

How to suck up to your CFO

and other career advice



They really do only care about the bloody Benjamins

- Top line growth
 - Making (more) money
- Bottom line growth
 - Making money efficiently
- Squeezing your budget
 - Do the same or more with less
 - Do it faster



But..... How can we make money?

- We are like BASF
 - We don't necessarily make the product,
 - Sometimes we make the things that make the product.
- If we make the product, we don't sell the product
 - We don't use the products either



We create chemistry

WE DON'T MAKE A LOT OF THE PRODUCTS YOU BUY.
WE MAKE A LOT OF THE PRODUCTS YOU BUY BETTER.

And..... How can I grow the bottom line?

- I'm just the IT guy
 - We already have too much work
 - There are not enough developers.
 - We cant work any more hours
 - Our servers are old we cant afford new ones
 - We cant afford to fill the generator's gas tank



All you need is....

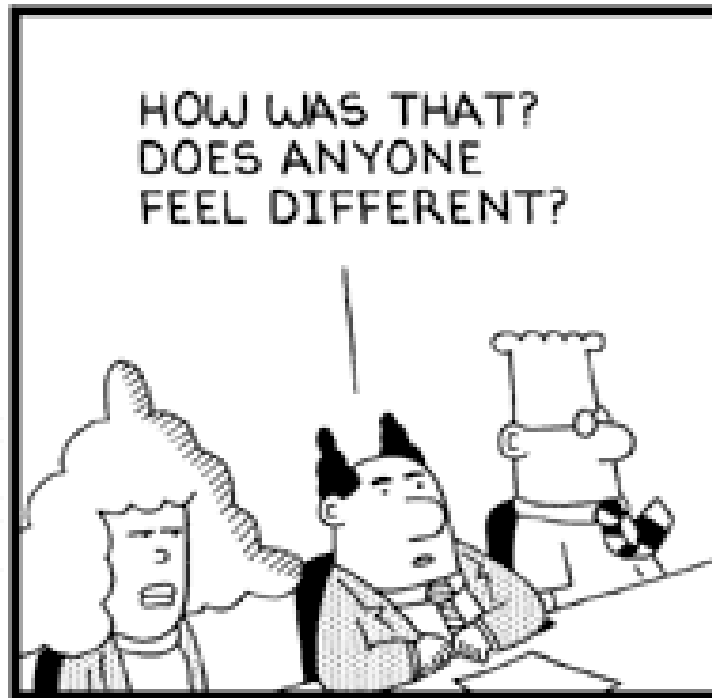


All you need is
Faith, Trust
and **LOT** bit of
Pixie dust

My Promise (today): No marketing bullshit.



www.dilbert.com scottadams@aol.com



1-7-05 © 2005 Scott Adams, Inc./Dist. by UFS, Inc.



We all need a little therapy

It's a three step program:

1. Act
2. Empower
3. Trust



Action - Taming The Pink Unicorn

- Omg please don't talk to me about:
 - Dev Ops
 - Agile
 - Acronym du Jour
- Cherry pick the flavor you need
- Make sure it is
 - Flipping business value focused
 - Damn Iterative
 - Totally Test driven
 - Automated to hell and back



Deliver Business Value (Incrementally)

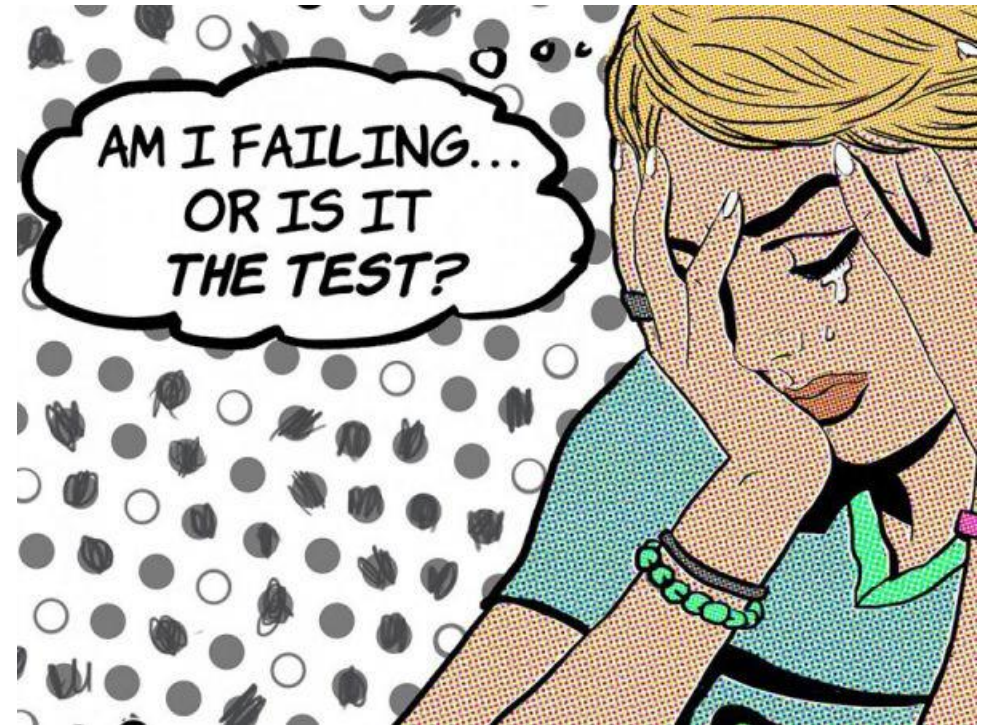
- Techniques to conquer:
 - Helping the user find the requirement
 - Thin slicing
 - Minimal viable product
- If it doesn't get deployed ... it didn't happen
- If they cant see it ... it didn't happen
- If it doesn't add value for them.... the wrong thing happened



**STAY
FOCUSED
AND
GET STUFF
DONE**

Test Driven

- Write your (unit) tests first
- I don't want to hear (again) why not
 - Yes it really does save time and money
 - Remember bottom line growth?
 - Fire the naysayers
- Unit test
- Input-Output
- UI regression
- Integration
- Code Analysis



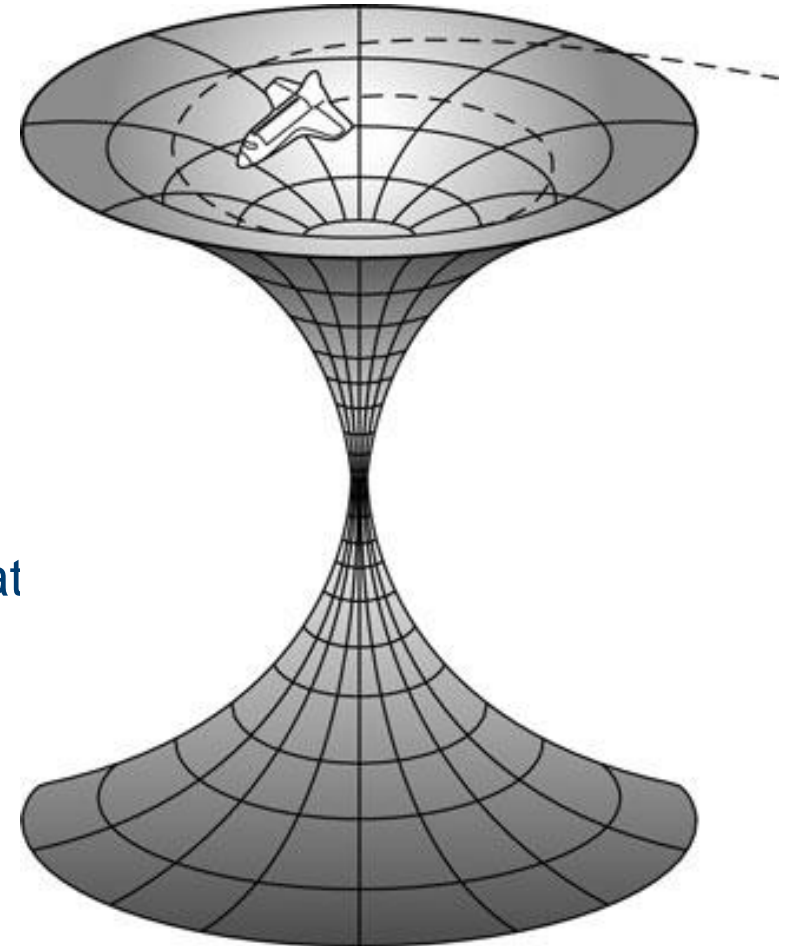
Automate

- Do a task once – fine
 - Do a task twice – okay
 - Do a task three times – shame on you
-
- Build
 - Deploy
 - Test
 - Promote
 - Roll back
-
- Not just Software also Infrastructure as Code



The cost of the context switching wormhole

- Where does it come from?
 - The constant tsunami of support
- Measure it
 - Support ticketing/email system (JitBit)
 - Baseline workload
- Fix it
 - Rotate per iteration *Defect Dude(s)* – Triage, Communicat
- Outcome:
 - Less context switching – greater efficiency
 - Knowledge sharing - breadth
 - Celebrate the zero day



Let the lunatics run the asylum aka Empowerment

- Steering committee
 - Business stakeholders
- Agree upon (your percentages)
 - 10% Support/Break Fix (Defect Dude)
 - 60% New Feature (Top Line)
 - 30% Maintenance/efficiency (Bottom Line)
- Use T-Shirt sizes to prevent μ -management
- Review and prioritize backlog – assign to %
- Let them fill the iteration



The process is the police

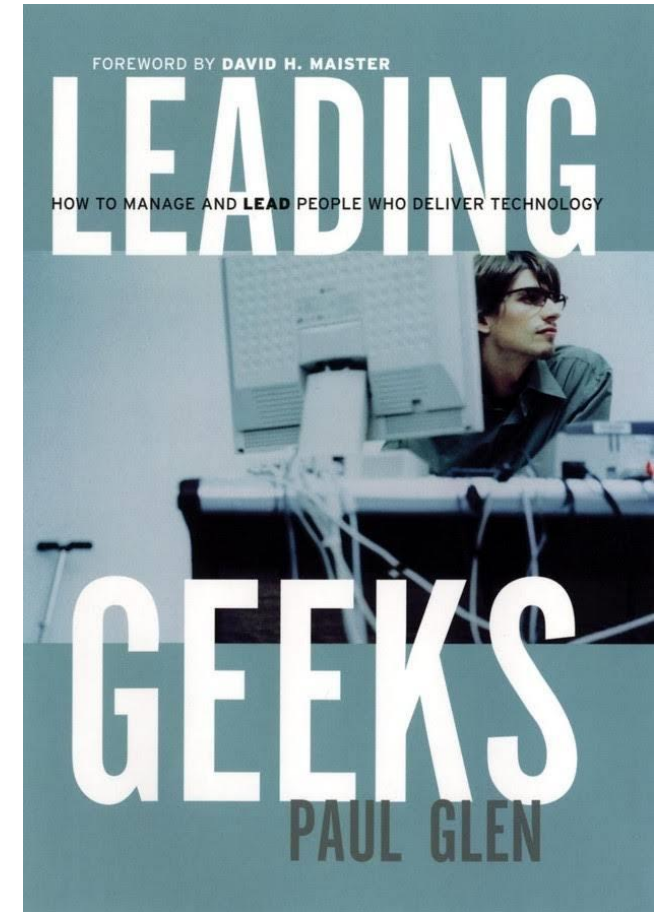
- Push back on the “special” people
 - The process protects you
- Be involved early as a developer
 - Understand the business need and value
- Ask what, not how
 - Don't let them tell you how to do your job



The role of you the IT Leader

- Protect the team from themselves and others
- Provide them the resources
- Feed them occasionally*

*Read Leading Geeks by Paul Glen



Communicate - It really is this easy:

- What did you do.
 - Support
 - Maintenance
 - New Features
- What didn't you do?
 - You are human
- What are you going to do.
 - And remind them when



The Trust Phase

- If you do what you say you will do

Repeatedly

- They may just begin to trust you



So back to the whole CFO thingy

- Please may I have some more money to spend on:
 - DevOps*
 - Cloud Infrastructure*
 - People

*Buzzword bingo points



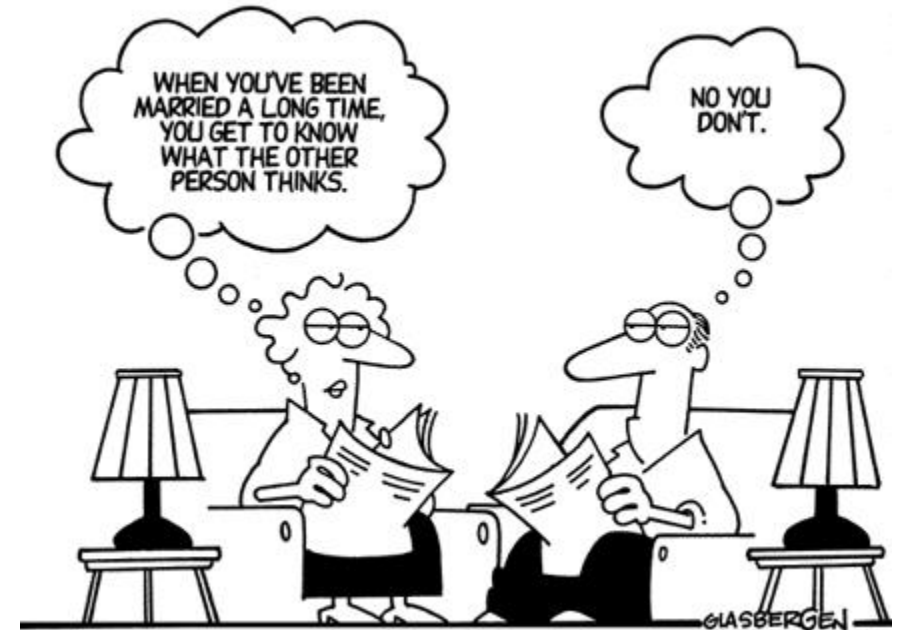
Talk their talk – learn their language

- Top Line
 - Make more money
- Bottom Line
 - Make better profit/margin %
- ROI
 - Return on investment
 - $(\text{What it will gain us} - \text{What it will cost us}) \div \text{what it will cost us}$
- NPV
 - Net Present Value
 - ROI factoring in multiple years and changing value of money over time



What works best?

- I need your approval to purchase a new server.
- We would like to automate the way we deploy.
- We can accelerate new features and bugs by adopting DevOps.
- We can decrease our time to market while at the same time making expense savings that will provide a one year ROI (or NPV) of \$X
- We can help meet this years top line growth goal whilst decreasing our expenses next year. We estimate a one year ROI of \$X. This will require an incremental expense of \$Y this year as we establish a new method of deployment automation.



Justification

- Decreased time to market (earlier revenue)
- Infrastructure cost reduction
- Maintenance cost reduction
- Move from capital (and depreciation) to expense (+ve or -ve)
- Operations efficiency (space, power, time, labor)
- Headcount reduction
- Increasing internal function efficiency (other team's headcount)



Take-aways

- Alignment of initiative with the objectives
- Use aligned language
- Everything saves or makes money
- Can you move away from being a cost center?



How to suck up to your CFO

and other career advice

