Sometimes the constraint isn't really a constraint.

Tales from the Bit Mines

Map Legend feature



No 0 Rates



With 0 rates

A real example from my life, where the feature was to add a map to a display that shows seeds per acre to plant in a field. The story involved adding a black box that represented 0 seeds/acre only if a prescription had rates of 0.

Getting this fact from the file loading layer through multiple layers to the mapping code would have been complex and potentially error-ridden.

Tales from the Bit Mines

Map Legend feature



No 0 Rates



With 0 rates

What if we just always showed the 0 rates version?
Save time, deliver faster, reduce waste

So I asked my product owner if I could just always show the black box. If there are no 0 rates, then nothing on the map gets represented, but is there a real value proposition to not showing the 0 rates in the legend?

The PO agreed (double-checking with superiors just in case), I saved a bunch of effort and time, and for all I know that work will never happen.

Tales from the Bit Mines

Continent colors in geography game

- Asia = yellow?
- Africa = black?
- Americas = red?



I am working on a geography game for a client, and I was asked to assign these colors to the regions of the world.

I raised a concern, citing companies like Apple, Dove, Sony and more who got into trouble with inadvertent racism.

Tales from the Bit Mines

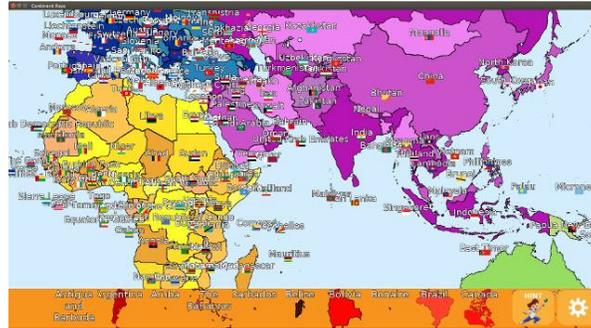
Continent colors in geography game

“We have decided to go ahead and change the colors on Asia and Africa. Since this game consultant is now the seventh person who’s brought it up even though obviously we were aware but ... I know you mentioned it early on. So we are finally taking everyone’s advice and will change Asia to magenta ... and Africa to yellow tones.”

After being told not to change the colors, I got this email one day to hear that they had a seventh person point this issue out.

Tales from the Bit Mines

Continent colors in geography game



Less inadvertent racism, more inclusive app

So now the app is more inclusive.

Harry Potter's no Code Monkey

From Chapter 20 of Goblet of Fire

"So... got any ideas how you're going to get past your dragon yet?" said Moody.

"No," said Harry.

"Well, I'm not going to tell you," said Moody gruffly. "I don't show favoritism, me. I'm just going to give you some good, general advice. And the first bit is - play to your strengths."

"I haven't got any," said Harry, before he could stop himself. "Excuse me," growled Moody, "you've got strengths if I say you've got them. Think now. What are you best at?"

Harry tried to concentrate. What was he best at? Well, that was easy, really --

"Quidditch," he said dully, "and a fat lot of help --"

"That's right," said Moody, staring at him very hard, his magical eye barely moving at all. "You're a damn good flier from what I've heard."

"Yeah, but..." Harry stared at him. "I'm not allowed a broom, I've only got my wand..."

"My second piece of general advice," said Moody loudly, interrupting him, "is to use a nice, simple spell that will enable you to get what you need."

Harry looked at him blankly. What did he need?

"Come on, boy..." whispered Moody. "Put them together... it's not that difficult..."

I was reading this passage to my 9 year old recently, and I love this demonstration of a lack of control. He thinks all he's allowed to bring to the first task of the Triwizard Tournament is his wand, but Prof. Moody reminds him that he can do more than merely stand there with a wand.

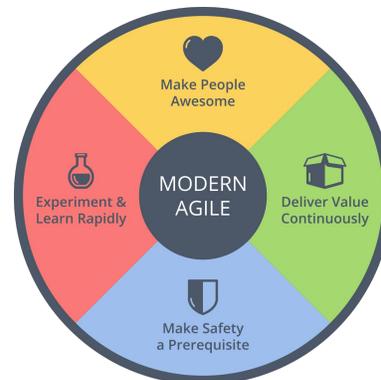
Harry eventually realizes he could use a summoning charm to summon his broomstick and use his awesome

Safety

- Company culture needs to be there
- Need to be able to experiment
- Failure can't be a death sentence
- Learning orgs vs fear-driven orgs

ModernAgile.org:

- Make Safety a Prerequisite



Ok, so developers should feel empowered to take back some control over their work, but organizations need to also adapt to this situation.

Look up Bowen family systems theory if you want to hear about how tough it is for change to happen in a family, which I think applies to an organization as well. There will always be pushback if someone starts acting differently.

Safety

- Applying for a job?
 - Ask questions
 - You're interviewing the company, too
 - What kind of company is it?
 - Command & Control monkey cage?
 - Learning environment?

As a developer, if you're applying for a job, remember that while you're being interviewed, you are also interviewing the company.

Try to find out if this company is the kind of place that expects to treat you as a code monkey or as a first-class citizen with a brain.

The future of software development

Automation, AI/ML, etc.
Do you have a job in the future?

Not if all you do is “write code” and follow spec.

Not if all you sell is
“experience in programming languages.”

You are not a code monkey.

With AI and machine learning, there's been a lot of talk about how a lot of jobs that used to be safe from automation no longer are.

While most software development tends to be manually created by individuals and teams, some problems can be solved in a potentially faster way by throwing data and algorithms together.

So if clinging code is all you sell, do

The future of software development

You solve problems.

You are a first-class citizen in your organization.

You leverage your talents to help
your organization do more in the same time
or do the same in less time.

Start acting like it, and your organization will treat you like it.

Your future is brighter because you solve problems. You make your organizations more efficient, and that's what they really care about.

And when developers can demonstrate it, organizations can expect it, and the relationship between the business and IT can be less dysfunctional and much brighter.

Game Over

Questions? Comments?

Gianfranco Berardi

<https://www.GBGames.com/>

gberardi@gbgames.com

Twitter: @GBGames

plus.google.com/+GbgamesLLC

facebook.com/GBGamesBlog

Thank you.