UiO : University of Oslo

IN5431 – spring 2026

CIO Toolbox 2:

Projects, products and design thinking

Note: All images in the presentation are copyrighted and cannot be used or redistributed separately

Agenda

- Background
- Organizing implementation: *Projects*
- "Design thinking“ : an explorative approach to making choices
- Organizing implementation: *Product teams*

From last week: On making choices

- For any organization and most CIOs, making informed *choices* is among the most important abilities.
- Choices come in all variants, from
  - "what will be the best approach to establish a new office in Sweden: hire employees on our own, or purchase a company?"

  to

  - "which PC vendor should we choose?"

From last week: Business case

- A structured and quantitative *business case* is a popular approach to structure decision making.
- For a given alternative approach, the business case estimates “net present value”:
  - Annualized cash flow, i.e., the expected income minus expected cost - per year.
  - The risk, i.e. the probability that either income or cost differ from the estimates

Alternative A , risk premium is 2%

$$NPV_A = \sum_{i=1}^{5} \frac{100\,000}{(1{,}12)^i} - 0 = 360\,478$$

Alternative B , risk premium is 10%

$$NPV_B = \sum_{i=1}^{5} \frac{200\,000}{(1{,}20)^i} - 350\,000 = 248\,122$$

From last week: The «plus/minus»-method

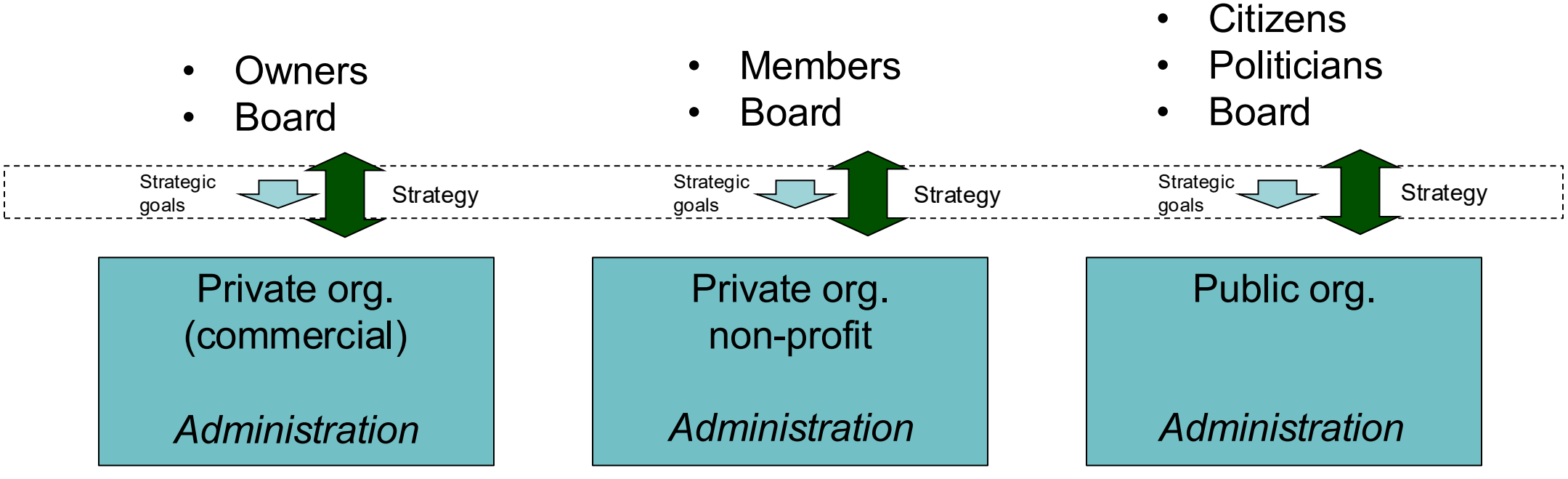
	K2 Order in own house	K10 Coordinated regional centers	K12 Order in the educational sector	K14 The knowledge center
Initial phase	5	10	10	15
Investment	50	100	70	125
Tax financial cost	10	20	15	30
Sum	65	130	95	170
Yearly mainenance	9	8	12	10
Rank cost	**1**	**3**	**2**	**4**
Reduced effort, consumers	+	++	++	+++
Reduced effort, data producers	0	++	+	++
Increased use of data	++	+++	+++	+++
Correct use of data	++	+++	+++	++++
Improved data protection	++	+++	+++	++++
Rank benefits	**4**	**2**	**3**	**1**
Risk	Low	Medium /high	Medium	High
Real options	High	Medium	High	Low
Overall evaluation	**3**	**2**	**1**	**4**

Adapted from https://www.regjeringen.no/contentassets/0ff78b4f861b43e8a7e52d5f02dbbd34/sluttrapport-fremtidig-deling-av-data-1.1.pdf (in Norwegian)
See also https://dfo.no/sites/default/files/Fagomr%C3%A5der/Utredninger/Veileder-i-samfunnsokonomiske-analyser.pdf (in Norwegian), section 3.4.8

Recall from our lecture on strategy that the result of a strategy process is an action plan

- This **action plan** contains:
  1. Activities to change or develop the current operation to meet strategic goals
  2. Assigned responsibility for these activities
  3. Expected ordering and timeframe for executing the activities (roadmap)
  4. Estimates/budgets
- Note: in modern organizations, there is a huge debate regarding what level of detailed planning one should perform. This can have a great impact on the action plan with respect to the level of detail, e.g. whether the activities are detailed in advance or simply outlined – and how precise the estimates are expected to be.
- However, it does not change the fundamental requirements for a working plan.
- Actions to implement strategic change could be projects, reorganizations or simply other priorities in day-to-day operation.

Different organizations



Regardless of organization type or strategy, there is a set of *expectations* from the surrounding stakeholders towards the Administration.

From the CIO toolbox: Projects

Tool	Typical purpose
Business case	Prioritization of digital services and funding
Alternative analysis	Vendor selection and product choice
Design thinking	• Prioritization of digital services and funding • Vendor selection and product choice
IT Architecture	Analyze and structure the IT portfolio – both within and among systems and services
Projects	Plan and organize development
Product teams and agile methods	Plan and organize development
IT governance	How to distribute responsibility for IT among organizational units?

We use the term «CIO toolbox» informally and also subjectively. Although many CIOs use many of these tools, few will use them all.

The main purpose of the methaphor «toolbox» is to emphasize that the tools must not be regarded goals in themselves. They are only meaningful if they serve their purpose.

There is a lot of relevant material we do not cover, management tools exist for everything from (organizational) change management to security testing.

Goodumbrellas Inc

- Production in Vietnam
- Retailers all over the world

For decades, customers have purchased umbrellas through retailers all over the world.

Goodumbrellas Inc

- Production in Vietnam
- Retailers all over the world

http://www.umbrella-heaven.com



For decades, customers have purchased umbrellas through retailers all over the world. The new strategy aims to establish a direct sales web channel – www.umbrella-heaven.com – and that within 3 years, at least 30% of the sales occurs through www.umbrella-heaven.com. The goal is to keep more of the profits.

This is a strategic decision:

For decades, customers have purchased umbrellas through retailers all over the world. The new strategy aims to establish a direct sales web channel – www.umbrella-heaven.com – and that within 3 years, at least 30% of the sales occurs through www.umbrella-heaven.com. The goal is to keep more of the profits.

These are activities in an action plan:

- Develop a web shop
- Establish their own warehouse with capability of shipping small orders on demand
- Integrate with shipping providers to allow users to track process of their order
- Establish a capability for online support and return to consumers

To achieve this, Goodumbrellas Inc must:

- Develop a web shop
- Establish their own warehouse with capability of shipping small orders on demand
- Integrate with shipping providers to allow users to track process of their order
- Establish a capability for online support and return to consumers



Legal issues

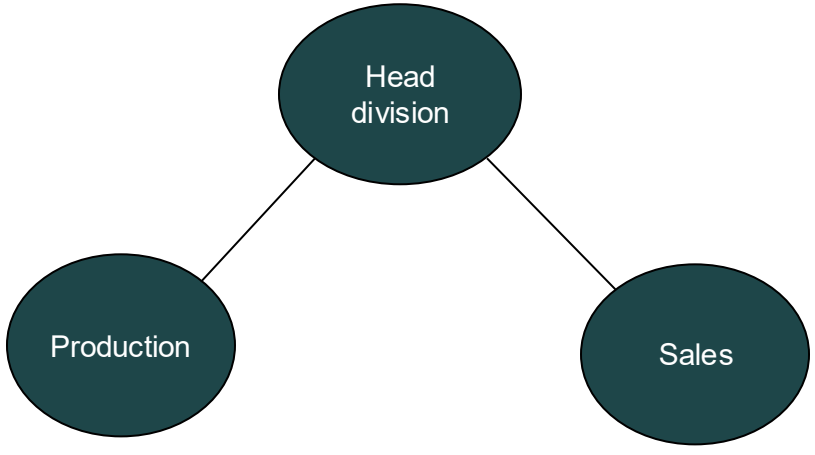
Risk



Time

Coordination challenges

And what about organization?



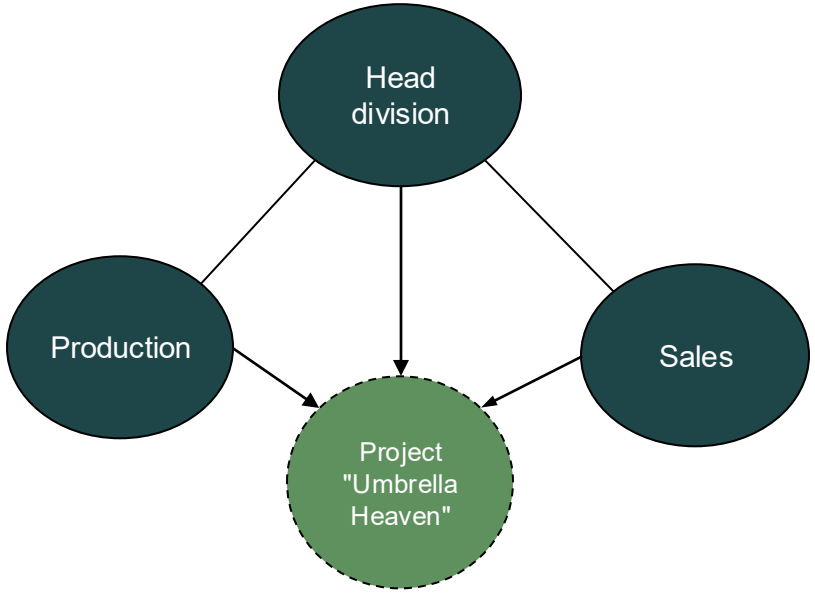
Project Umbrella heaven

- The leader group of Goodumbrellas Inc decides to establish a project to ensure the development and launch of www.umbrella-heaven.com is as predictable and smooth as desired
- The question is: how?

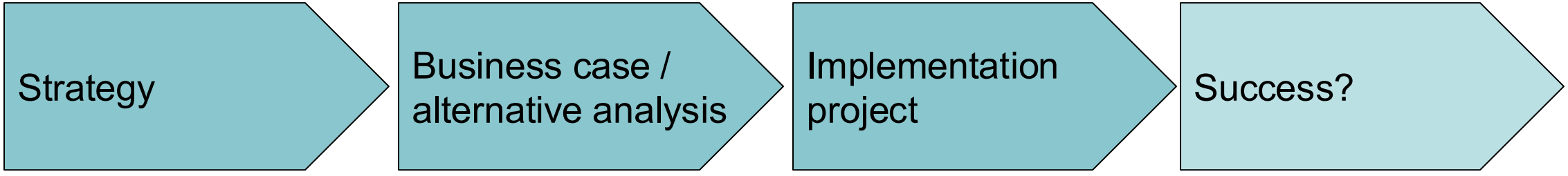


What is a project?

"A project is a temporary organization established to deliver one or more specified results or products within a specified period."



From idea to implementation: the project approach



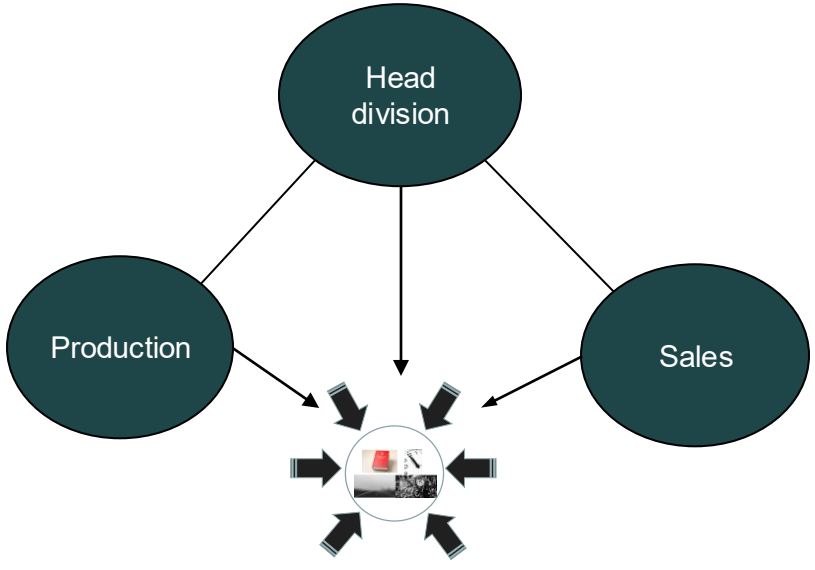
Advantages of the project approach

- Multiple stakeholders can be easily involved in prioritization and planning at all stages
- Complex dependencies among different organizations can be resolved through careful planning and involvement

What are the disadvantages?

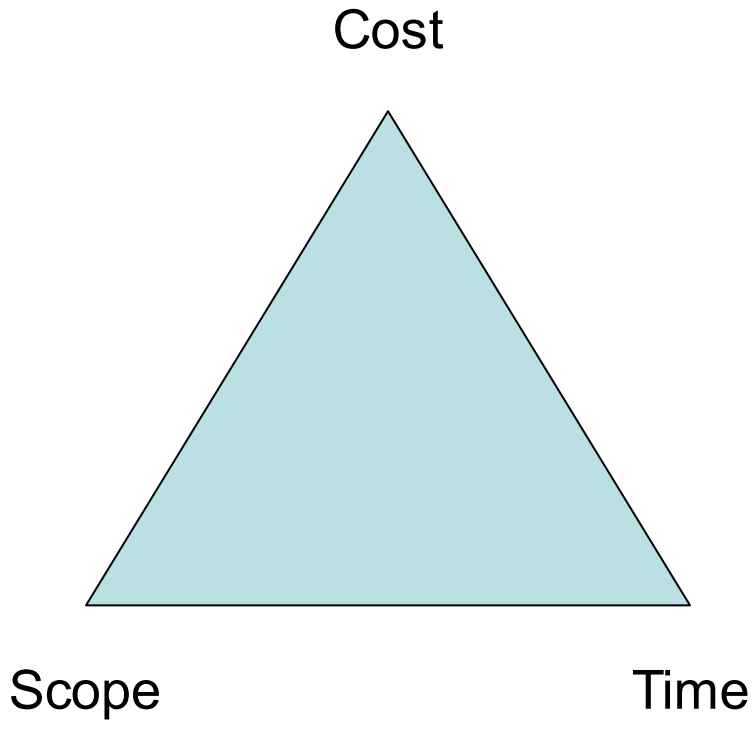
Challenges of the project situation

- There are always *expectations* behind every project
- Funding is granted based on estimated costs and benefits – which is again based on assumptions
- In the presence of *risk*: how to differentiate between unrealistic expectations and incompetent execution?



«The triple constraint»

Projects with fixed cost, scope and time are particularily vulnerable to disappointment.



Selection of agile project management approaches based on project complexity
(Pinciroli 2024)



Received: 1 July 2023 | Revised: 8 July 2024 | Accepted: 9 July 2024
DOI: 10.1002/smr.2716
RESEARCH ARTICLE - METHODOLOGY
Software: Evolution and Process WILEY

Selection of agile project management approaches based on project complexity

Fernando Pinciroli
Universidad Nacional de San Juan, San Juan, Argentina
Correspondence
Fernando Pinciroli, Universidad Nacional de San Juan, Av. Jose Ignacio de la Roza Oeste 590, San Juan, Argentina.
Email: fpinciroli@iinfo.unsj.edu.ar

Abstract
Managing software development projects is a complex endeavor due to the constant emergence of unforeseen events that deviate from initial expectations. A competent project leader is not just someone who follows the planned course but also adept at handling and minimizing inconveniences, ultimately striving to achieve results that align as closely as possible with the desired outcome. However, individuals involved in technological development often cling to familiar tools that have previously yielded positive outcomes, even when those tools may not be the best fit for the current project context. The Agile Manifesto has significantly transformed project management, infusing the discipline with a fresh perspective. Nevertheless, there remain several challenges to overcome. In this article, we aim to provide a guide that addresses these difficulties and minimizes their impact. We explore the selection of key factors that adequately describe a project's complexity, which can subsequently be used in conjunction with the Cynefin framework to categorize management strategies, techniques, and tools based on their applicability to specific complexities. Additionally, we offer insights on adapting project management approaches throughout the project life cycle in response to changes in reality, utilizing the dynamics outlined by the Cynefin framework. Finally, we present suitable strategies, techniques, and tools for agile project management based on the complexity context assigned by the Cynefin framework.

KEYWORDS
agile project management, agnostic agile, complexity management, Cynefin framework

1 | INTRODUCTION

The management of software development projects is a complex process[1] due to the constant emergence of unforeseen events that challenge the initial expectations. A competent project leader is not solely focused on executing the planned tasks but also adept at resolving any inconveniences that may arise in order to achieve a result that aligns as closely as possible with the initial expectations.[2]

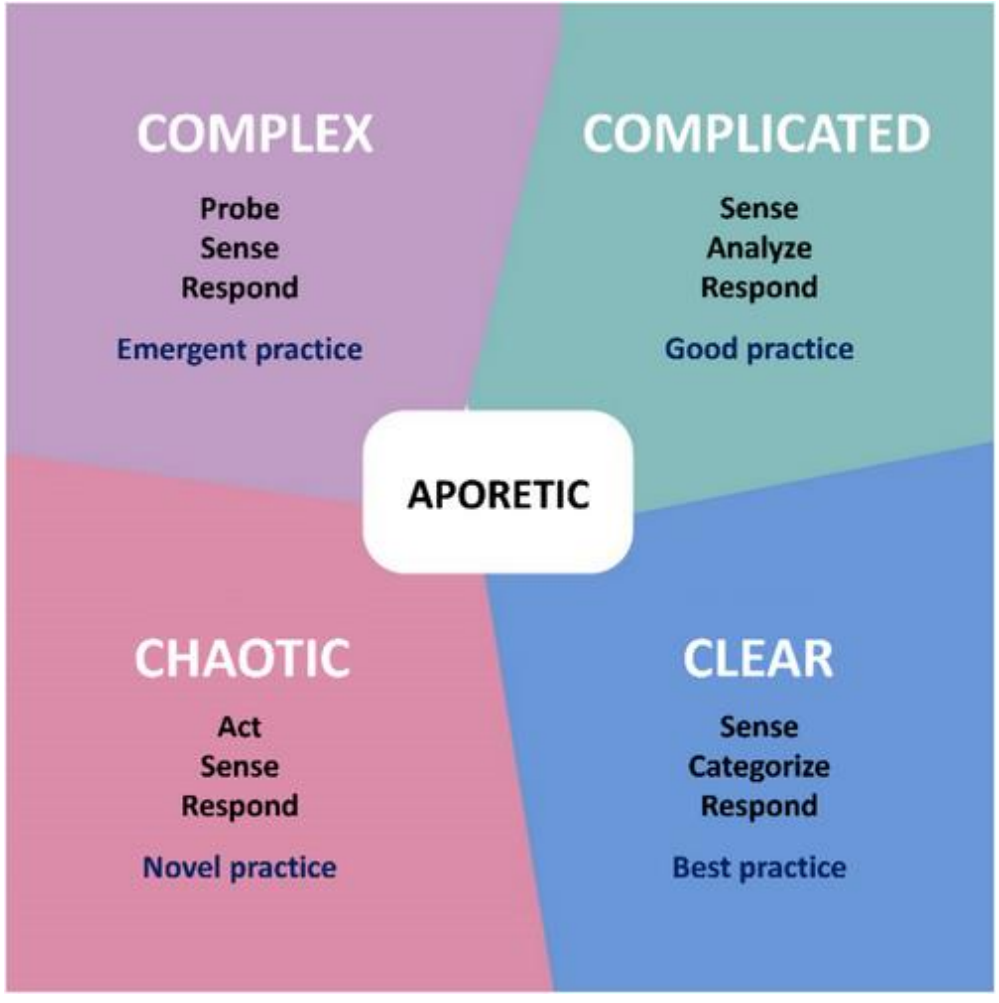
In essence, the ability to adapt to change and effectively handle the unforeseen becomes paramount in managing software development projects. Making optimal decisions based on the evolving reality that unfolds throughout the project is crucial.

Conversely, individuals involved in technological developments often exhibit a tendency to rely on tools that have proven successful in the past, even if they may not be the most suitable choice for a particular situation. Moreover, even when the project context changes, these familiar tools are often preferred over others. Daniel and Daniel elaborate on how mental models influence project management approaches.[3]

Similarly, software development is inherently complex due to the intrinsic nature of software itself. Brooks states that complexity is not an accidental but an essential characteristic of software.[4] This complexity aligns with the characteristics exhibited by complex systems.[5] During the

Cynefin[1)] (*pronounced kuh-NEV-in*) is a framework to support organizations and leaders to choose the the right management approach based on context.

- A *clear* situation means that the issues occuring are typically well known, and can be solved by previously agreed and often written procedures.
- In a *complicated* situation, a lot of non-trivial decisions have to be made – however, the cause-and-effect relationships are still possible to analyze in advance. This is said to be the «domain of experts»
- In a *complex* situation, several important factors influencing the outcome are unknown, and experimentation is typically necessary to look for an approach to move towards the desired outcome.
- In a chaotic situation, there is typically an emergency which requires immediate action to move into a more stable state.

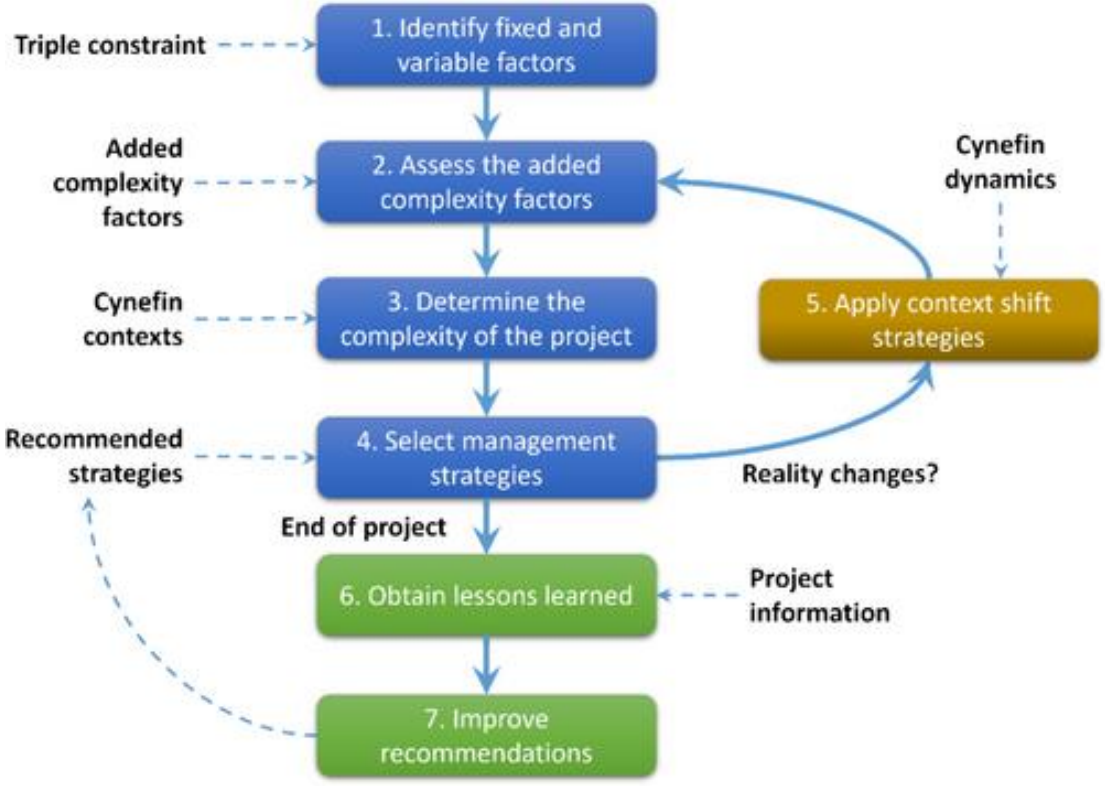


[1)]Snowden, David J., and Mary E. Boone. "A leader's framework for decision making." Harvard business review 85.11 (2007): 68.

The article presents a method to choose the right project management approach based on:

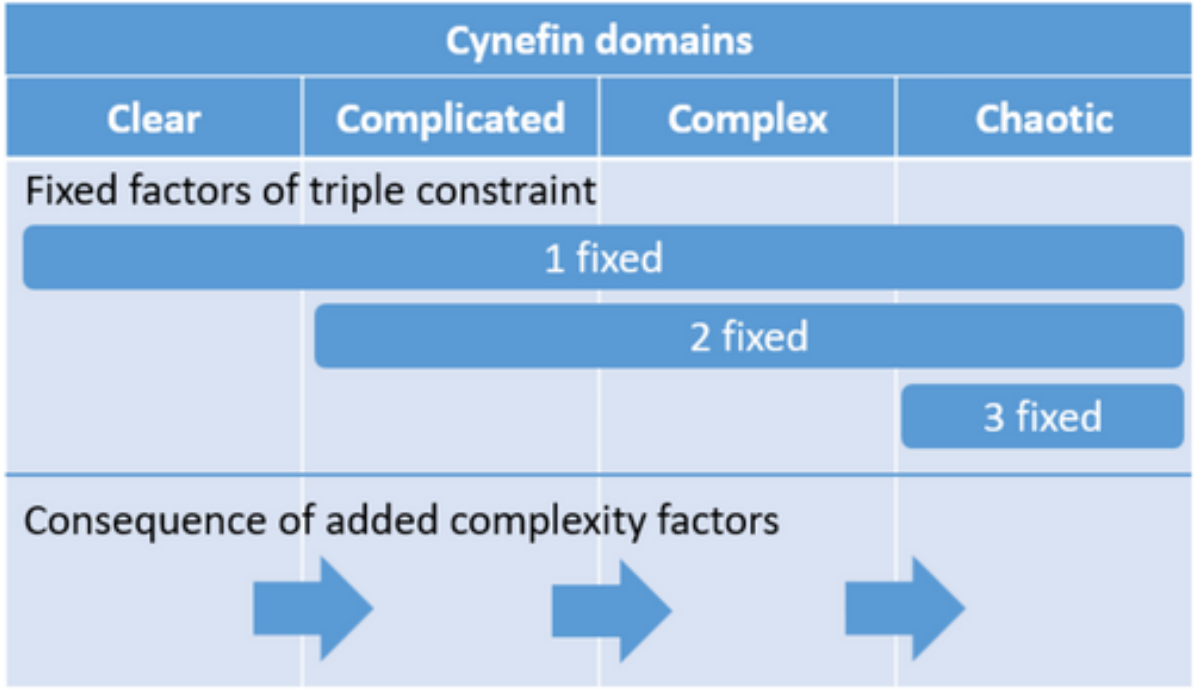
1) How many of the triple constraints elements *time, cost* and *scope* are fixed?
2) What other complexity factors are affecting the project?

The project is mapped to a Cynefin-context based on (1) and (2). Typical management approaches for each context is presented.

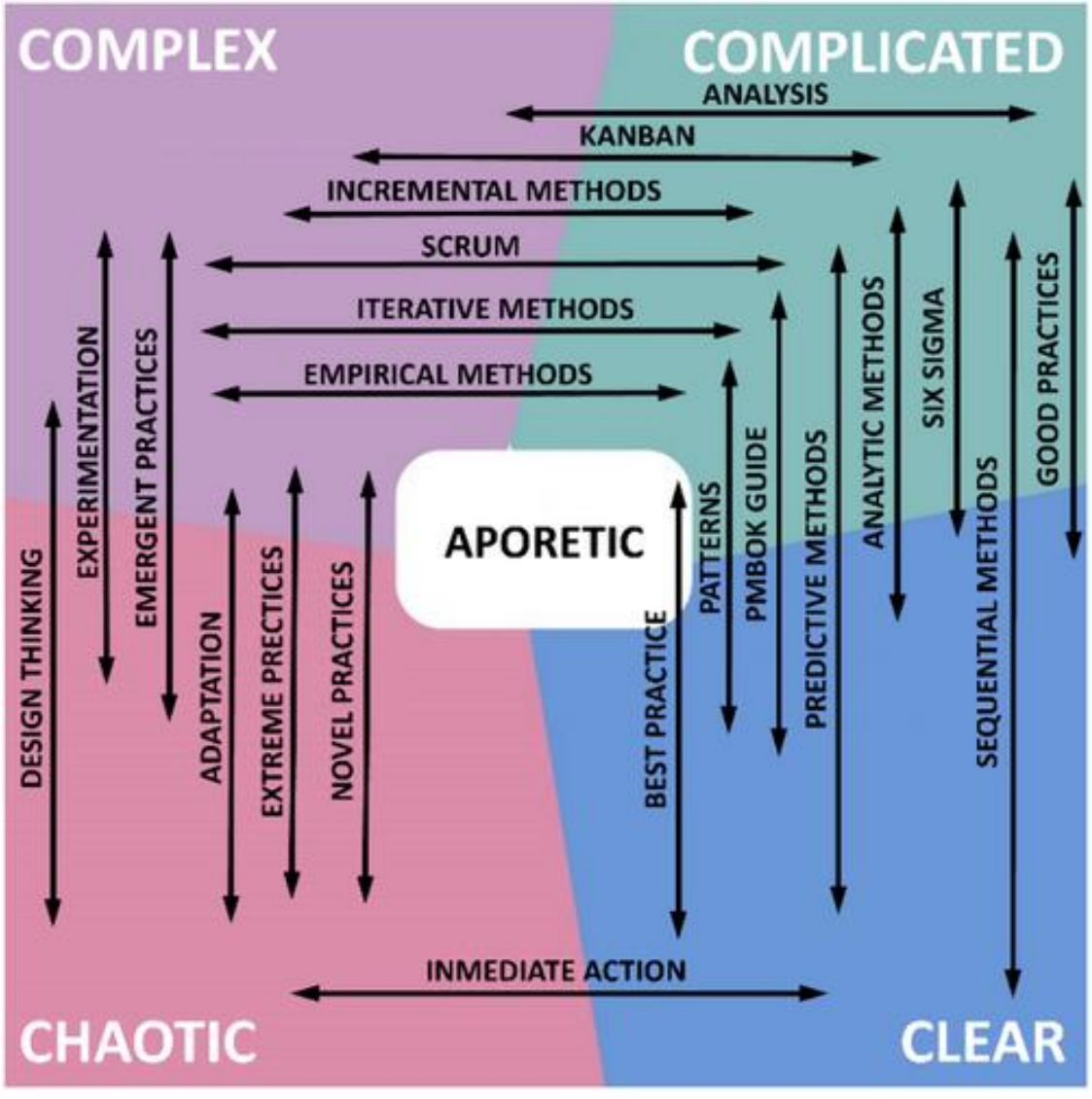


Complexity factors

Factor	Meaning
Team skills	Team skills required for the project: level of demand for specific technical knowledge, seniority, experience, domain knowledge of the problem, team maturity level, etc.
Project's objective	Level of definition, coherence, and consensus of the project objective.
Project size	Size of the project scope.
Project duration	Temporal duration of the project.
Selected process for conducting the project	It aims at the traditional classification between heavy or lightweight processes, based mainly on the number of prescriptions the process has. Today, we can talk about traditional, agile, or hybrid processes.
Stability of requirements	Degree or probability of stability of requirements over time: high or low.
Client involvement	Level of client involvement or availability: high or low.
Complexity	Range of complexity: very high, high, medium, low, very low.
Documentation requirements	Quantity, detail, comprehensiveness, coverage, and level of validation of the documentation required for the project.
Project cost	Level of economic investment required to complete the project.
Technology involved	Type of required technology: known or unknown, new or proven. Also, breakthrough, current, best of breed, practical. 28
Response to changes	Requirement for flexibility and response times to changes.
Interdependences	Degree of interdependence between parts of the problem or resources.
Project risks	Quantity, rigor, and consequences of the risks that may affect the project, as well as possible alternatives to mitigate them.



Context	Fixed factors of the triple constraint	Number of added complexity factors
Clear	1	2
	2	0
Complicated	1	3
	2	2
Complex	1	3+
	2	2
	3	0
Chaotic	1	4+
	2	2+
	3	1+


COMPLEX
COMPLICATED
ANALYSIS
KANBAN
INCREMENTAL METHODS
SCRUM
ITERATIVE METHODS
EMPIRICAL METHODS
EXPERIMENTATION
EMERGENT PRACTICES
DESIGN THINKING
ADAPTATION
EXTREME PRECTICES
NOVEL PRACTICES
APORETIC
BEST PRACTICE
PATTERNS
PMBOK GUIDE
PREDICTIVE METHODS
ANALYTIC METHODS
SIX SIGMA
SEQUENTIAL METHODS
GOOD PRACTICES
INMEDIATE ACTION
CHAOTIC
CLEAR

The CIO toolbox

Tool	Typical purpose
Business case	Prioritization of digital services and funding
Alternative analysis	Vendor selection and product choice
Design thinking	• Prioritization of digital services and funding • Vendor selection and product choice
IT Architecture	Analyze and structure the IT portfolio – both within and among systems and services
Projects	Plan and organize development
Product teams and agile methods	Plan and organize development
IT governance	How to distribute responsibility for IT among organizational units?

We use the term «CIO toolbox» informally and also subjectively. Although many CIOs use many of these tools, few will use them all.

The main prupose of the methaphor «toolbox» is to emphasize that the tools must not be regarded goals in themselves. They are only meaningful if they serve their purpose.

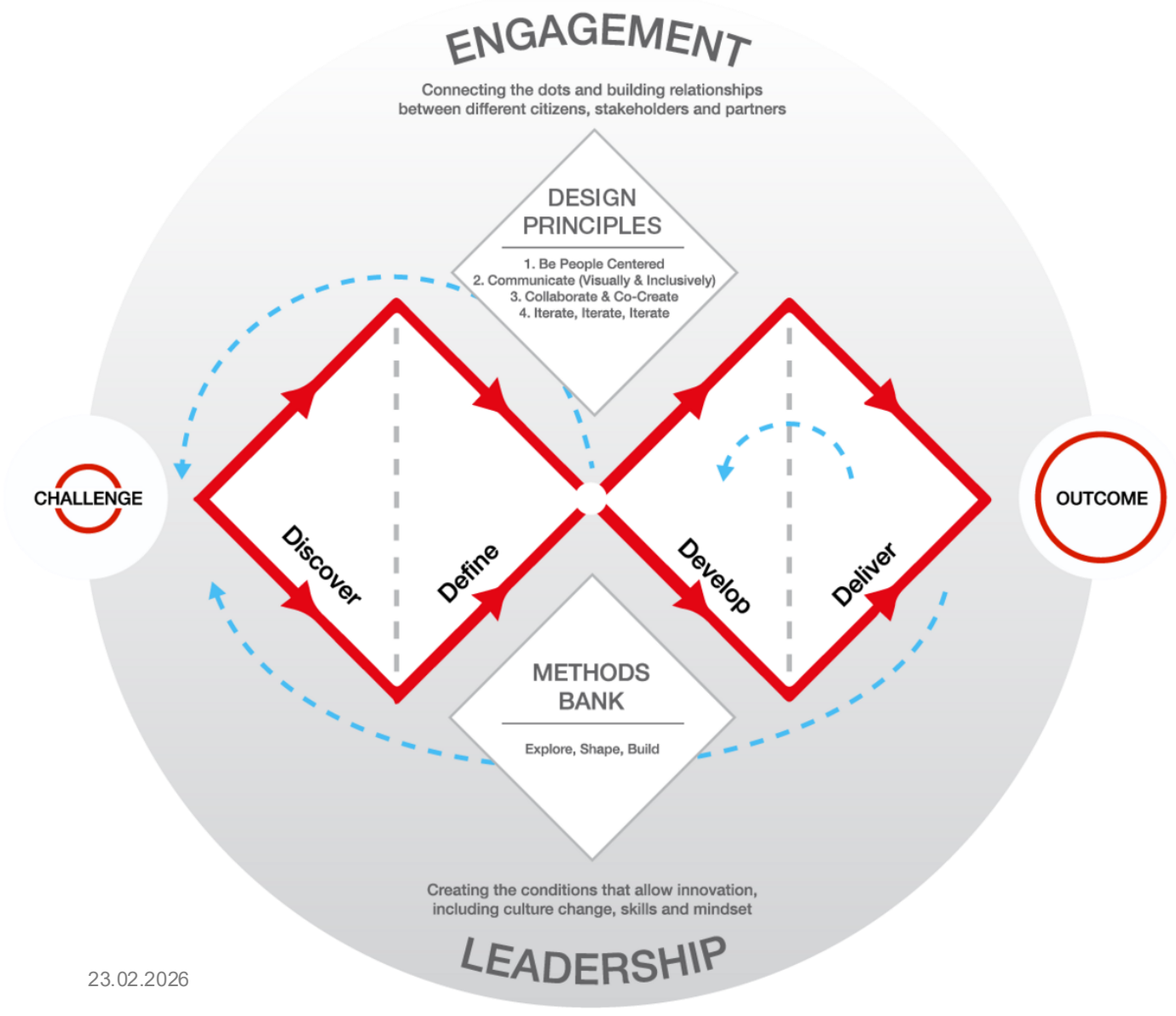
There is a lot of relevant material we do not cover, management tools exist for everything from (organizational) change management to security testing.

Design thinking: an explorative approach to making choices

Design Thinking (DT) is not only now a motor for innovation promoted by designers, but it **offers new models of processes and toolkits which help to** improve, accelerate and visualise every creative process, carried out not only by designers, but in multidisciplinary teams in any kind of organisation. The new use of the term DT, specifically the combination of "thinking" and "design", offers fields such as Innovation Management the opportunity to **apply design tools to other problem-solving-contexts not directly related with the appearance and functionality of artefacts, but with the form of businesses, services and processes**. **Design Thinking** today is not only a cognitive process or a mindset, but **has become an effective toolkit for** any innovation process, **connecting the creative design approach to traditional business thinking, based on planning and rational problem solving.**

Tschimmel, Katja. "Design Thinking as an effective Toolkit for Innovation." *ISPIM Conference Proceedings*. The International Society for Professional Innovation Management (ISPIM), 2012.

Example: the double diamond



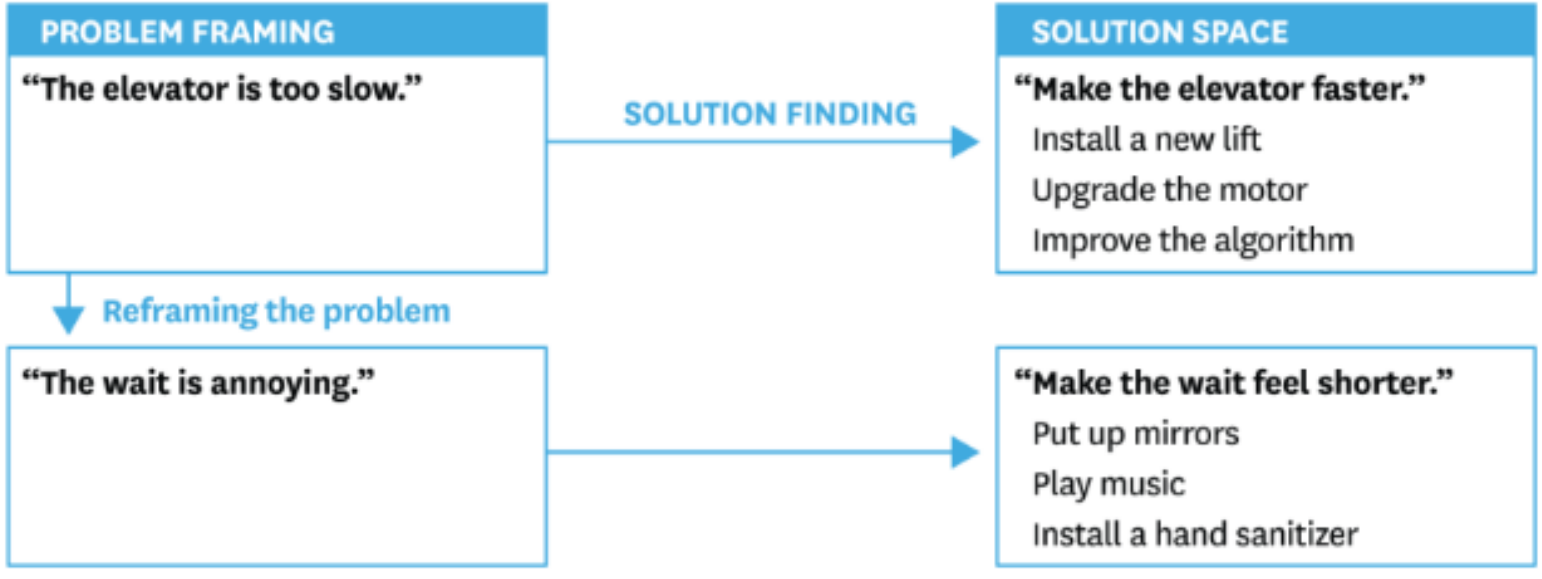
The four phases:

- **Discover**: The first diamond helps people understand, rather than simply assume, what the problem is. It involves speaking to and spending time with people who are affected by the issues.
- **Define**: The insight gathered from the discovery phase can help you to define the challenge in a different way.
- **Develop**: The second diamond encourages people to give different answers to the clearly defined problem, seeking inspiration from elsewhere and co-designing with a range of different people.
- **Deliver**:Delivery involves testing out different solutions at small-scale, rejecting those that will not work and improving the ones that will.

Source:

Framework for Innovation - Design Council

The «Discover» phase



"Note that the initial framing of the problem is not necessarily wrong. Installing a new lift would probably work. The point of reframing is not to find the "real" problem but, rather, to see if there is a better one to solve. In fact, the very idea that a single root problem exists may be misleading; problems are typically multicausal and can be addressed in many ways."

Source:
Wedell-Wedellsborg, Thomas. "Are you solving the right problems." Harvard Business Review 95.1 (2017): 76-83.

The CIO toolbox: Product teams and agile methods

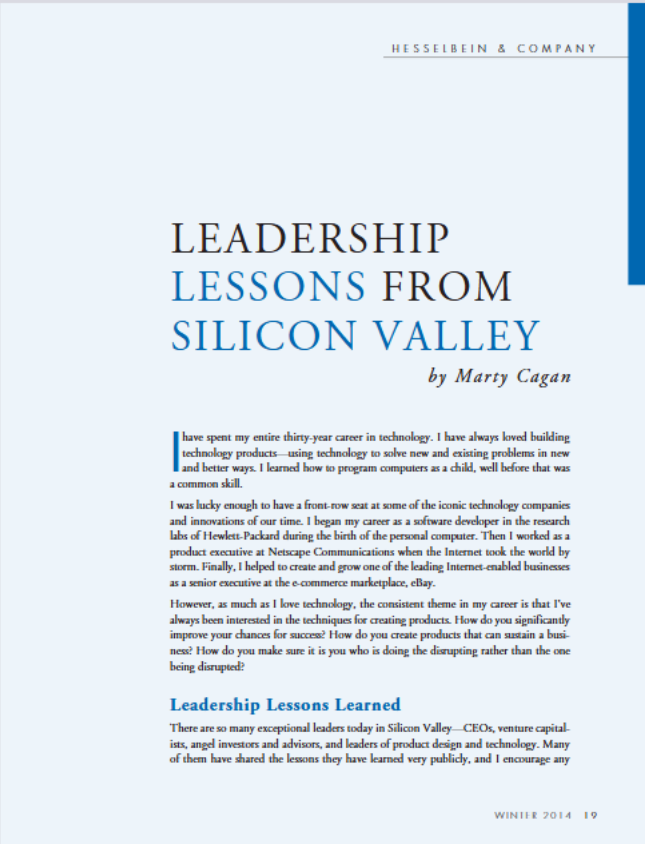
Tool	Typical purpose
Business case	Prioritization of digital services and funding
Alternative analysis	Vendor selection and product choice
Design thinking	• Prioritization of digital services and funding • Vendor selection and product choice
IT Architecture	Analyze and structure the IT portfolio – both within and among systems and services
Projects	Plan and organize development
Product teams and agile methods	Plan and organize development
IT governance	How to distribute responsibility for IT among organizational units?

We use the term «CIO toolbox» informally and also subjectively. Although many CIOs use many of these tools, few will use them all.

The main purpose of the methaphor «toolbox» is to emphasize that the tools must not be regarded goals in themselves. They are only meaningful if they serve their purpose.

There is a lot of relevant material we do not cover, management tools exist for everything from (organizational) change management to security testing.

Product teams and agile development



HESSELBEIN & COMPANY

LEADERSHIP LESSONS FROM SILICON VALLEY

by Marty Cagan

I have spent my entire thirty-year career in technology. I have always loved building technology products—using technology to solve new and existing problems in new and better ways. I learned how to program computers as a child, well before that was a common skill.

I was lucky enough to have a front-row seat at some of the iconic technology companies and innovations of our time. I began my career as a software developer in the research labs of Hewlett-Packard during the birth of the personal computer. Then I worked as a product executive at Netscape Communications when the Internet took the world by storm. Finally, I helped to create and grow one of the leading Internet-enabled businesses as a senior executive at the e-commerce marketplace, eBay.

However, as much as I love technology, the consistent theme in my career is that I've always been interested in the techniques for creating products. How do you significantly improve your chances for success? How do you create products that can sustain a business? How do you make sure it is you who is doing the disrupting rather than the one being disrupted?

Leadership Lessons Learned

There are so many exceptional leaders today in Silicon Valley—CEOs, venture capitalists, angel investors and advisors, and leaders of product design and technology. Many of them have shared the lessons they have learned very publicly, and I encourage any

WINTER 2014 19

«I'd like to share with you what I personally consider the ten most important lessons of how Silicon Valley creates new products that drive disruption"

Note: this is not The Truth. However, it is an important perspective: Marty Cagan wrote the books *Inspired, Empowered* and *Transformed* – on how to achieve efficient, product-oriented organizations

Product teams and agile development

Ten Leadership Lessons from the Silicon Valley

1. Changing the world—or disrupting your industry—requires a strong product vision and a committed, inspired, and passionate team.
2. Know what you can't know.
3. Know what your customers can't know.
4. Make product discovery a core competency.
5. Create and empower dedicated product teams.
6. Focus on outcome not output.
7. Design is not something added at the end—it must be a part of the product from Day 1.
8. Foster true collaboration.
9. Embrace pivots.
10. Nurture a culture of innovation.

Important takeaways:

- Storytelling: inspire passion through a strong product vision (see lesson 1)
- Design thinking (see lesson 2, 3, 4, 7 and 9)
- Ensure continuity by creating lasting *product* teams instead of projects (see lesson 5)
- Expect pivotal change (see lesson 9)
- Give team members autonomy to innovate and explore – focus on outcome, not output (lesson 5, 6, 10)

Challenges of autonomous product teams (examples)

- How to balance the need for alignment and team autonomy in complex organizations and multiteam environments?
- What is the right degree of autonomy in different contexts (and how to measure it)?
- How should agile practices be adjusted to promote effectiveness in cross-functional teams?



Autonomous agile teams:

Challenges and future directions for research

Viktoria Stray
University of Oslo, SINTEF
Norway
stray@ifi.uio.no

Nils Brede Moe
SINTEF
Norway
nilsm@sintef.no

Rashina Hoda
University of Auckland
New Zealand
r.hoda@auckland.ac.nz

ABSTRACT

According to the principles articulated in the agile manifesto, motivated and empowered software developers—relying on technical excellence and simple designs—create business value by delivering working software to users at regular short intervals. These principles have spawned many practices. At the core of these practices is the idea of autonomous, self-managing, or self-organizing teams whose members work at a pace that sustains their creativity and productivity. This article summarizes the main challenges faced when implementing autonomous teams and the topics and research questions that future research should address.

CCS CONCEPTS

• Software and its engineering → Software development process management, Programming teams, Agile software development

KEYWORDS

autonomy, self-organizing, self-management, teams, coordination, collaboration, communication, continuous learning, agile practices, software engineering, research agenda

ACM Reference format:

Viktoria Stray, Nils Brede Moe, and Rashina Hoda. Autonomous agile teams: Challenges and future directions for research. XP 2018, 19th International Conference on Agile Software Development. In *Proceedings of XP'18 Companion, Porto, Portugal, May 2018*, 5 pages. https://doi.org/10.1145/3234152.3234182

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.

XP '18 Companion, May 21–25, 2018, Porto, Portugal
© 2018 Copyright is held by the owner/author(s).
ACM ISBN 978-1-4503-6422-5/18/05.
DOI: 10.1145/3234152.3234182

1 INTRODUCTION

To succeed in complex environments, organizations developing software have to find ways to support and regulate their teams' autonomy according to the environmental demands and limitations. Furthermore, they have to take into consideration the degree of change and uncertainty, and that there is no one-size-fits-all autonomy approach [1]. We have researched the topic of autonomous teams in agile software development for some time [2-4], but the process of designing, supporting, and coaching autonomous agile teams is still not adequately addressed and understood in the context of software development organizations. Further, there is a need for new knowledge on how companies should organize for the right level of team autonomy and utilize autonomous agile teams to attain better performance, productivity, innovation, and value creation, and thus, increase competitiveness. One emerging question is "How can organizations give autonomous agile teams the authority and competence to set directions for new products so that organizations can deliver innovative and high-quality software more rapidly?"

To address the challenges of implementing autonomous teams, we organized the first international workshop on autonomous agile teams at XP 2018 (The 19th International Conference on Agile Software Development) to understand better the specific challenges and to suggest a future research agenda. The goal of the workshop was to facilitate knowledge sharing about the current practice of autonomous agile teams and deepen the knowledge about practices and strategies that enable autonomous teams. We use the label "autonomous teams" as a synonym for "self-organizing teams," "self-managing teams," and for "empowered teams."

1.1 Autonomous teams: the origins

The concept of autonomous teams is not new and has been studied and described from various perspectives in the past [2]. One of the earliest references dates back to the Tavistock group's study of English coal miners [5]. From this socio-technical perspective, autonomous teams were described as teams of 10-15 cross-trained individuals guided by the corporate vision, motivated by peer-pressure, and taking on

Stray, V., Moe, N. B., & Hoda, R. (2018). Autonomous agile teams: challenges and future directions for research. *Proceedings of the 19th International Conference on Agile Software Development: Companion*, 1–5. https://doi.org/10.1145/3234152.3234182

Tool	Typical purpose
Business case	Prioritization of digital services and funding
Alternative analysis	Vendor selection and product choice
Design thinking	• Prioritization of digital services and funding • Vendor selection and product choice
IT Architecture	Analyze and structure the IT portfolio – both within and among systems and services
Projects	Plan and organize development
Product teams and agile methods	Plan and organize development
IT governance	How to distribute responsibility for IT among organizational units?

Are these tools good?

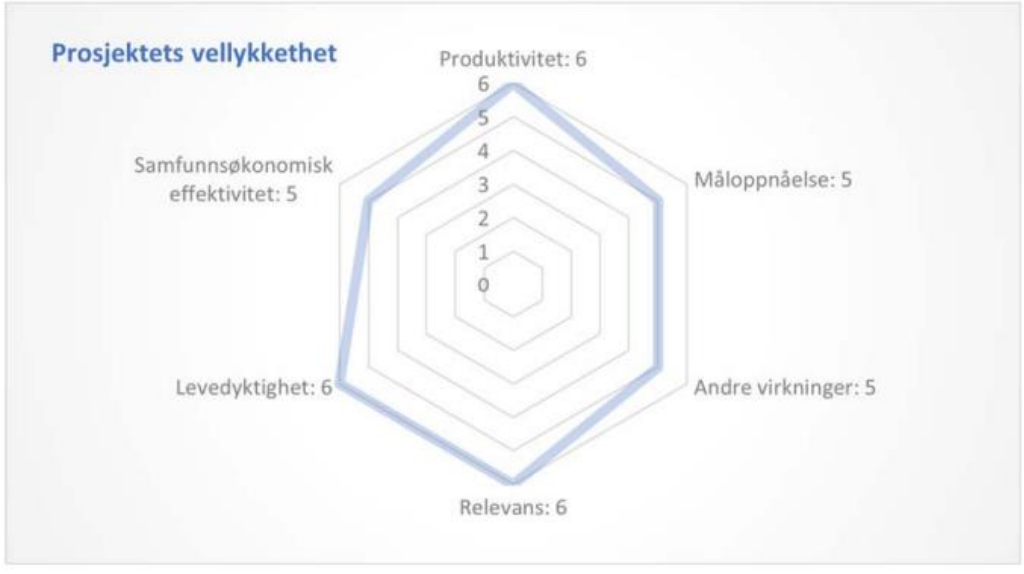
And these bad?



Statens vegvesens prosjekt ble omtalt som en skandale i årevis. Nå revolusjonerer det bilsektoren.

Å redde Veivesenets store IT-prosjekt fra totalhavari har kostet mange år og kroner, men nå er det blitt en suksess. Prosjektlederen mener andre offentlige digitaliseringer kan lære av det de har vært gjennom.

Kilde: Aftenposten



Kilde: Ettervaluering av prosjektet Autosys kjøretøy
SINTEF – NTNU Concept

Key characteristics:

- «Modernization»-project with a business case
- Agile elements within the project

Parts of the scope (self-service functions) delivered through a (relatively) autonomous product team

SPACEMAKER

Norske Spacemaker kjøpes av software-giganten Autodesk

17.11.2020 11:41:50 CET | Spacemaker | Pressemelding

Del

I dag ble det ble det offentliggjort at det norske AI-selskapet Spacemaker kjøpes opp av Autodesk [NASDAQ: ADSK] for 2,2 milliarder kroner. I tillegg kommer selskapets kontanter og et aksjeprogram for ansatte på ca. 185 millioner kroner. Spacemaker, som ble startet i 2016, bruker kunstig intelligens for å optimalisere eiendomsprosjekter og bidrar til å utvikle mer bærekraftige byer og nabolag over hele verden. På kort tid har selskapet vokst til 115 ansatte fordelt på kontorer i syv land. Nå blir de en del av Autodesk, et av verdens største selskaper innenfor AEC-programvare.



De tre gründerne av Spacemaker. Fra venstre; Carl Christensen (CTO), Håvard Haukeland (CEO) og Anders Kvåle (arbeidende styreformann). Foto: Spacemaker

Kilde: NTB

Spacemaker was built with autonomous product teams

«This document was created on the belief that instead of making strict rules on how we build products and services in Unified Design @ Autodesk, we believe in alignment on how, by creating some guiding principles. We strongly believe that time to value is best served by autonomy and thereby letting our people choose the right tool for the job every time. In that we also say that we choose a little chaos over control and big platform architectures.
Why is autonomy important, and what is it, really? It's not about autonomy for autonomy's sake, or because it makes people in the squad happy - it's about **time to value."**

Source GitHub - spacemakerai/unified-design-tech-core-beliefs: Core beliefs, our guiding principles for time to value; speed, quality, reliability and agility

Tool	Typical purpose
Business case	Prioritization of digital services and funding
Alternative analysis	Vendor selection and product choice
Design thinking	• Prioritization of digital services and funding • Vendor selection and product choice
IT Architecture	Analyze and structure the IT portfolio – both within and among systems and services
Project organization	Plan and organize development
Product teams and agile methods	Plan and organize development
IT governance	How to distribute responsibility for IT among organizational units?

There is no evidence that one approach is – generally – superior.

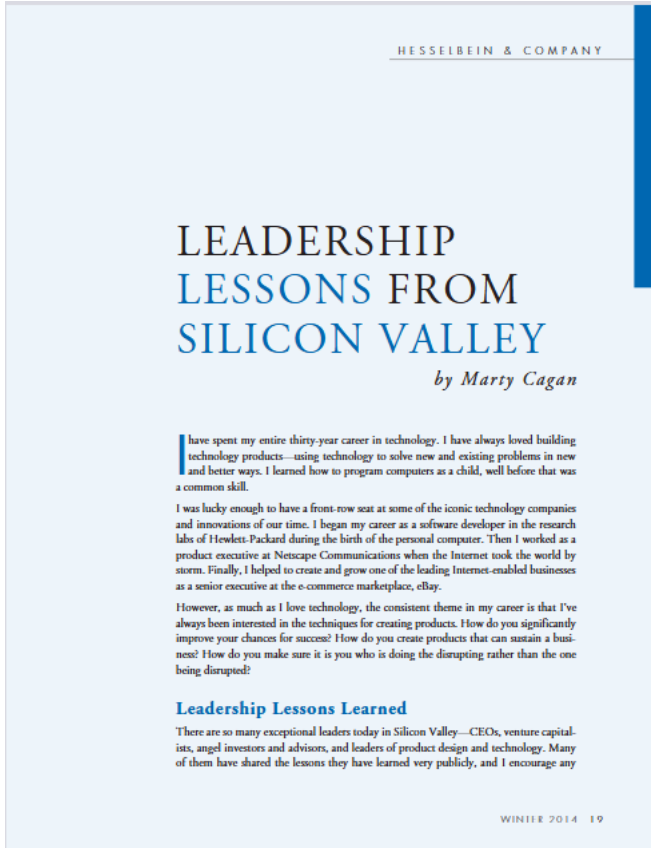
As usual, it depends a lot of context: improving the *operational backbone* of your organization may require other tools than developing new *digital offerings*. You will hear more about this in *Designed for digital.*

Are these tools good?

And these bad?

(19) Design thinking for every endeavour | Robyn Richardson | TEDxCreativeCoast - YouTube

Readings



HESSELBEIN & COMPANY

LEADERSHIP LESSONS FROM SILICON VALLEY

by Marty Cagan

I have spent my entire thirty-year career in technology. I have always loved building technology products—using technology to solve new and existing problems in new and better ways. I learned how to program computers as a child, well before that was a common skill.

I was lucky enough to have a front-row seat at some of the iconic technology companies and innovations of our time. I began my career as a software developer in the research labs of Hewlett-Packard during the birth of the personal computer. Then I worked as a product executive at Netscape Communications when the Internet took the world by storm. Finally, I helped to create and grow one of the leading Internet-enabled businesses as a senior executive at the e-commerce marketplace, eBay.

However, as much as I love technology, the consistent theme in my career is that I've always been interested in the techniques for creating products. How do you significantly improve your chances for success? How do you create products that can sustain a business? How do you make sure it is you who is doing the disrupting rather than the one being disrupted?

Leadership Lessons Learned

There are so many exceptional leaders today in Silicon Valley—CEOs, venture capitalists, angel investors and advisors, and leaders of product design and technology. Many of them have shared the lessons they have learned very publicly, and I encourage any



Received: 1 July 2023 | Revised: 8 July 2024 | Accepted: 9 July 2024
DOI: 10.1002/smr.2716

RESEARCH ARTICLE - METHODOLOGY

Software: Evolution and Process WILEY

Selection of agile project management approaches based on project complexity

Fernando Pinciroli

Universidad Nacional de San Juan, San Juan, Argentina

Correspondence
Fernando Pinciroli, Universidad Nacional de San Juan, Av. Jose Ignacio de la Roza Oeste 590, San Juan, Argentina.
Email: fpinciroli@dinfo.unsj.edu.ar

Abstract

Managing software development projects is a complex endeavor due to the constant emergence of unforeseen events that deviate from initial expectations. A competent project leader is not just someone who follows the planned course but also adept at handling and minimizing inconveniences, ultimately striving to achieve results that align as closely as possible with the desired outcome. However, individuals involved in technological development often cling to familiar tools that have previously yielded positive outcomes, even when those tools may not be the best fit for the current project context. The Agile Manifesto has significantly transformed project management, infusing the discipline with a fresh perspective. Nevertheless, there remain several challenges to overcome. In this article, we aim to provide a guide that addresses these difficulties and minimizes their impact. We explore the selection of key factors that adequately describe a project's complexity, which can subsequently be used in conjunction with the Cynefin framework to categorize management strategies, techniques, and tools based on their applicability to specific complexities. Additionally, we offer insights on adapting project management approaches throughout the project life cycle in response to changes in reality, utilizing the dynamics outlined by the Cynefin framework. Finally, we present suitable strategies, techniques, and tools for agile project management based on the complexity context assigned by the Cynefin framework.

KEYWORDS
agile project management, agnostic agile, complexity management, Cynefin framework

1 | INTRODUCTION

The management of software development projects is a complex process[1] due to the constant emergence of unforeseen events that challenge the initial expectations. A competent project leader is not solely focused on executing the planned tasks but also adept at resolving any inconveniences that may arise in order to achieve a result that aligns as closely as possible with the initial expectations.[2]

In essence, the ability to adapt to change and effectively handle the unforeseen becomes paramount in managing software development projects. Making optimal decisions based on the evolving reality that unfolds throughout the project is crucial.

Conversely, individuals involved in technological developments often exhibit a tendency to rely on tools that have proven successful in the past, even if they may not be the most suitable choice for a particular situation. Moreover, even when the project context changes, these familiar tools are often preferred over others. Daniel and Daniel elaborate on how mental models influence project management approaches.[3]

Similarly, software development is inherently complex due to the intrinsic nature of software itself. Brooks states that complexity is not an accidental but an essential characteristic of software.[4] This complexity aligns with the characteristics exhibited by complex systems.[5] During the

Summary

Summary

- Projects are ad-hoc organizations used to organize development. They are intended to solve a specific problem within a given budget and timeframe.
- Design-thinking is an explorative approach both to decision making and development.
- Autonomous product teams, focusing on outcome not output, has been a key factor to the success of most modern startups.
- Both projects or autonomous product teams can be useful or troublesome.
- However, any approach will benefit from passion and storytelling.

