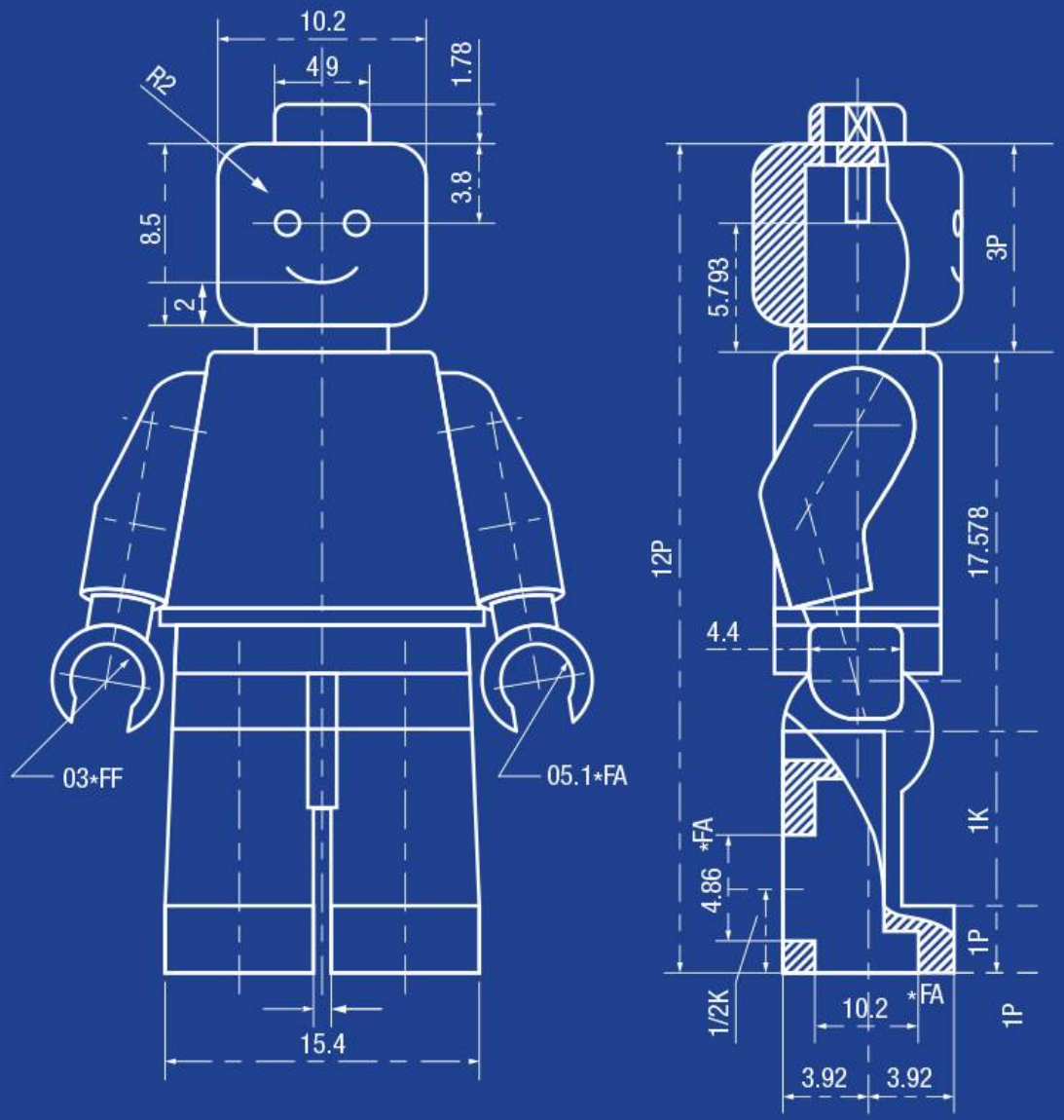
Open Source Software Requirements Engineering and Product Managment

ETSN15 - JOHAN LINÅKER, PHD


10.2
4.9
1.78
R2
3.8
8.5
2
03*FF
05.1*FA
15.4
12P
5.793
3P
17.578
4.4
1K
*FA
4.86
1P
1/2K
10.2
*FA
1P
3.92
3.92

About Johan

What is Open Source Software?

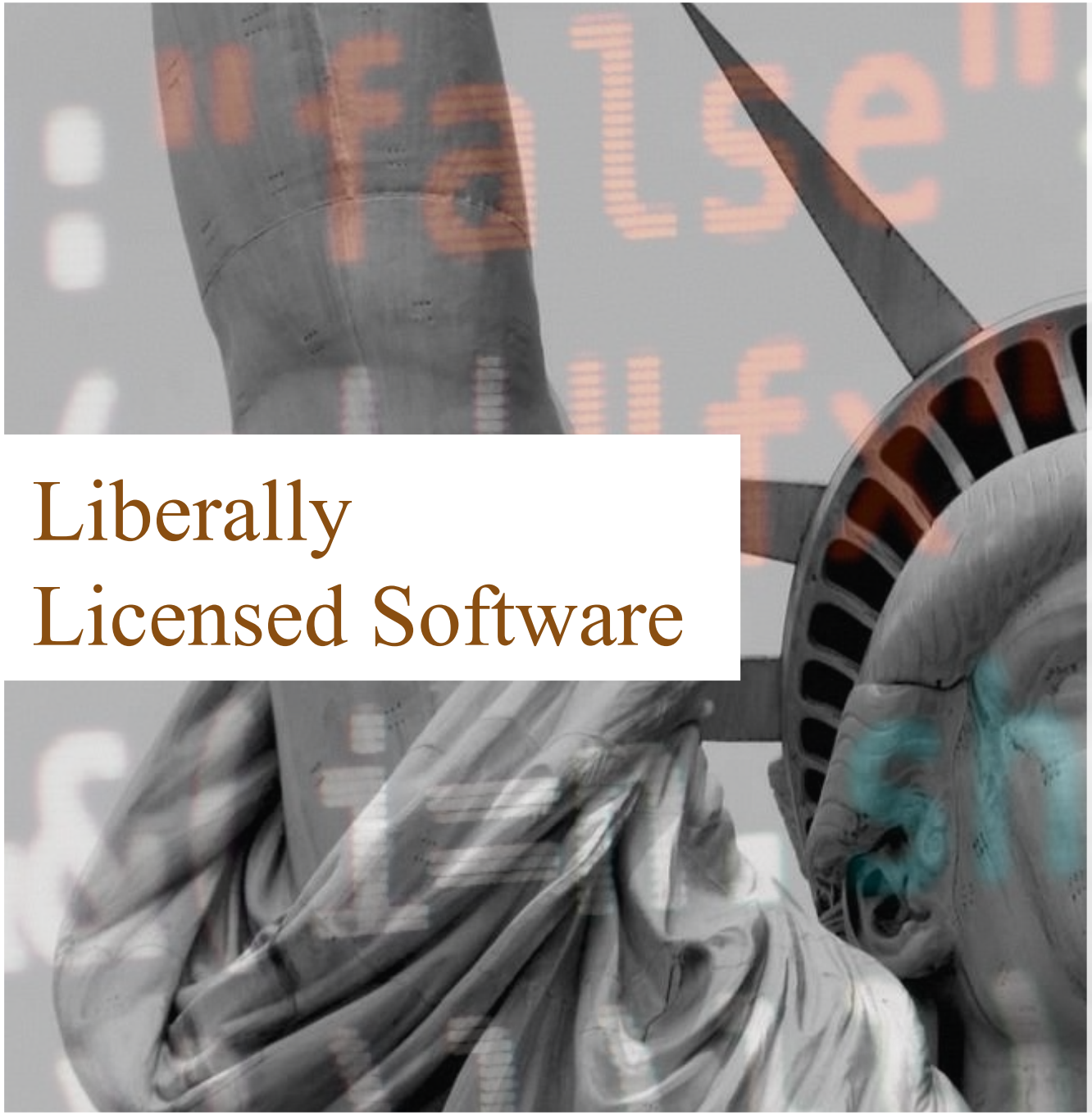
Liberally Licensed, Collaboratively Developed Software*

* https://medium.com/@stephenrwalli/there-is-no-open-source-business-model-cdc4cc20238

Liberally Licensed, Collaboratively Developed **Software***

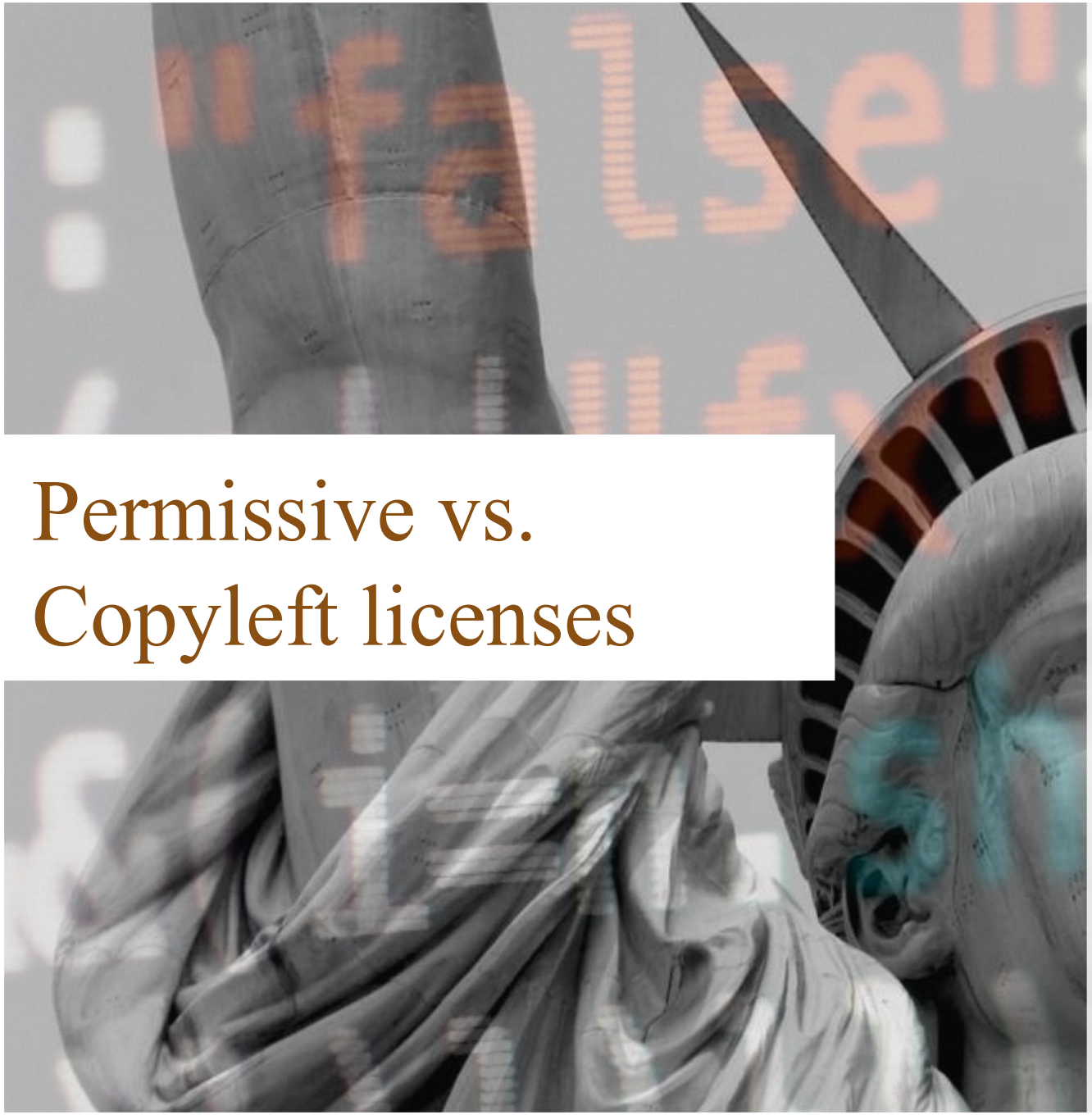
* https://medium.com/@stephenrwalli/there-is-no-open-source-business-model-cdc4cc20238


Liberally
Licensed Software



Liberally Licensed Software

- Software available under an open source license
  - License that follows the Open Source Definition and is approved by the Open Source Initiative (http://opensource.org)
- Software which you may inspect, use, modify and redistribute
- As long as you don't discriminate against any user och use-case
- Different conditions apply per license requirements
- Copyright holder still holds its copyright



Permissive vs. Copyleft licenses

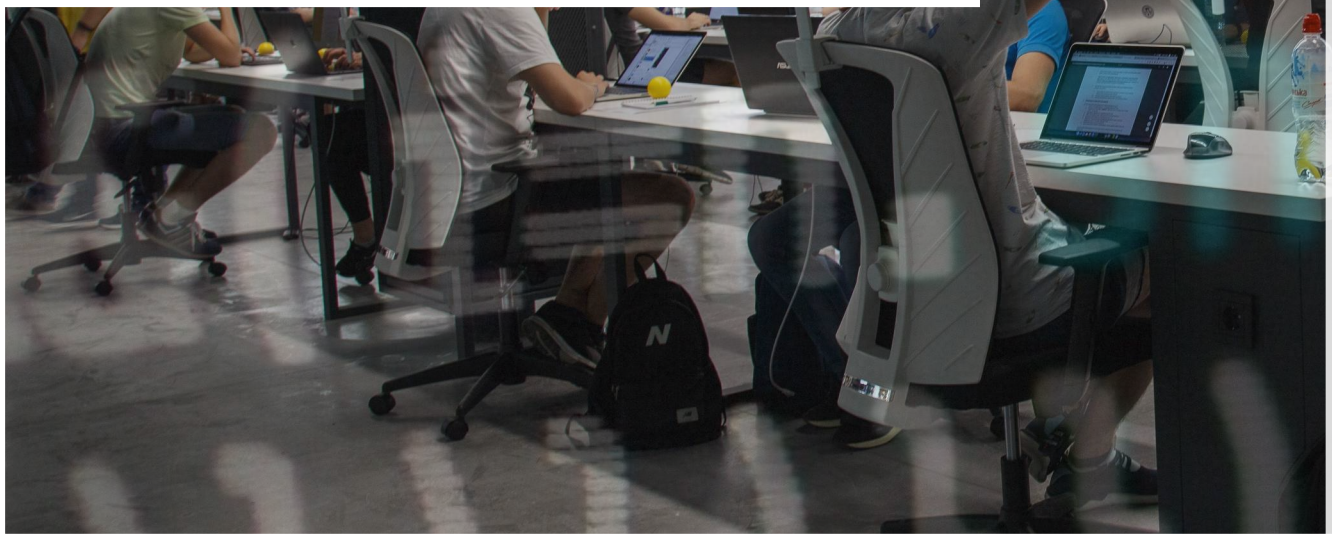
- Permissive licenses – do whatever you want, as long as you recognize the copyright holder
  - E.g., MIT, BSD, Apache
- Copyleft licenses – Above + share any modifications, additions and connecting code under same license.
  - GPL 2, GPL 3, AGPL
- Permissive common for standardiszations and collaboration on non-differentiating software
- Copyleft common when copyright holder wants to capture value back

Liberally Licensed, **Collaboratively Developed Software***

* https://medium.com/@stephenrwalli/there-is-no-open-source-business-model-cdc4cc20238

Collaboratively Developed Software

Collaboratively Developed Software

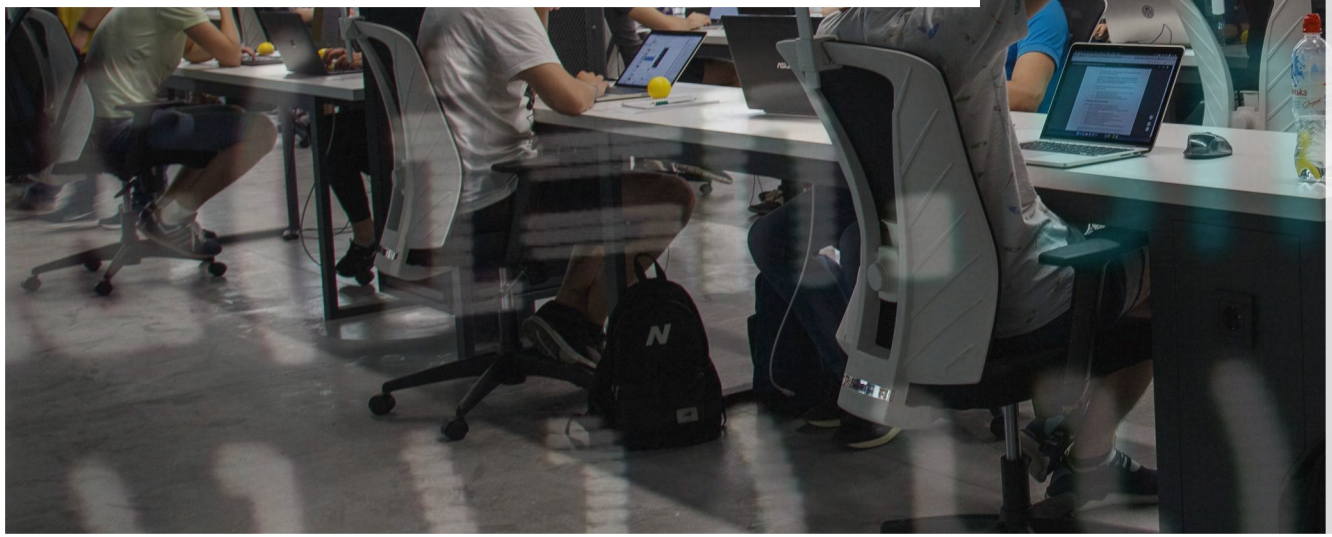


- Software developed as projects by networks of individuals and organizations, aka. **Open Source Communities**
- ”Members” of the community commonly both users and developers
- Are united by a common vision and goal around the Open Source Software.

Incentives, benefits and opportunities

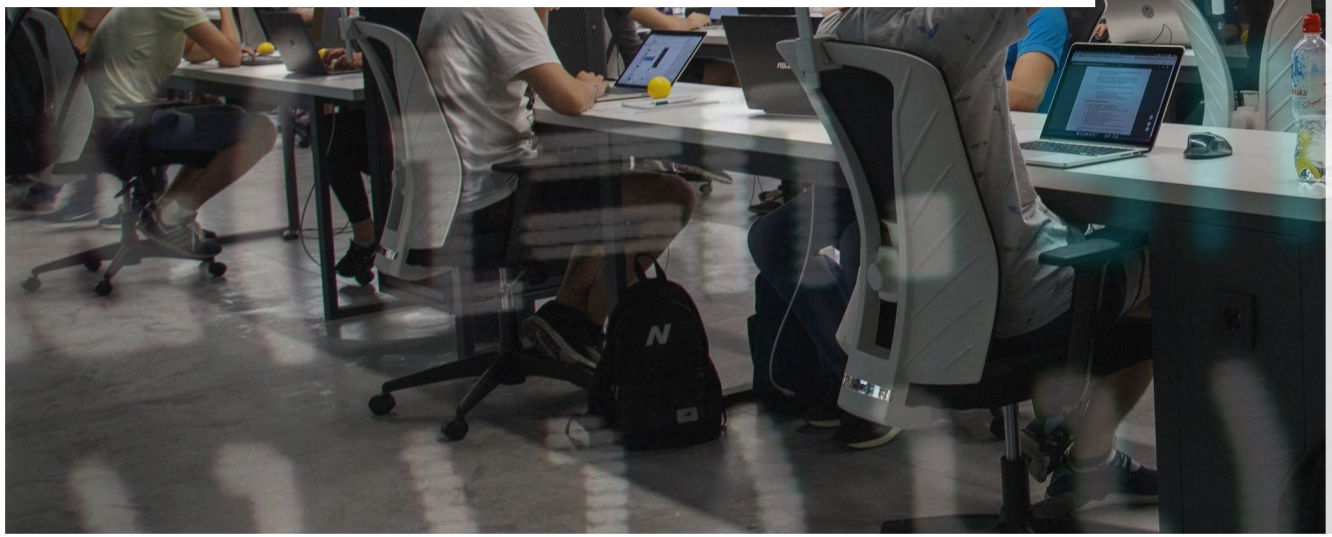
- Incentives may differentiate
  - Individuals:
    - Sense of belonging,
    - recognition for contributions,
    - solves paintpoint,
    - build CV
  - Companies:
    - Lower costs,
    - increased innovation,
    - branding and PR,
    - strategic tool

Costs, risks and complexities



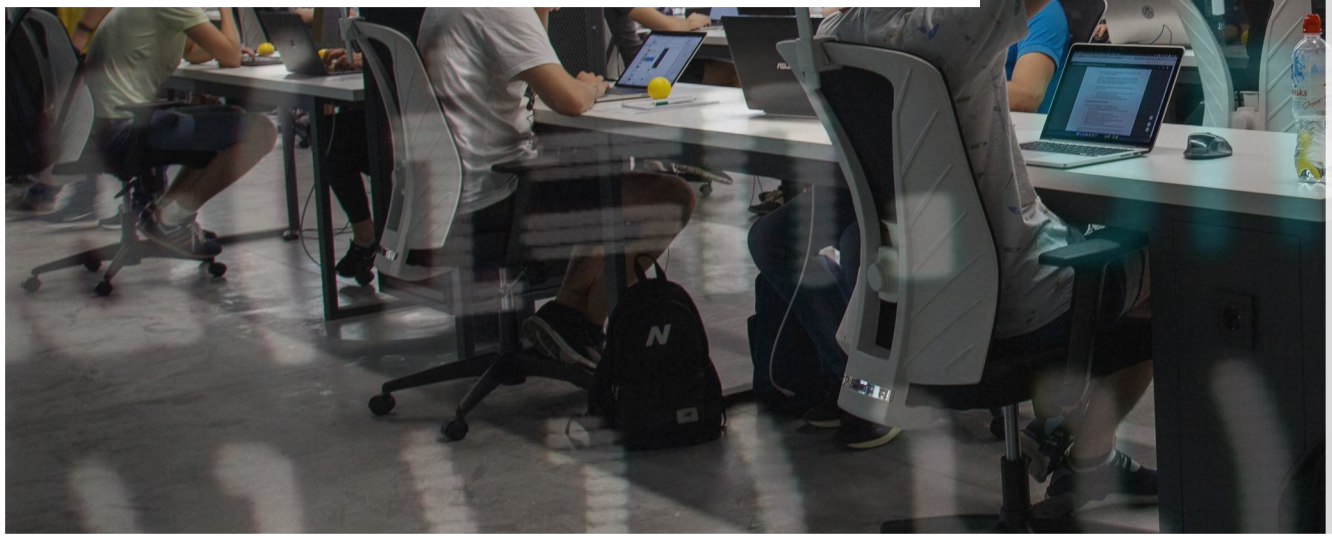
- Connection to business model and internal operations
- Differentiating functionality, competitive edge and commoditization
- Sensitive IPR and pantents
- Internal budget and resource constraints
- Modularity and technical architecture
- Reputation as a good OSS citizen
- Health of OSS community
- Influence on OSS community

Technical and non-technical contributions



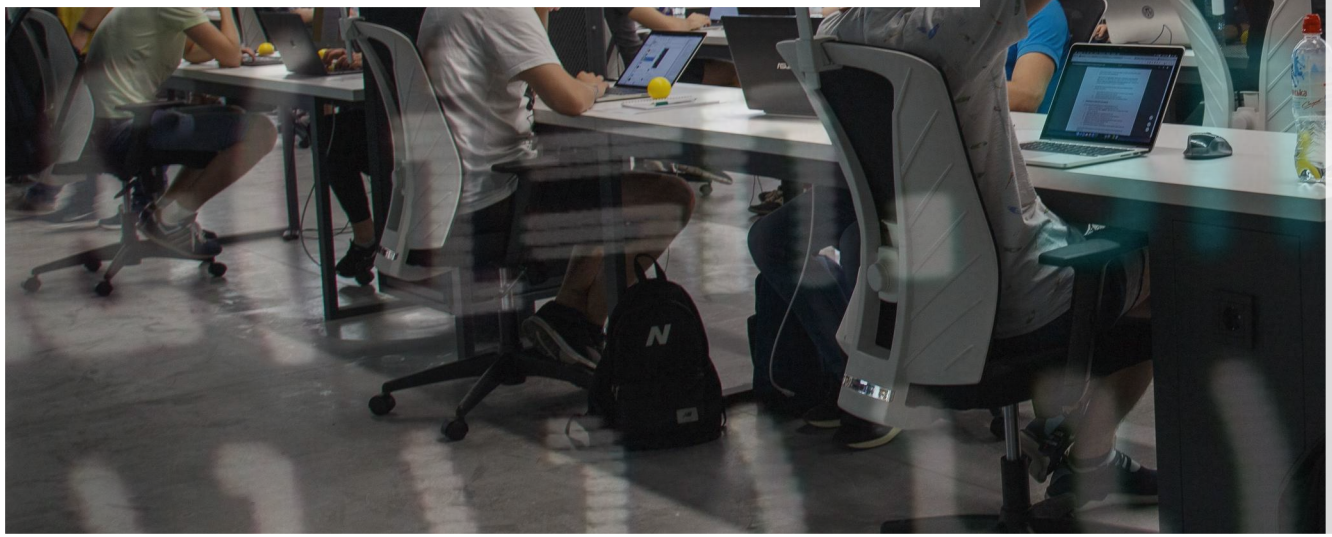
- A contribution can come in many forms…
  - Code contributions (bugfixes, improvements, features, test cases)
  - Design and prioritization input
  - Reporting and triaging of bugs
  - Suggesting improvements and new functionality
  - Writing documentation and manuals
  - Answering questions and joining discussions
  - Arranging events and meetups
  - Writing blog-posts and creating content for social media
  - Offering server space and cloud computing
  - Testing and quality assurance
  - Monetary support

Development process



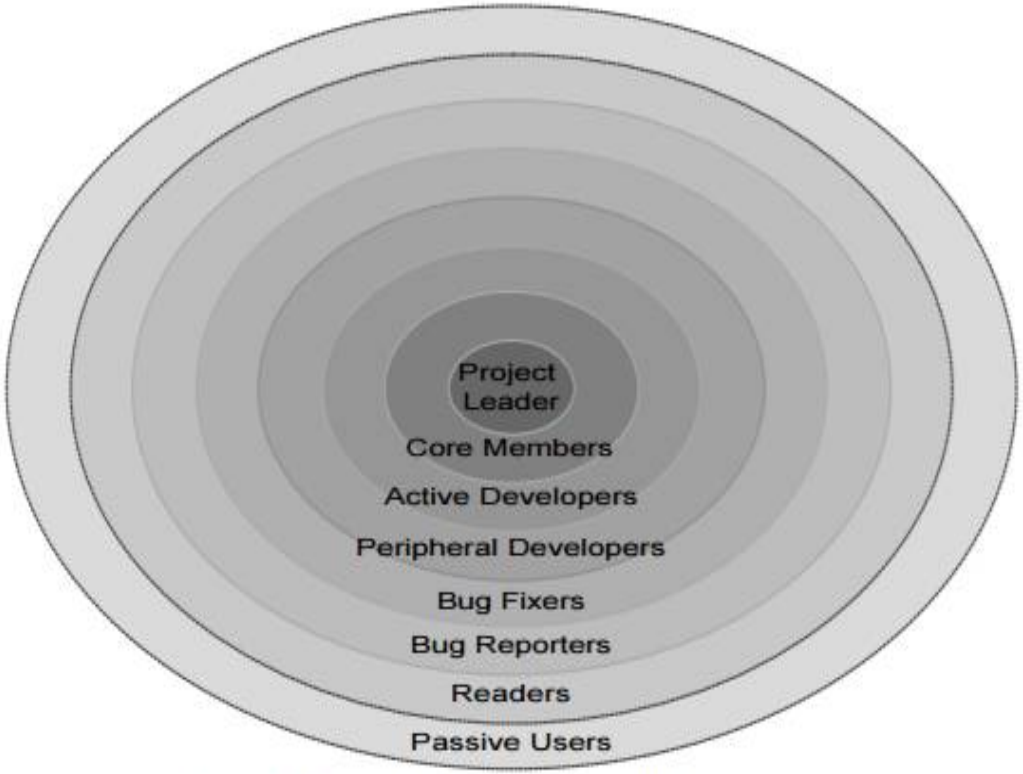
- Informal processes - often specific for each community
- Collaborative development
- Transparent and open discussions on issues (e.g., bugs reports, feature requests) and roadmap
- Stored and available via online and open infrastructure
- Decentralized requirements management
- Community full of stakeholders, all with their own agendas (which may not always align)

Development Infrastructure



- Means of synchronous and asynchronous communication
  - E.g., bulliten boards, mailing-lists, real-time chat
- Ticket system for reporting, discussing and managing issues
  - E.g., Jira, Bugzilla
- Knowlegde base/Wiki e.g., for road map, documentation, and communiy governance
- Source code repository with code review and potentially build functionality

Community Structure and Governance



Community Structure and Governance

Leadership

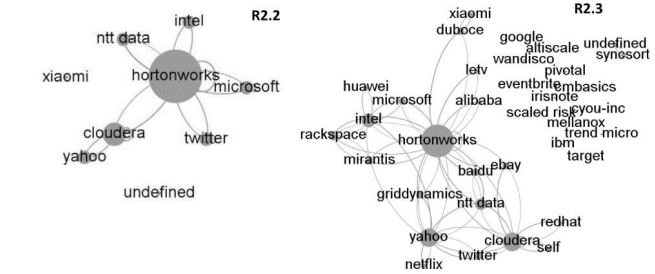
Maintainership Maintainership Maintainership

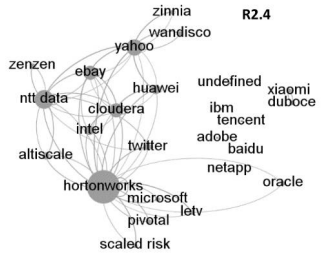
Committers Committers Committers Committers
Committers Committers Committers Committers

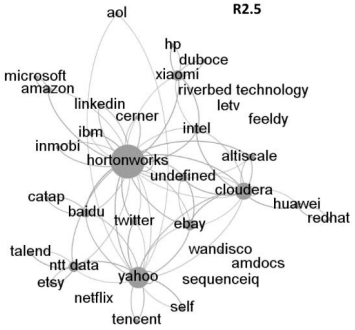
Contributors Contributors Contributors Contributors Contributors
Contributors Contributors Contributors Contributors Contributors

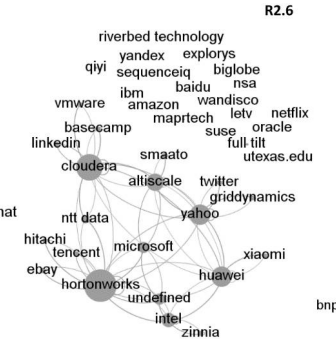
Users Users Users Users Users Users Users Users Users Users
Users Users Users Users Users Users Users Users Users

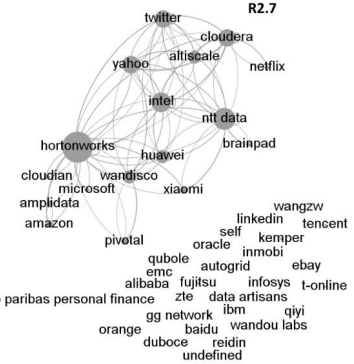
Community constantly evolving







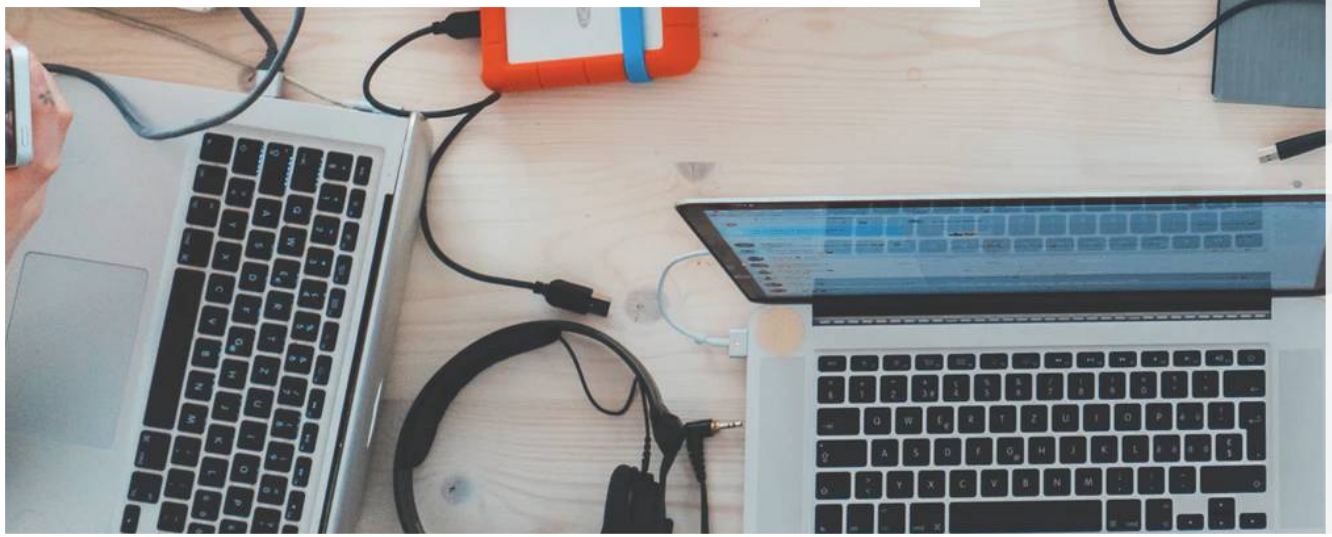




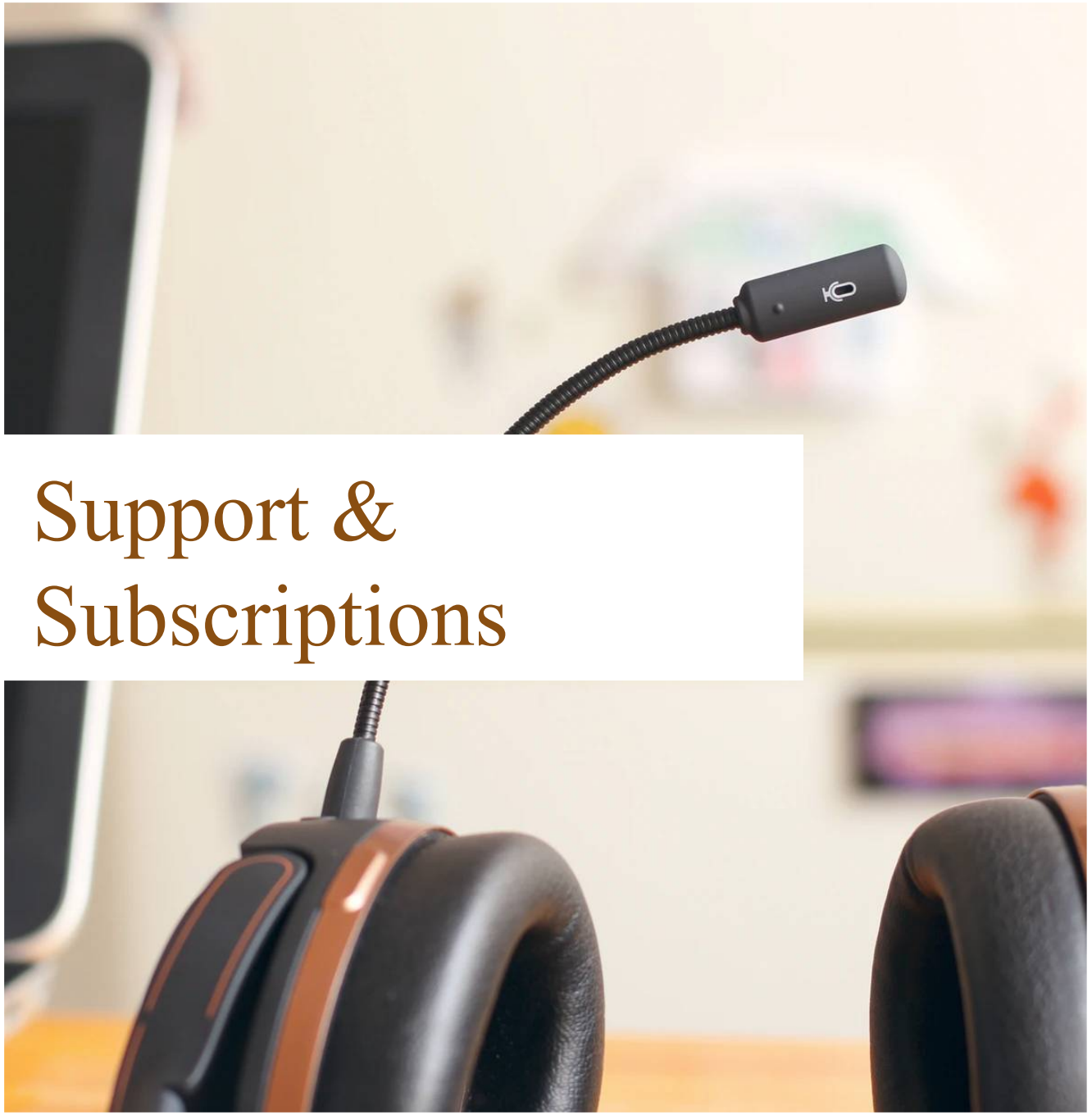
Relationship with community

- Different types of relationships
  - Symbiotic
    - Win-win for both firm and community
    - Contributing to influence projects according to internal agenda and improve health to mitigate security risks
  - Commensalistic
    - Gain for firm, community indifferent
    - Use project and doing ligther contributions. Project in line with internal agenda and healthy with others already supporting it.
  - Parasitic
    - Firm free-riding on community.
    - Using as is not giving anything back. Worst case expecting free work for nothing in return. Looked down on from communities.

Different use cases



- Adapt community engagement based on connection to internal business goals and needs, e.g., connection to business model
- Open Source Business Patterns
  - Support & Subscriptions
  - Open Core & Proprietary Extensions
  - Dual-License
  - X-as-a-Service
  - Data Driver
  - Product Enabler
  - Infrastructure & Development
  - ...



Support & Subscriptions

redhat.

HORTONWORKS®
POWERING THE FUTURE OF DATA™

CANONICAL

Open Core & Prop. extensions

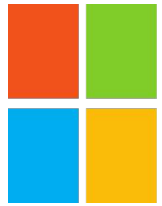
docker

GitHub

cloudera®

Dual-Lience

X-as-a-Service



Microsoft

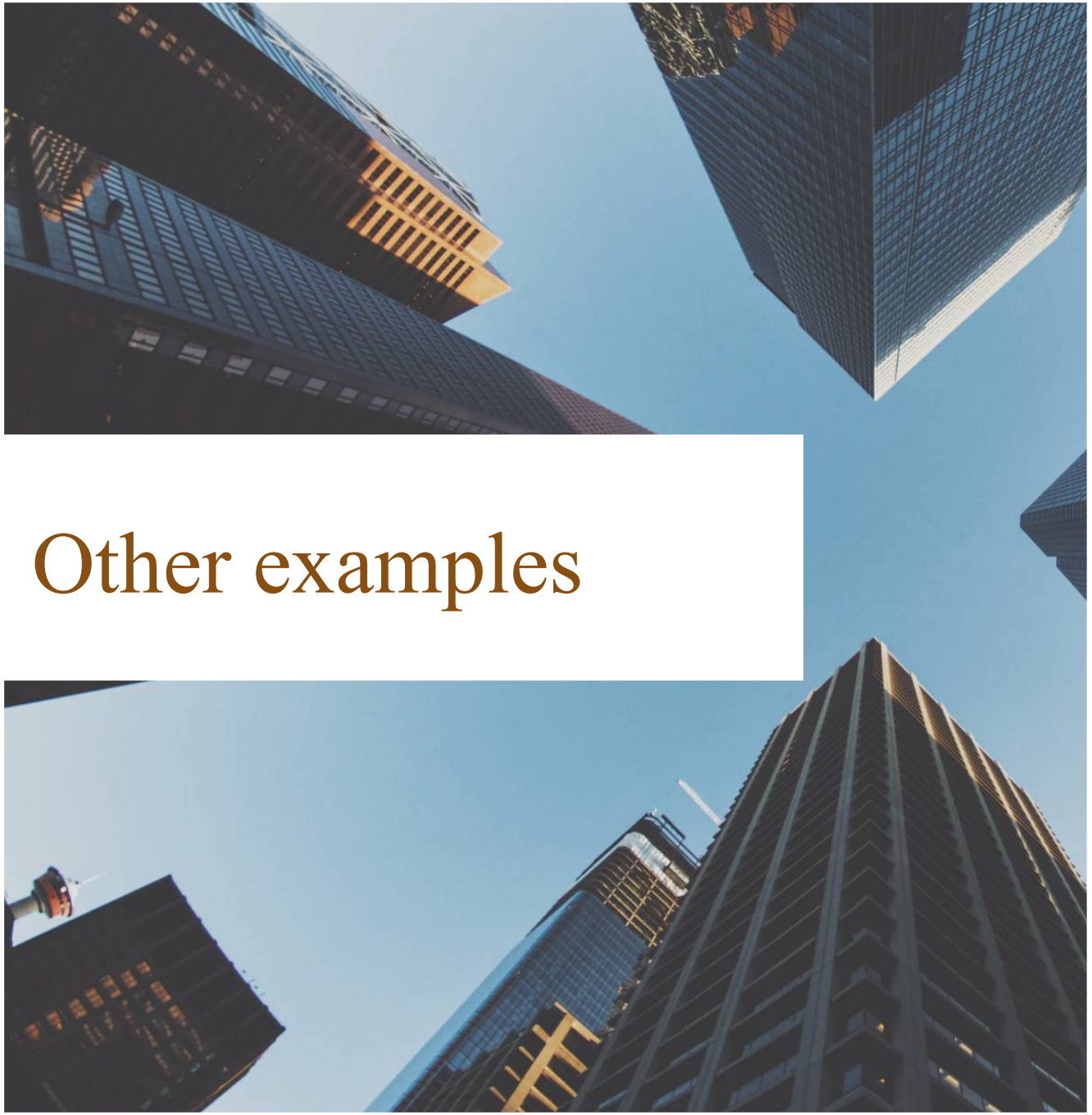
aws

Data Driver

Product enabler

Infrastructure & development

http://todogroup.org

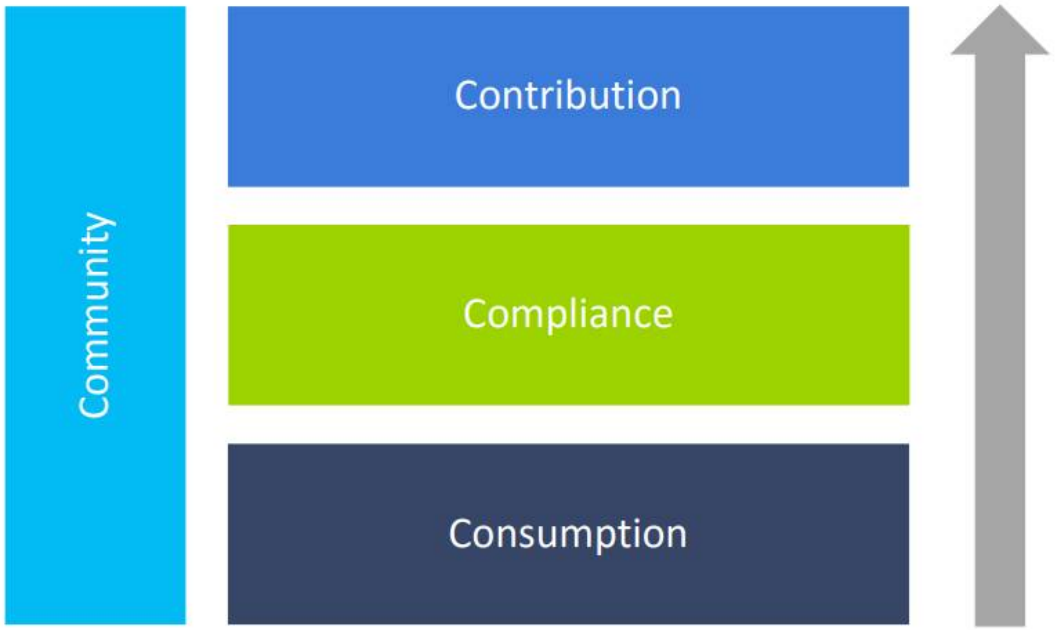


Other examples

TODO

amazon.com · AUTODESK · box
CapitalOne · Dropbox · facebook
GitHub · Google · Hewlett Packard Enterprise
HUAWEI · KHANACADEMY · IBM
Microsoft · NATIONAL INSTRUMENTS · NETFLIX
PayPal · Pinterest · salesforce
SanDisk · SAUCELABS · Square
stripe · Twitter · @WalmartLabs
vmware · YAHOO!

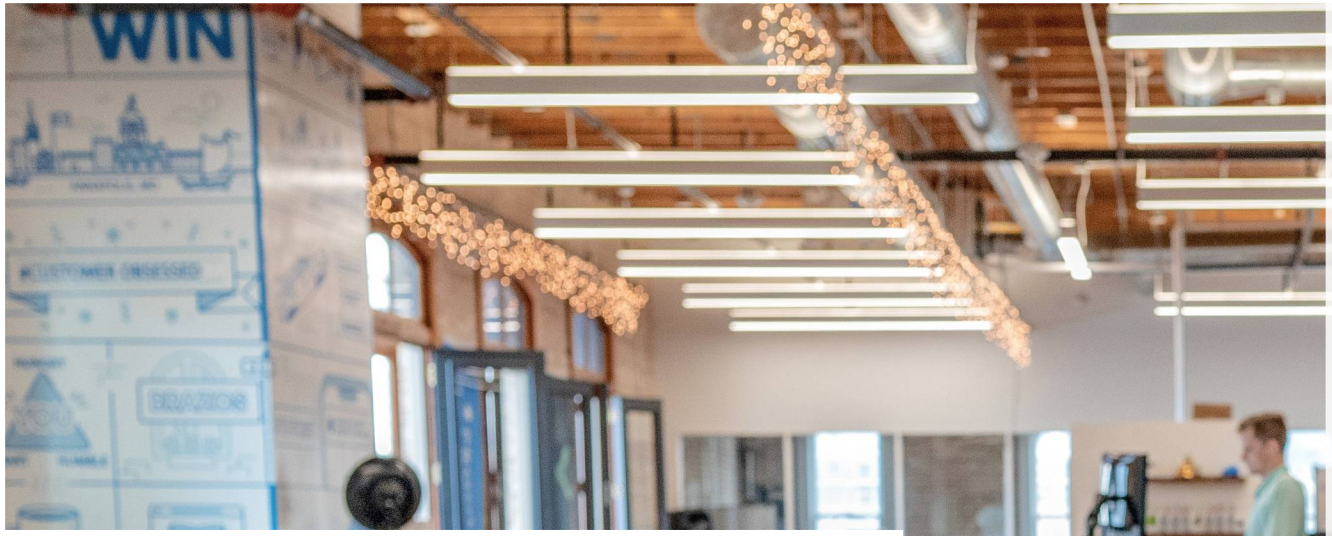
OSS Operations in Companies



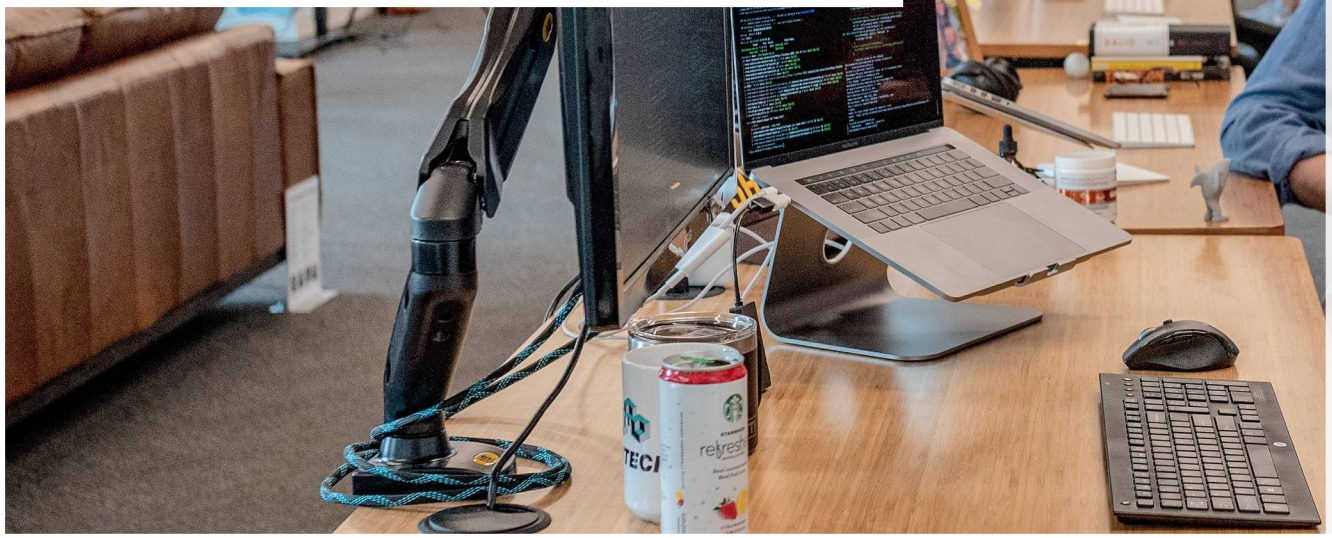
Software Product Management

What makes up a product?

- Can e.g., consist of physical/virtual objects, services, humans, or combinations
- Can be categorized into Packaged SW, SaaS, and Embedded SW
- Often part of a larger portfolio of products
- Engineering, Business and Legal perspectives



What makes up a product manager?



- Can be characterized as a “Mini-CEO” (or any software startup entrepreneur)
- Responsible for product success
- Drafts the product strategy and plans and executes it
- Connects to marketing, corporate management, R&D, product development, legal, finance...

Software Project Management

- The product manager is the customer to the project managers
- A project has a defined start and endpoint and a specific set of deliverables

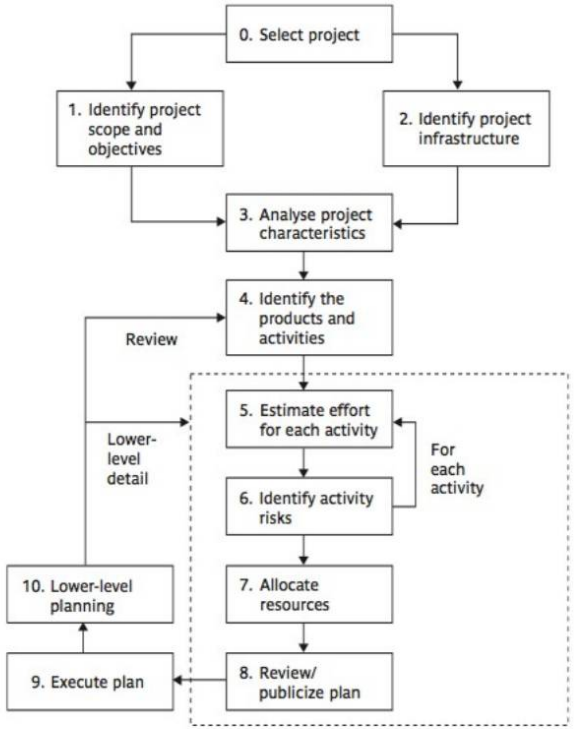
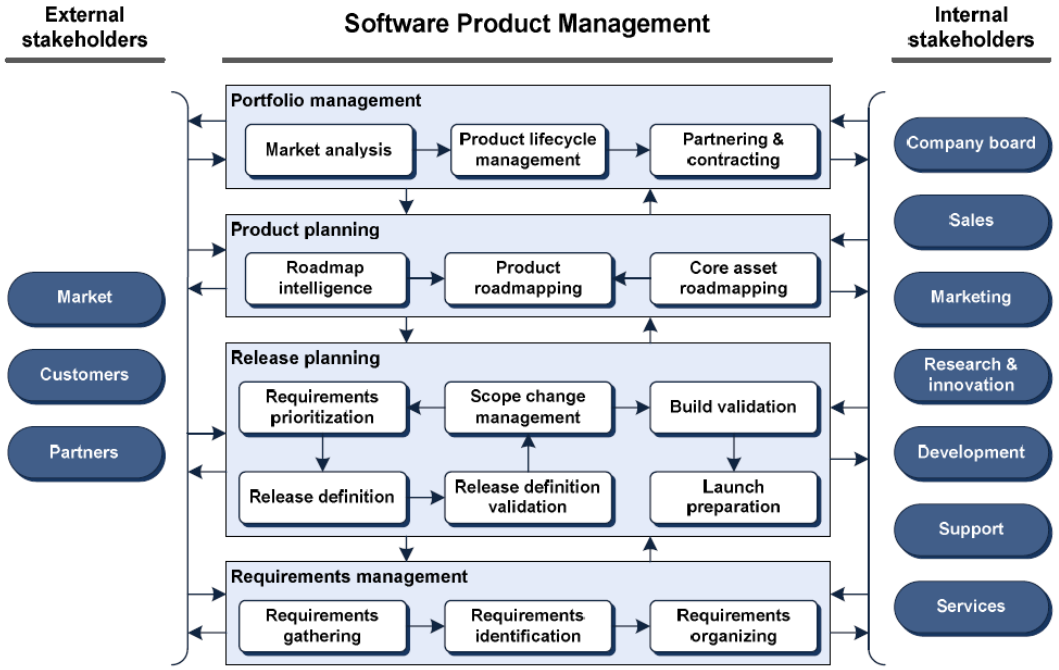


fig3.1 Ch 3 *Software Project Management (5th Ec*

Software Product Management



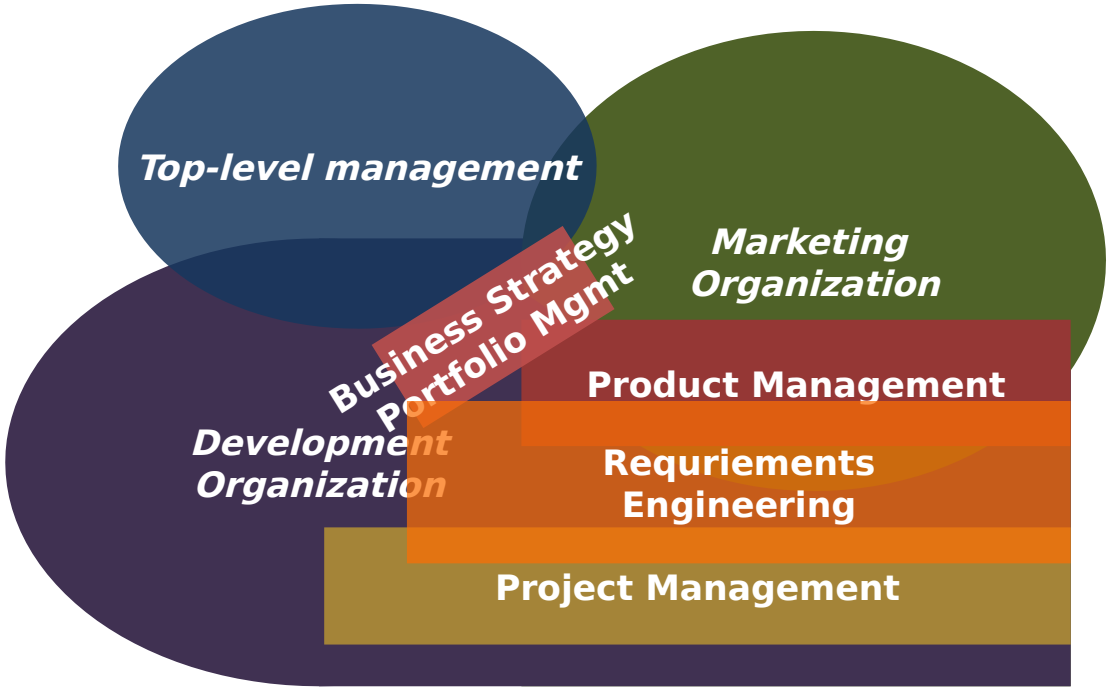
Software Product Management

Strategic Management	Product Strategy	Product Planning	Development	Marketing	Sales and Distribution	Service and Support
Corporate Strategy	Product Positioning and Definition	Product Lifecycle Management	Engineering Management	Marketing Planning	Sales Planning	Services Planning and Preparation
Portfolio Management	Delivery Model	Roadmapping	Project Management	Customer Analysis	Channel Preparation	Services Provisioning
Innovation Management	Sourcing	Release Planning	Project Requirements Engineering	Opportunity Management	Customer Relationship Management	Technical Support
Resource Management	Business Case and Costing	Product Requirements Engineering	Quality Management	Marketing Mix Optimization	Operational Sales	Marketing Support
Market Analysis	Pricing			Product Launch	Operational Distribution	Sales Support
Product Analysis	Ecosystem Management			Operational Marketing		
	Legal and Intellectual Property Rights Management					
	Performance and Risk Management					
Participation	Core SPM		Orchestration			

Fig. 3 The software product management body of knowledge (SPMBoK) (ISPMA 2012)

[PRMAN]

Software Product Management



LUND
UNIVERSITY