

15 awesome tips for a great culture in fast-growth startups

Julien Simon, CTO @ Viadeo



15 awesome tips for a successful culture in fast growing startups

Julien Simon, CTO @ Viadeo



Life in the tech trenches

Julien Simon, CTO @ Viadeo



Who am I ?

- 20+ years in R&D teams,
from smartcards to web platforms
- VP Eng @ DigiPlug, CTO @ Pixmania,
VP Eng @ Criteo, CTO @ Aldebaran Robotics
- Grew teams from 8 to 60+ people
- Managed teams up to 150 people

I've seen things you wouldn't believe

(and done a few myself, I have to admit)

My only goal tonight is to help you avoid some of them.



Disclaimer (ooooh): I'm going to be rude, harsh and cynical.

Feel free to disagree 100% 😊

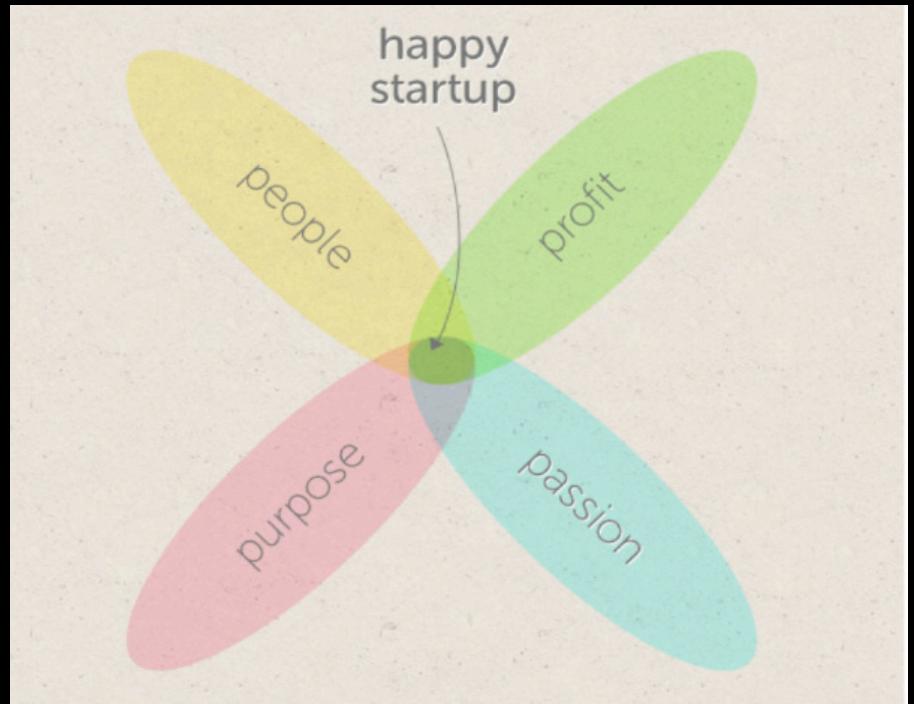
The beginning of the end

Your company is running fine,

You've got 5-10 fine engineers,

Management says you have
to hire 10 more,

Celebrate now, because...



SMOKEY, MY FRIEND



YOU ARE ENTERING A WORLD OF PAIN

Yes, winter is coming



Hiring
Tools
Methodology
Processes
Technology

And the worst of all..
Politics

They will creep in your company (no, you're not special)
Guard the wall, because if you don't, who will?
And if you don't want to do it, you picked the wroooooong job, buddy.

Hiring

We should really stop doing it



Hiring : default answer is « no »

- Can you convincingly answer :
 - « Why this candidate? » « What does (s)he bring that we don't have? »
 - « Why now ? » « What if we wait a few more months ? »
 - No Asshole Rule ?
- How ?
 - Coding interviews : « Cracking the Code Interview », Codility, Project Euler, etc.
 - « Do you code outside of work? » : Github, open-source projects, Stack Overflow
 - Reference checks !
- Caveats
 - Rockstars: « This guy was awesome at Google/Microsoft/etc. We must have him ». Maybe, maybe not. Different company, different game.
 - Dev leads: raise the bar. Then raise it again. Cost of mistake is sky high.
 - Hire « anyway » (especially juniors). No, no and no : wrong hire → more problems.
 - Made a mistake ? Fix it ASAP. Things NEVER « get better ». Minimize damage to the code base and to team morale.

Yes, I'm a bitter middle-aged man

Passion for a job well-done has become an exception (do you hear me, Gen Y?).

Proper Computer Science skills (Knuth, Meyers, GoF, etc) are so rare it's scary.

Once again : GIVE THEM A LONG, HARD LOOK & KEEP THE BAD ONES OUT!



Tools

« The Joel Test: 12 Steps to Better Code » (2000)

The Joel Test

1. Do you use source control?
2. Can you make a build in one step?
3. Do you make daily builds?
4. Do you have a bug database?
5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
9. Do you use the best tools money can buy?
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?

Your real score is the one I'd find if I audited you ;)

Bonus: « Joel Test for Programmers » (article on simpleprogrammer.com)

The Joel Test

1. Do you use source control?
2. Can you make a build in one step?
3. Do you make daily builds?
4. Do you have a bug database?
5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
9. Do you use the best tools money can buy?
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?

Real-life examples

... from leading companies.

Makes you wonder about the other ones, huh?

#1 : « *Yes, we use SVN and everyone commits to TRUNK* »

#2 : « *Yes, Gérard does it on his PC and puts a ZIP file on our filer* »

#3 : « *Yes, well no, not for the last 6 months* »

#4 : « *Yes, we use a custom version of FlynSpy 0.8* ». Variant: « *Excel works great* »

#5 : « *No, Marketing won't let us* »

#7 : « *Why? We are an Agile team* »

#10 : « *Yes, we test in production because it's more convenient* »

There is no excuse for not getting this right.

This is priority #1 for the CTO.

Methodology

Yes, you need one



Methodology

- Agile, Agile, Agile
 - Agile has become a cult (like ISO9001 30 years ago). Be a practitioner, not a priest.
 - Yes, it's OK to adapt Scrum/Kanban to your own context. Whatever works.
 - The Agile Manifesto (2001) is the light in the dark. Stay on the path, you'll be fine.
- Multi-discipline teams
 - Engineering + ops + designers + product managers working as one.
 - A unicorn? Not at Viadeo (one of the reasons I joined, actually).
 - Awesome, but VERY VERY hard to get right.
 - Long-term CxO commitment mandatory.
- Lean product development
 - Lean, MVP, walking skeleton: fine, but make sure you eventually deliver something consistent. Sum of demos and PoCs != Product
 - « Fail fast », « move fast, break stuff », « trial and error »: fine too, but make sure you have solid tests and the right KPIs or else, how will you know you failed?

Processes: skip this slide if...

(you're totally delusional about the sorry state of your engineering practice)

Bugs are always assigned and solved in time,

Your code scales endlessly,

Monitoring always catches production issues,

You never run out of budget
(or hosting space, or servers),

Your Disaster Recovery plan is just a click/script away,

The same problem never happens twice,

Etc. etc.



If not... processes, then

- Yes, I hate them as much as anyone,
all the more if they're arbitrary and inefficient.
- Wikipedia says : « a collection of activities that takes one or more kinds of input
and creates an output that is of value to the customer ».
- I sez : «For better or worse, I know only one way to do some things right.
And one way is all I need ».
- Sit down, write some simple, proven rules that prevent real problems
and make sure they're enforced every day (ass kicking may be required).
- Could it be that this is what the 'C' in 'CTO' stands for? Hmm?

Technology

The 'T' in 'CTO', remember?



WHAT DO YOU MEAN

« LOAD ON THE MYSQL MASTER IS 252.4 » ?

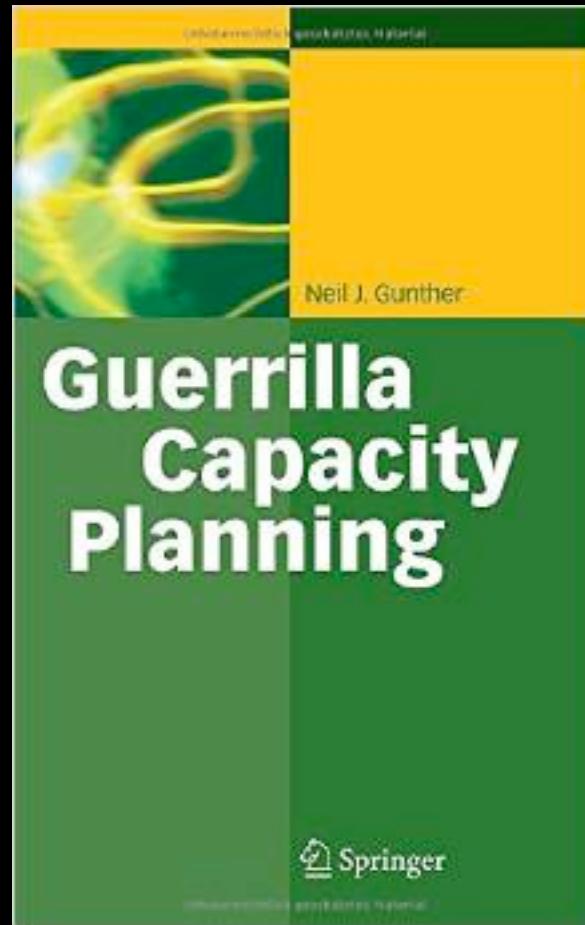
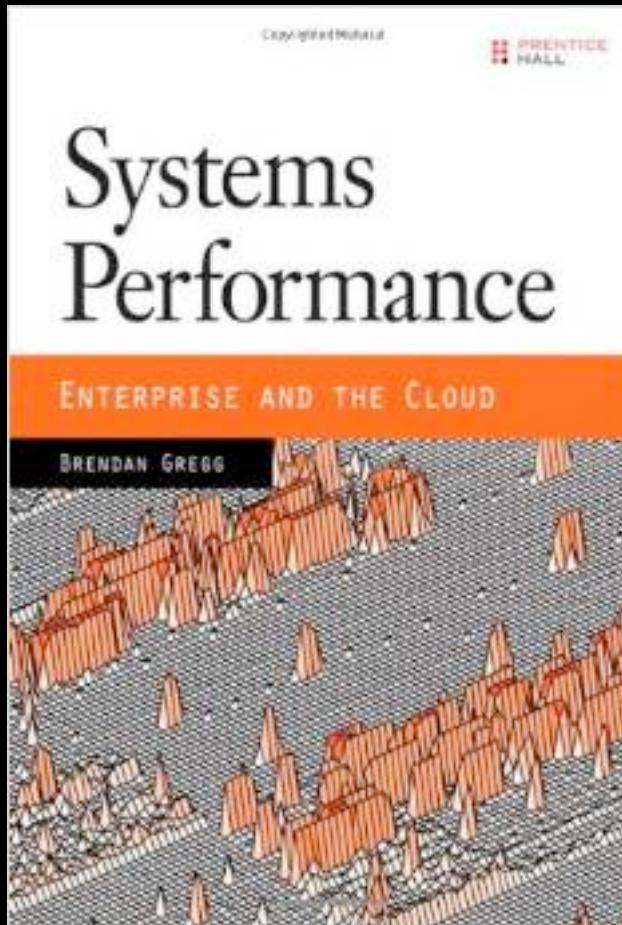
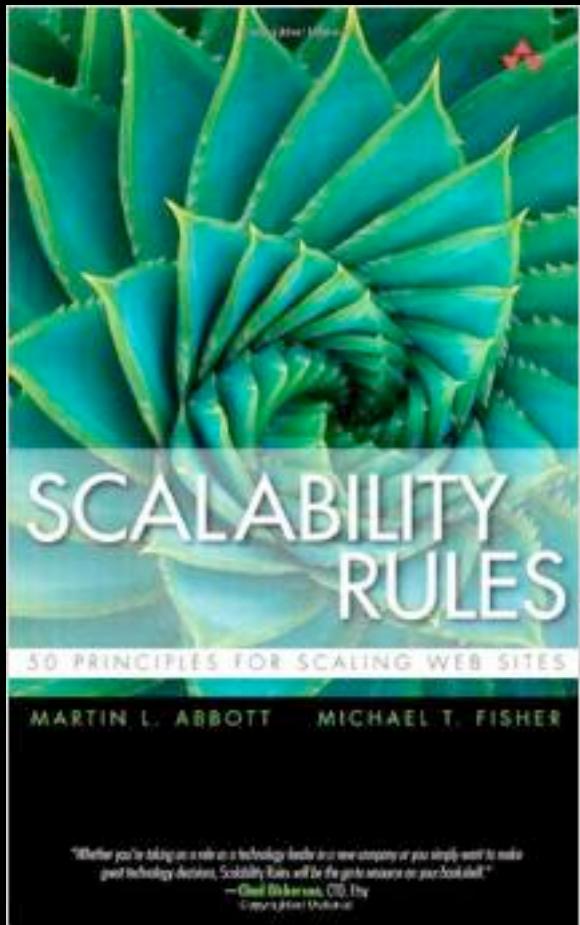
Technology : the big secret (ooooh)

1. Focus on business needs: don't build cathedrals, don't get « lost in the Bazaar » (great article by Poul-Henning Kemp)
2. Identify top challenges: time to market? UI? Perf? Security? Don't know?
3. List candidate technologies, expecting them to last at least a year (think 10x)
4. KPIs, benchmarks, PoC: educated guess is OK, random decision isn't!
5. Implement, deploy and monitor
6. Anything on fire ?
 - Can it be fixed by code optimization/refactoring?
 - If not (are you really sure?), can it be fixed with new technology?
 - Yes: you need a new building block in your stack, GOTO 2.
 - No: WTF? Are you scared? Man up! Not moving = death
 - If there is absolutely no other way, add servers... but it won't work forever!

KISS, DRY, and watch out for NIH

Must...read...books

because you don't really learn anything useful in school or at work
and they'll look great on your desk ;)



Ever heard these?

Real-life examples that will restore your faith in well-educated, well-paid engineers

- Trolls: « *Java is for pussies. Real men use C++* »
- Lunatics: « *Erlang is the bomb. Can't you see, old man?* »
Also works with Haskell, Clojure, etc.
- Living in the past: « *SQL Server has always worked for us* »
- New boss: « *Let's rewrite everything in ...* »
- Boss buddies: « *My ex-colleagues at Microsoft would like to meet you about Azure* »
- **And the worst of all**, fanboys & hipsters: « *guys, HackyLib v0.1 has just been pushed to Github. It's totally awesome. Spotify, Netflix and Valve are already using it in production. Let's use it too!* »
- Variant: the latest trend your boss read on Business Insider, HBR or worse: « *We need a Big Data strategy* » → DIE, DIE, DIE

SAY SPOTIFY AGAIN

I DARE YOU

memegenerator.net

You need to make YOUR OWN choices and build YOUR OWN platform.
Ideology, conservatism or trends won't get you anywhere.

Which leads me to...

Politics

« You've got to watch them, be quick or be dead » Iron Maiden



Politics

- CTO = tech leader + people manager + executive (that's what the 'O' means)
 - You have to be all three. Able to? Want to? Allowed to? Encouraged to?
 - In a web company, technology cannot take a back seat (willingly or not).
 - Don't be the tech dude who « executes » while the big boys « strategize ».
 - Let tech be heard. No one is going to do it for you... You owe it to your team!
 - Try to work efficiently with non-tech managers, but don't forget: most of these guys were TAUGHT politics and they LOVE it.
 - The odds are against you, so be smarter, run faster... and pick your fights.
 - Engineering = teamwork, transparency, facts, continuous improvement.
 - Politics = ego, lies, twisting facts, status quo.
- NO POLITICS allowed inside the IT team. Zero, none, zilch, nada. Get it?
- Especially between you and your team. Remember that asshole manager you had to work for in a previous company? Don't be him. Simple as that.

Fear is not an option

Whatever happens, kick ass, stay true and try to have fun



Thank you very much for listening

Happy to answer questions or
listen to comments.



ROCK ON

Staying in touch:

julsimon@viadeoteam.com

Twitter: @julsimon

Blog: juliensimon.blogspot.com

And of course on Viadeo...
and LinkedIn ;)