

Empirical research on Software Ecosystem (SeCo) and Software Startup

Anh Nguyen Duc

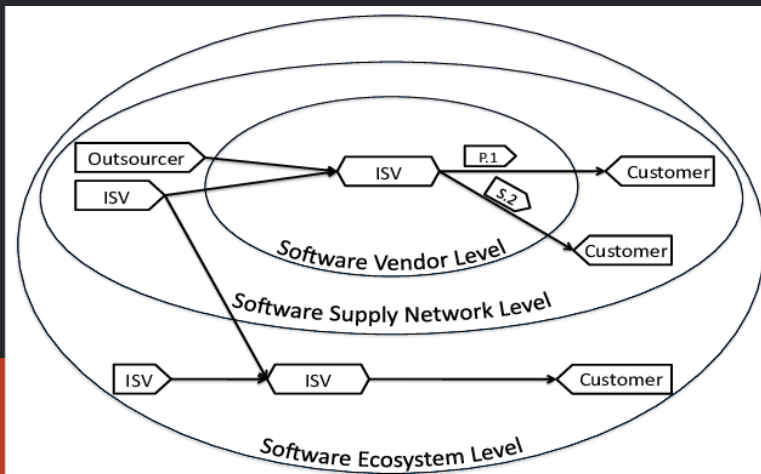
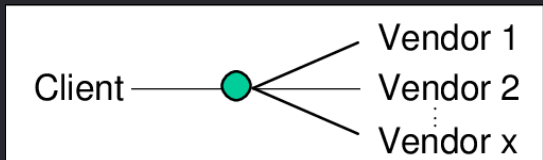
Agenda

- Software Ecosystem (SECO)
- Software Startups

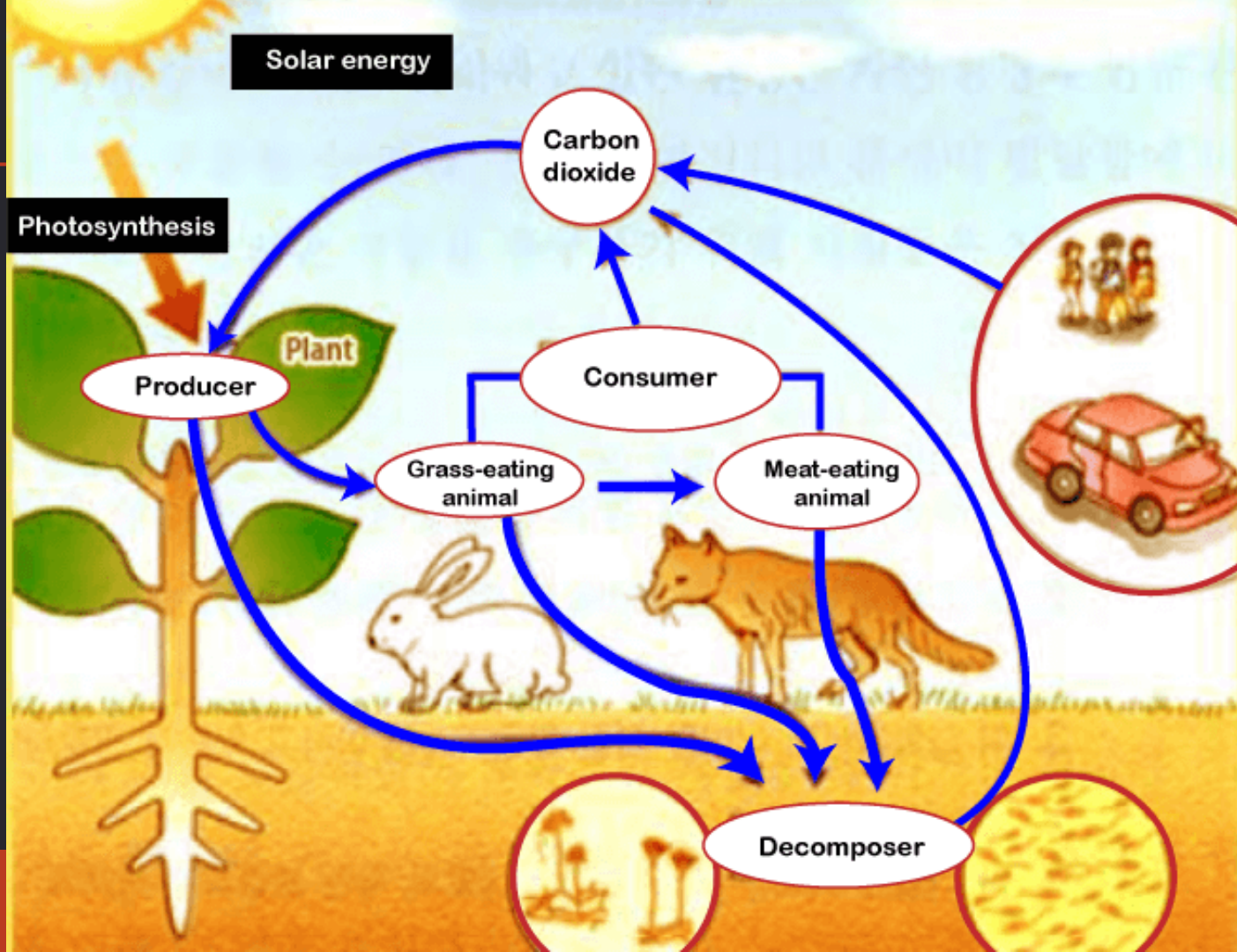
Software Ecosystem

Introduction

- ❑ Software ecosystems is an emerging trend within the software industry
- ❑ The shift from closed organizations and processes towards open structures
- ❑ Ecosystem metaphor is used to understand relationships among stakeholders



Biological ecosystem

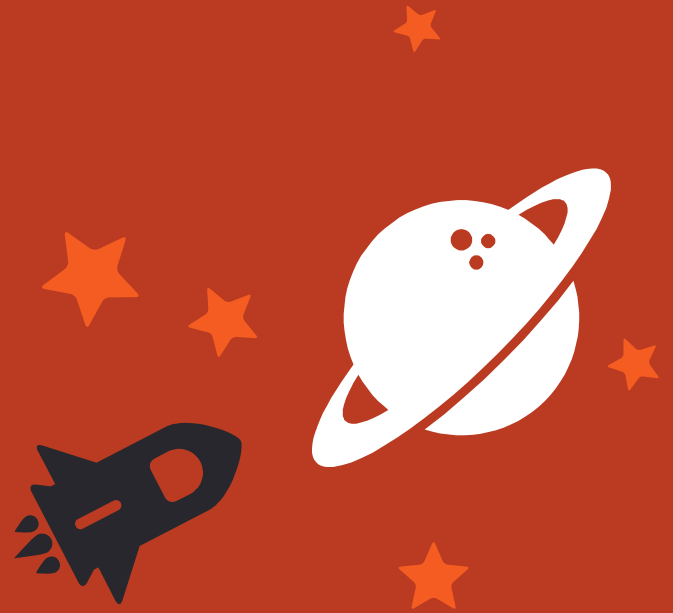


Definitions of Software Ecosystem (SeCo)

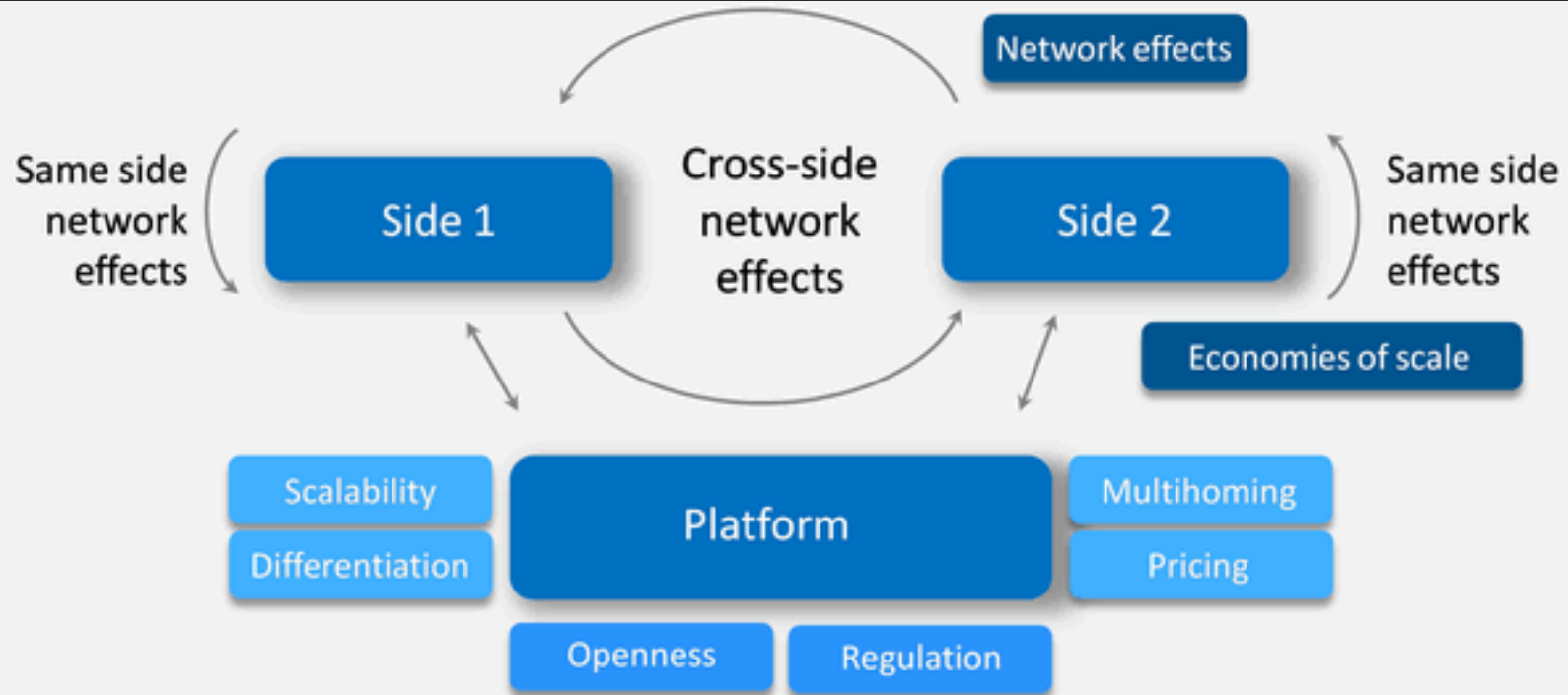
- ❑ SeCo as an informal network of independent units that have a positive influence on the economic success of a software product and benefit from it (Kittlaus and Clough)
- ❑ SeCo as consisting of the set of software solutions that enable, support, and automate the activities and transactions by the actors in the associated social or business ecosystems and the organizations that provide these solutions (Bosch)
- ❑ a set of actors functioning as a unit and interacting with a shared market for software and services, together with the relationships among them (Jansen)
- ❑ Three shared concepts stand out in these definitions:
 - (1) actors, organizations and businesses,
 - (2) networks and social or business ecosystems, and
 - (3) software.

Menti.com

Examples of Software
Ecosystems you know?



Foundation – two-sided market theory



Software Ecosystem (SECO)

- App stores: Apple, Google, Microsoft, etc



Software Ecosystem (SECO)

- Service providers on top of Shopify ecosystem

Integrated Back Office



Shipping

Inventory

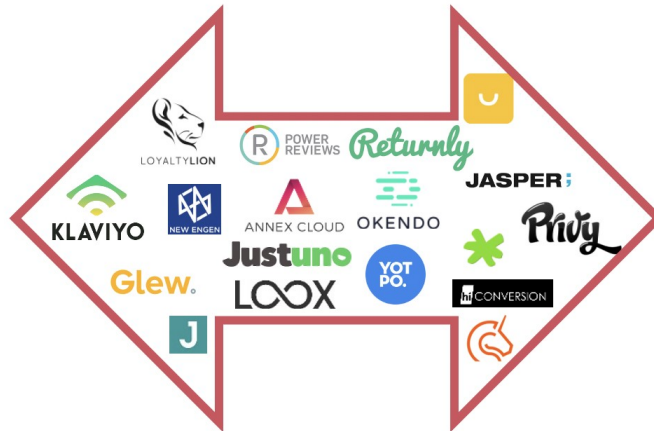
Payments

Analytics

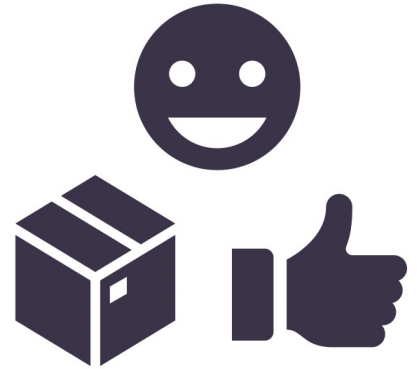
Order
Management

Promotions

Front Office Software Vendor



Customer



Other examples of Software Ecosystems

- **Standards** - XML, BPM, OSGi, J2EE, Corba, SEPA, etc.
- **Products** - OpenOffice, Microsoft Word, SAP BusinessOne, Grand Theft Auto, etc.
- **Hardware** - Playstation 3, HTC Diamond, PDAs, BMW 5 series, etc.
- **Platforms** - .Net, Facebook, Android, OS X, etc.

Types of SeCo

Example: The Apple iOS software ecosystem is based on a software platform and coordinated by a privately owned entity with a commercial extension market to which participants can submit extensions after making a payment .

- ❑ Core technology
 - software platform,
 - software service platform,
 - software standard
- ❑ Coordinated by
 - Community
 - Private organization
- ❑ Extension market
 - no extension market,
 - a list of extensions,
 - an extension market,
 - a commercial extension market,
 - multiple extension markets
- ❑ Accessibility
 - for free,
 - after a screening,
 - after making a payment

Characteristics of SeCos

- ❑ **Complexity** - In a platform ecosystem with numerous actors, complexity must be controlled somehow to reduce risk of gridlocks, unpredictable ripple effects and co-innovation problems
- ❑ **Productivity** - A network's ability to consistently transform technology and other raw materials of innovation into lower costs and new products. Simple to measure: return on invested capital.
- ❑ **Robustness** - Should be capable of surviving disruptions such as unforeseen technological change.
- ❑ **Niche creation** - the ecosystem's capacity to increase meaningful diversity through the creation of valuable new functions or niches.

A healthy Software Ecosystem

- ❑ A stable and sustainable platform and mechanism to govern the platform
- ❑ Developers, software firms making benefits from collective innovation
- ❑ Consumers receive high quality services/ products, opportunities for value co-creation

A healthy Software Ecosystem

- ❑ Sustainability is the key!

- ❑ How to design and maintain a sustainable software ecosystem ?



A healthy Software Ecosystem

❑ Sustainability is the key!

- ❑ A viable business model
- ❑ A well-organized inter-organizational interaction model
- ❑ A paradigm supporting Collaboration
- ❑ A mechanism to allow Coopetition

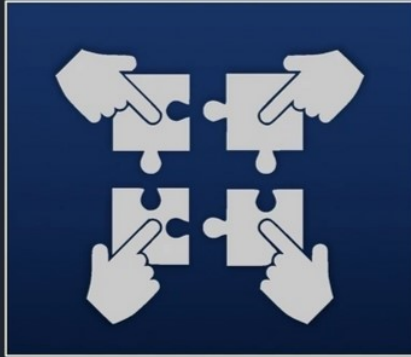


The 3C in SeCo

COOPERATION



COLLABORATION

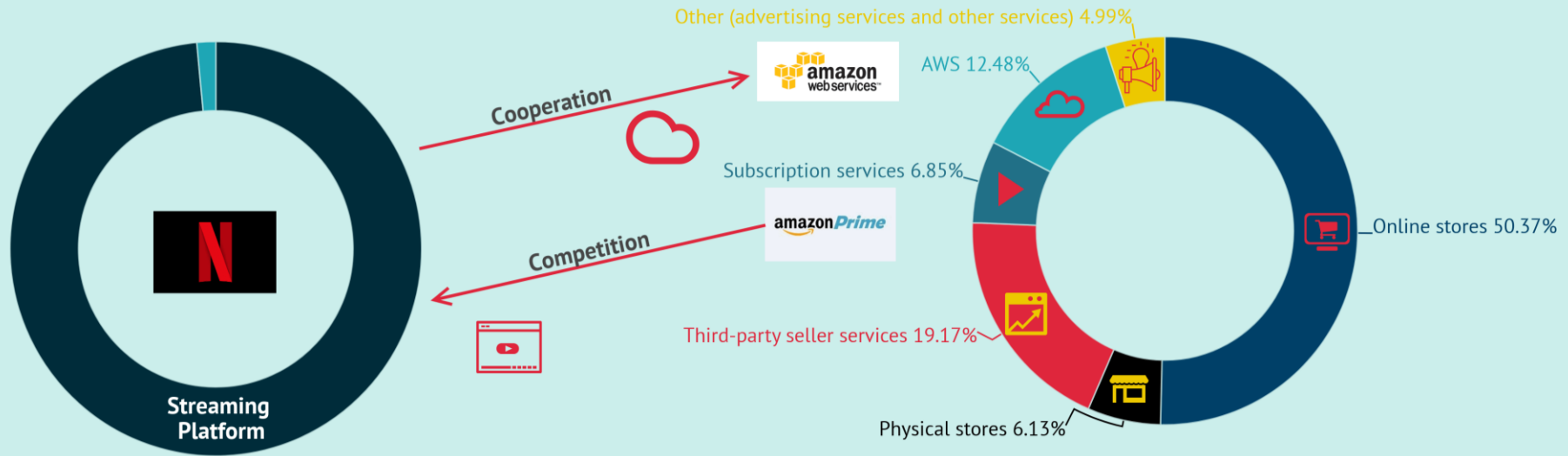


COOPETITION

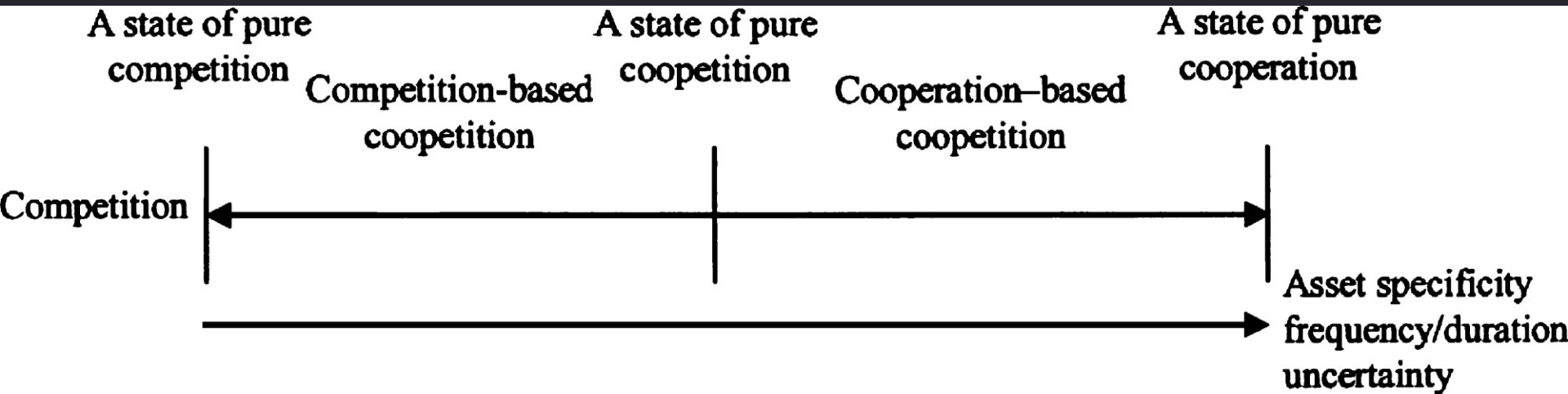


Coopetition in a nutshell

Netflix uses AWS cloud services while compete with Amazon Prime

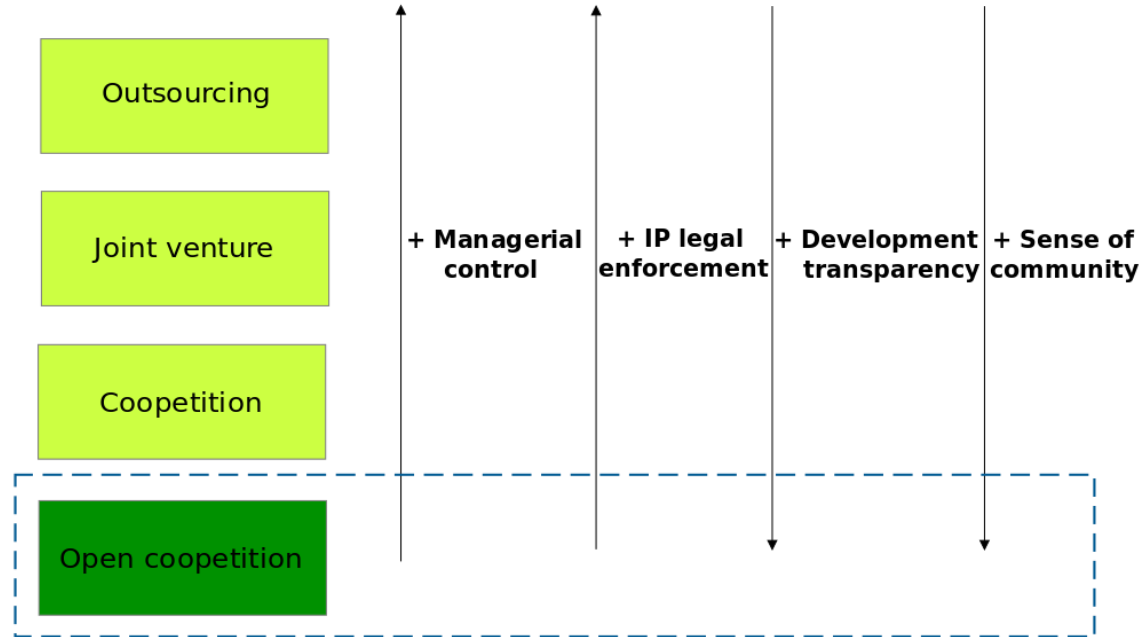


A model of coopetition

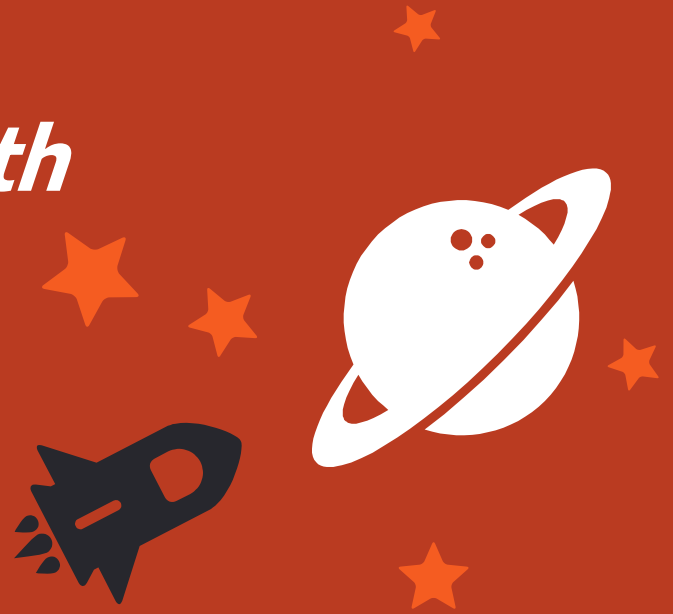


Open cooperation

Coopetition + Open Source



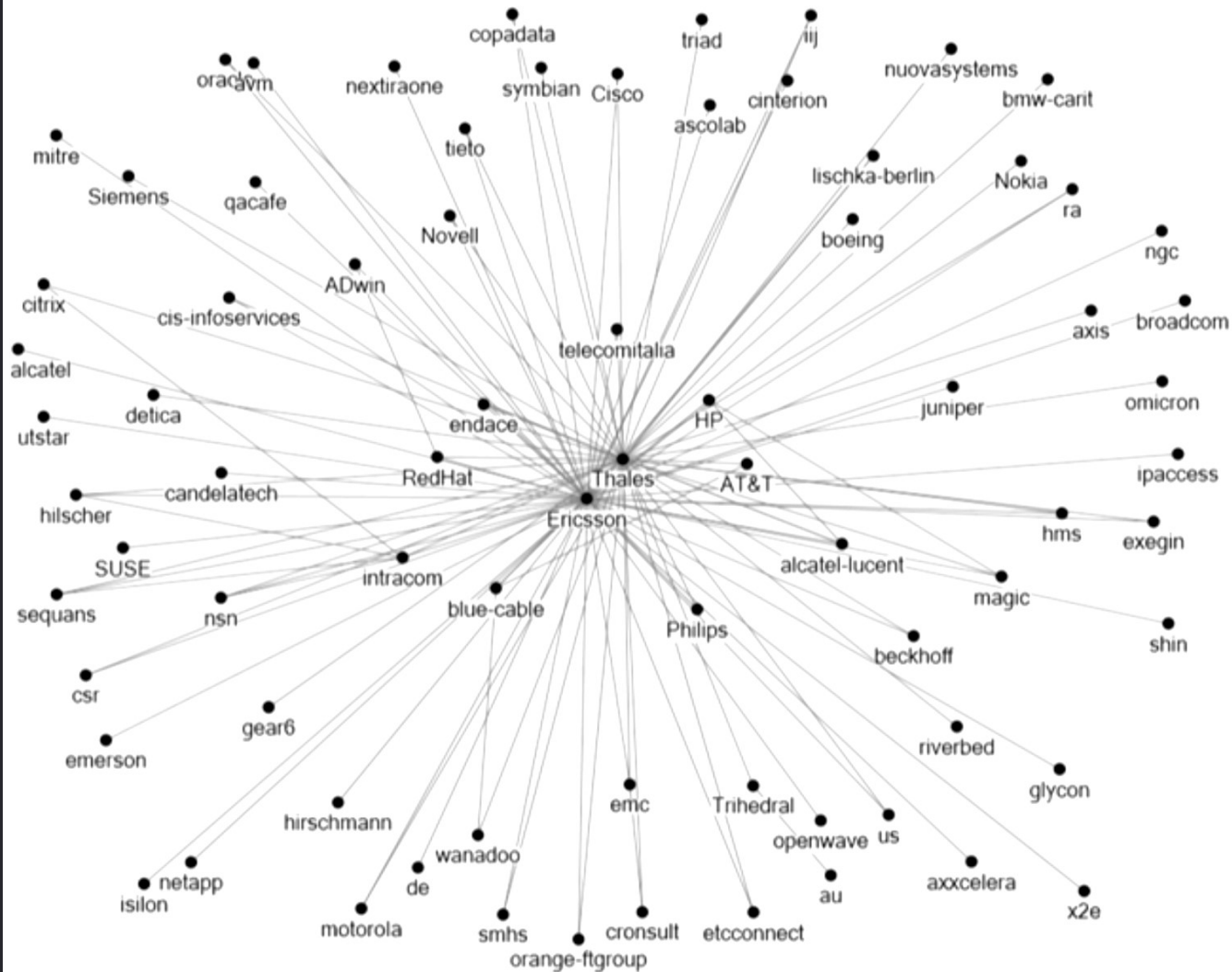
How do commercial companies maintain both collaboration and competition in an open source software ecosystem?



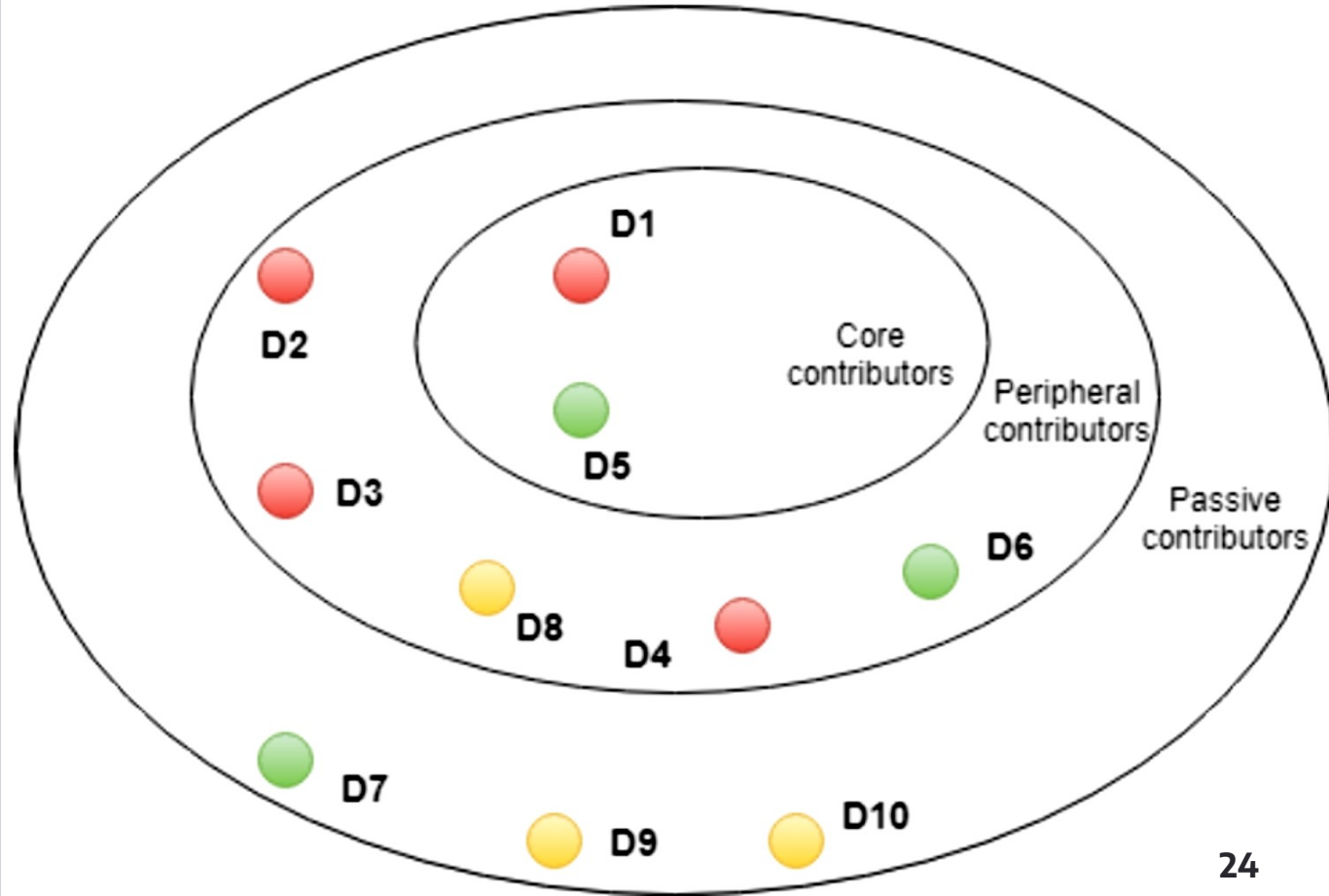
Our cases



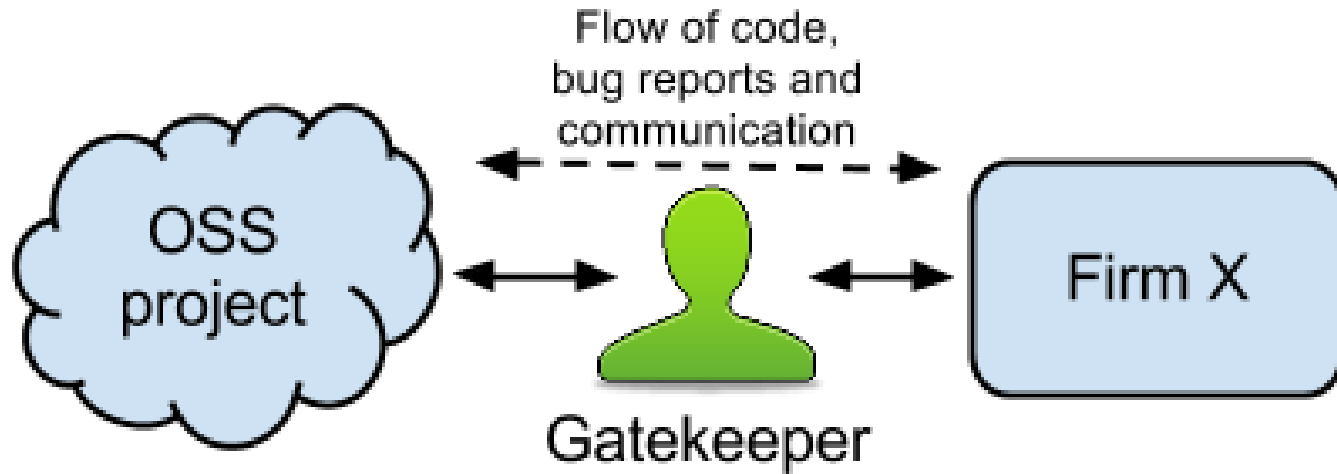
Wireshark ecosystem - social analysis



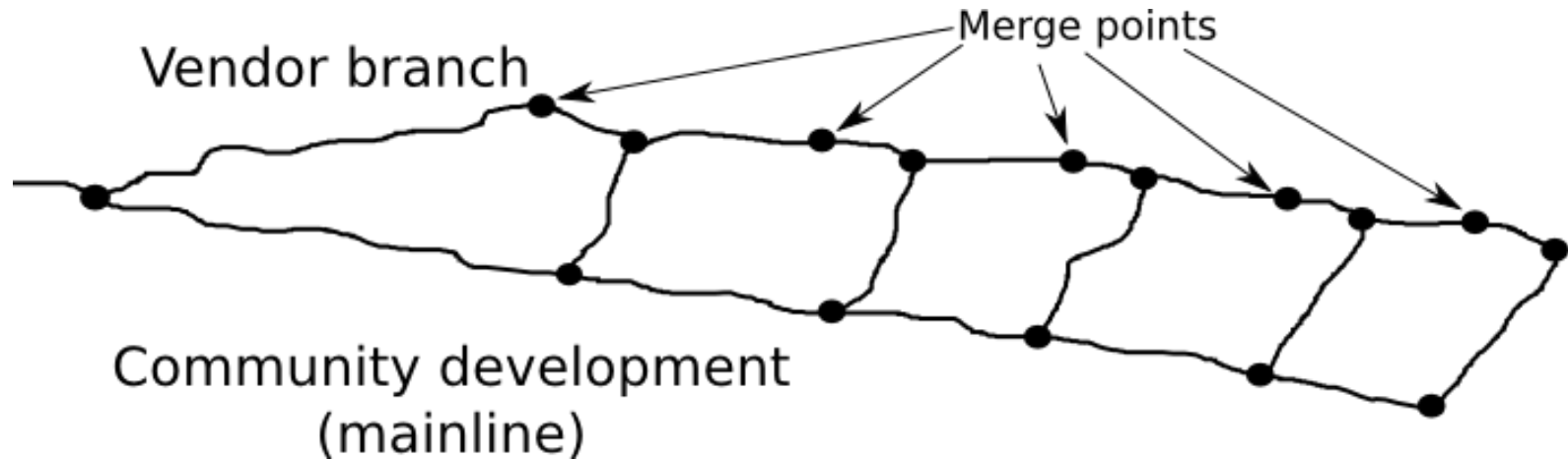
Wireshark ecosystem - conceptual layers



How do commercial firms manage coopetition with other firms in such context?

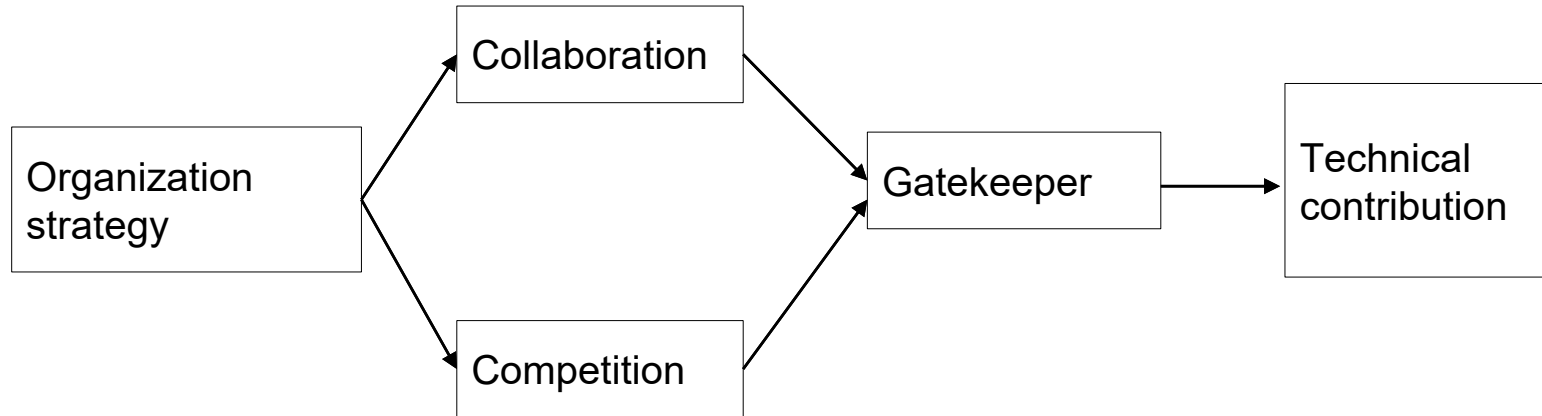


How do commercial firms manage coopetition with other firms in such context?



- Open-core sourcing policy
- Upstream first - get features integrated into open source projects before integrating them into firm's product

A model of open cooperation in SECO



Software Startups

Software startups

Startup companies are unique:

- Little or no operating history
- Limited resources
- Multiple influences
- Dynamic technologies and markets
- Scalable business model

And more ...

Software startups

Startup companies are unique:

- Software in value proposition
- Innovation focus
- Lack of resources
- High uncertainty
- Time-pressure
- Highly reactive
- Rapidly evolving
- Temporary organizational state
- Little/ no history of stable operations
- Product development and business development in parallel
- Seek for product-market fit
- Strong presence of entrepreneurial personalities
- **60%** of startups do not survive in the first five years
- **75%** of venture capital funded startups fail
- Little rigor and relevance exist in the studies about software startups
- Do not focus on investigating issues and challenges

Menti.com

Name a software startup
that you know in Norway



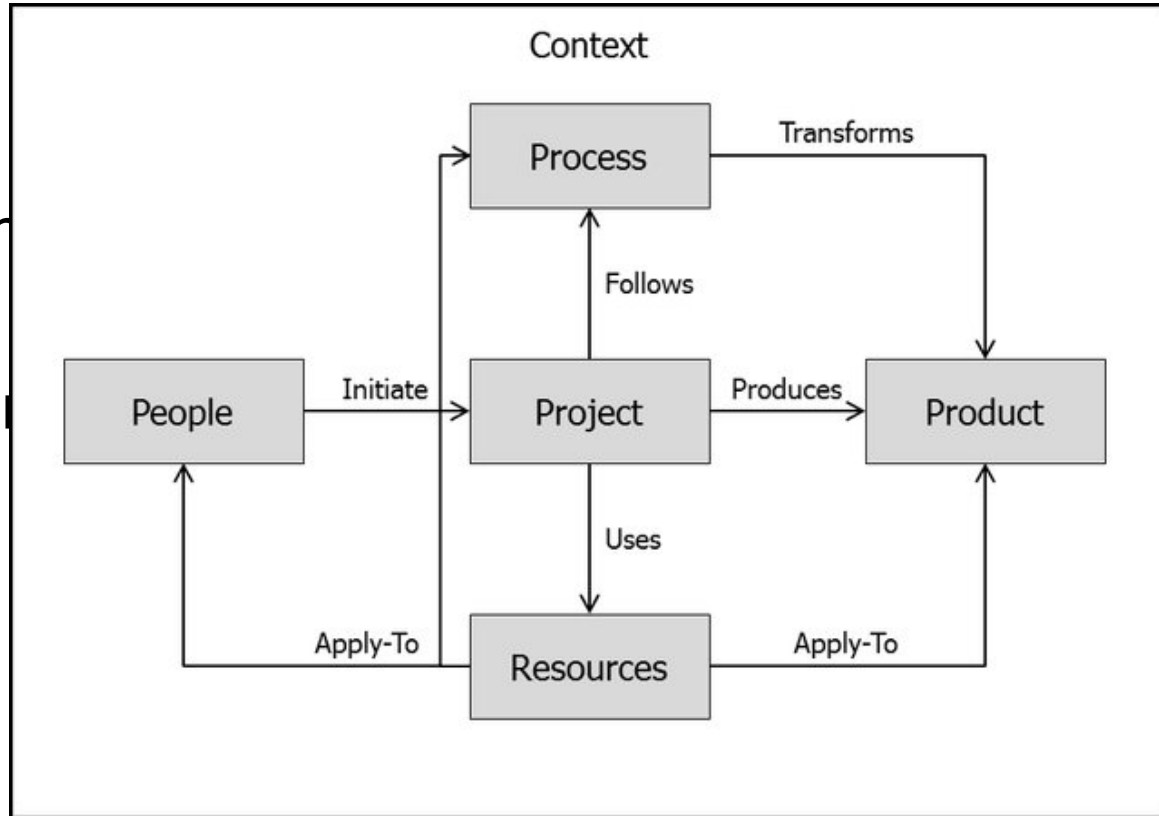
Software startups

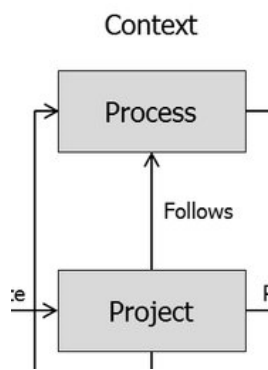


1. J. Melegati, A. Goldman, F. Kon, and X. Wang, "A model of requirements engineering in software startups," *Information and Software Technology*, vol. 109, pp. 92-107, May 2019
2. V Berg, J Birkeland, A Nguyen-Duc, IO Pappas, L Jaccheri, Software startup engineering: A systematic mapping study, *Journal of Systems and Software* 144, 255-274, 2018

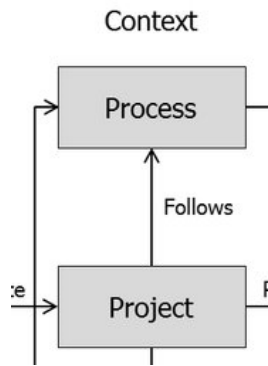
Software startup research

- Empirical
- Reveal and reflect on phenomena

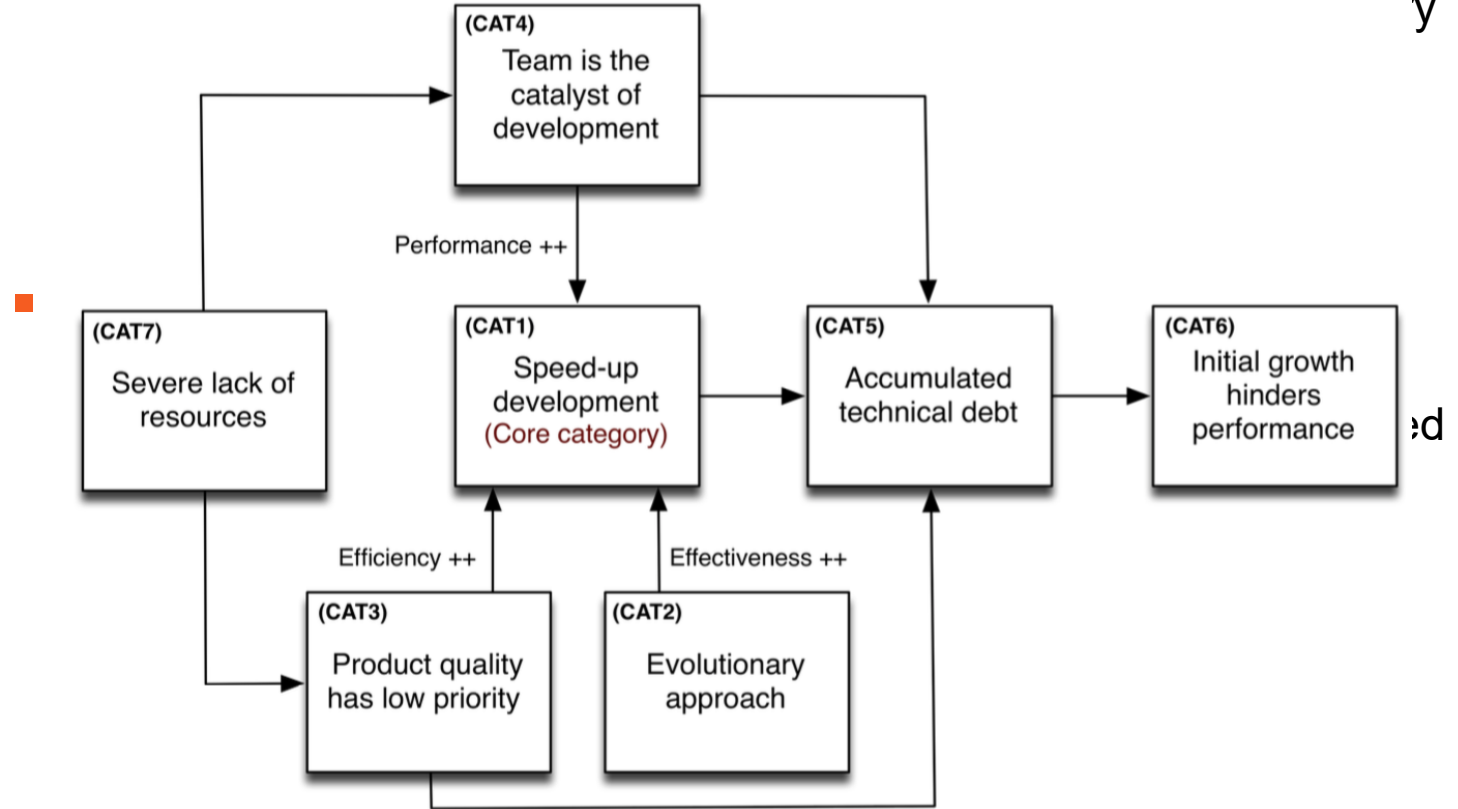




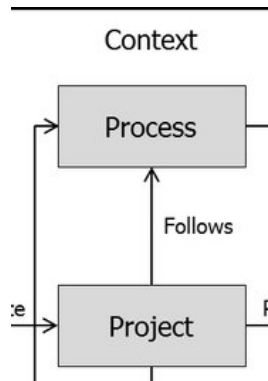
- **Technical debt:**
 - ▣ when development teams take actions to expedite the delivery of a piece of functionality or a project which later needs to be refactored
 - ▣ Intentional vs. unintentional
 - ▣ prioritizing speedy delivery over perfect code
- Technical debt as integral parts of software startups: the need to shorten time-to-market, by speeding up the development through low-precision engineering activities, is counterbalanced by the need to restructure the product before targeting further growth



Technical debt:



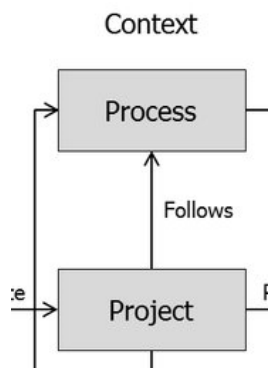
1. C. Giardino, N. Paternoster, M. Unterkalmsteiner, T. Gorschek, and P. Abrahamsson, "Software Development in Startup Companies: The Greenfield Startup Model," IEEE Transactions on Software Engineering, vol. 42, no. 6, pp. 585–604, Jun. 2016



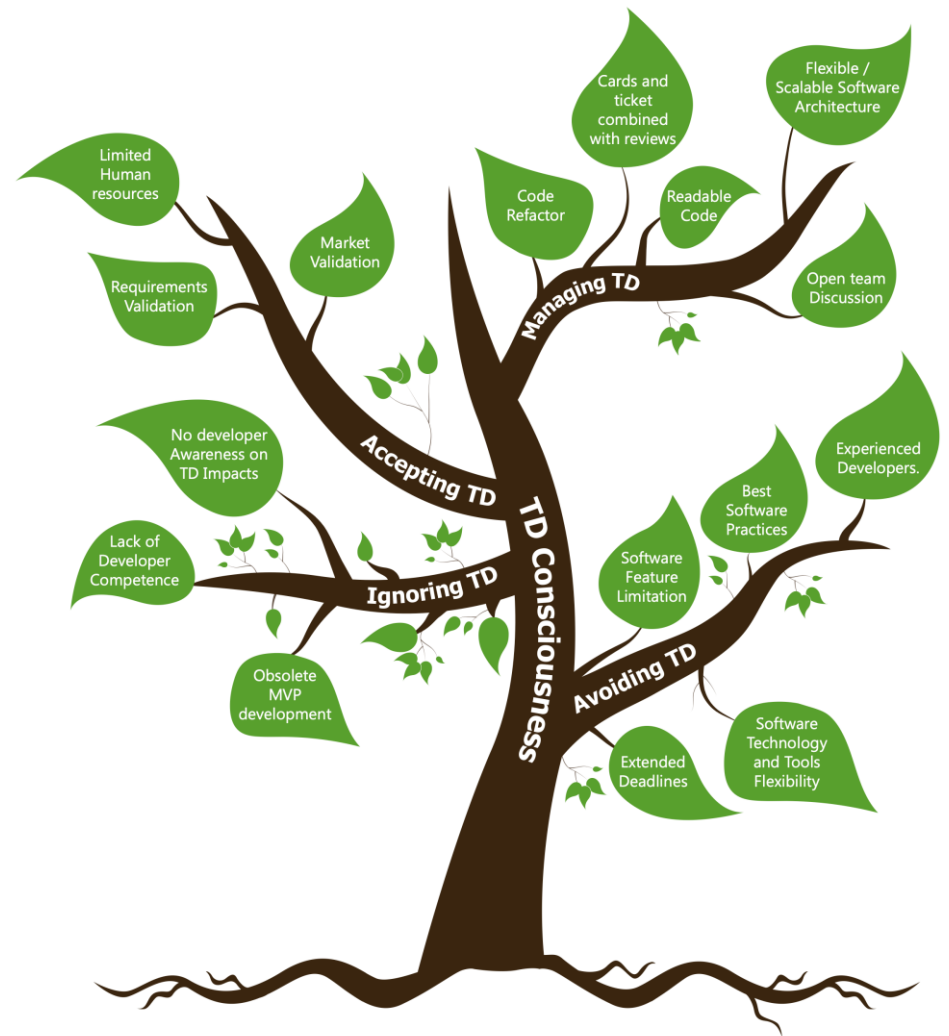
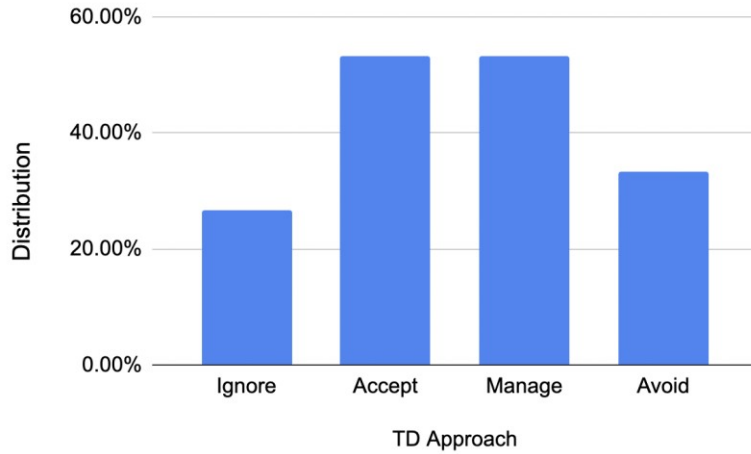
■ Technical debt – four facet model in software startups



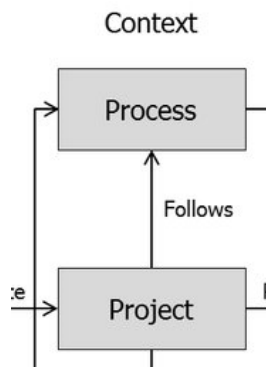
1. Cico et al. Technical Debt Approach and Consciousness in Startups Transitioning to Growth Phase - A Four Facet Model. Submitted to IEEE Access, 2022



Technical debt – fo

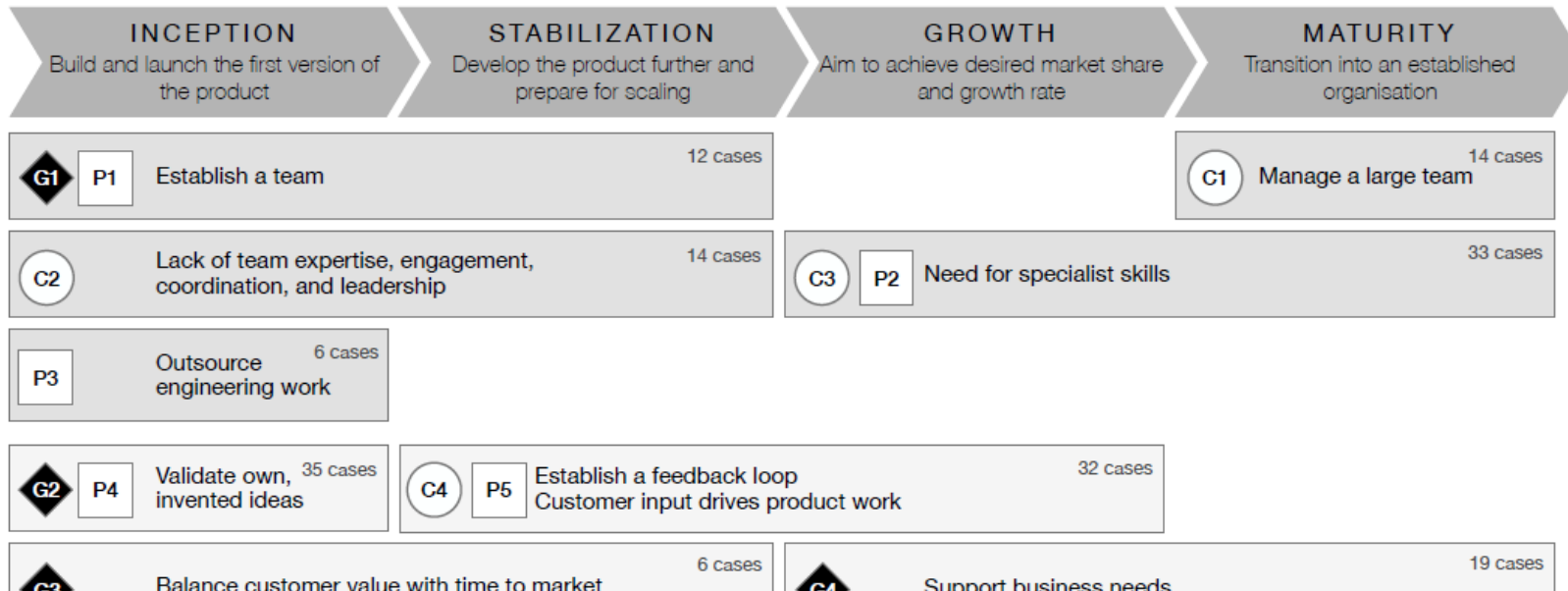


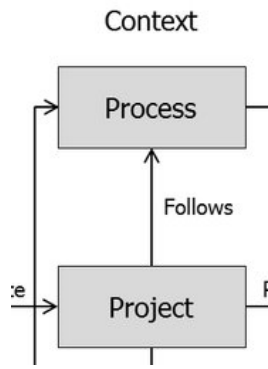
1. Cico et al. Technical Debt Approach and Consciousness in Sta Access, 2022



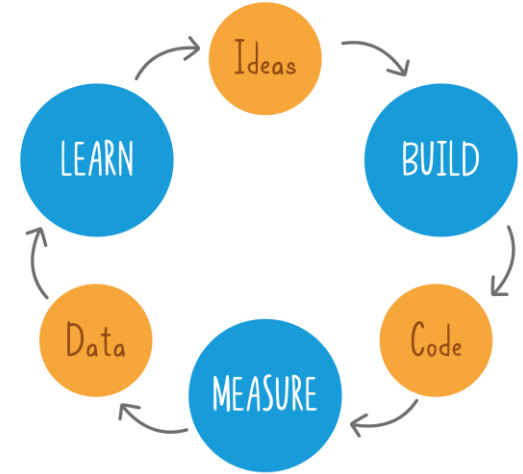
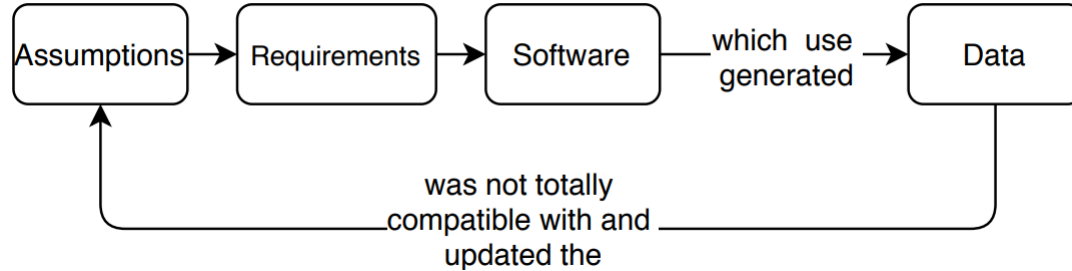
- Domain knowledge, technical expertise, and teamwork are the key components in the early stages of a start-up. Shortages of any of these components can be compensated with specific practices
- Key difference and difficulty to practice software engineering in

Start-up life-cycle phases

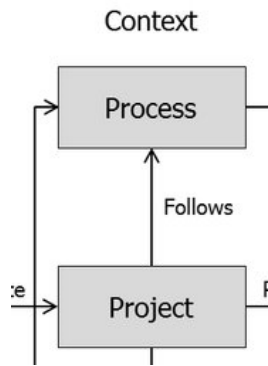




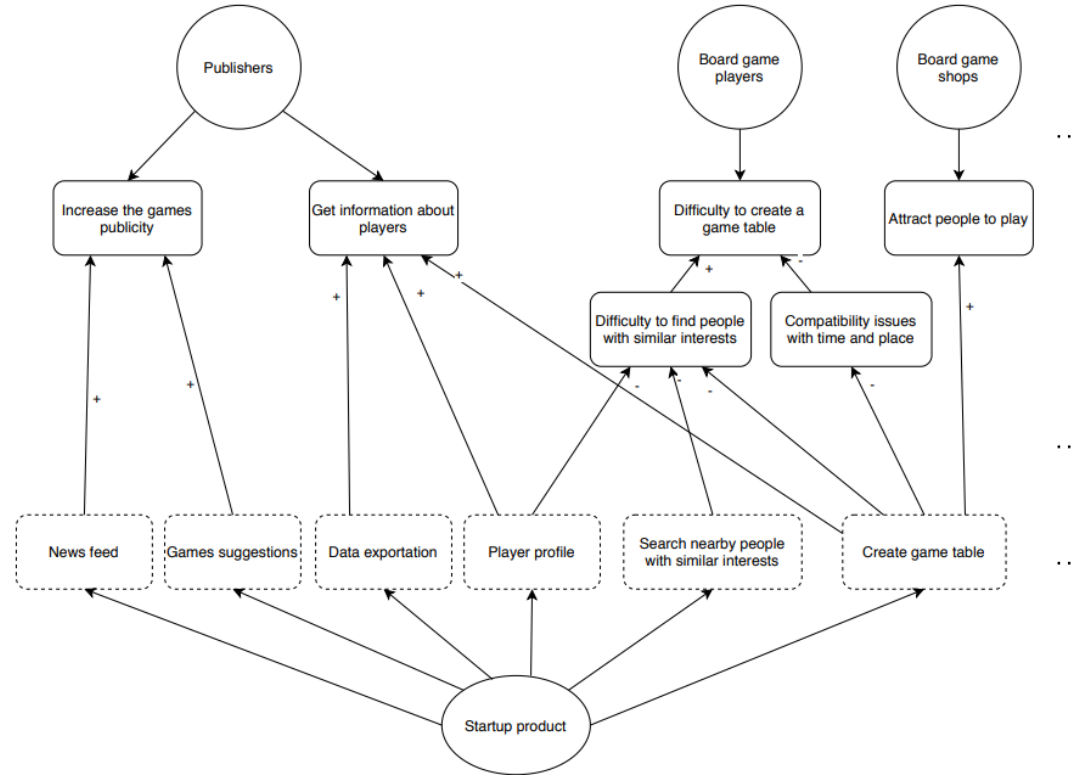
- **Hypothesis engineering** in software startups
 - ▣ There are always assumptions
 - ▣ Decisions are often driven by the instinct of startup founders
 - ▣ Lean startup – is this really used?



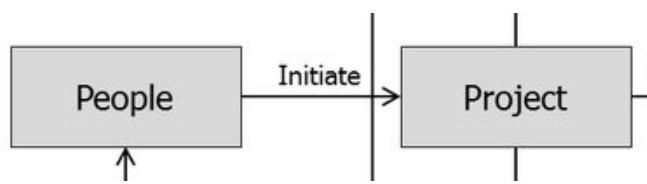
1. J. Melegati, H. Edison, and X. Wang, "XPro: a Model to Explain the Limited Adoption and Implementation of Experimentation in Software Startups," IEEE Transactions on Software Engineering, pp. 1–1, 2020
2. E. Ries, *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, 1 edition. New York: Currency, 2011.



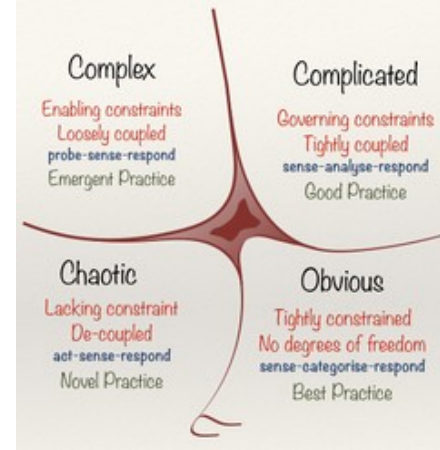
- Cognitive map to visualize the hypotheses and their relationships



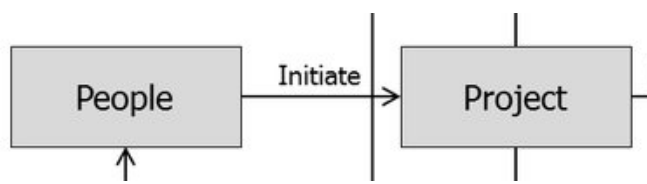
1. J. Melegati, H. Edison, and X. Wang, "XPro: a Model to Startups," IEEE Transactions on Software Engineering, . . .
2. E. Ries, *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, 1 edition. New York: Currency, 2011.



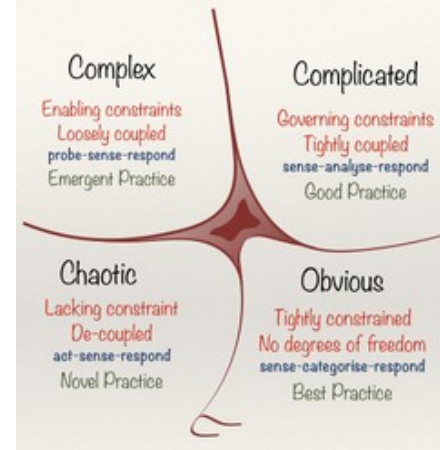
- People:
 - Startup Founders
 - Stakeholders in Startup Ecosystems
- Cynefine-based categories of startups' decisions:
- Simple decisions:
 - Setting up social media accounts for the startup
 - Focusing on acquiring funding
 - Creating a common team calendar



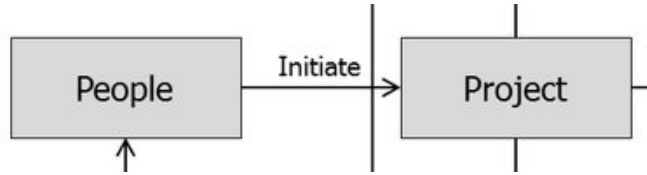
Nguyen Duc, A., Shah S. and Abrahamsson, P. (2016). Towards an early stage software startups evolution model. Euromicro SEAA



- Cynefine-based categories of startups' decisions:
- Complicated decisions:
 - Practicing public speaking
 - Deciding on different pricing models
- Complex decisions:
 - Validating the most important features
 - Gathering a team

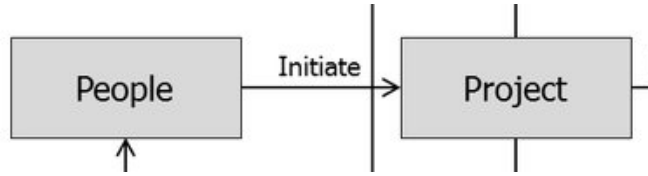


Nguyen Duc, A., Shah S. and Abrahamsson, P. (2016). Towards an early stage software startups evolution model. Euromicro SEAA

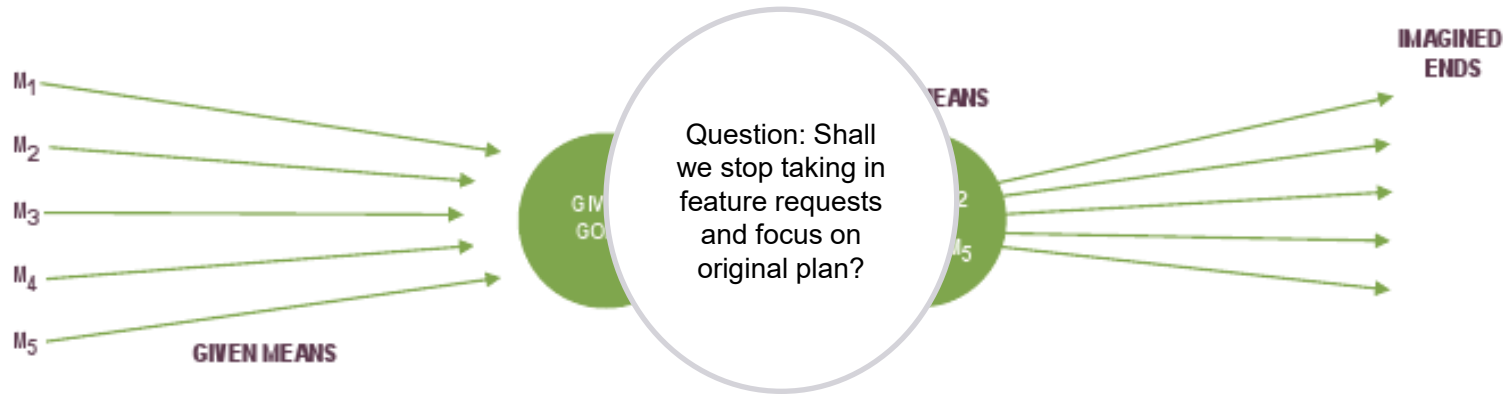


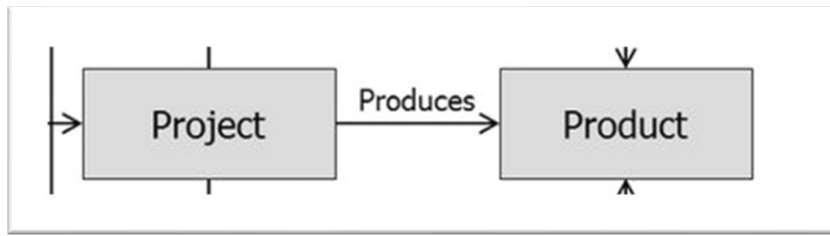
- Two logics of making decisions: Effectuation and Causations
- Effectuation:





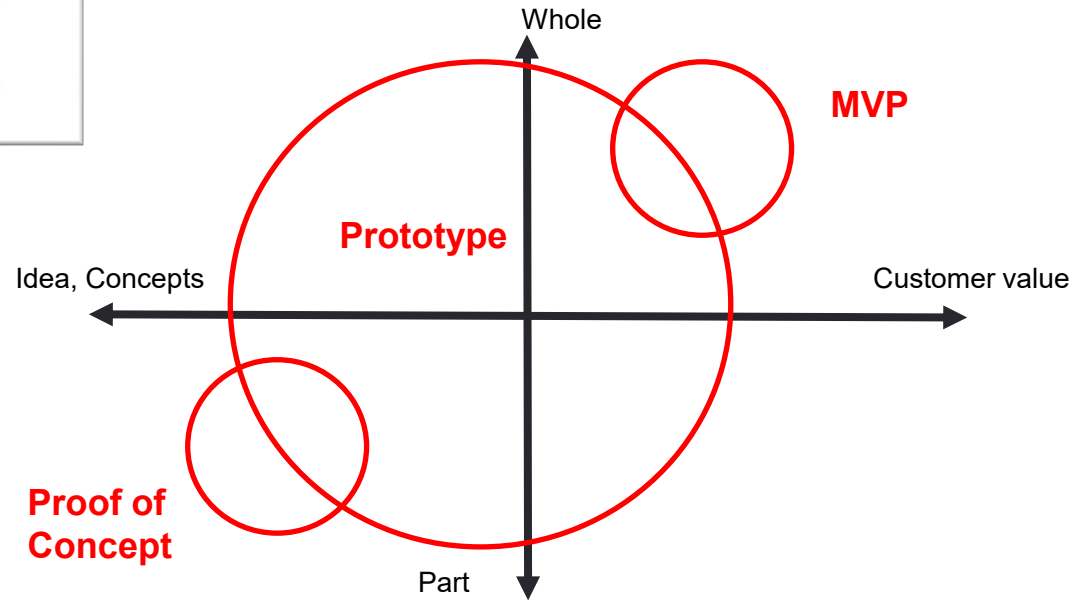
- Two logics of making decisions in startup projects:
 - ▣ Will they apply to decisions at technical and product-level as well?

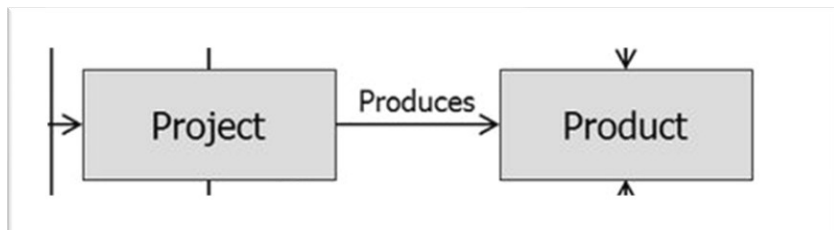




■ What - Minimum Viable Product

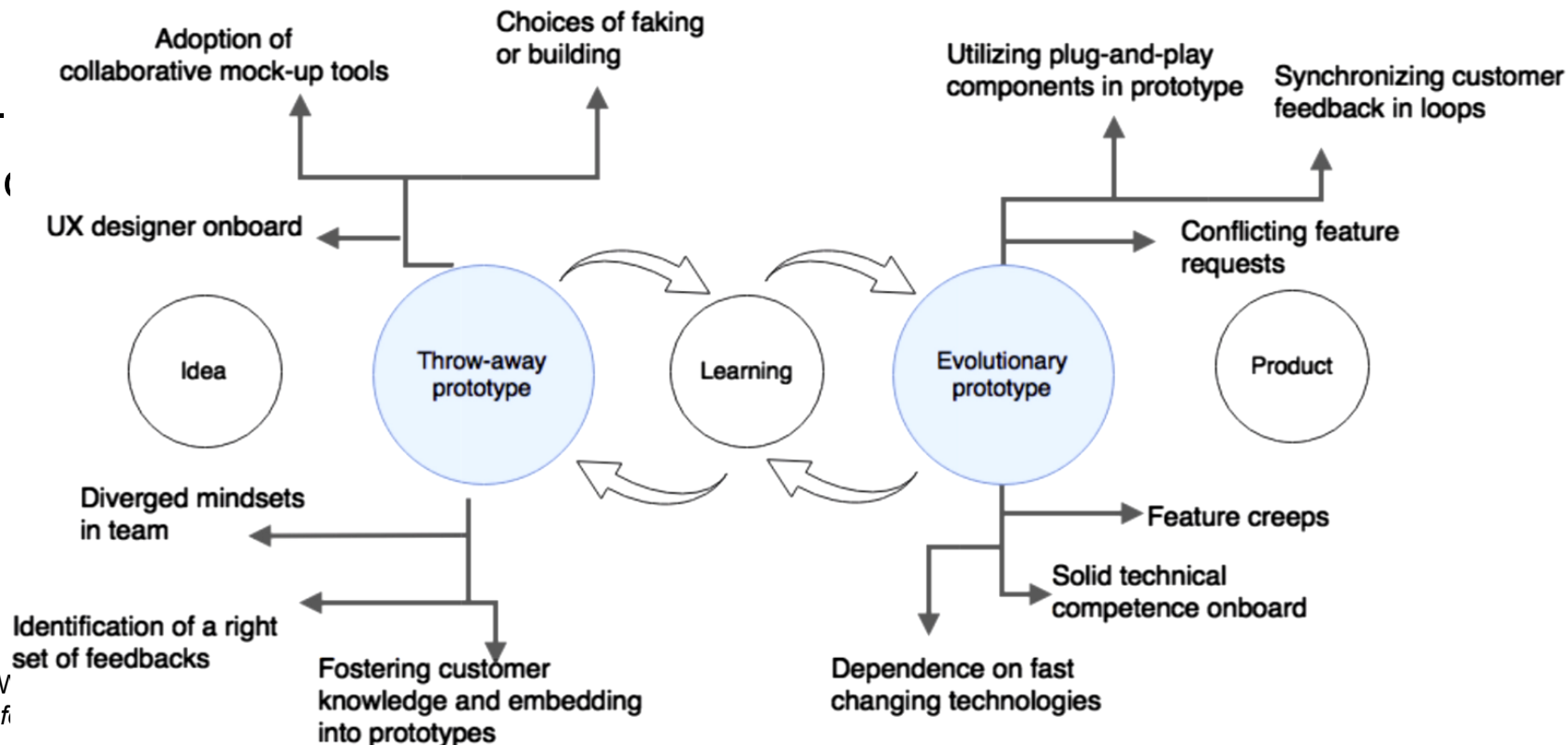
- ... as a prototype
- ... as a communication tool
- as a learning mechanism
- as documentation approach





How - Product

- ...
- ...



Thank you! Questions?

Anh Nguyen-Duc
angu@usn.no

