



THE EVOLUTION OF REQUIREMENTS PRACTICES IN SOFTWARE STARTUPS

Catarina Gralha¹, Daniela Damian², Anthony Wasserman³, Miguel Goulão¹, João Araújo¹

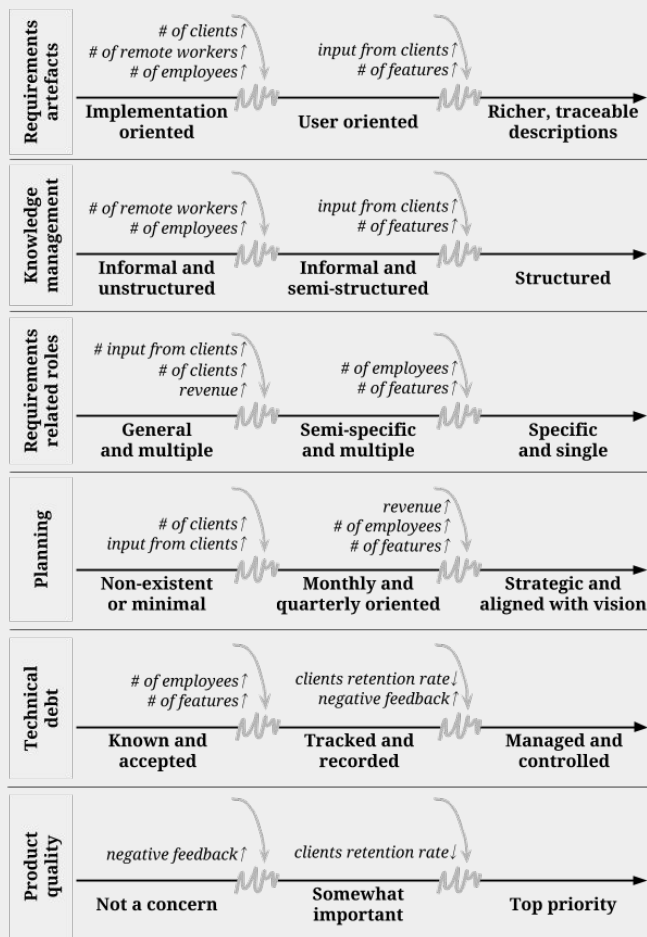
¹ Universidade Nova de Lisboa, Portugal

²University of Victoria, Canada

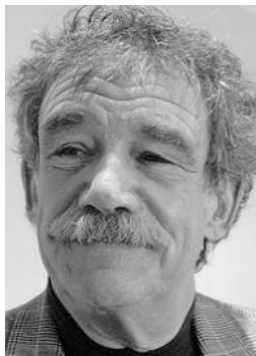
³Carnegie Mellon University, Silicon Valley, USA

June 1st, 2018

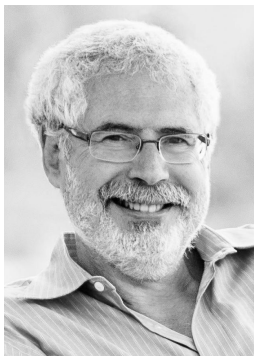
REQUIREMENTS PRACTICE EVOLUTION IN STARTUPS



DEFINITION OF A **STARTUP**



Bob Dorf



Steve Blank

“An organisation in search of a scalable, repeatable, profitable business model”

“A human institution designed to create a new product or service under conditions of extreme uncertainty”



Eric Ries

IN THIS STUDY, A **STARTUP** IS A **COMPANY** THAT...



...has recently spun-off from a large company



...is still at a stage without a solid revenue stream



...has not yet gone public

EMERGING COMPANIES



CURRENT STATUS OF RESEARCH

Very little is known about how these emerging companies discover, prioritise and manage information about requirements over time

Requirements Engineering in Software Startups: a Grounded Theory approach

Jorge Midegami and Allanah Gershman
Department of Informatics
University of York
j.midegami@york.ac.uk, a.gershman@york.ac.uk

Abstract—Software startups face a very distinctive market: their users deliver high uncertainty, volatility in the demand for their products, and a high rate of churn. In this paper, we explore how these startups discover, prioritise and manage information about requirements over time. We use a grounded theory approach to explore how these startups discover, prioritise and manage information about requirements over time. We use a grounded theory approach to explore how these startups discover, prioritise and manage information about requirements over time.

Keywords: requirements engineering, software development, software startups, grounded theory

1. INTRODUCTION

Startups are at the vanguard of the technology revolution that is reshaping the world for the 21st century. In this high-growth environment, software development is a critical success factor for many startups. However, the software development process is a complex and dynamic one, and startups face many challenges in this regard. In this paper, we explore how these startups discover, prioritise and manage information about requirements over time. We use a grounded theory approach to explore how these startups discover, prioritise and manage information about requirements over time.

A detailed description of software requirements engineering is beyond the scope of this paper. However, we provide a brief overview of the key concepts and terminology used in this field.

Hunter-Gatherer Cycle: A Conceptual Model of the Evolution of Software Startups

Arin Nguyen-Gauk
Department of Computer and Information Science
University of Oslo
arin.nguyen-gauk@iis.uio.no

Abstract—Software startups are increasingly important in the global economy. However, there is a lack of understanding of how these startups discover, prioritise and manage information about requirements over time. In this paper, we propose a conceptual model of the evolution of software startups, based on a grounded theory approach. We use a grounded theory approach to explore how these startups discover, prioritise and manage information about requirements over time.

Keywords: software startups, requirements engineering, hunter-gatherer cycle, conceptual model

1. INTRODUCTION

Software startups are increasingly important in the global economy. However, there is a lack of understanding of how these startups discover, prioritise and manage information about requirements over time. In this paper, we propose a conceptual model of the evolution of software startups, based on a grounded theory approach. We use a grounded theory approach to explore how these startups discover, prioritise and manage information about requirements over time.

A detailed description of software requirements engineering is beyond the scope of this paper. However, we provide a brief overview of the key concepts and terminology used in this field.

Software Development in Startup Companies: The Greenfield Startup Model

Carrie Ganssler, Nishu Palamare, Michael Ustunhanian, Michael IEEE, Tony Ganssler, Amir, IEEE, and Paula Altshuler, Amir, IEEE

Abstract—Software startups are a new type of organization that are disrupting the traditional software development process. They are characterized by their focus on innovation, their rapid growth, and their ability to adapt to changing market conditions. In this paper, we propose a conceptual model of the software development process in startup companies, based on a grounded theory approach. We use a grounded theory approach to explore how these startups discover, prioritise and manage information about requirements over time.

Index Terms—Software development, startup, greenfield startup

1. INTRODUCTION

Software development is a complex and dynamic process that is constantly evolving. In this paper, we propose a conceptual model of the software development process in startup companies, based on a grounded theory approach. We use a grounded theory approach to explore how these startups discover, prioritise and manage information about requirements over time.

A detailed description of software requirements engineering is beyond the scope of this paper. However, we provide a brief overview of the key concepts and terminology used in this field.

Model (GSM). We capture the underlying phenomena of software development in startup companies. We use a grounded theory approach to explore how these startups discover, prioritise and manage information about requirements over time.

A detailed description of software requirements engineering is beyond the scope of this paper. However, we provide a brief overview of the key concepts and terminology used in this field.



RESEARCH QUESTIONS

1

How do **requirements practices** change **over time** in *emerging companies*?

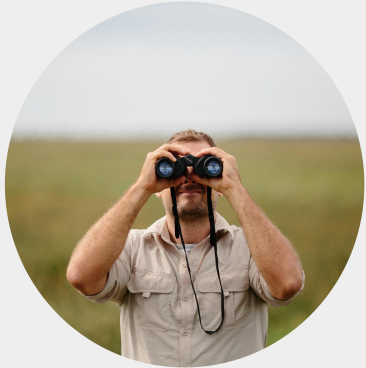
2

What **factors** and **turning points** drive those changes?

STUDIED 16 STARTUP COMPANIES

Age	1..10
Countries	     
Domain	      
Employees	1..10 to 51..60
Roles	     

GROUNDNED THEORY: DATA COLLECTION



6 full-day
observations



14 project meetings
attendance



8 focus
groups



18 semi-structured
interviews

GROUNDNED THEORY: DATA ANALYSIS



open coding

*identification
of patterns*



constant
comparison



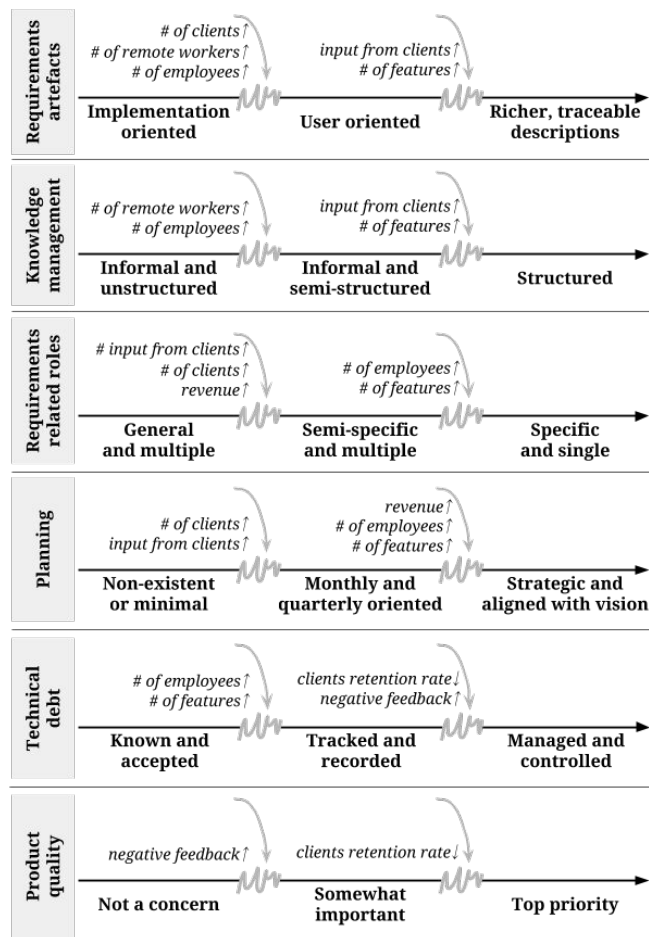
axial coding

*identification
of relationships*



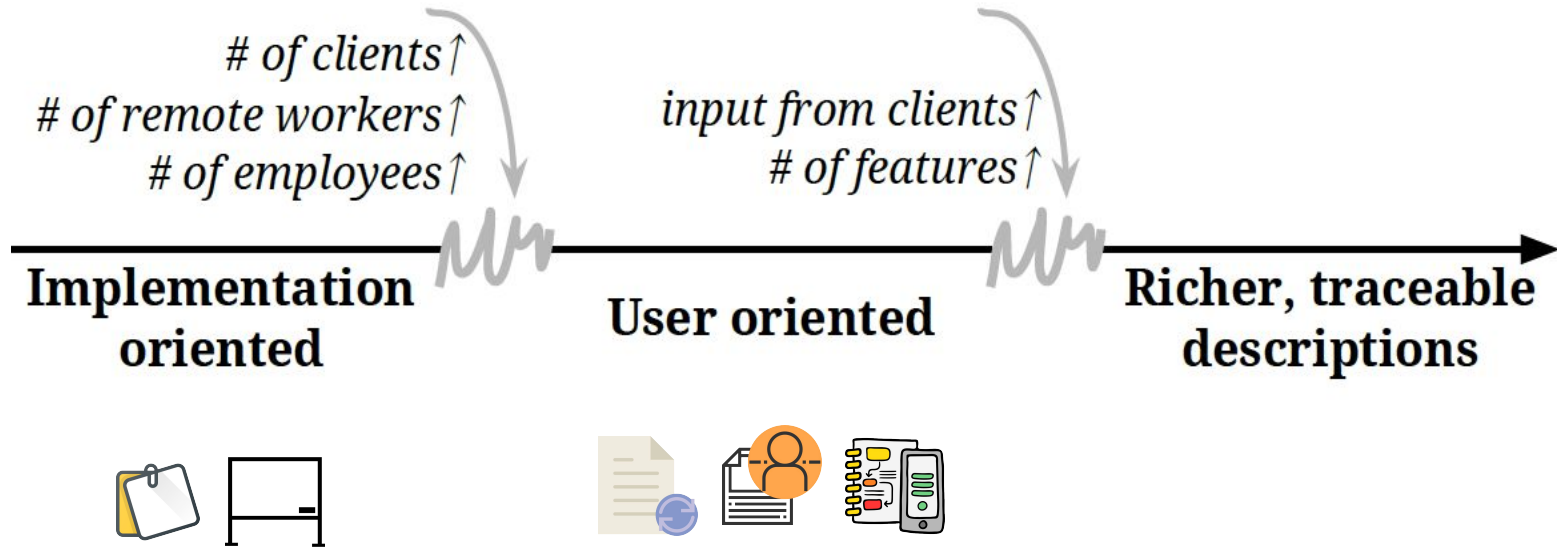
theoretical
saturation

REQUIREMENTS PRACTICE EVOLUTION



REQUIREMENTS ARTEFACTS:

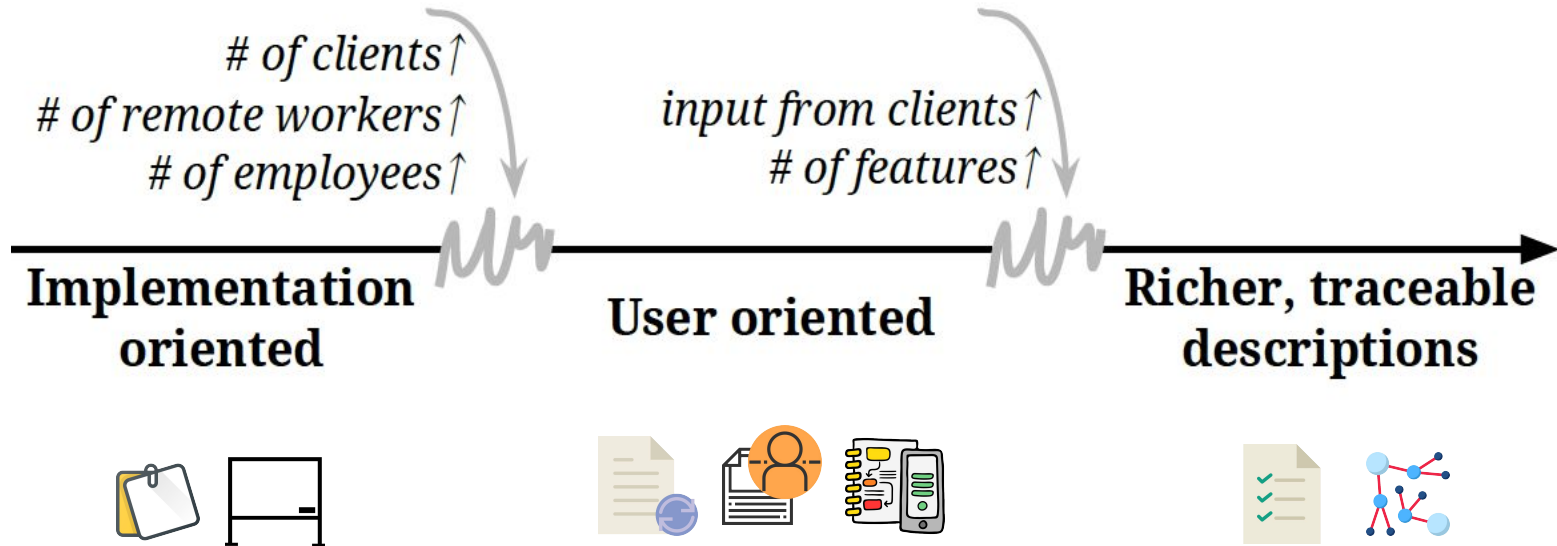
content of information and user orientation matters



*“It was mainly because we had more and more **clients** (...) We need to know their **needs** when we are writing code, so (...) **user stories** are important”*

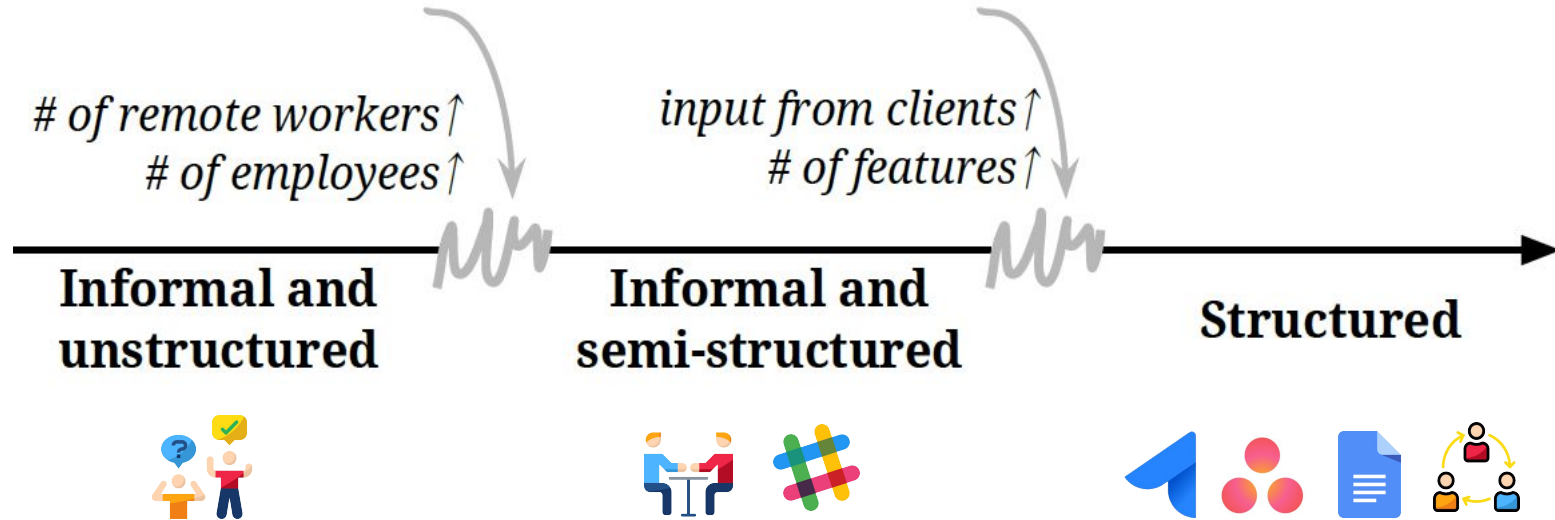
REQUIREMENTS ARTEFACTS:

content of information and user orientation matters



KNOWLEDGE MANAGEMENT:

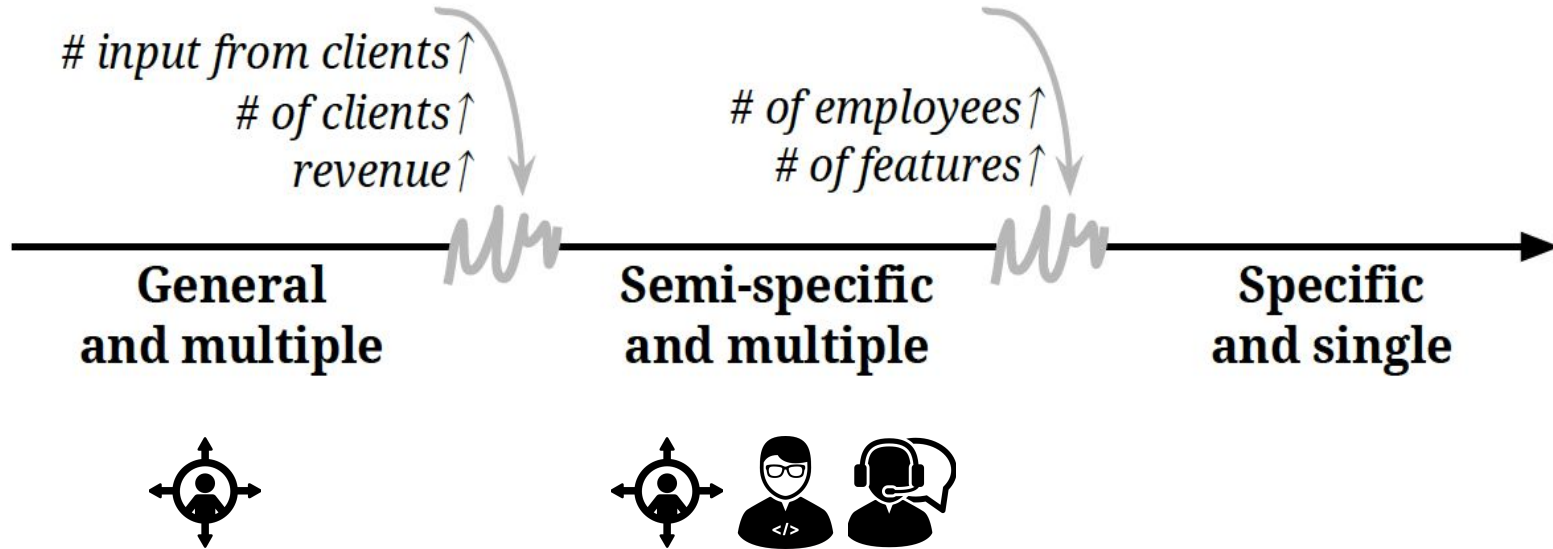
project communication and documentation matters



*“It’s possible to have good practices and improve the **knowledge dissemination** earlier because the tools are there (...) but there were more important things to do.”*

REQUIREMENTS-RELATED ROLES:

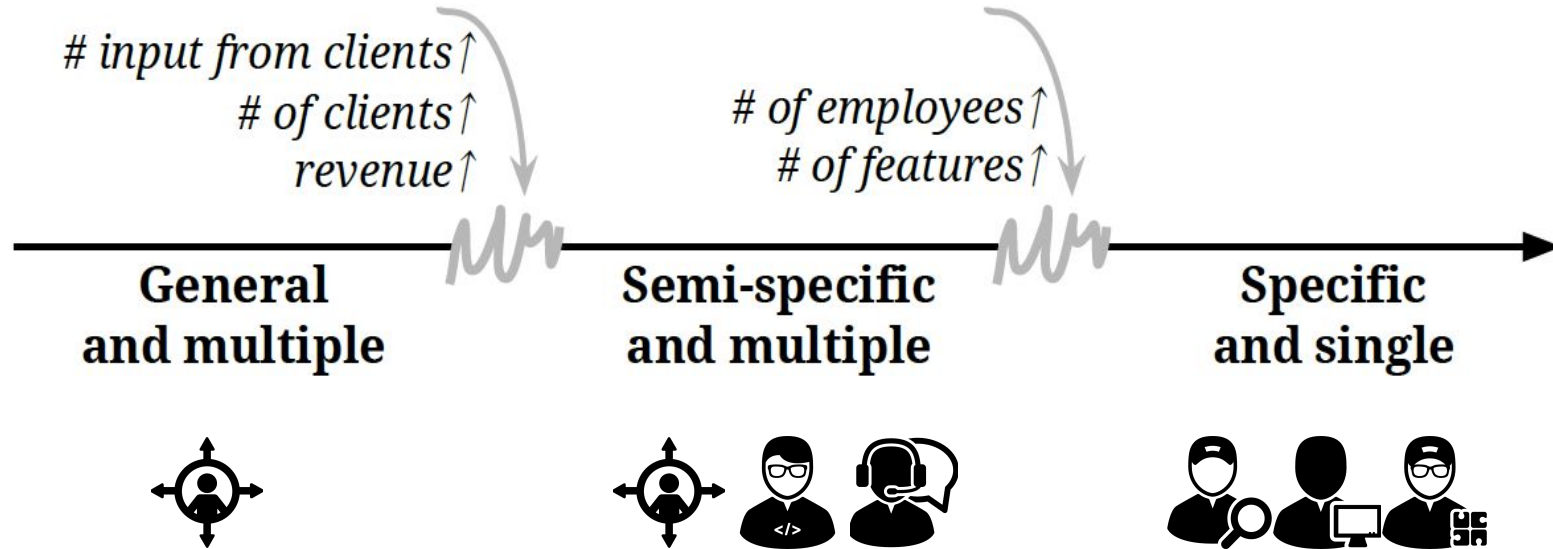
focusing on **customer-facing** roles matters



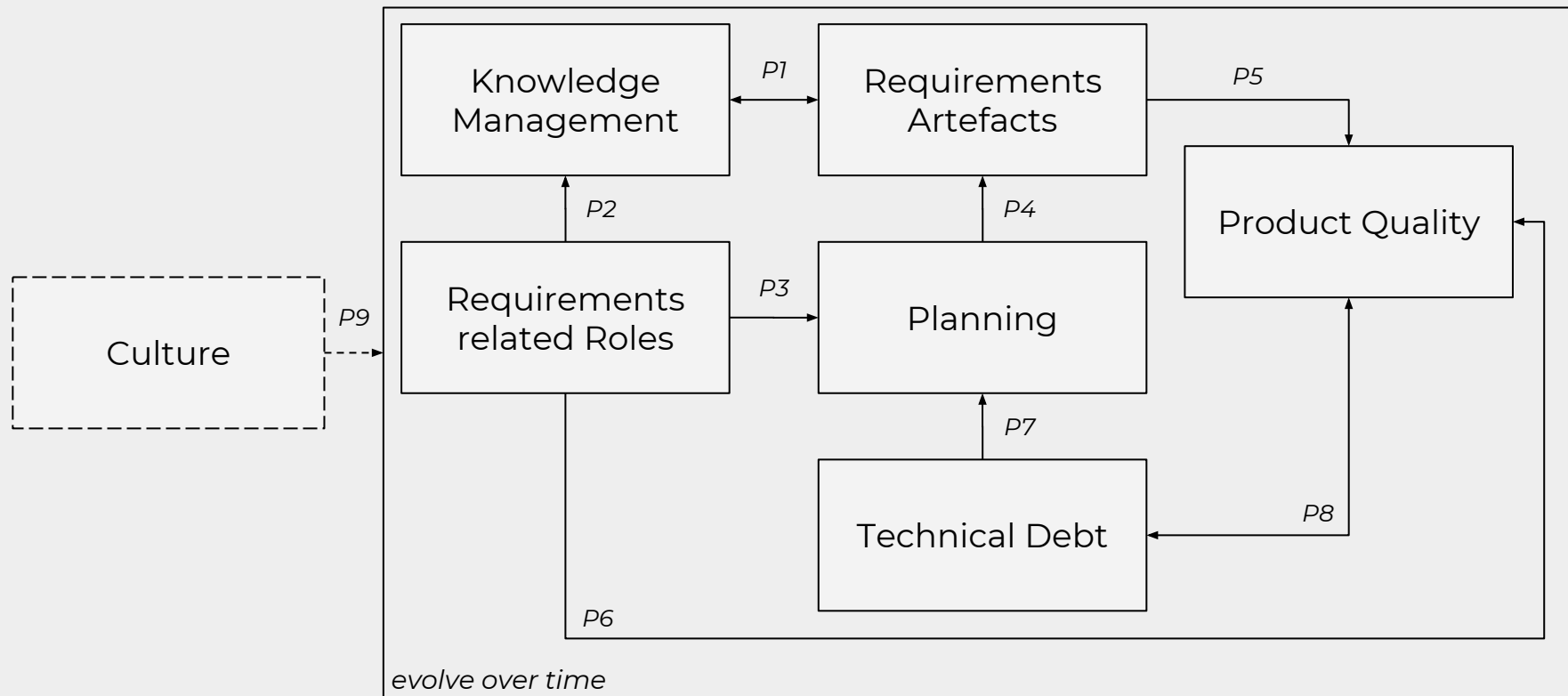
*“We started hiring more people for specific roles.
We had **developers** (...) we hired a **client
success manager** to stay on track of all of our
clients. We still need to be more specialised.”*

REQUIREMENTS-RELATED ROLES:

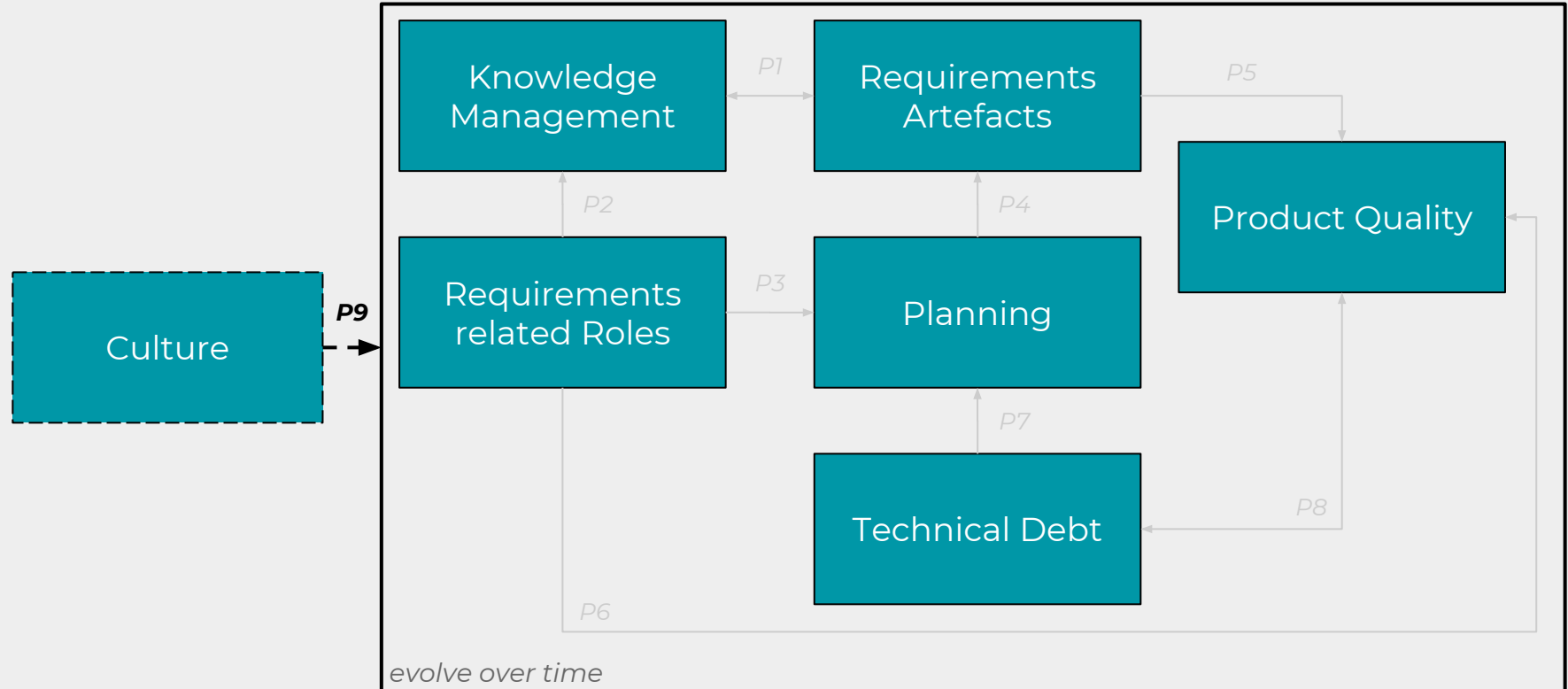
focusing on customer-facing roles matters



RELATIONSHIPS AMONG DIMENSIONS IN OUR THEORY



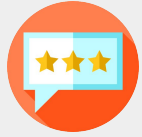
All changes are influenced by a combination of company **culture** and the **co-founders' backgrounds**



TOWARDS RE FOR STARTUPS



no initial idea of a final product



client and market feedback matters



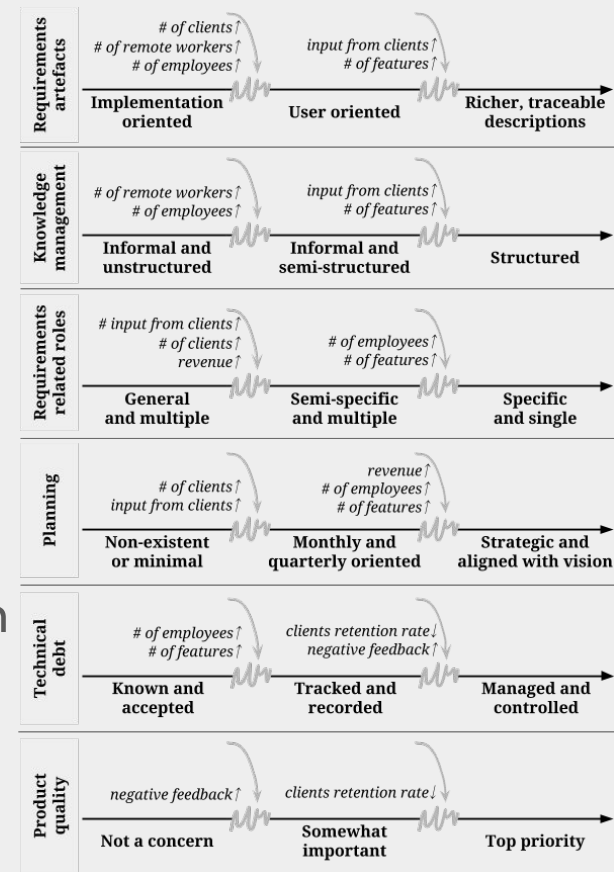
evolution towards a structured, plan-, documentation- and client-oriented approach



reactive changes, when they bring benefits



pragmatic lightness towards an *engineering* of requirements



IS **EVOLUTION** ALONG THE 6
DIMENSIONS FUNDAMENTAL TO
THE **SUCCESS** OF A STARTUP?

NO

but it helps!

Requirements artefacts

1 = Implementation oriented, 2 = User oriented,
3 = Richer, traceable specifications



Product quality

1 = Not a concern,
2 = Somewhat important,
3 = Top priority

Knowledge management

1 = Informal and unstructured,
2 = Informal and semi-structured,
3 = Structured

Technical debt

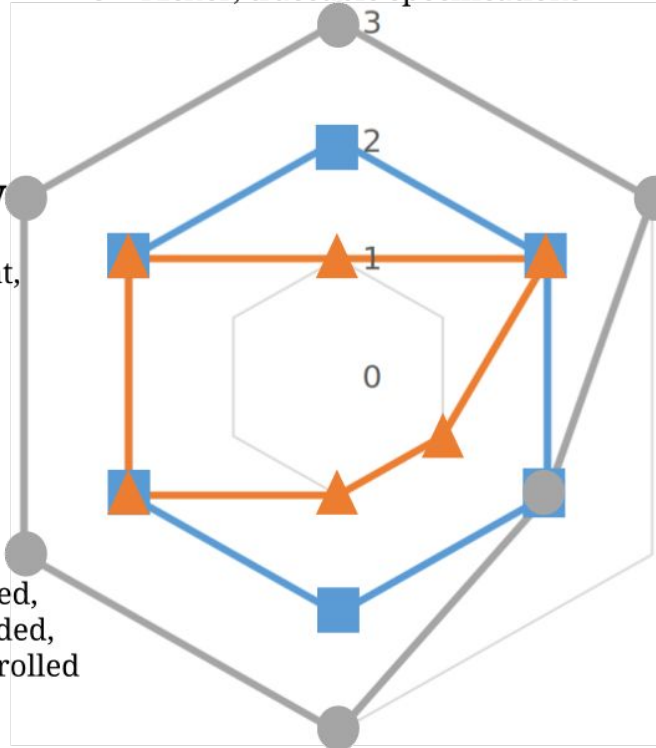
1 = Known and accepted,
2 = Tracked and recorded,
3 = Managed and controlled

Requirements-related roles

1 = General and multiple,
2 = Semi-specific and multiple,
3 = Specific and single

Planning

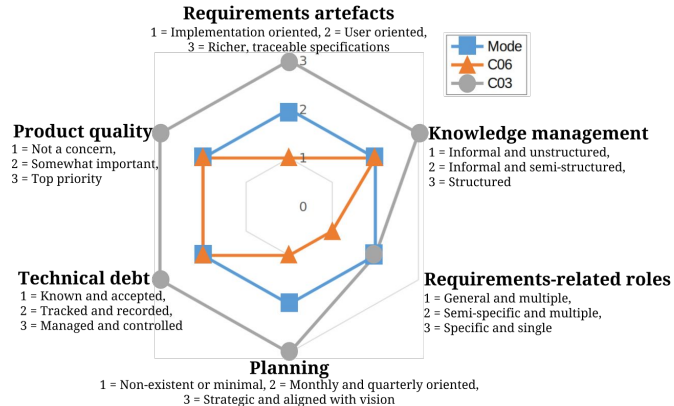
1 = Non-existent or minimal, 2 = Monthly and quarterly oriented,
3 = Strategic and aligned with vision



IMPLICATIONS FOR PRACTICE

track evolution

*place itself along
each dimension*



IMPLICATIONS FOR PRACTICE

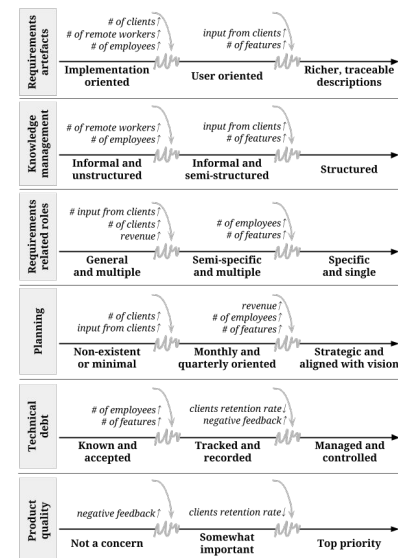
track evolution

*place itself along
each dimension*



plan ahead

*insight about how to
address a turning point*



IMPLICATIONS FOR PRACTICE

track evolution

*place itself along
each dimension*



plan ahead

*insight about how to
address a turning point*



culture

*CEO,
co-founders*



IMPLICATIONS FOR PRACTICE

track evolution

*place itself along
each dimension*



plan ahead

*insight about how to
address a turning point*



culture

*CEO,
co-founders*



people and
their beliefs

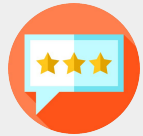
change over time



TOWARDS RE FOR STARTUPS



no initial idea of a final product



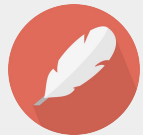
client and market feedback matters



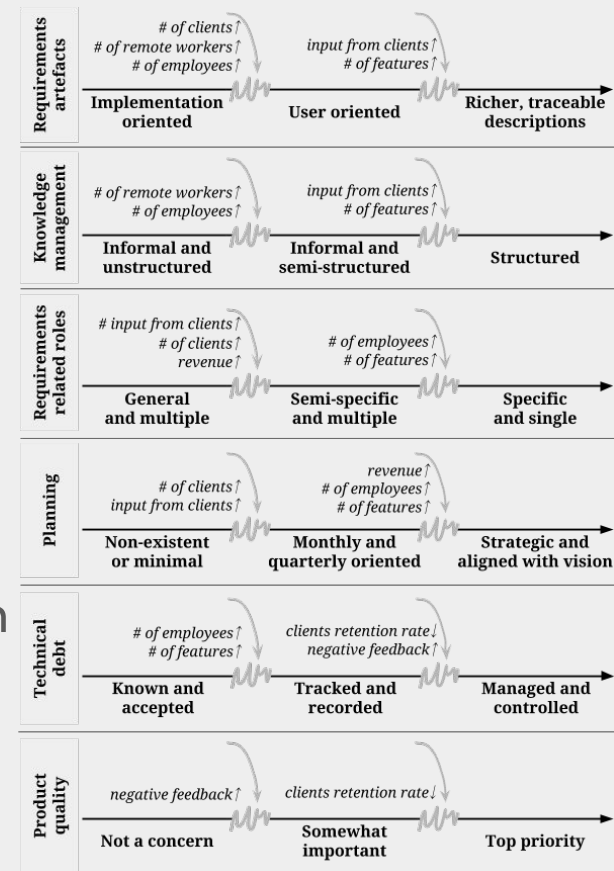
evolution towards a structured, plan-, documentation- and client-oriented approach



reactive changes, when they bring benefits



pragmatic lightness towards an *engineering* of requirements



THANK YOU

QUESTIONS?

