Betriebswirtschaftlich relevante Erfolgsmetriken von Social Coding-Programmen in deutschen Großunternehmen

Open/Inner Source Success Metrics that satisfy upper management and do not frustrate developers

Johannes Nicolai (jonico@github.com)


agile
development
software
change
work
methods
principles
techniques
lifecycle
must
process
processes
team
code
project
working
business
instead
projects
rate
significant
documentation
productivity
stakeholders
specifications
detailed
product
using
agility
integration
month
manifesto
manufacturing
resulted
serve
hard
adopting
throughout
features
impact
activity
evidence
defect
valuable
difficult
behavior
knowing
adopt
teams
however
supporting
following
lines
may
people
waterfall
early
study
iterative
continuous
changing
functional
systems
requirement
environment
found
collaboration
releasing
system
progress
phase
give
per
one
also
common
transition
used
value
clients
activities
respond
experience
little
design
conceptual
often
tools
approach
amount
source
come
factors
offer
discuss
need
quality
larman
less
developers
defects
communication
various
managers
first
frequently
new
guide
feedback
make
even
writing
develop
showing
experts
advantage
structure
initial
complete
iterations
expected
goal
plan
better
help
many
lies
within
complexity
material
malleable
define
including
specification
claim
weeks
discourage
possible
movement
recognize
always
month
integration
using
agility
serve
productivity
manifesto
stakeholders
specifications
client
analysis
comfort
resulted
manufacturing
features
impact
activity
hard
adopting
throughout
product
detailed


I DON'T ALWAYS DO AGILE
Wie wäre Kalenderwoche 53?
BUT WHEN I DO, I DO IT LIKE WATERFALL
memecrunch.com

Input for this presentation

zalando

SAP

BMW

DEUTSCHE BÖRSE GROUP

exact

SIGNAVIO

Allianz

AUTODESK

SOCIETE GENERALE

DAIMLER

Continental

HYPOPORT

Agenda

- GitHub Kurz-Einführung
- Open/Inner Source-Erfolgsmetriken von Allianz, SAP, Continental, Autodesk, Exact, SocGen
  - Ramp Time
  - Developer Happiness
  - Deployment Frequency (und das größte Missverständnis hinter Inner Source)
- Zusammenfassung / Q&A

The #1 Developer platform on the planet

Most contributions **1.1B in 2018**

Most developers **40M**

Highest growth **8M new devs in 2018**

Most Repos **110M**

Most activity **200M PRs, 800M API requests daily**

Most students **1.1M**

Most organizations **2.2M**

Most secure **27M vulnerability alerts in 2019**

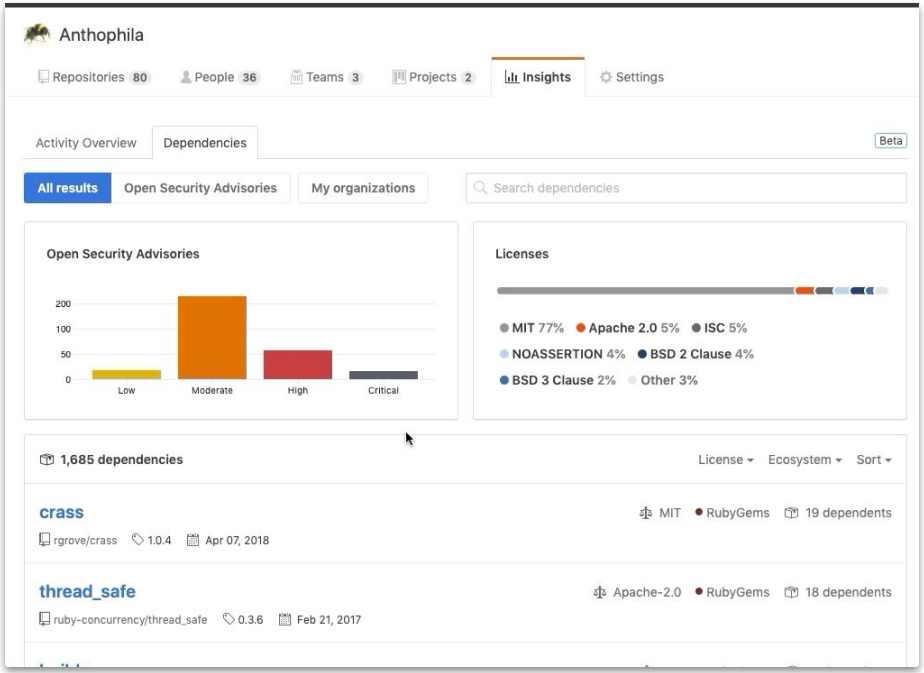
Global Rank **50th most visited website worldwide**

Dependency insights

Understand open source dependencies and how they impact your business.

Drill-down to discover which dependencies have **security advisories** or risky **licenses** (e.g. GPLv3)

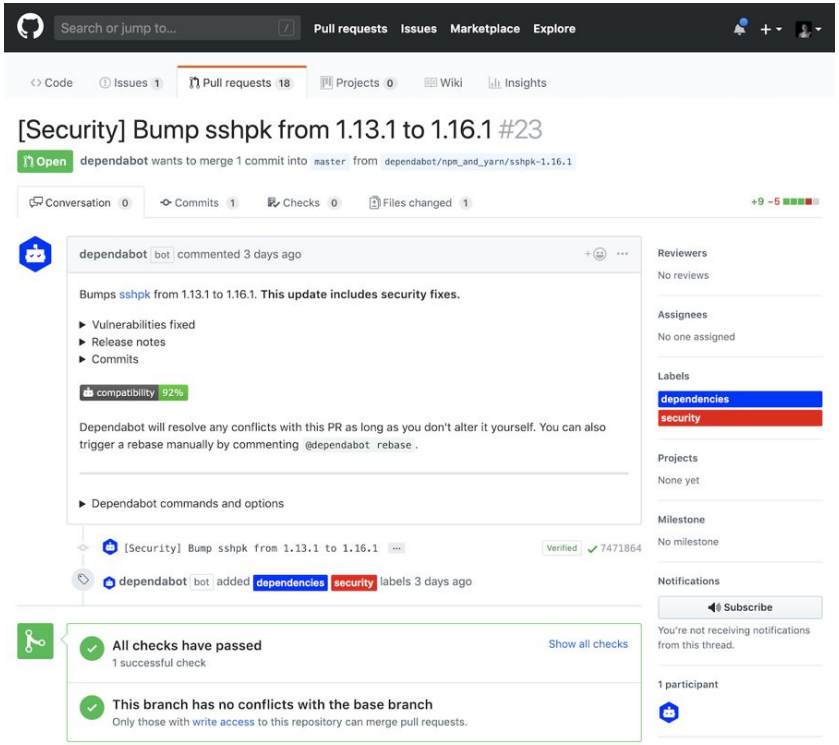
- Identify the repositories
- Take corrective actions



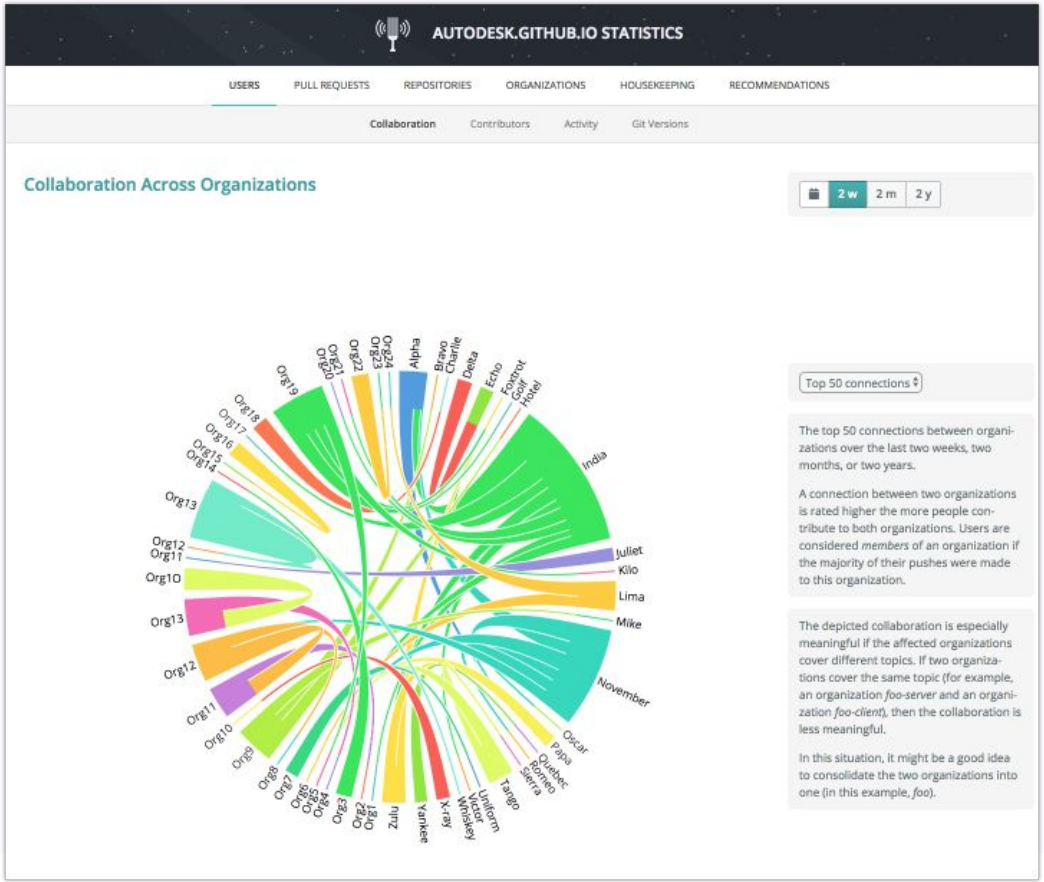
Automatic Security Fixes

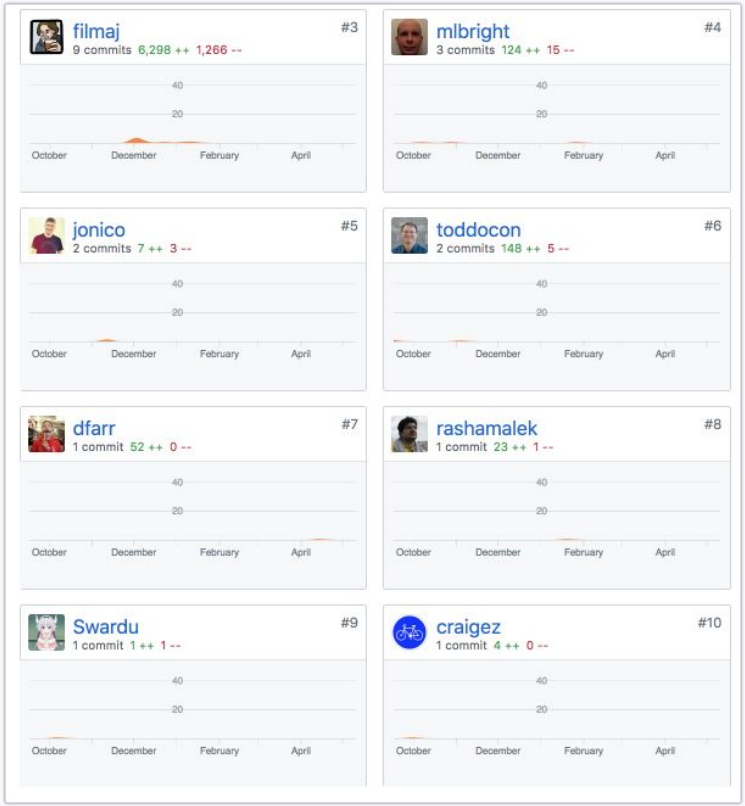
Keep your code secure and up-to-date

- Remediation patches -> pull-requests
- Confidence Score on merge
- Planet scale “update” workflows



Visualizing cross-org collaboration





Businesses Using GitHub Today

Allianz Global ADP (Agile Delivery Platform)

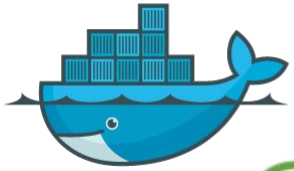
Jenkins

CLOUDFOUNDRY

Pivotal Elastic Runtime

OPENSHIFT® by Red Hat®

ANSIBLE



docker

Test Automation

Selenium

JFrog Artifactory

sonarqube

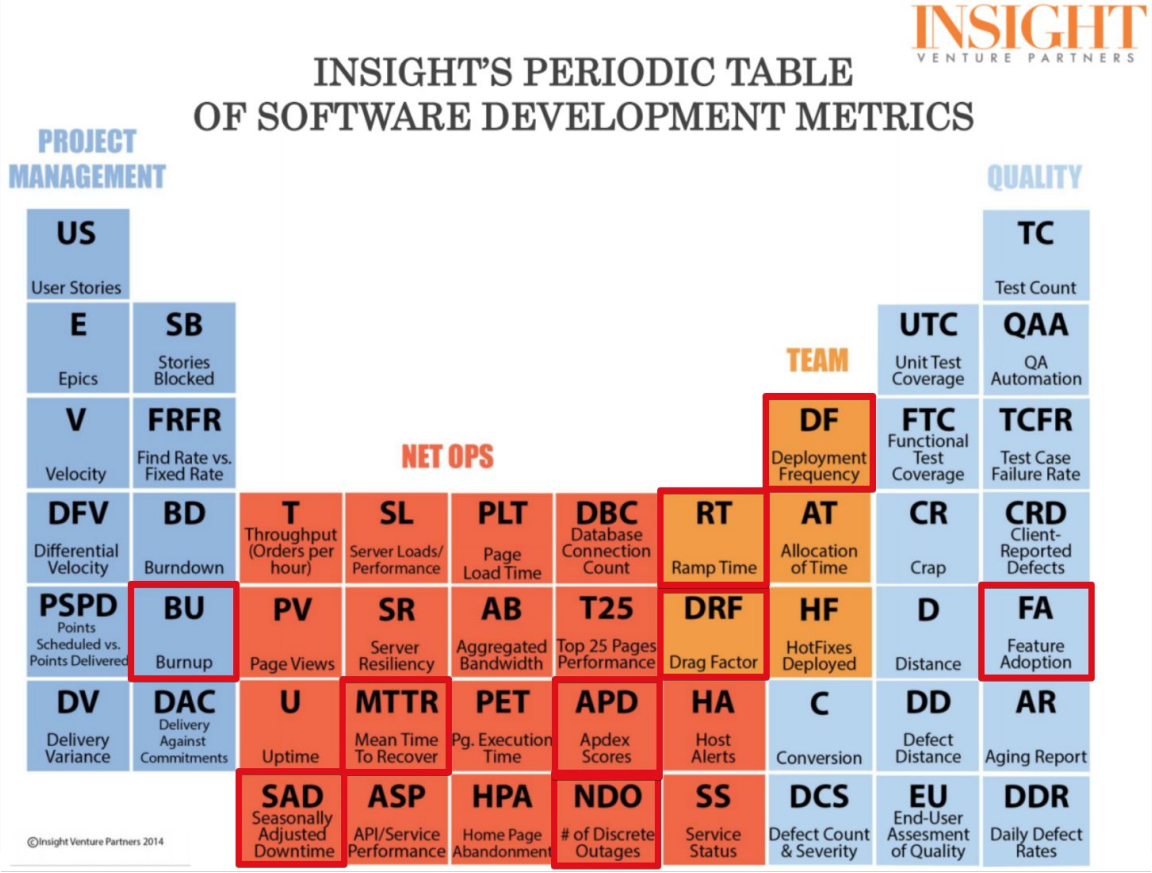
Metrics that make sense

1. Tied to specific economic benefits and hard to game

2. High level enough to allow comparisons

3. Agreeable on their usefulness

Metrics, metrics, metrics



Team Onboarding/Training KPIs: RT

Associated GitHub Value Proposition: Community, Time to Delight

Description

RT: Ramp Time - Length of time required for a new developer to push code to production for the first time.

Economic Impact

The normal onboarding time varies from company to company, but typically consists between 2 weeks and 8 weeks. With a previous familiarity of GitHub and its workflows, it can typically be cut in half, saving 2 to 4 weeks of on boarding time per developer. Our evidence from previous customers even suggests that only 30 hours are needed in average to get a developer fully up to speed with GitHub's work flows and tooling.

```
<number developers on boarded / switching platforms > / 52 weeks * <number of weeks saved> * #average dev salary
```

Training on proprietary version control, collaboration and issue tracking system typically requires a dedicated trainer (team). With GitHub, only minimal training is required which is freely available and can be done in a self-paced, self service manner:

```
<number developers needing training> / <class size> * <average cost of training>
```

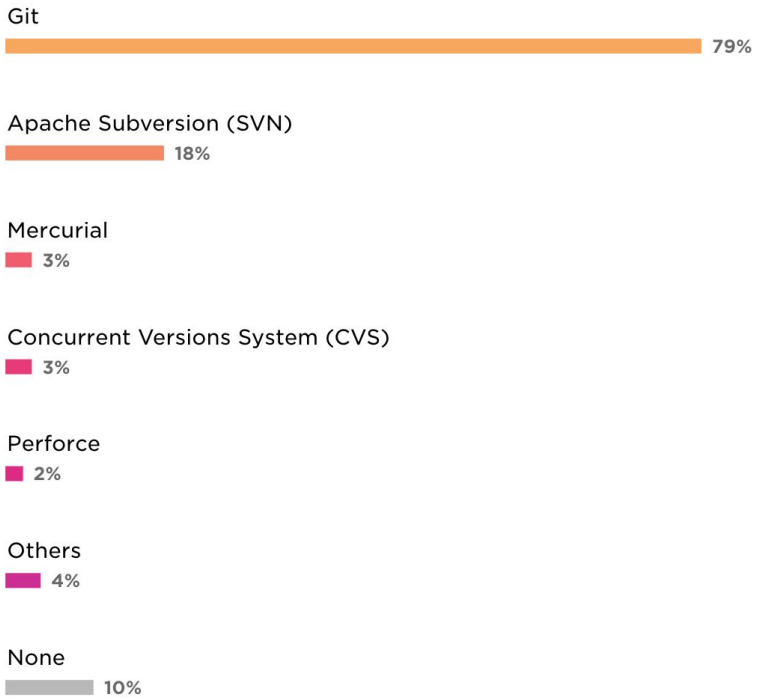
How GitHub can help

Legacy vs modern tools used in Inner Source

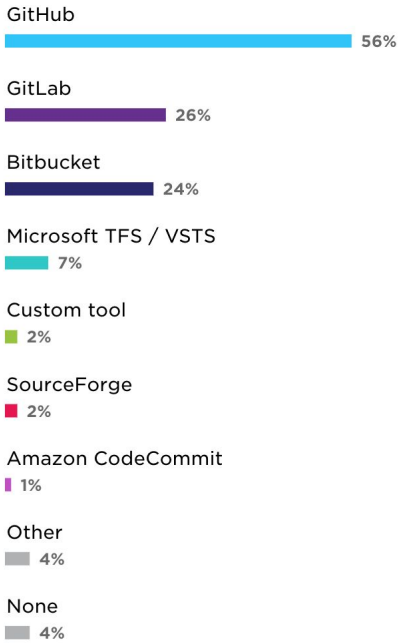
VS

JetBrain Developer Surveys 2017 & 2018

Which Version Control systems



Which Version Control Services do you regularly use, if any?



https://www.jetbrains.com/research/devecosystem-2017/team-tools/

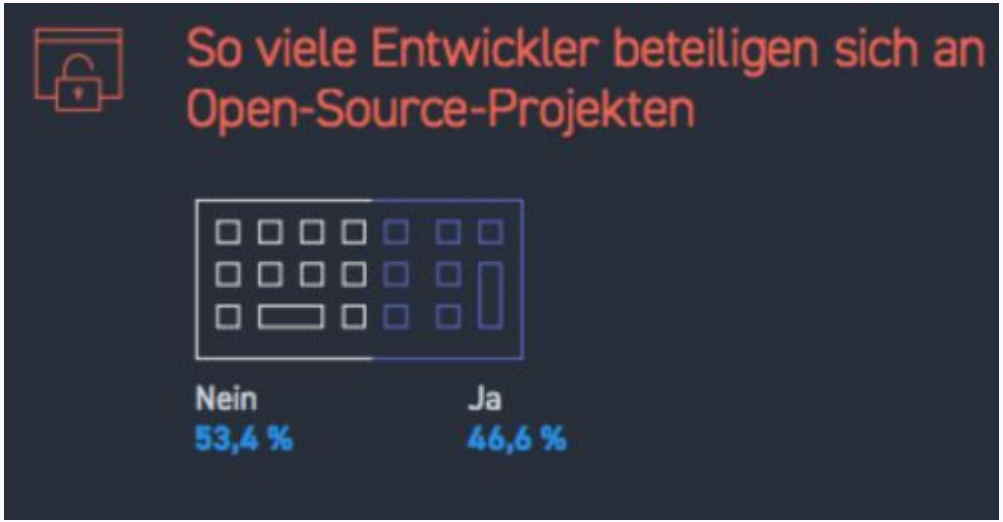
German results from StackOverflow Survey

STACK OVERFLOW ENTWICKLERUMFRAGE 2018

Prioritäten bei Arbeitgeberwahl

- Tools
- Team
- Gehalt
- Weiterentwicklung

StackOverflow Survey (German subset, 6k devs)



https://insights.stackoverflow.com/survey/2018/#work-how-do-developers-assess-potential-jobs

Developer Happiness

Team Talent Attraction KPIs: DH

Associated GitHub Value Proposition: Community, Invention

Description

DH - Developer Happiness: Employee Net Promoter Score - Based on a scale of zero to ten, how likely is it you would recommend this company as a place to work?

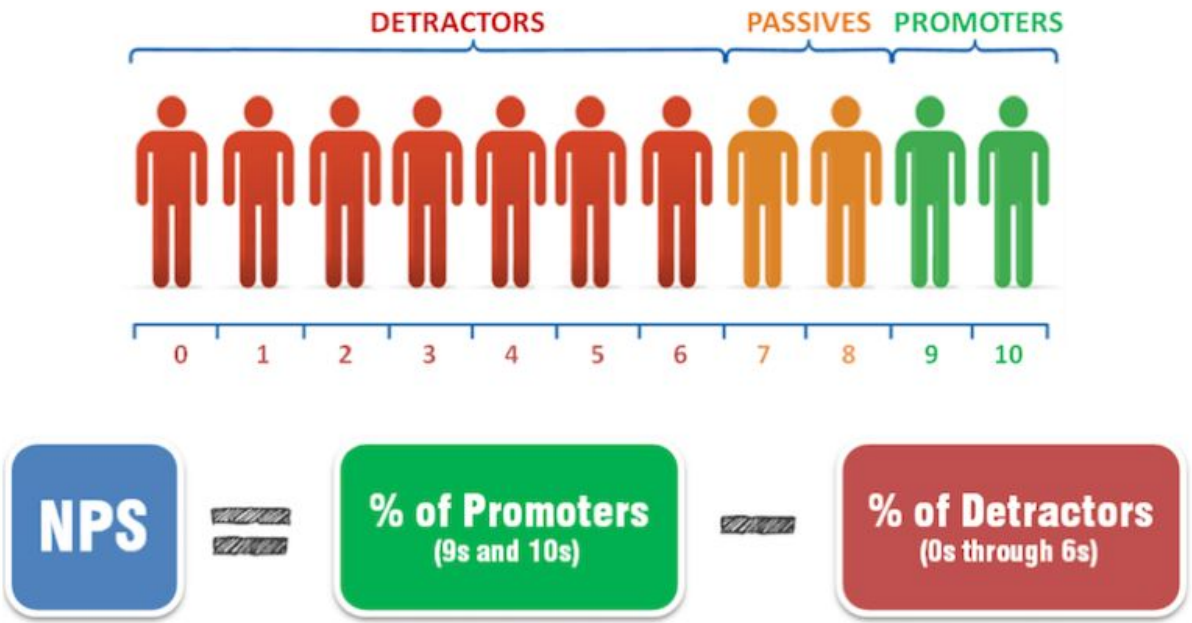
Economic impact

Attracting and retaining top talent is probably the most important objective of any engineering organization as more and more companies see their main ability to compete in the quality and speed of their IT based innovations. The costs for rehiring, retraining and lost business opportunities because of the missing capabilities to execute often add up to 3 to 6 months of salary for the position.

```
attrition rate * <average monthly salary of a developer> * <number developer> * 6 = costs related with low developer happiness
```

How GitHub can help

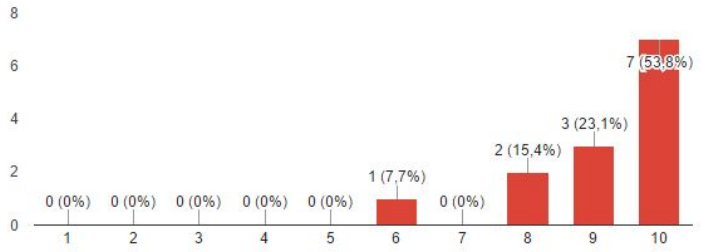
Developer Happiness



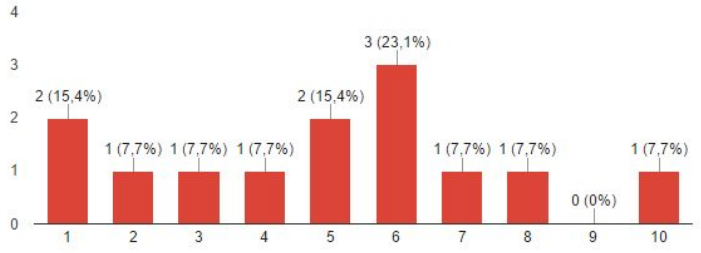
 https://www.employeeconnect.com/blog/complete-guide-employee-net-promoter-score-enps/

The voice of the people

On a scale of 1-10, how likely is it that you would recommend Exact as an employer, if GitHub would be the source control system for everyone?
(13 reacties)



On a scale of 1-10, how likely is it that you would recommend Exact as an employer, if TFS would be the source control system for everyone?
(13 reacties)



GitHub is used by 40 million developers and trusted by more than 100,000 organizations.

Easier onboarding and easier to attract talent.

= exact

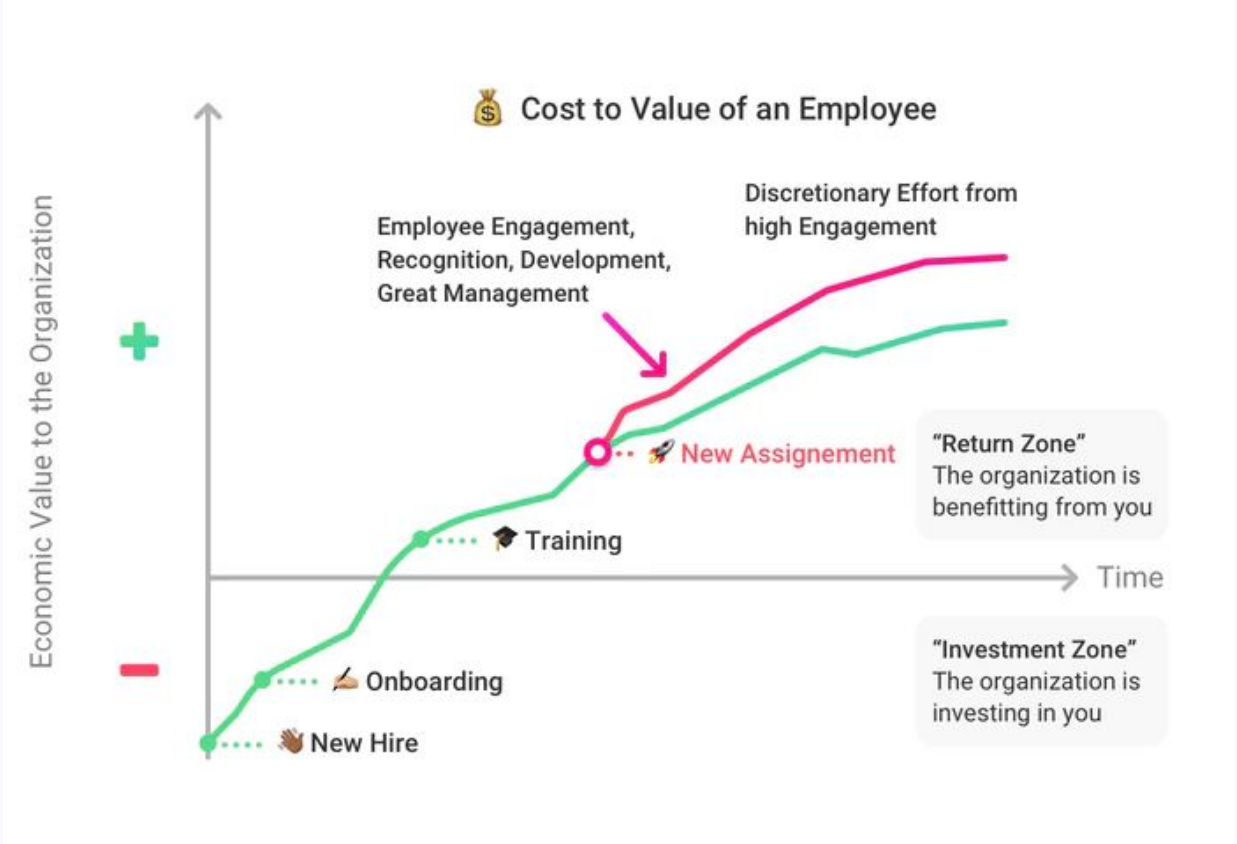
Economic impact on employee retention

As an example, if you are a **150 person company with 11% annual turnover**, and you spend $25k on per person on hiring, $10k on each of turnover and development, and lose $50k of productivity opportunity cost on average when refilling a role, then your annual **cost of turnover** would be about **$1.57 million**.

Reducing this by just **20%**, for example, **would** immediately **yield over $300k** in value.

https://lattice.com/blog/how-much-does-employee-turnover-really-cost/

Economic impact on employee retention



https://lattice.com/blog/how-much-does-employee-turnover-really-cost/

What have those people in common?

besessener,
basejumpa,
randomByte,
marti4ka,
larsxschneider,
drivingThrillhouse,
SWCraftsMan,
moltob, zemunk ,
matz3, sengaya,
dannola, tacurran

They like to show their contributions to the world

Thomas Aidan Curran
tacurran

Software developer

Unfollow

Block or report user

Software developer
Berlin, Germany
mail@thomasaidancurran.com

Overview | Repositories 19 | Projects 0 | Stars 22 | Followers 47 | Following 32

Popular repositories

workshop-dbg
Forked from aeneasr/workshop-dbg
Go ★1 1

DB-Story
JavaScript ★1

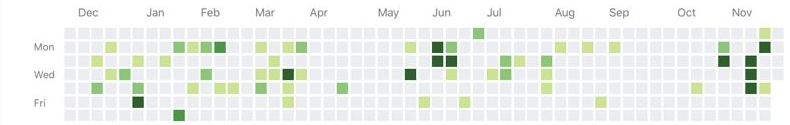
180612-cebitgithubfuture
Three ways
★1

derrick
Derrick Metadata Management System

flynn
Forked from flynn/flynn
A next generation open source platform as a service (PaaS)
Go

book
Forked from GoBootcamp/book
Source code of the companion book/website
TeX

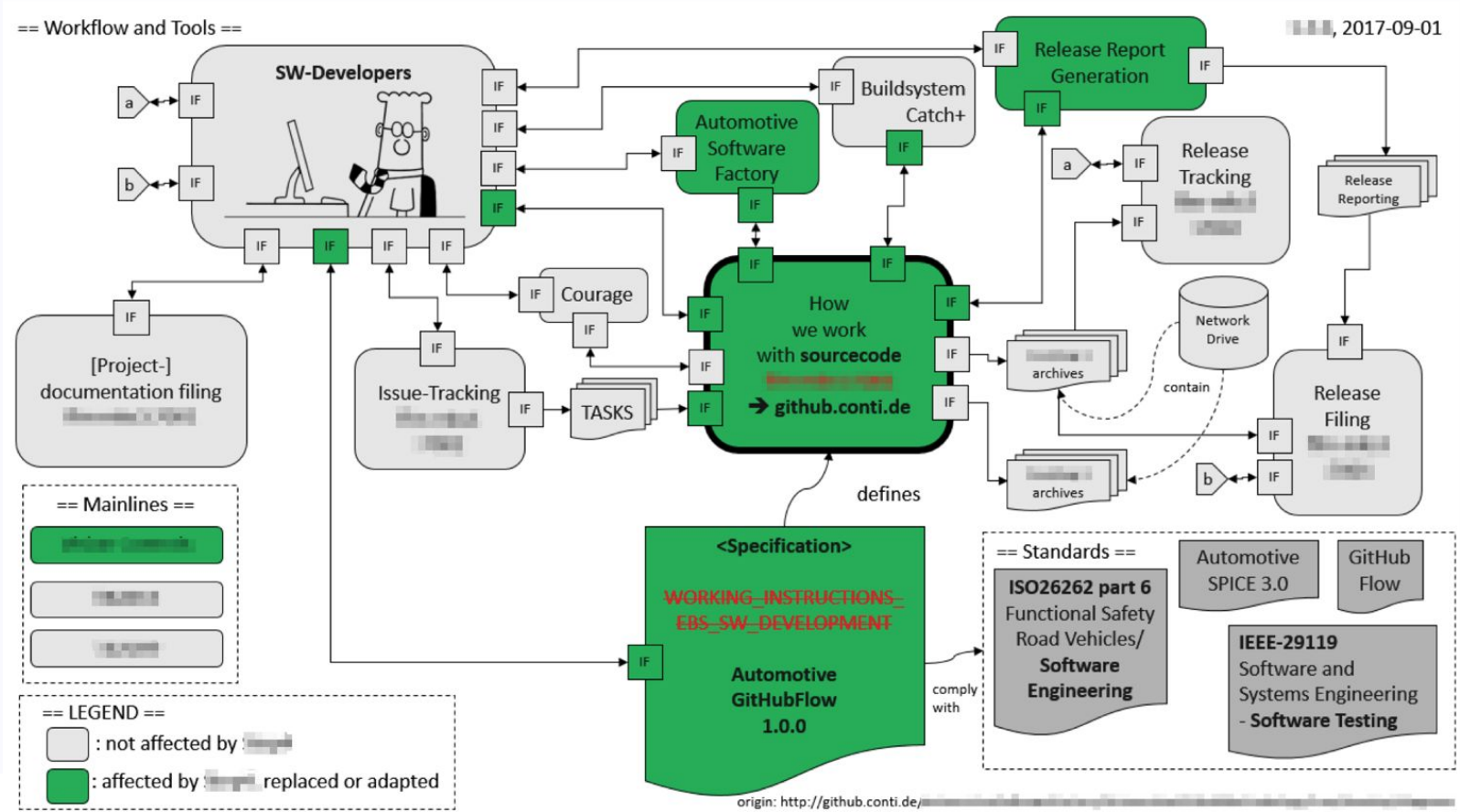
142 contributions in the last year



They are team leads and CxOs of fortune 500

https://github.com/customer-stories/societe-generale

Passion to use, create & teach modern tools

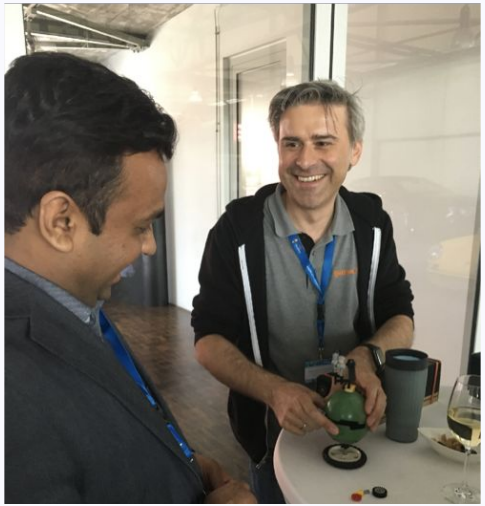


Passion to use, create & teach modern tools

Passion to create, use & teach modern tools

Continental's Build Indicator

Recruiting



Team Delivery KPIs: DF, HF

Associated GitHub Value Proposition: Invention, Collaboration

Description

DF: Deployment Frequency- number of times per day/week/month that a team deploys.

HF: Hot Fixes - Count of HotFixes over a given interval

Economic impact

The number of times a team can deploy (either to production or an internal system), correlates to how fast a team can

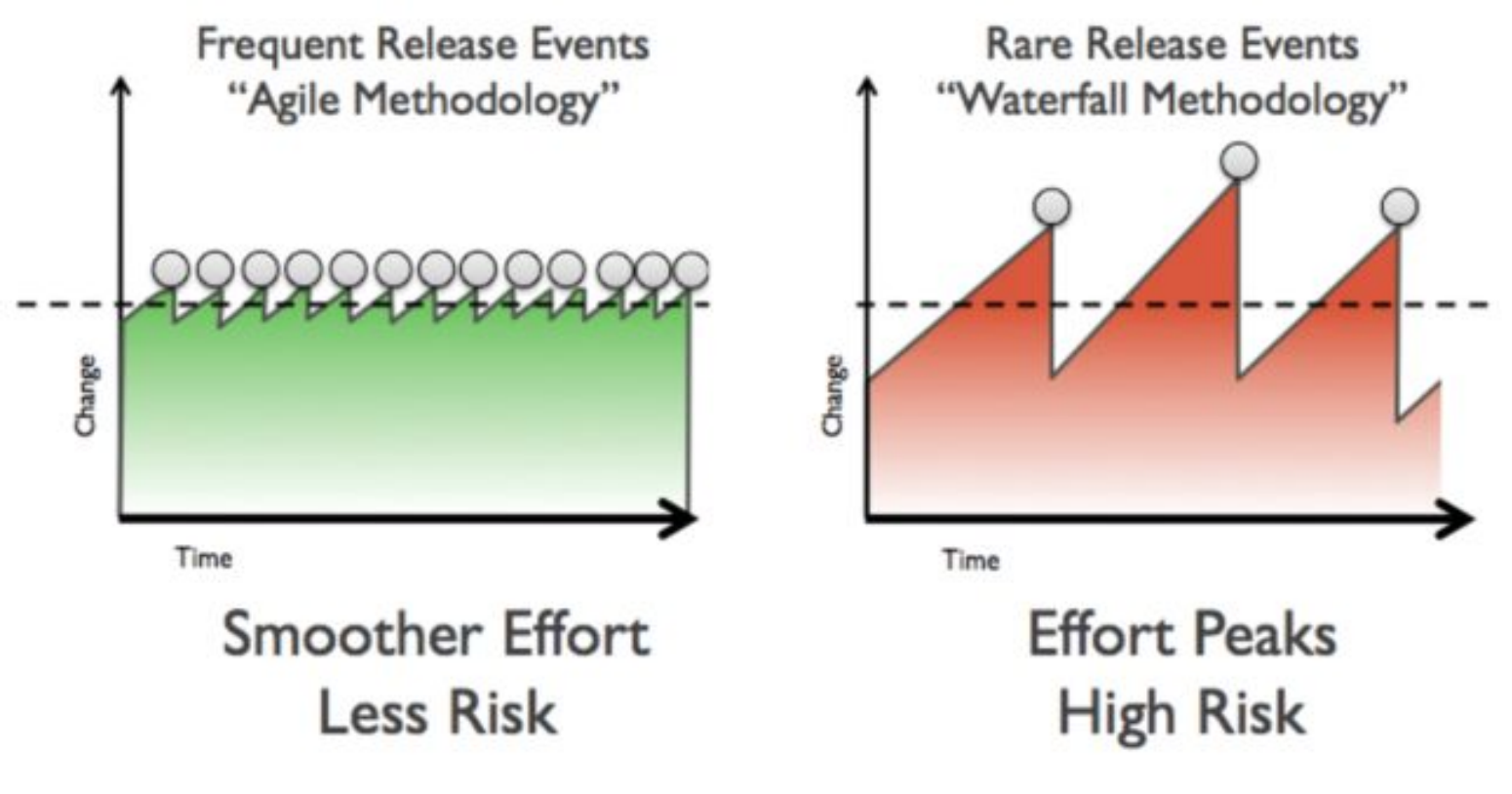
- fix an issue in production (reducing risk, HF)
- change course based on customer feedback (validated learning, create the right product)
- deliver new features to the customer (time to market)
- execute a deployment fearlessly

The shorter it takes to go through the cycle from the initial scheduling of work to actual deployment, the higher the deployment frequency. To shorten the cycle time without compromising quality and audit compliance

Agile Manifesto

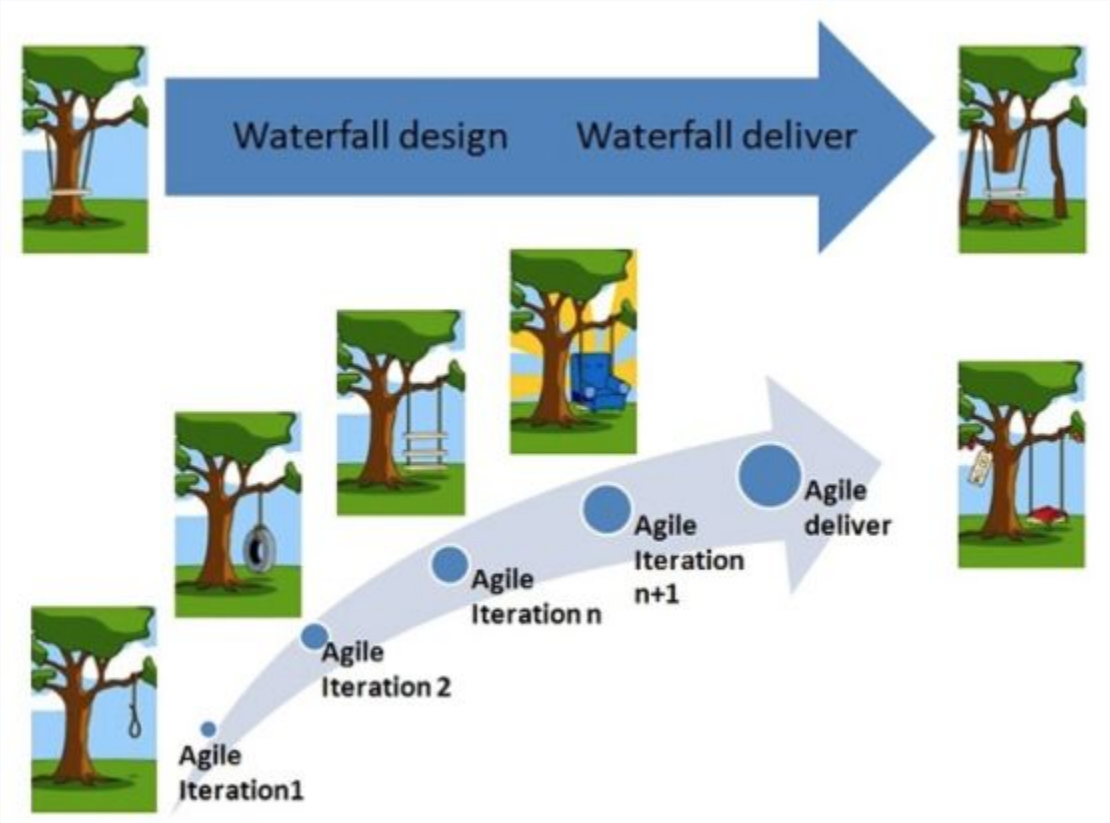
> *Our highest priority is to satisfy the* ***customer*** *through early and* ***continuous delivery*** *of* ***valuable software***

Why deployment frequency does matter



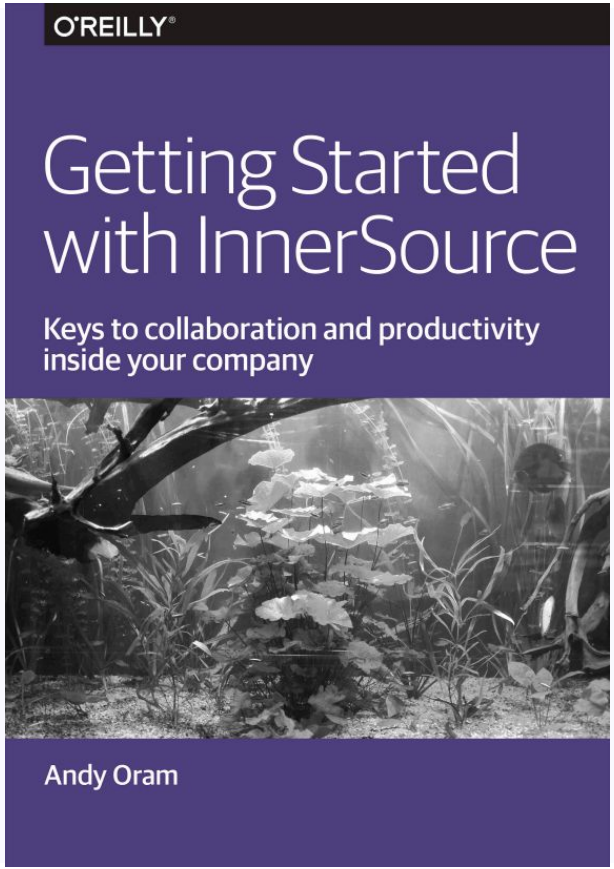
http://en.wikipedia.org/wiki/File:Agile-vs-iterative-flow.jpg

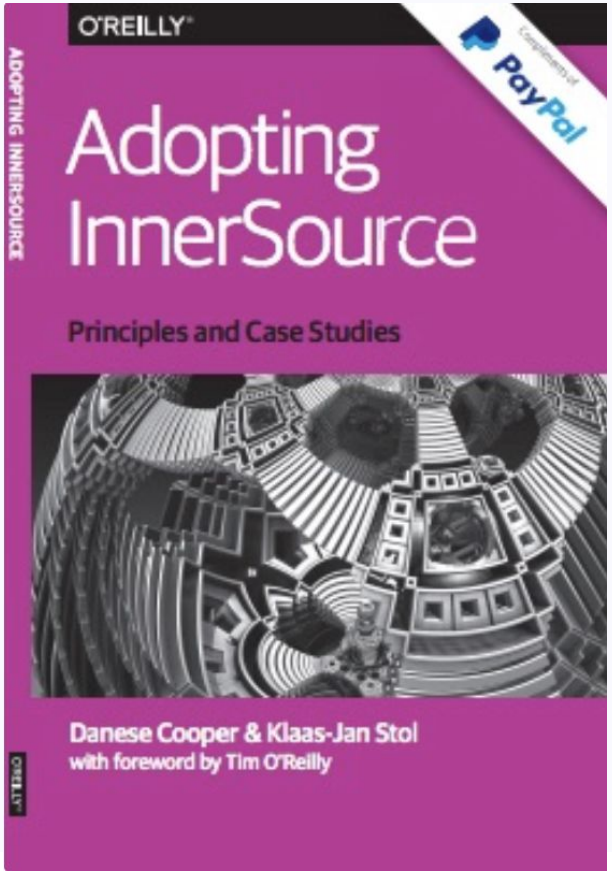
Why deployment frequency does matter



https://www.codeproject.com/Articles/1064114/Agile-Software-Development-Basics?msg=5177236

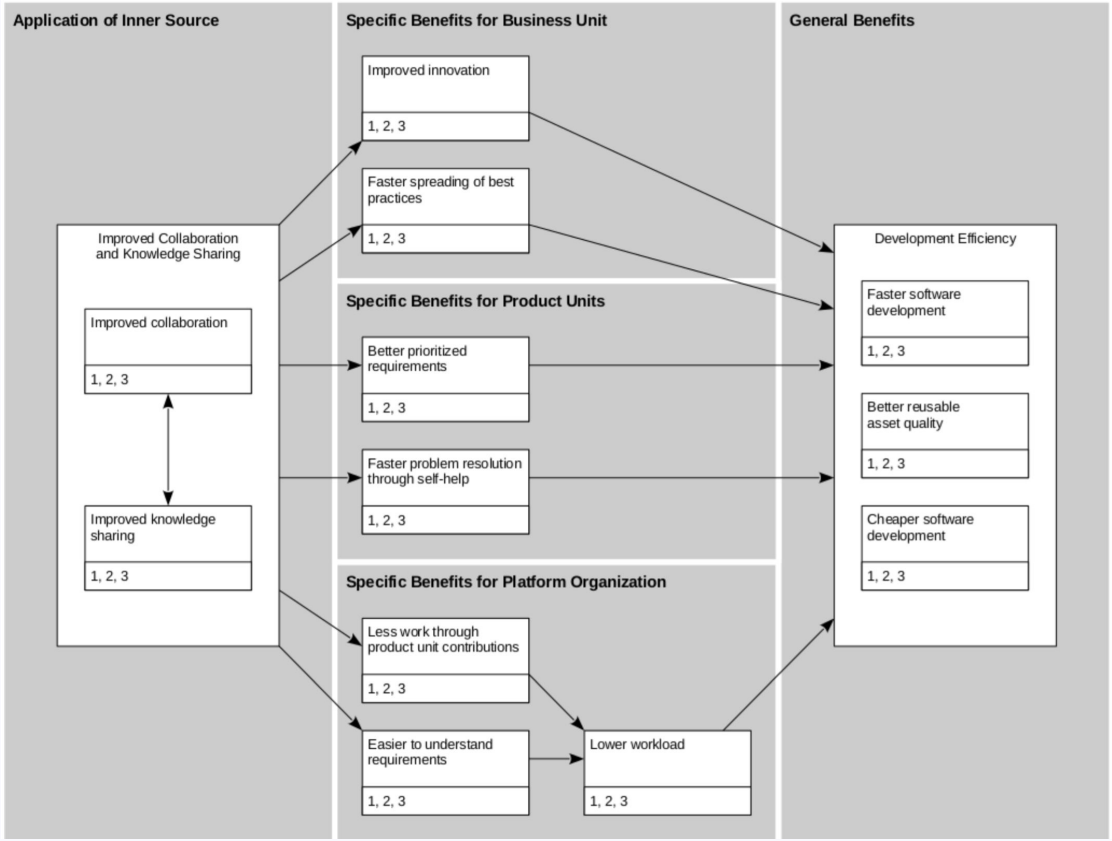
Inner Source





https://innersourcecommons.org/resources/books/adoptinginnersource/

Scientific validation



 Source: http://dirkriehle.com/uploads/2015/04/TR-CS-2015-02-Inner-Source-Full-Web.pdf

Inner Source Practitioners DACH

How does Inner Source work?

Visibility: all internal software projects are by default visible to all employees

Fork/Branch: anyone can create a copy of a project to make changes freely

Pull Requests: people outside a project team can suggest changes and contribute to the project

Testing/CI: every proposed change is automatically tested and the result is shown in the pull request

Issues: there is a public issue tracker in which everyone can submit a bug or ask a question

Documentation: all software projects include a Readme that describes what the software does, how to run it and how to contribute to it.

Benefits of Inner Sourcing

Cross-team collaboration

Increased shipping velocity

Improved code quality

Worldwide scalability

Developer happiness

Transparency

Common challenges for Inner Source

- Culture change- It won't happen overnight and needs buy in on all levels

- **Manager Fears**
  - **Fear of losing control (best developers contribute to other projects)**
  - **Fear of missing performance goals because of locally optimized metrics**

- Developer Fears
  - Fear of doing development with the entire company watching
  - Fear of follow up / maintenance work / reviewing all day

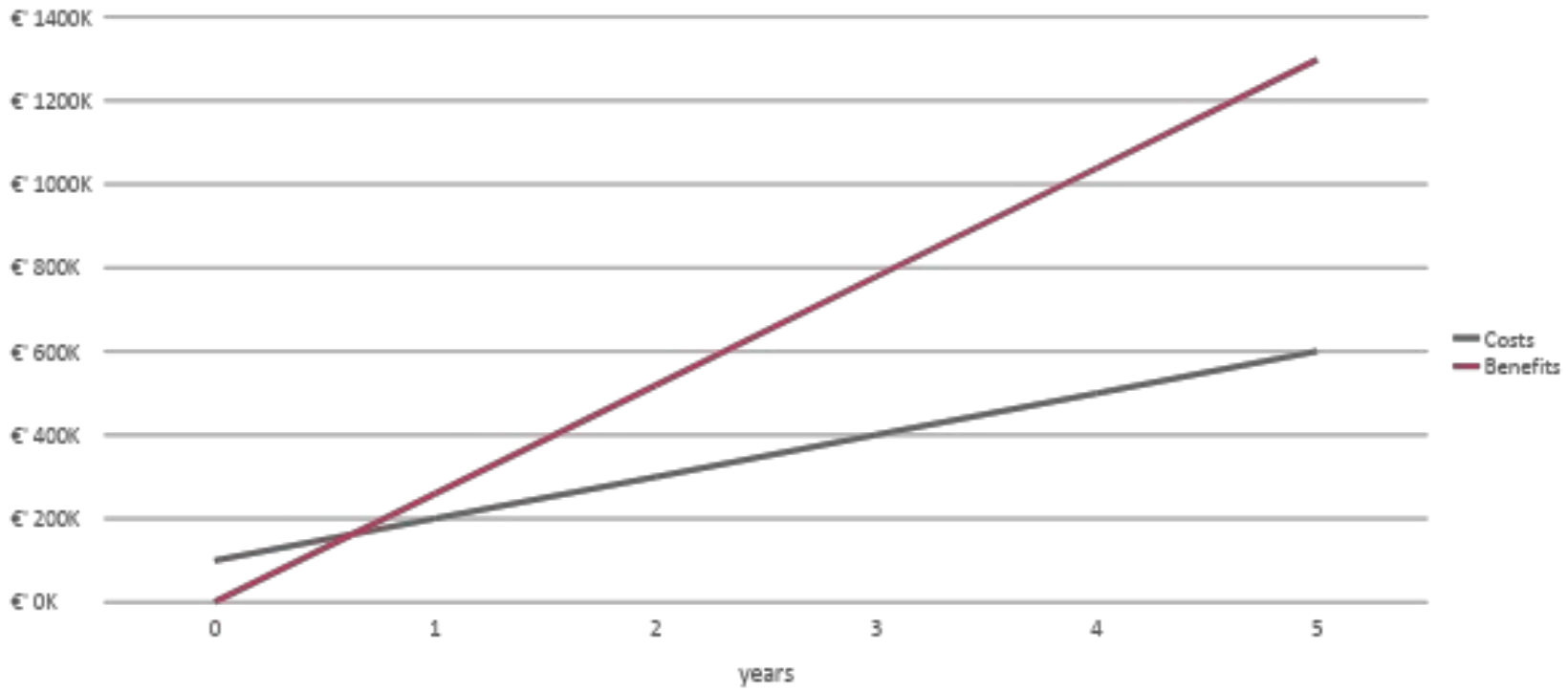
- Legal - Sharing information and contributing to other projects has to be ok

- Architecture - The more modular, less coupled the software, the better
- Test automation - Investments into fully automating have to be made - agile practices help

Inner source is not about arbitrary contributions

> *Inner source is **NOT** about contributing to all kind of internal projects **just because you can** do so, **but because** your team is working on **cross-functional features** that need to change components out of your core team's responsibility.*
>
> *In a world **before Inner source**, you would have to **wait for many product owners** of other components to prioritise your request which may take **forever** or might never happen. **Now**, our teams can **propose the needed changes themselves** and most likely **get them merged** if they follow the contribution guidelines which are a mandatory part of every inner source project.*
>
> *The best argument for Inner source is that **other teams** are motivated to **make your own software better** and less work stays on your plate.*
>
> *Inner Source Evangelists at Zalando*

Efficiency gains led to a convincing business case



 © 2017 EXACT

Summary

Now you know how to measure fun

Joe Duffy
@xjoeduffyx

Follow

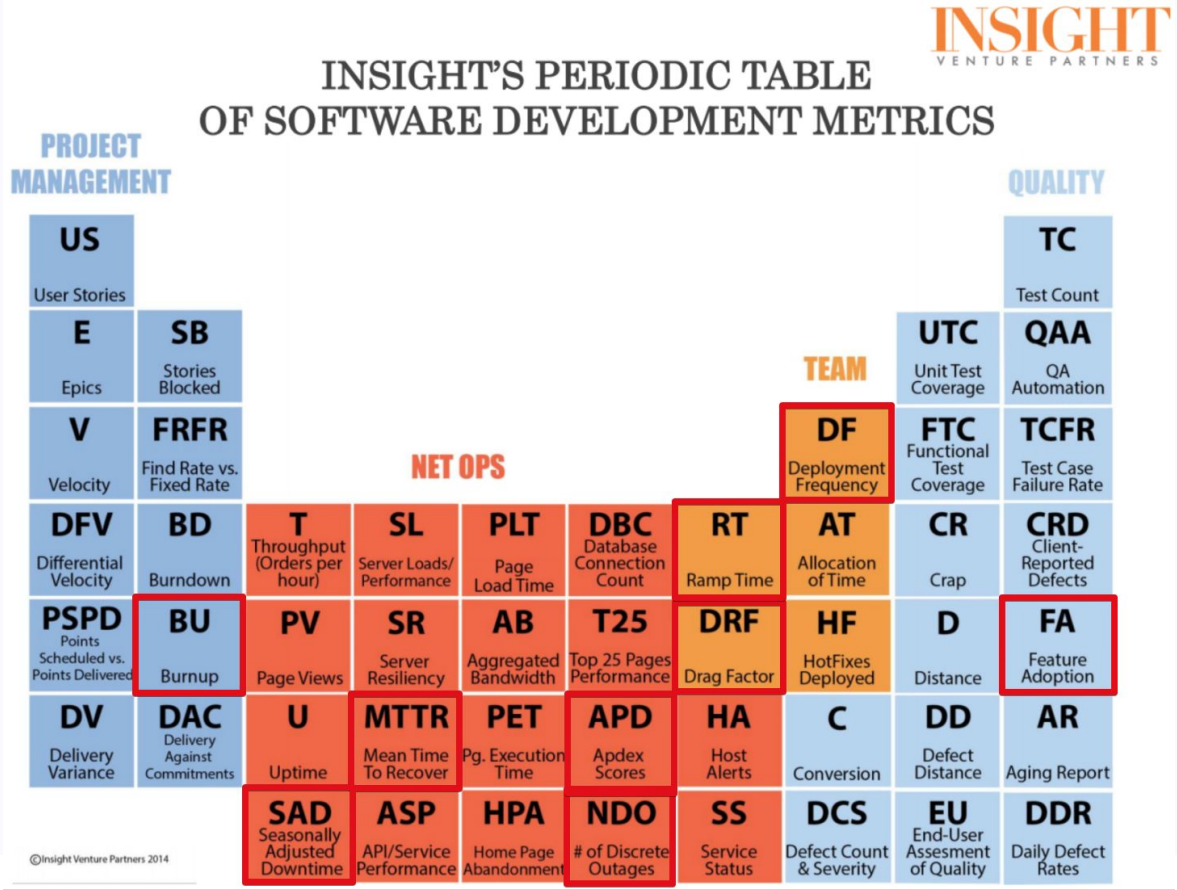
Having watched many teams at @Microsoft move to @github, I can honestly say that social coding often delivers 10x more productivity and fun.

RETWEETS 50 LIKES 89

8:19 PM - 8 Jun 2016

50 89

Use the same metrics as the VCs



Benefits of Open and Inner Sourcing

Cross-team collaboration

Increased shipping velocity

Improved code quality

Worldwide scalability

Developer happiness

Transparency

Questions