Lightweight Chaos

For Distributed Teams

Aras Pranckevičius
at BuildStuffLT 2013

What process?

- I don’t know!
- Use whatever makes sense!
- well that’s it, kthxbye

srsly tho

- Here’s what has (and has not) worked
- At Unity Technologies
- Where many are allergic to “process”

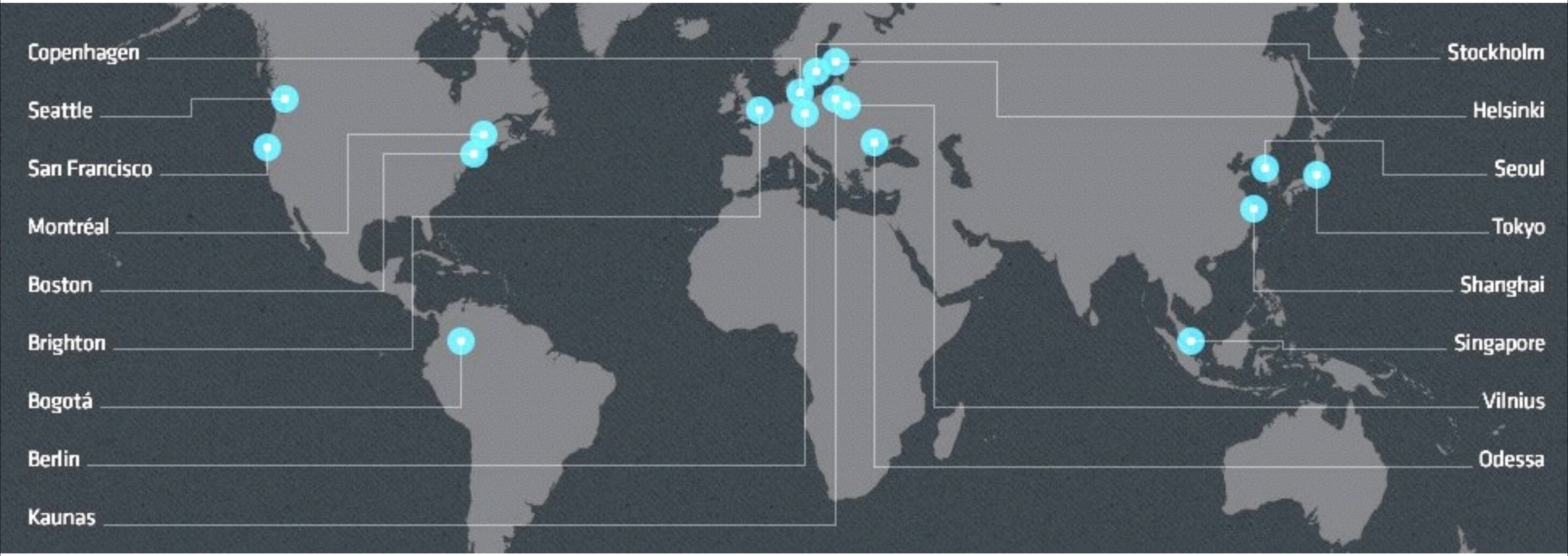
/me

- Coding since 1996, paid to do that since 2000.
- Mostly a computer graphics programmer.
- Employee #4 at Unity Technologies, since 2006.
- @aras_p
- http://aras-p.info

Unity

- Game development engine & toolset.
- http://unity3d.com
- 2.5 million users.
- Thousands of games.
- Games on 1 billion mobiles, 250 million PCs.

Distributed



- 17 offices
- ~20 more random one-person places

Why distributed?

- Happened by accident
- Easier to find talent w/o relocation
- Smaller offices can be more productive
- Specialize based on location

Specifics

- 120..150 engineers *(depending on how you count)*
- Mostly one product
- 1.4 MLOC codebase, mostly C++ & C#
- Some code is 12 years old

Specifics

- No written specification
- Close to zero design docs

Product / Project

- We’re making *product*, not *project*
- We decide *what* we build*
- We decide *when* we build it*

- Makes many things easier

* mostly

Process TL;DR

- What process?
- Everyone hates “The Man”
- Depends on situation
- Let teams & people decide what to use

Caveats

- Can only work if people *care* about what you build
    - Someone will step up to do *boring* tasks since the result is worth it

Fancy quotes!

- Guidelines, not rules
- Only add process when not having it becomes unbearable
- Be suspicious of silver bullets
- The only constant is change
- Most approaches are overhyped, but probably have one good idea in them

Heretic quotes?

- Processes are for managing mediocre people
- In a group of people, stupidity adds up, but intelligence multiplies*
- To maintain intelligence level, only hire people who are better than current average!

* in 0..1 range

3 person company

- Zero process
- Zero hierarchy
- Everyone knows everything
- Everyone decides everything
- Everyone does everything

15 person company

- More specialization
- Still everyone knows everything
- Still zero hierarchy & process

40 person company

- Teams forming
- We got one “manager”!

150 person company

- Dunbar’s number
- “this will kill your startup”
- …it did not

~320 in company now

- 120-150 engineers
- Can’t know everyone
- Not a single person knows everything
- Have to have *some* hierarchy
- 5 engineer managers right now

How we work?

- Parts of “how” might change at any time
- Quite chaotic
- Barely any “rules”

Teams

- 2..10 people
- Usually has a “lead”
    - Lead is main point of contact
- Workflow completely up to each team

Team process

- Some use Scrum, Kanban, …
- Some don’t use anything in particular
- Some things are quite common among teams now…

Common things

- “bugfix week” - whole team only fixes bugs for a week
- “technical debt week” - refactoring, cleanups etc.
- “fridays are for fun” - on Fridays everyone does whatever they want

Common things

- Major feature releases each 2 months (wish)… 4-5 months (reality)
- Feature work before alpha
- Tweaking, feedback, polish, fixes during alpha
- Final small tweaks, majority of fixes during beta
- Final shipstopper fixes during RC

Who does decisions?

- Whoever wants to
- Talk with others
- Persuade others
- Argue with others

Who’s responsible?

- Everyone is
- Can not say “they told me to do this stupid shit”
    - Should have said “this is stupid shit”!

Responsibility is hard

- Takes a while to accept that *you* are responsible
- “Why is this still broken?!”
  - Because *you* did not fix it
    - …or did not convince others to fix it ;)

What does a lead do?

- First point of contact for the team
- Direction if things turn bad
- Has to be *respected* by the team!

Tech vs people lead

- Technical lead != people lead
  - Very different skill sets!
- Too often career path for great coders is “be a people manager”

Can this really work?!

- So far it does!
- github sounds similar
    - http://zachholman.com/talk/how-github-no-longer-works/
- Valve sounds similar
    - http://media.steampowered.com/apps/valve/Valve_NewEmployeeHandbook.pdf

Will this work?

- We thought it would break down at
  - 20
  - 50
  - 100
  - …
- Some change happened, but not essence

Will this work at 1000?

- Who knows
- 1000 engineers probably too much for 1 product?
  - But then, we thought 100 would be too much as well…

But it’s unpredictable!

- Yes
- You can still guesstimate
- Our deadlines are self-imposed anyway

Tools: IM

- Skype right now
  - Not ideal, but kind of works
  - Good: sending files around
  - Bad: discovery
- Alternatives: Hangouts, Hipchat, Campfire, IRC, …

Tools: Wiki

- Confluence
  - It’s pretty good
- Google docs
  - Excellent for realtime editing
- Guidelines, plans, status, docs, …

Tools: pairing

- TeamViewer
- Alternatives: Hangouts, Skype, RDP, …

Tools: planning

- Trello
- Asana, Greenhopper, Pivotal Tracker, Basecamp, MS Project (!)

What did not work

- Mandatory Monday Meetings
    - Supposed to be quick info & status updates
- Does not work at 15 offices
- Found them useless

What did not work

- Trying to force everyone to “work harder”
  - At some point progress *felt* slow
  - Maybe people aren’t working hard enough?
- Was one of biggest fuckups we made

People do work hard!

- The slowdowns are due to *inefficiency*
- Automate
- Reduce duplication
- Better communication
- Better teaching & learning
- Get better HW/SW :)

What did not work

- Trying same process for all teams
  - “Everyone do Scrum from now on”
- Team sizes & composition is different
- Teams work on different things & schedules

Recap

- Do not force process
  - Just let people do work
- You do give up some control
  - It was an illusion anyway

Well that’s it!

Questions?