



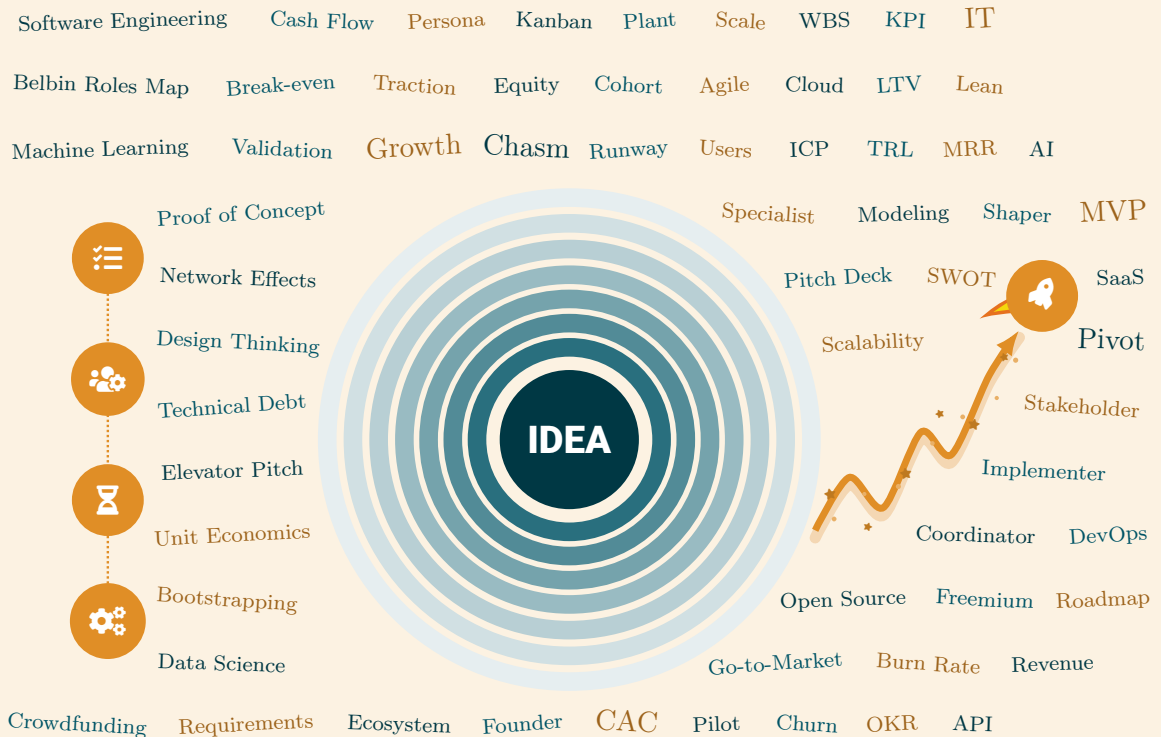
Tamara Papić | Pavle Dakić

Entrepreneurship, Innovation Management and Business Skills in the IT Sector

A Step-by-Step Innovation Practicum

From the first idea to a defended business plan

First Edition • 2026



Singidunum University

Tamara Papić

Pavle Dakić

**ENTREPRENEURSHIP, INNOVATION
MANAGEMENT AND BUSINESS
SKILLS IN THE IT SECTOR
– A STEP-BY-STEP INNOVATION
PRACTICUM**

1st edition

Belgrade, 2026

**ENTREPRENEURSHIP, INNOVATION MANAGEMENT
AND BUSINESS SKILLS IN THE IT SECTOR
– A STEP-BY-STEP INNOVATION PRACTICUM**

Authors:

Tamara Papić

Pavle Dakić

Reviewers:

Marko Tanasković

Mladen Veinović

Publisher:

SINGIDUNUM UNIVERSITY

32 Danijelova Street, Belgrade

www.singidunum.ac.rs

For the publisher:

Milovan Stanišić

Prepress:

Pavle Dakić

Tamara Papić

Cover Design:

Pavle Dakić

Tamara Papić

Year:

2026

Circulation:

eBook - digital copy

Printed by:

Singidunum University, Belgrade

ISBN: 978-86-7912-869-0

Copyright © 2026 Singidunum University, Belgrade and its authors. All rights reserved. No part of this work covered by the copyright herein may be reproduced, transmitted, stored or used in any form or by any means graphic, electronic, or mechanical, including but not limited to photocopying, recording, scanning, digitizing, taping, Web distribution, information networks, or information storage and retrieval systems, without the prior written permission of the publisher.

*“Only those who will risk going too far can possibly find
out how far one can go.”*

T. S. Eliot, 1931

*“Innovation is the specific instrument of entrepreneurship; the act
that endows resources with a new capacity to create wealth.”*

Peter F. Drucker, 1985

This practicum was developed for the courses *Practical Research and Business Skills*, *Entrepreneurship in Software Engineering*, *Innovation Management*, and *Entrepreneurship and Innovations in AI*.

Typeset by the authors with L^AT_EX, using the standard `book` document class at a 24 cm trim. Body text is set in Computer Modern (T1, `cm-super`); titles, covers and display type are set in Roboto, with icons from Font Awesome 5. Figures, diagrams, the concentric-ring motif, and the part-illustrations were produced as original vector graphics with TikZ/PGF. Cover and page design, including the original cover and part-illustration graphics, by the authors. References are formatted with `natbib` in the `IEEEtran` style; the index is generated with `imakeidx` and the glossary with `glossaries`. The PDF was produced with pdfL^AT_EX from T_EX Live 2026, using `tikz`, `tcolorbox`, `hyperref` with `hyperxmp`, `microtype`, `geometry`, `fancyhdr`, `titlesec` and `pdfpages`, among others. Printed and bound in Belgrade by Singidunum University. First Edition, 2026.

Preface

This practicum grew out of a simple observation made over many years of teaching entrepreneurship, innovation management, and business skills to software and data engineering students: our students rarely lack ideas, and they almost never lack technical talent. What they lack is a *guided path* — a structured way to walk an idea from a vague spark to a defended, numbers-backed business plan, making smart mistakes along the way rather than fatal ones.

The practicum you are holding is that path. It is structured as the development of one real innovation — yours — from the first idea to a complete business plan and pitch deck. Every chapter gives you a compact theoretical foundation, a set of guided steps, concrete tasks with templates, a worked example from real student projects, and a recurring exercise that no other practicum we know of offers: the *Concentric Rings*, a systems-thinking model in which you gradually build a map of your venture as a living system and learn to see how a change in one part of it propagates through all the others.

The book serves four courses at once: *Practical Research and Business Skills*, *Entrepreneurship in Software Engineering*, *Innovation Management*, and *Entrepreneurship and Innovations in AI*. Their syllabi overlap far more than students usually realise; the Course Coverage Matrix on page viii shows exactly where each course lives in these pages.

We wrote this book to be used, not read. Write in it, fill the templates, argue with the examples, and let your rings move.

Authors
Tamara Papić, Pavle Dakić
Belgrade, 2026

Predgovor

Ovaj praktikum proizašao je iz jednostavnog zapazanja stečenog tokom mnogogodišnjeg podučavanja preduzetništva, inovacionog menadžmenta i poslovnih veština studentima softverskog i inženjerstva podataka: naši studenti retko nemaju ideje, i gotovo nikada im ne nedostaje tehnički talenat. Ono im nedostaje je *vođen put* — struktuirani način da se ideja od nejasne iskre dovede do branjenog, brojkama potkrepljenog poslovnog plana, praveći po putu pametne, a ne kobne greške.

Praktikum koji držite u ruci je taj put. Struktuiran je kao razvoj jedne prave inovacije — vaše — od prve ideje do kompletnog poslovnog plana i prezentacionog materijala. Svako poglavlje daje vam kompaktan teorijski osnov, niz vođenih koraka, konkretne zadatke sa šablonima, rešen primer iz stvarnih studentskih projekata i ponavljajuću vežbu koju nijedan drugi praktikum koji poznajemo ne nudi: *Koncentrični prstenovi*, model sistemskog mišljenja u kojem postepeno gradite mapu vašeg poduhvata kao živog sistema i učite da vidite kako se promena u jednom delu prenosi kroz sve ostale.

Knjiga istovremeno služi četiri kursa: *Practical Research and Business Skills*, *Entrepreneurship in Software Engineering*, *Innovation Management* i *Entrepreneurship and Innovations in AI*. Njihovi programi se daleko više preklapaju nego što studenti obično shvataju; Matrica pokrivanja kurseva na stranici viii tačno pokazuje gde se svaki kurs nalazi na ovim stranicama.

Napisali smo ovu knjigu da se koristi, a ne čita. Pišite u njoj, popunjavajte šablone, osporavajte primere i dozvolite da se vaši prstenovi pomeraju.

Autori
Tamara Papić, Pavle Dakić
Beograd, 2026

How to Use This Practicum

The anatomy of every chapter

Each of the fourteen chapters follows the same rhythm, so you always know where you are:

1. **Learning outcomes** — what you will be able to do afterwards.
2. **First principles** — the theory you need, no more and no less, with references for deeper reading.
3. **Guided steps** — numbered actions that apply the theory to *your* venture, aligned with the Disciplined Entrepreneurship methodology developed at MIT.
4. **Tasks & exercises** — individual and team assignments with explicit deliverables.
5. **Template** — a fill-in canvas or table; all templates are reproduced on single pages in the Templates Appendix.
6. **Worked example** — a real student project walking through the same step.
7. **Systems Thinking Lab — The Concentric Rings** — the signature exercise of this practicum (see below).
8. **AI angle** — how artificial intelligence changes this particular step, in a short box; Chapter 13 is devoted to AI ventures in full.

The Concentric Rings

Your venture is not a checklist; it is a system. The idea sits at the core, surrounded by seven further rings: customer and value, business model, product and assumptions, team and organization, project and risk, financials, and — as the outermost ring — the market, partners and ecosystem you do not control but must engage (the realm of open

innovation). Starting in Chapter 1, you will add one ring per chapter on the workbook sheet at the back of the book. In several chapters you will also *shift a ring* — deliberately change one element — and trace how the disturbance travels through the whole system. By Chapter 14 you will hold a complete, defensible system map of your venture, alongside the business plan and the pitch deck.

What you will finish with

The practicum produces three capstone artifacts, each with its grading rubric in the Appendix:

1. a complete **Business Plan** (assembled in Chapter 11, refined afterwards);
2. a complete **Pitch Deck** with a live pitch delivery (Chapter 12);
3. your **Concentric Rings system map** (defended in Chapter 14);
4. a **complete management toolkit** — the set of all seventeen templates you filled along the way (idea matrix, canvases, risk table, Gantt, budgets...), ready to be reused on every future venture.

Individual or team track

All tasks are designed for a team of two to four, and every team task also works individually. If you work alone, treat the Belbin exercise in Chapter 8 as a map of the roles you will have to cover yourself — or deliberately not cover.

Course Coverage Matrix

Chapter	PRBS	ESE	IM	AI
1. Entrepreneurial Mindset & Opportunity	•	•	•	•
2. Market Research & Opportunity Analysis	•	•		•
3. Customer Discovery & Value Proposition	•	•		•
4. Business Models & Open Innovation		•	•	•
5. Innovation Strategy & the Innovation Process			•	•
6. Product, MVP & Testing Assumptions	•	•	•	•
7. Competitive Positioning & Strategy	•	•	•	
8. Founding Team, Roles & Leadership		•	•	
9. Legal Structure & Intellectual Property		•	•	•
10. Project & Risk Management for Startups		•	•	
11. Financials & the Business Plan	•	•	•	
12. Fundraising & Pitching	•	•		•
13. Building Ventures in the Age of AI		•	•	•
14. Growth, Internationalization & Exit		•	•	

PRBS — Practical Research and Business Skills; ESE — Entrepreneurship in Software Engineering; IM — Innovation Management; AI — Entrepreneurship and Innovations in AI.

Contents

Preface	vi
Predgovor	vii
How to Use This Practicum	viii
Introduction: The Venture as a System	1
 I From Idea to Opportunity	 5
1 Entrepreneurial Mindset and Opportunity Recognition	7
1.1 Everything Starts from the Idea	7
1.1.1 Idea validation	8
1.1.2 Pivot	8
1.1.3 Motivation: mission, vision, values	8
1.1.4 Target customer first	9
1.2 Creativity and Innovation	9
1.2.1 Types of innovation	9
1.3 Where Ideas Come From	11
1.3.1 Idea-generation methods	11
1.3.2 “Why now?”	12
Chapter references	17
 2 Market Research and Opportunity Analysis	 18
2.1 The Market Research Process	18
2.2 Market Segmentation	19
2.3 Primary Market Research (PMR)	20
2.4 Selecting the Beachhead Market	20
2.5 The Option Prioritization Matrix (Opportunity $\times$ Feasibility)	21
2.6 Sizing the Market: TAM–SAM–SOM	23
2.7 Competition, Substitutes, Trends	24

Chapter references	27
3 Customer Discovery and Value Proposition	28
3.1 End User First, Customer Second	28
3.2 The Ideal Customer Profile (ICP)	29
3.3 Persona and the Full Life-Cycle Use Case	30
3.4 Jobs, Pains, Gains — and the Value Proposition Canvas .	31
3.5 Quantify the Value Proposition	33
Chapter references	36
 II Designing the Venture	 37
4 Business Models and Open Innovation	39
4.1 What a Business Model Is	39
4.2 The Nine Building Blocks	40
4.3 Five Patterns Worth Stealing	43
4.4 Open Innovation: Designing the Permeable Boundary . .	43
Chapter references	47
5 Innovation Strategy and the Innovation Process	48
5.1 Innovation Is a Strategy Before It Is a Process	48
5.1.1 The innovation audit	49
5.2 Managing the Process: Funnel, Stage-Gate, Lean	49
5.2.1 Idea management	51
5.3 R&D and New Product Development (NPD)	51
5.4 Diffusion: How Innovations Actually Spread	52
5.5 Measuring Innovation	53
Chapter references	56
6 From Idea to MVP: Prototyping and Experiments	57
6.1 Assumptions Before Products	57
6.2 Prototyping: Ask Questions with Artefacts	58
6.3 The MVP: A Testing Machine, Not a Demo	59
6.4 Technology Readiness Levels: How Real Is the Technology?	59
6.5 Designing Experiments That End Debates	60
Chapter references	64

7	Competitive Positioning and Strategy	65
7.1	Define Your Core	65
7.2	Know the Arena: Mapping Competitors	66
7.2.1	The competitive positioning chart	66
7.3	Escape the Comparison: The Blue Ocean Lens	67
7.4	Your Business Model in Its Environment	68
7.5	From Position to Plan: First 10 Customers and Follow-on Markets	68
7.6	SWOT That Ends in Decisions	69
	Chapter references	72
III	Building the Organization	73
8	Founding Team, Roles and Leadership	75
8.1	Why Teams Beat Heroes	75
8.2	The Equity Conversation — Have It Now	76
8.3	The Human System: OB in Miniature	76
8.4	Belbin’s Nine Team Roles	77
8.5	Leading a Technology Team	78
8.6	Founder Burnout and Team Sustainability	79
	Chapter references	82
9	Legal Structure and Intellectual Property	83
9.1	Legal Form Is a Strategy Decision, Not Paperwork	83
9.2	The Menu of Legal Forms	84
9.3	Intellectual Property: What You Actually Own	85
9.3.1	The IP traps that actually hit student teams	85
9.4	The Protection Plan	86
	Chapter references	89
10	Project and Risk Management for Startups	90
10.1	Work Deserves a Container	90
10.2	Lifecycles: Project and Product	91
10.3	Waterfall, Agile, and the Student Hybrid	91
10.4	Planning Tools: WBS, Gantt, Critical Path, Kanban	93
10.4.1	WBS: decompose before you schedule	93

10.4.2 Gantt: sequence and time	93
10.4.3 Critical path: where delay is fatal	94
10.4.4 Kanban: steer the week	94
10.5 Risk Management: Managing the Future Before It Manages You	96
10.6 Time Management for Founders	98
Chapter references	102

IV From Plan to Growth 104

11 Financials and the Business Plan 106

11.1 Money Is a Design Variable	106
11.2 The Project Budget: From WBS to Money	107
11.3 Three Statements, One Survival Question	108
11.4 COGS and Gross Margin in Software	108
11.5 Unit Economics: LTV, CAC, and the 3:1 Rule	109
11.6 Break-Even and Runway: The Two Dates That Matter . .	110
11.7 ROI: The Investor’s Question, Asked of Yourself	111
11.8 The Business Plan: Assembling the System	112
Chapter references	116

12 Fundraising and Pitching 117

12.1 Money Has a Voice	117
12.2 The Funding Ladder	118
12.3 Equity, Valuation, Dilution — Founder Arithmetic	119
12.4 The Ask and the Use of Funds	119
12.5 The Pitch Deck: Twelve Slides, One Story	120
12.6 The Elevator Pitch: Sixty Seconds of Truth	121
12.7 Pitching Is a Performance	122
12.8 Due Diligence: What They Will Check	122
Chapter references	125

13 Building Ventures in the Age of AI 126

13.1 AI Changes the Raw Material, Not the Discipline	126
13.2 Three Ways AI Enters a Venture	127
13.3 Build, Borrow, or Blend	128

13.4 The Data Strategy: Your Real Asset	128
13.5 The Economics of AI Products	129
13.6 Responsible AI: License to Operate	130
13.7 AI Inside the Founding Workflow	131
Chapter references	134
14 Growth, Internationalization and Exit	136
14.1 The Long Game	136
14.2 Growth: Scale What Repeats	137
14.3 Internationalization: The World Is the Follow-On Market	138
14.4 Exit: Designing the End at the Beginning	138
14.5 The Sale Process, Briefly	139
14.6 Failure and Turnaround	140
Chapter references	144
A The Management Toolkit: Templates	145
The Concentric Rings Workbook Sheet	145
Template 1 — Idea Validation Matrix	148
Template 2a — Market Research Worksheet & TAM–SAM–SOM	150
Template 2b — Option Prioritization Matrix (Opportunity × Feasibility)	152
Template 3a — Persona Canvas (B2C & B2B variants)	154
Template 3b — Value Proposition Canvas	156
Template 4a — Business Model Canvas	157
Template 4b — Open Innovation Flow Map	158
Template 5a — Innovation Funnel Map	160
Template 5b — Innovation KPI Table	162
Template 6a — Assumption Map	163
Template 6b — Experiment Card	165
Template 7a — SWOT / TOWS Matrix	166
Template 7b — Competitive Positioning Chart	168
Template 8a — Team Charter	170
Template 8b — Belbin Roles Map	172
Template 9a — Legal Form Decision Table	173
Template 9b — IP Inventory	174
Template 10a — Risk Management Table	175

Template 10b — WBS / Gantt Skeleton	176
Template 10c — Kanban Board	178
Template 11a — Project Budget Sheet	179
Template 11b — Unit Economics Sheet	180
Template 12a — The Ask / Use of Funds	182
Template 13 — Business Plan Skeleton	183
Template 14 — Pitch Deck Skeleton	184
Template 15a — Data Strategy Sheet	185
Template 15b — Responsible AI Checklist	187
Template 16a — Growth & Internationalization Sheet	188
Template 16b — Exit Plan	190
B Capstone Grading Rubrics	192
Rubric B1 — Business Plan	192
Rubric B2 — Pitch Deck & Delivery	193
Rubric B3 — Rings Map & Toolkit	193
C Glossary	194
D Startup Gallery: Six Ventures to Dissect	199
BabyFM — one technology, two markets	199
UrbiGo — the smart garden	200
Strawberry energy — the bench that is not the business	200
Anora — the smart glove for the blind	201
Inclusio — access as a service	201
CodeKido — security products from Niš	202
E A Student Business Plan, Condensed: ZenTutor	203
F The Rings Volvelle: A Paper Instrument	208
Declaration on the Use of Artificial Intelligence	213
Bibliography	214
Index	224
Author Biographies	226

Introduction: The Venture as a System

Why most startups fail — and what systems thinking changes

Ask a room of first-time founders why startups fail and you will hear single causes: no market need, ran out of cash, wrong team, outcompeted. The lists are accurate and useless at the same time, because they treat each cause as independent. In reality these factors are coupled: a wrong team builds the wrong product; the wrong product fails to reach the right customer; the missing customer revenue burns the cash; the cash pressure breaks the team further. Failure is rarely an event. It is a *cascade* through a system.

Systems thinking — the discipline of seeing wholes, relationships, and feedback loops rather than isolated parts — has a long and productive history in management science. Senge placed it at the heart of the learning organization as its “fifth discipline” [1]; Meadows gave us its working vocabulary of stocks, flows, and feedback loops [2]; Wheatley showed that organizations behave less like machines and more like self-organizing living systems, held together by meaning, information, and relationships rather than by command [3]. Sastry and Penn demonstrated that failures inside projects almost always have systemic roots, and that the productive response is to design small, deliberate experiments from which the system can learn [4]. And in the study of technological innovation, Dodgson, Gann and Salter insist that innovation is not a department or an event but a firm-wide process embedded in an external network of partners, suppliers, customers and regulators [5].

Entrepreneurship, seen this way, is the act of *growing a new system* inside an existing one. That is the view this practicum takes — and it is why

we added one instrument you will not find in standard startup manuals.

The Concentric Rings model

Imagine your venture as a set of concentric rings (Figure 1). At the very core sits the **idea** — together with its innovation type, because an incremental product improvement and a disruptive platform are different organisms from day one. Around it, ring by ring, the venture acquires substance:

- R1. Idea & innovation type** — the core: what you believe, and what kind of new it is.
- R2. Customer & value proposition** — for whom, and why they should care.
- R3. Business model** — how value is created, delivered, and captured.
- R4. Product & assumptions** — what you actually build, and what must be true for it to work.
- R5. Team & organization** — who does it, and how they are structured and led.
- R6. Project & risk** — how the work is planned, executed, and de-risked.
- R7. Financials** — what it costs, what it earns, and what it needs.
- R8. Market, partners & ecosystem** — the outer ring you do not control: competitors, regulators, platforms, and open-innovation partners.

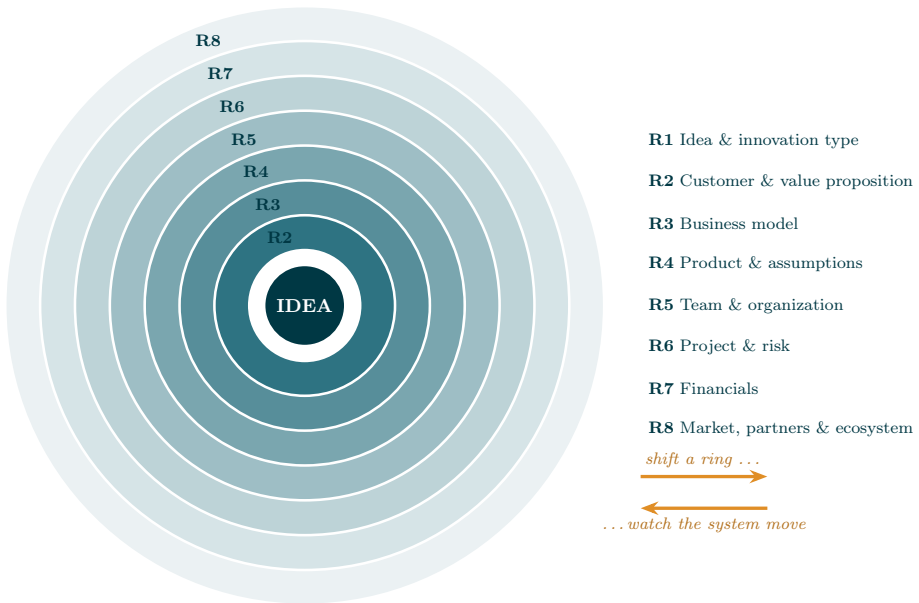


Figure 1: The Concentric Rings model. Moving any ring “to the right” — changing one element — propagates effects through every other part of the venture system.

Two rules govern the model. First, **every ring touches every other ring**: you cannot change the customer without touching the product, the channels, the revenue logic, and the team skills. Second, **the outer ring touches everyone else, but you do not control it**: the ecosystem answers back. This is where open innovation lives — the deliberate use of external ideas, partners, data, and channels, and the deliberate release of some of your own [5, 6].

In the Systems Thinking Lab of each chapter you will first *fill* a ring for your own venture, then *shift* something and watch the consequences. The goal is not a pretty diagram; it is a habit of asking, whenever you change anything: *what else moves?*

Open innovation: the permeable outer ring

Classical entrepreneurship teaching draws the firm as a fortress: ideas are generated inside, protected, and pushed out. The open innovation

paradigm reversed the picture: valuable knowledge is distributed across customers, suppliers, universities, competitors and communities, and the winning firms are those that manage *flows* across their boundary rather than hoarding stocks inside it. For a software startup this is almost trivially true — you build on open-source stacks, public APIs, cloud platforms, pretrained models, app-store distribution and community feedback. Your outer ring is permeable whether you like it or not; the only question is whether you design that permeability or suffer it. Chapter 4 gives you the tools to design it.

How the journey is organized

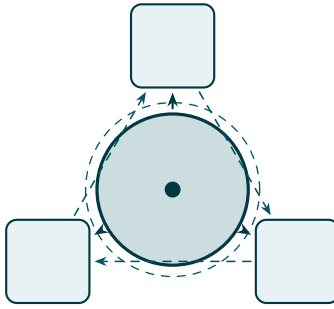
The fourteen chapters walk the rings from the core outward and back again. **Part I** (Chapters 1–3) takes you from an idea to a validated customer opportunity. **Part II** (Chapters 4–7) designs the venture: business model, innovation strategy and process, the product and its riskiest assumptions, and competitive position. **Part III** (Chapters 8–10) builds the organization: team, legal form and intellectual property, project and risk management. **Part IV** (Chapters 11–14) turns design into numbers and momentum: financials and the business plan, fundraising and pitching, the specifics of building with AI, and finally growth, internationalization and exit.

The guided steps inside the chapters follow the logic of Disciplined Entrepreneurship — the market-first, evidence-driven sequence developed at MIT and taught worldwide [7, 8] — enriched with the Lean Startup’s Build–Measure–Learn loop [9] and the Fail Better method for designing productive experiments [4]. You do not need to read those books before starting; where their ideas appear, we tell you exactly where to go deeper.

One last thing before Chapter 1. Drucker, the person who effectively invented management as a discipline, insisted that performance in any institution begins with a clear definition of what the institution is *for* [10]. Your venture is no different. Everything that follows is easier if you can already answer, in one sentence: *what will exist in the world when your startup succeeds that does not exist today?* Keep that sentence close. It is the seed of your first ring.

PART I

From Idea to Opportunity



In this part

Every venture begins twice: first as an idea in someone's head, then as an opportunity tested against reality. In this part you generate and screen ideas (what makes an idea worth a semester?), research the market it must live in (who buys, how many, why now?), and discover the customer behind the market (which persona, which pains, which gains?). By the end, the core and the first inner rings of your system map — idea, customer, value — are no longer guesses; they carry evidence.

Chapters in this part: 1 Entrepreneurial Mindset and Opportunity Recognition · 2 Market Research and Opportunity Analysis · 3 Customer Discovery and Value Proposition

Entrepreneurial Mindset and Opportunity Recognition

Learning outcomes

After this chapter you will be able to:

- explain what an entrepreneurial idea is, and what it is not;
- distinguish creativity from innovation and classify an innovation by type and degree;
- name the five most common sources of new venture ideas and apply three idea-generation methods;
- answer the “Why now?” question with evidence rather than enthusiasm;
- screen a portfolio of your own ideas with a scoring matrix and defend the choice of one;
- place your chosen idea at the core of your Concentric Rings map.

1.1 Everything Starts from the Idea

An **idea** is a thought or suggestion about a possible course of action. That is all it is — and that is precisely the point. Ideas are cheap, abundant, and fragile; what turns an idea into a venture is everything that happens after the first sentence “what if we...”. Experienced investors therefore say they invest in teams and markets, not in ideas. Yet every team and every market story begins with one, so we begin there too.

A useful idea for a startup has three properties you can test immediately.

It *names a problem* (someone’s pain, not your invention), it *names who has the problem* (a customer you could, in principle, phone this week), and it *suggests why you* (your insight, skill, or unfair advantage). If any of the three is missing, what you have is not yet an idea for a venture — it is a wish.

1.1.1 Idea validation

Validation is the discipline of converting belief into evidence, in the cheapest possible way, before you build. The classical sequence — market research, problem confirmation, planning — has been radically compressed by the Lean Startup movement: instead of asking “*can this product be built?*”, ask “*should it be built, and for whom?*” [9]. You will practice validation properly in Chapters 2, 3 and 6; for now, hold one rule: **an idea you have not tried to kill is an idea that will try to kill you later.**

1.1.2 Pivot

A **pivot** is a structured change of direction that keeps one foot in what you have already learned. Nokia began as a paper mill; Pinterest grew out of a failed shopping app; Instagram was stripped out of an overloaded check-in product; Airbnb pivoted from air mattresses at conferences to any home anywhere [9, 11]. The founders we admire did not so much “get it right” as *turn* well. Keep that in mind when you place your idea in the core ring today: the core can move, and the system is designed to move with it.

1.1.3 Motivation: mission, vision, values

Why become an entrepreneur at all? The honest literature separates the truths from the myths. The myths are being your own boss, financial independence by thirty, managing your own time, and glamour. The truths are responsibility that does not clock out, stress that follows you home, and founders’ depression that few discuss. On the other side of the ledger sits a chance to maximize impact in a way few careers allow [11, 12]. Your **mission** (what you do every day), **vision** (what the world looks

like when you succeed), and **values** (what you will not do even when it pays) are not wall decoration. They are the load-bearing elements of your team ring in Chapter 8, and the first thing experienced co-founders and investors probe.

1.1.4 Target customer first

You can change your startup idea, but you cannot change your customer — at least not cheaply. Think about your “public” before your product: who feels the problem, how often, how strongly, and what they do about it today. The discipline of focusing on customer needs before product features is the backbone of the Disciplined Entrepreneurship sequence you will follow from Chapter 2 onward [7].

1.2 Creativity and Innovation

Creativity is the production of new ideas, independent of their eventual applicability. **Innovation** is the transformation of new ideas into useful products or services — value actually delivered to someone [5, 13]. The distinction matters practically: creativity exercises make you a better generator of options; innovation management (Chapter 5) makes you a better selector and executor.

1.2.1 Types of innovation

Innovations differ in degree and in kind, and the difference shapes your entire strategy (Figure 1.1).

- **By degree:** *advanced (revolutionary)* innovations create entirely new capabilities (penicillin, the airplane, the internet); *technological* innovations embody them in products (the personal computer); *ordinary* innovations extend and refine what exists — most startups live here, and there is no shame in it [13].
- **By impact on the market:** *incremental* innovation improves along dimensions the market already values; *disruptive* innovation enters with a different value package — typically cheaper, simpler, more accessible — and rewrites the market’s rules [5, 9].

- **By object:** *product* innovation changes what is offered; *process* innovation changes how it is made or delivered (and is often invisible to the customer — and to competitors, which can be an advantage).
- **By boundary:** *closed* innovation relies on internal resources; *open* innovation deliberately collaborates with external partners, importing and exporting knowledge [5, 6]. Chapter 4 is devoted to the open variant.

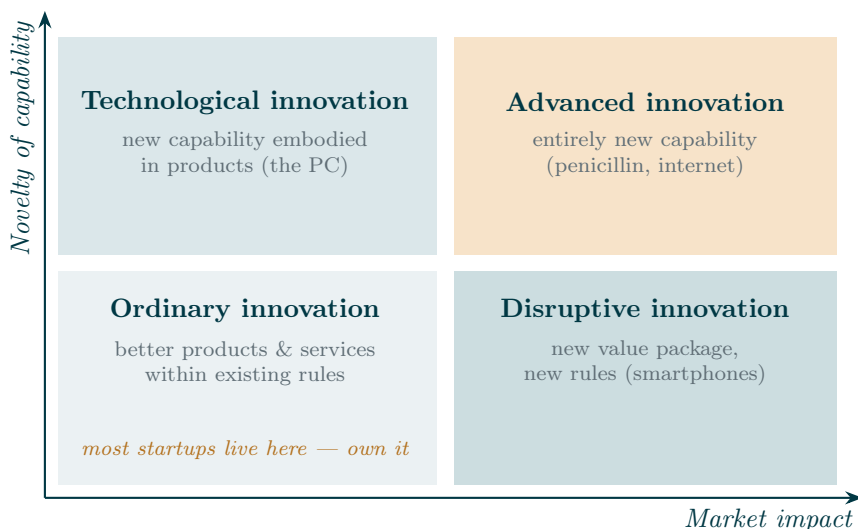


Figure 1.1: Innovation types by novelty and market impact. Locating your idea on this map early prevents a revolutionary’s spending plan attached to an ordinary innovation — and vice versa.

A word on what “new” even means. In industrial markets a product may count as new after modest design changes; from the consumer’s standpoint, novelty is measured by how much behavior must change; from the company’s standpoint, by the volume of improved technology and the newness of segmentation [13]. Keep all three readings in mind — your customer, your competitor and your investor use different rulers.

1.3 Where Ideas Come From

Research on entrepreneurs consistently finds five dominant sources of new venture ideas [13]:

1. **Clients** — complaints, requests and workarounds of people you already serve or observe;
2. **Existing products and services** — frustration with what is on the market today;
3. **Distribution channels** — seeing what flows, what stalls, and what is missing on the shelf;
4. **State administration** — regulation changes, public tenders, open data, subsidy programs;
5. **Research and development** — results and technologies looking for a problem to solve.

Notice that four of the five are *outside your head*. Ideas are found in contact with the world more often than in solitude — one more reason this practicum sends you to real interviews already in Chapter 2.

1.3.1 Idea-generation methods

Three methods cover most of what teams need at this stage:

- **Brainstorming** — a group method for volume: suspend judgment, build on others' ideas, aim for quantity, then cluster. Works best with a sharp prompt (“for student X, problem Y, ten ways to make it disappear”).
- **Focus groups** — a structured conversation with 6–10 individuals from the target segment, guided by a moderator through their current behavior — not their opinions about your solution.
- **Problem analysis** — start from the problem, not the solution: pick a process you know well, list where it wastes time, money, or attention, and generate ideas that remove the waste.

1.3.2 “Why now?”

Every credible opportunity answers **Why now?** — what changed in statistics, technology, competition, or available solutions that makes this the right moment [12]. If the answer is “nothing changed”, ask why a capable incumbent has not already done it. Sometimes the honest answer is timing: smartphones, cheap cloud, or — today — capable AI models each opened windows that did not exist five years earlier.

Guided steps

Work through these steps for your own venture. Keep all outputs — they feed directly into Chapter 2.

Step 1. Generate. Using problem analysis on one domain you know well, produce at least five distinct venture ideas. One sentence each: problem — who — why you.

Step 2. Classify. For each idea, mark its innovation type (degree, market impact, object, boundary) using Figure 1.1.

Step 3. Screen. Score all five ideas in the Idea Screening Matrix (Template 1). Be brutal with evidence, generous with curiosity.

Step 4. Choose one. Pick the idea with the best total score *that you are also willing to live with for a year*. If the top score fails the willingness test, take the second.

Step 5. Write your “Why now?”. One paragraph, at least two external facts. This paragraph will reappear in your pitch deck in Chapter 12.

Tasks & exercises

Task 1.1 (individual). Generate five ideas following Step 1. Deliverable: five one-sentence ideas in the matrix format.

Task 1.2 (individual). Fill the Idea Screening Matrix for all five ideas. Deliverable: completed matrix + 100-word defense of your chosen idea.

Task 1.3 (individual). Write your “Why now?” paragraph with two verifiable external facts (cite sources). Deliverable: one paragraph.

Task 1.4 (team). Run a 20-minute brainstorming session for one team member’s idea using the rules above; produce ten *variations* of the same idea (different customer, different channel, different pricing logic). Deliverable: variation list.

Task 1.5 (individual). Draft your mission, vision, and values — three sentences total. Deliverable: the three sentences; they will be stress-tested in Chapter 8.

Template

Template 1 — Idea Validation Matrix (this practicum’s original instrument; reproduced full-page in the Appendix). Score each idea 1–5 per dimension; multiply by weights; totals rank the portfolio. In Chapter 2 the Cambridge Option Prioritization Matrix takes the winning idea deeper (Opportunity $\times$ Feasibility scoring with anchored statements).

Dimension	Weight	Idea A	Idea B	Idea C	...
Idea — problem is real, frequent, clearly stated	25%				
Market — size, growth, “why now” evidence	20%				
Competition — intensity, visible whitespace	15%				
Feasibility — can we build an MVP this semester?	15%				
Required resources — access to skills, data, money	15%				
Originality — differentiation vs. existing solutions	10%				
Weighted total	100%				

Scoring anchors: 1 = pure guess, works against us; 3 = partial evidence; 5 = verified with external facts. A dimension scored below 2 cannot be compensated by high scores elsewhere — treat it as a

red flag, not an average.

The elevator pitch: saying the idea in one breath

Before a matrix can score an idea, the idea must be *sayable*. The course format for this is the **elevator pitch** — sixty seconds, six moves, no slides [12]: *company* (who you are), *problem or market need* (the pain, in the customer’s words), *solution* (what you build, one sentence), *target users* (who pays, who uses), *advantages* (why you, not the obvious alternative), and *added value* (what the world gets that it did not have). If any of the six is missing, the idea is not yet an idea — it is a wish. The pitch is also the validation matrix read aloud: problem feeds the *Idea* row, users feed *Market*, advantages feed *Competition* and *Originality*. In Chapter 12 you will deliver this pitch to investors; you draft it now because a blurry pitch means a blurry idea.

Worked example

ZenTutor — from elevator pitch to a filled matrix. A student team’s pitch, mapped onto the six moves: “*ZenTutor is a personalized tutoring marketplace and mobile app [company] that solves the unorganized search for trusted tutors, the scheduling chaos and the high cost of private lessons [problem] by connecting students and parents with verified tutors through one app with easy scheduling and group-session options [solution], aimed at students aged 12–18 and their parents [users]. Unlike generic platforms, ZenTutor offers personalized learning packages, real in-person sessions and group discounts [advantages] — affordable, transparent tutoring that actually fits a school week [added value].*”

Scored on this chapter’s matrix, as the team defended it in class:

Dimension (weight)	Score	Evidence the team showed
Idea (25%)	4	Exam stress and tutor-search friction recur every term; pain stated in the customer's words.
Market (20%)	4	Private tutoring spend grows; urban parents are mobile-first (sized in Chapter 2).
Competition (15%)	2	Generic platforms and local agencies are crowded; differentiation still thin. <i>Red flag to fix.</i>
Feasibility (15%)	3	Marketplace MVP is buildable, but two-sided (needs tutors <i>and</i> students from day one).
Required resources (15%)	3	Tutor supply reachable via campuses; marketing budget modest; scheduling logic non-trivial.
Originality (10%)	3	Personalized packages + in-person focus is a fresh bundle, not a new category.
Weighted total	3.30	Worth pursuing <i>if</i> the competition row improves by Chapter 7's positioning work.

Notice the discipline: the pitch supplied the words, the matrix forced the evidence, and the weakest row — competition, scored 2 — became the team's homework instead of staying hidden inside an enthusiastic paragraph.

Worked example

ClockMEup — deconstructing a student idea. A student team's elevator pitch read: "*ClockMEup develops a hardware and software platform for waking up on time, solving the problem of being always late due to heavy sleep, with a personalized set of features on one device, for students, postgraduates and juniors.*" Run it through this chapter's lens:

- *Problem*: chronic lateness caused by heavy sleep — real, frequent, and the team felt it themselves (problem analysis source).
- *Customer*: students and young professionals — identifiable and reachable (a campus is a beachhead waiting to be named; see

Chapter 2).

- *Why now*: cheap sensors, smartphone ubiquity, and post-2020 normalization of connected home devices.
- *Innovation type*: ordinary product innovation with an incremental hardware+software bundle — which correctly told the team to compete on execution and design rather than on a technological moat.
- *Pivot readiness*: when hardware timelines slipped, the team could pivot to a software-only alarm with the same persona — one foot kept in everything learned.

Notice how much strategy is already visible in one paragraph, once you know how to read it.

Systems Thinking Lab — The Concentric Rings

Fill Ring R1 — the core. On your rings workbook sheet, write in the center: (a) your idea in one sentence; (b) its innovation type from Figure 1.1; (c) your “Why now” fact that is strongest.

Then shift it. Imagine a funded competitor launches the same idea tomorrow, aimed at a *different* customer segment. On the map, draw an arrow from the core outward and answer in writing: which rings would you have to move first — customer (R2), product (R4), or channels on the outer ring (R8)? One short paragraph is enough; you are rehearsing the habit, not solving the case.

Record. Keep the sheet visible in every later chapter. The map you start today is the one you will defend in Chapter 14.

AI angle

AI changes this chapter twice. First, *as a source of ideas*: every large-model capability released in the last three years created a wave of “why now” openings — your Task 1.1 may legitimately start from “what can models do now that they could not do last year?”. Second, *as an ideation tool*: use an AI assistant as a brainstorming partner for Task 1.4, but apply the same matrix to its suggestions as to yours

— models generate volume, you supply evidence. In Chapter 13 we will look at what makes an AI idea defensible rather than merely possible.

Chapter references

For deeper reading on this chapter’s foundations: Dodgson, Gann & Salter, *The Management of Technological Innovation*, chapters 1–3 [5]; Ries, *The Lean Startup*, Part One [9]; Altman et al., *How to Start a Startup*, Lecture 1 [11]; Aulet, *Disciplined Entrepreneurship*, introductory chapters [7]; course lectures *Entrepreneurship in SE* (intro) and *Opportunity Analysis* [12–14]; Avlijaš & Avlijaš, *Preduzetništvo* (the Serbian-language companion textbook) [15]; Kraus et al. on digital entrepreneurship as a research field [16]; Duening, Hisrich & Lechter, *Technology Entrepreneurship*, on taking innovation to the marketplace [17]. Osterwalder & Pigneur, *Business Model Generation* (open vs. closed innovation) [6].

Market Research and Opportunity Analysis

Learning outcomes

After this chapter you will be able to:

- run a seven-step market research process for a new venture;
- segment a market and select a beachhead market with defensible arguments;
- conduct and document primary market research (PMR) interviews;
- size a market with TAM–SAM–SOM, top-down and bottom-up;
- score and plot competing market options on the Opportunity × Feasibility matrix;
- add the customer ring (R2) to your Concentric Rings map and trace its couplings.

2.1 The Market Research Process

Enthusiasm is a terrible market researcher. It finds confirming evidence, ignores the rest, and reports back that the idea is brilliant. The antidote is process. A disciplined market research effort follows seven steps [18]:

1. **Understand the need for research** — what decision depends on this information?
2. **Define research goals** — what exactly must be true for the decision to be yes?

3. **Define the information to collect** — quantities (sizes, prices, rates) and qualities (motives, frustrations, workarounds);
4. **Identify sources of information** — desk research, public statistics, industry reports, and above all people;
5. **Check source–goal compatibility** — a report about “global trends” rarely answers “will 30 students in Belgrade pay for this?”;
6. **Gather information** — with dates and sources recorded, so it can be challenged;
7. **Analyze and conclude** — the conclusion is a *decision*, not a summary: enter, adapt, or abandon.

Notice the order: goals before sources. Most failed research starts with whatever data is easy to find and works backwards to a conclusion that was desired from the start.

2.2 Market Segmentation

You cannot research “the market” — there is no such thing. There are only segments: groups of potential customers who share a problem, a context, and a buying logic. Segmentation divides the market so that one group can be served *specifically* better than anyone serves them today. The Disciplined Entrepreneurship sequence starts exactly here: before you fall in love with a product, you map who might care at all [7, 19].

Useful segmentation questions for software ventures [20]:

- For whom are we solving a problem? Are they individuals or legal entities?
- If companies: which industry, size, revenue, geography, decision maker?
- If individuals: what demographics, behaviors, and contexts unite them?
- Who are the people who will *appreciate* our value proposition — not merely tolerate it?
- What characteristics do the people *searching* for a solution like ours share?

A good segment passes three tests: its members share the same painful problem; they can find and pay for a solution; and you can actually reach them with the channels available to a student team.

2.3 Primary Market Research (PMR)

Desk research tells you what the world looks like; only primary research tells you what *your customer's Tuesday* looks like. PMR means direct contact — interviews, observations, shadowing — with people in the segment. The rules that separate signal from theater [9, 19]:

- **Ask about past behavior, not future opinions.** “How did you solve this last month?” beats “Would you use an app that...” every time.
- **Do not pitch.** The moment you describe your solution, the interview becomes politeness measurement.
- **Follow the pain, not the script.** When a respondent's energy spikes, dig there.
- **Record verbatims.** The customer's own words will later write your marketing copy — and your pitch deck.

Ten honest interviews with the right segment teach more than a thousand-question survey of the wrong one.

2.4 Selecting the Beachhead Market

D-Day logic applies to startups: you do not open five fronts with the resources for one. A **beachhead market** is the single segment where you concentrate everything first — small enough to win, valuable enough to matter, adjacent enough to expand from [7, 19]. Criteria: customers in the segment *talk to each other* (word of mouth travels), the pain is urgent, budget exists, and you have credibility or access there.

But how do you choose among several plausible beachheads — or among several market options for the same product? Enthusiasm again votes for all of them. What is needed is an instrument that forces the trade-offs

into the open.

2.5 The Option Prioritization Matrix (Opportunity $\times$ Feasibility)

The Option Prioritization Matrix — this practicum’s adaptation of a scoring tool used in technology innovation workshops [21] — scores every market option on two families of factors:

Opportunity factors (how attractive is the option?): market size and growth, intensity of the unmet need, willingness to pay, impact on key customer relations, competitive whitespace, strategic upside.

Feasibility factors (how achievable is it for *us*?). Our adaptation restructures the original factor set for software ventures: differentiation, sustainability of competitive advantage (IPR, brand, switching costs, data moats), technical challenge, market knowledge, team capability, fit to channels and distribution, financial feasibility, strategic fit, and organisational backing.

Each factor is scored on an **anchored scale of 0–3–6–9–12** — the anchor statements matter, because they force evidence instead of vibes. Two examples from the scoring rubric:

Factor	Score 0 (anchor)	Score 12 (anchor)
Differentiation	No features better than competition	Several important features much better than competition
Market knowledge	Market size unsupported by data; requirements unchecked with customers	Market size known to $\pm 5\%$; customer understanding established and tested
Technical challenge	Key features not yet demonstrated by us or others	All features tested and fully functional
Finance	Extra funding required, source not identified	External funding available for the whole project

Summed scores place every option on a scatter plot: Opportunity on one axis, Feasibility on the other. The discussion that follows is the real deliverable — the plot is only its trigger.

Worked example

BabyFM — one technology, two markets, seven options.

BabyFM is a company developing a single monitoring technology that it brings to two distinct markets at once: baby monitoring (babyFM, in medical and non-medical variants) and pet monitoring (petFM). The same core technology, the same team — but seven very different ways to reach customers: veterinary clinics in Serbia, distributors in the USA, non-medical distributors in the USA, distributors in Qatar, non-medical distributors in the UAE, pharmacy distributors in Serbia, and private clinics in Serbia. With limited resources, “all seven” is not a strategy; the team used the Option Prioritization Matrix to understand their priorities and the real feasibility of each route. Scored and plotted, the options separated clearly:

Option	Feasibility	Opportunity
petFM — vet clinics, Serbia	102	67.5
petFM — distributors, USA	99	101
babyFM non-med — distributors, USA	91	98
babyFM — distributors, Qatar	95	79
babyFM non-med — distributors, UAE	101	84
babyFM — pharmacy distr., Serbia	106	57
babyFM — private clinics, Serbia	107	68

The Serbian clinic options were highly *feasible* but mediocre *opportunities* — easy to reach, hard to scale. The US distributor options inverted the profile. Once your own options are plotted, ask the questions the BabyFM team asked: *Which are our best options? Does the graph match what we expected? If not, why not? What would we rescore? What additional information do we need?* Note that the tool does not decide — it forces the decision to show its working.

2.6 Sizing the Market: TAM–SAM–SOM

Once the beachhead is chosen, quantify it. The standard three-level breakdown [18, 19]:

- **TAM** (Total Addressable Market) — everyone who could, in principle, buy a solution of this type;
- **SAM** (Serviceable Available Market) — the part your model, language, and geography can actually serve;
- **SOM** (Serviceable Obtainable Market) — what you can realistically win in the first years, given competition and your capacity.

Two ways to compute: **top-down** (start from published totals and narrow by justified percentages) and **bottom-up** (count reachable customers $\times$ realistic price — slower, and almost always more honest). Investors distrust top-down-only numbers for a reason: “1% of a huge market” is a wish, not a plan. Always try to produce the bottom-up check, even rough.

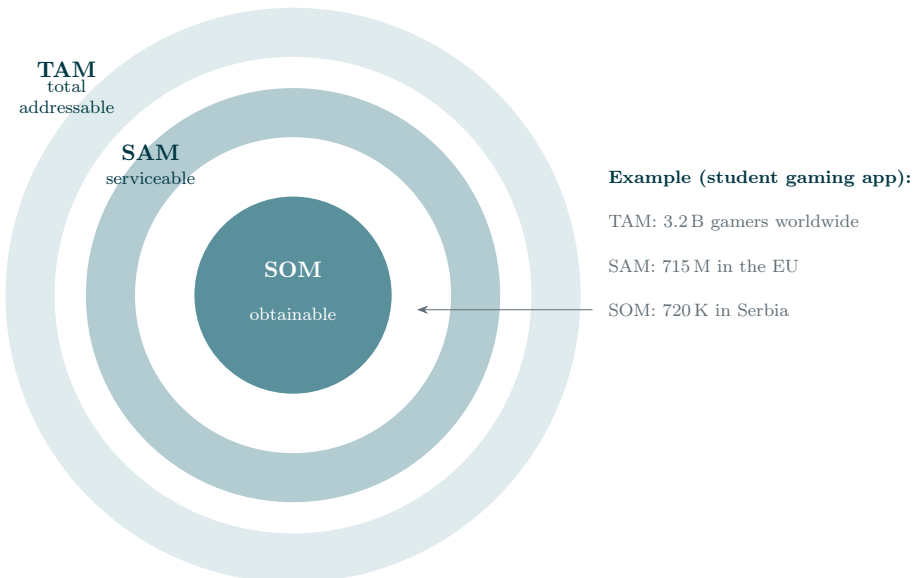


Figure 2.1: TAM–SAM–SOM: three nested markets. Numbers from a real student market-research slide.

Worked example**ZenTutor — sizing the tutoring market, both directions.**

The ZenTutor team (their pitch and profiles appear in Chapters 1 and 3) sized their market for Serbia first — a beachhead they could actually reach. The figures below are the team’s working estimates with every assumption shown; your own numbers must come from sources you cite (Task 2.4).

Top-down:

- **TAM $\approx$ EUR 50 M/year.** About 650,000 primary- and secondary-school pupils in Serbia; roughly one in four uses paid tutoring at some point in the year; average spend about EUR 35/month over a nine-month school year ($\approx$ EUR 315 per tutoring family). $650,000 \times 0.25 \times 315 \approx 51$ M.
- **SAM $\approx$ EUR 9.5 M/year.** Keep only what the model can serve at launch: urban cities with dense tutor supply ($\times 0.45$), mobile-first parents who buy services online ($\times 0.70$), and the exam-prep subjects ZenTutor starts with — math, physics, languages ($\times 0.60$). $50 \text{ M} \times 0.45 \times 0.70 \times 0.60 \approx 9.5$ M.
- **SOM $\approx$ EUR 300 K/year.** Three percent of SAM within three years, given established agencies and generic platforms: $9.5 \text{ M} \times 0.03$.

Bottom-up check: 900 active families $\times$ EUR 28 of monthly market-place revenue (commission on booked sessions plus learning-package subscriptions) $\times$ 12 months $\approx$ EUR 302 K. The two directions agree within rounding — which is exactly the point of doing both. Had the bottom-up come out ten times smaller, the top-down percentages would have been the first suspects.

2.7 Competition, Substitutes, Trends

Opportunity analysis is completed by three scans [13]: **competitors** (who solves the same problem the same way), **substitutes** (who solves it differently — including “doing nothing”, the most common competitor of all), and **trends** (technological, demographic, regulatory currents moving

in your favor or against you). The output is not a list but a judgment: is the whitespace you see real, or is it empty because three companies died there? Chapter 7 will take this into full competitive positioning.

Guided steps

Step 1. Segment. List 4–6 distinct market segments for your chosen idea (Section 2.2 questions).

Step 2. Interview. Run at least 10 PMR interviews across the two most promising segments (Section 2.3 rules).

Step 3. Score options. Fill the Option Prioritization Matrix for your 4–6 segments with anchored scores (Template 2b).

Step 4. Choose the beachhead. Plot, discuss, pick one — and write down what you would rescore with better data.

Step 5. Size it. Compute TAM–SAM–SOM for the beachhead, top-down *and* bottom-up (Template 2a).

Step 6. Scan. List competitors, substitutes, and three trends with sources.

Tasks & exercises

Task 2.1 (team). Segmentation map: 4–6 segments, each with the three tests from Section 2.2. Deliverable: one page.

Task 2.2 (individual). Conduct 5 PMR interviews (10 per team of two) with verbatims logged. Deliverable: interview log + three quotes that surprised you.

Task 2.3 (team). Option Prioritization Matrix: anchored scores for all options, scatter plot, and the five Cambridge discussion questions answered. Deliverable: matrix + plot + one-page discussion.

Task 2.4 (individual). TAM–SAM–SOM for your beachhead, both methods, sources cited. Deliverable: calculation sheet + funnel diagram.

Task 2.5 (team). Competitor/substitute map with a one-paragraph “is the whitespace real?” judgment. Deliverable: map + paragraph.

Template

Template 2a — Market Research Worksheet & TAM–SAM–SOM table (Appendix): research goals, sources, findings, and the market-sizing table (segment count $\times$ price, top-down percentages with justification).

Template 2b — Option Prioritization Matrix (Appendix): the full anchored rubric (Opportunity and Feasibility factors, 0–3–6–9–12) and the scatter plot grid for your options.

Systems Thinking Lab — The Concentric Rings

Fill Ring R2 — customer & value. On your map, write your beachhead segment and the single sharpest customer pain you verified in interviews.

Then shift it. Move your chosen beachhead “one segment to the right” — pick the runner-up segment from your scatter plot instead. In writing: what changes in R4 (product features), R6 (project plan), and R8 (channels, partners)? At least two concrete couplings per ring.

Record. Note one assumption about the customer that would, if wrong, force you to redo this ring. It joins the experiment backlog in Chapter 6.

AI angle

Desk research is where AI assistants genuinely save days: summarizing industry reports, extracting statistics with sources, drafting interview guides, clustering verbatim quotes by theme. Two cautions. First, AI cannot conduct your PMR — invented “user opinions” are worse than no data, and every number an AI gives you must be traced to a real source before it enters your TAM table. Second, your competitors use the same assistants; the edge is not in having the summary, but in the ten interviews nobody else bothered to do.

Chapter references

Aulet, *Disciplined Entrepreneurship*, Steps 1–4 [7]; StartSmart workshop materials, market segmentation and PMR [19]; Cambridge technology innovation workshop, option prioritization tool and BabyFM case [21]; course lectures *Opportunity Analysis* and *Business Plan* (market research section) [13, 18]; Luckey & Phillips, *Software Project Management For Dummies*, ch. 3 [22]; Hisrich, Peters & Shepherd, *Entrepreneurship*, chapters on marketing research and industry analysis [23]. Ries, *The Lean Startup* (validated learning before sizing) [9]; course lecture, *Business Model Canvas* [20].

Customer Discovery and Value Proposition

Learning outcomes

After this chapter you will be able to:

- build an end-user profile and an ideal customer profile (ICP) for B2C and B2B ventures;
- construct an evidence-based persona and a full life-cycle use case;
- map customer jobs, pains, and gains and design a matching value proposition;
- quantify your value proposition so it survives a sceptic's questions;
- apply design thinking to customer discovery;
- connect the customer ring (R2) and the value part of your system map, and test persona changes.

3.1 End User First, Customer Second

Chapter 2 gave you a beachhead *segment*. A segment is still a crowd; products are bought by people. The next step in the Disciplined Entrepreneurship sequence is therefore the **end user profile**: a precise description of the human (or the role, in B2B) who will actually use your product [7, 19]. Distinguish three roles that students routinely merge:

- the **end user** — who uses the product;
- the **economic buyer** — who pays (a parent, a procurement officer);
- the **champion** — who wants you to win inside their organization.

In a student app they may be one person; in B2B they are almost never. Your value proposition must speak to the user, your pricing to the buyer, and your sales motion to the champion.

3.2 The Ideal Customer Profile (ICP)

The ICP generalizes the end user into a checklist you can screen leads against. It looks different in consumer and business markets [12]:

B2C customer profile	B2B ideal customer profile
Age, location, job title	Industry segment, public or private
Income, purchase habits	Employees, revenue, turnover
Hobbies, goals	Growth rate, current state
Challenges, daily frustrations	Decision maker, situational factors

Table 3.1: ICP dimensions for B2C and B2B ventures.

The discipline here is specificity with evidence. “Young people who like fitness” is a horoscope, not a profile. “Women 24–32, first job after university, recently relocated, disposable income above X, currently maintaining friendships through messaging apps and finding it insufficient” — that is a profile you can find, count, and interview [19]. Figure 3.1 shows what filled-in profiles look like for two of the student ventures followed in this book.

B2C ideal customer profile	
TEMPLATE — fill in	STUDENT EXAMPLE — <i>Fridge Mind</i>
Who (age, life stage).....	urban professionals, 24–38
Location	cities above 100 000 people
Income & purchase habits.....	above median, often dual income
Hobbies & goals	order groceries online, plan weekly
Challenges & frustrations.....	food expires before anyone uses it
Where to find them	university campuses, delivery apps
<i>screenable against census, survey and PMR data</i>	

B2B ideal customer profile	
TEMPLATE — fill in	STUDENT EXAMPLE — <i>SportSpot</i>
Industry segment	sports & leisure facilities
Public or private	private, single location
Employees, revenue, turnover.....	5–50 employees
Growth rate, current state.....	stable, no online booking yet
Decision maker	owner or facility manager
Situational factors	seasonal peaks, cash-flow sensitive
<i>screenable against the business register (APR)</i>	

Figure 3.1: The ideal customer profile as a fill-in template (left half of each card, dotted lines for your answers) and as filled in by two student teams (right half). The point is not the template but the specificity: every line is a criterion you can count, search for, and interview against.

3.3 Persona and the Full Life-Cycle Use Case

A **persona** turns the profile into one concrete, named individual with a story: demographics, goals, frustrations, a day in their life, and the criteria by which they will judge you. The persona is not decoration; it is the reference object for every later decision. When the team argues about a feature, the question becomes “does Mina need this on her worst Tuesday?” — and the argument ends.

The **full life-cycle use case** then follows the persona through the entire arc: how they *become aware* of the problem and of you, how they *evaluate*,

purchase, onboard, use, get support, and eventually recommend or leave [7]. Gaps in this arc are silent killers: many student products have a fine “use” stage and no awareness stage at all — a product nobody discovers.

This is also where **design thinking** earns its place. Its five modes — *empathize, define, ideate, prototype, test* — are precisely a discovery discipline: empathize with the persona’s reality, define the problem from *their* point of view, and only then ideate solutions [24]. Notice how well it composes with what you already did: PMR interviews (Chapter 2) are empathizing; the persona and use case are the define stage; Chapters 5–6 will handle ideate–prototype–test in earnest.

3.4 Jobs, Pains, Gains — and the Value Proposition Canvas

What does the persona actually *hire* your product to do? The Value Proposition Canvas (VPC) splits the answer into two sides that must fit [6, 25]:

- **Customer side:** *jobs* (what they are trying to get done — functional, social, emotional), *pains* (what frustrates or blocks them), *gains* (what they hope to achieve or save);
- **Product side:** *products & services* (what you offer), *pain relievers* (how each pain is removed), *gain creators* (how each gain is produced).

Fit means every serious pain has a pain reliever and every important gain has a creator — and, crucially, that you have not built features nobody’s job requires. The canvas makes mismatches visible before you pay for them in code.

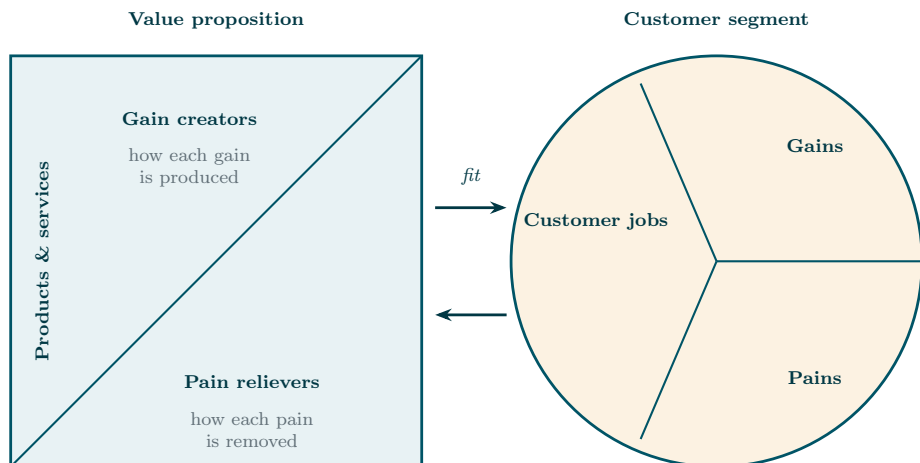


Figure 3.2: The Value Proposition Canvas: the offer (square) must fit the customer (circle) — pain relievers against pains, gain creators against gains.

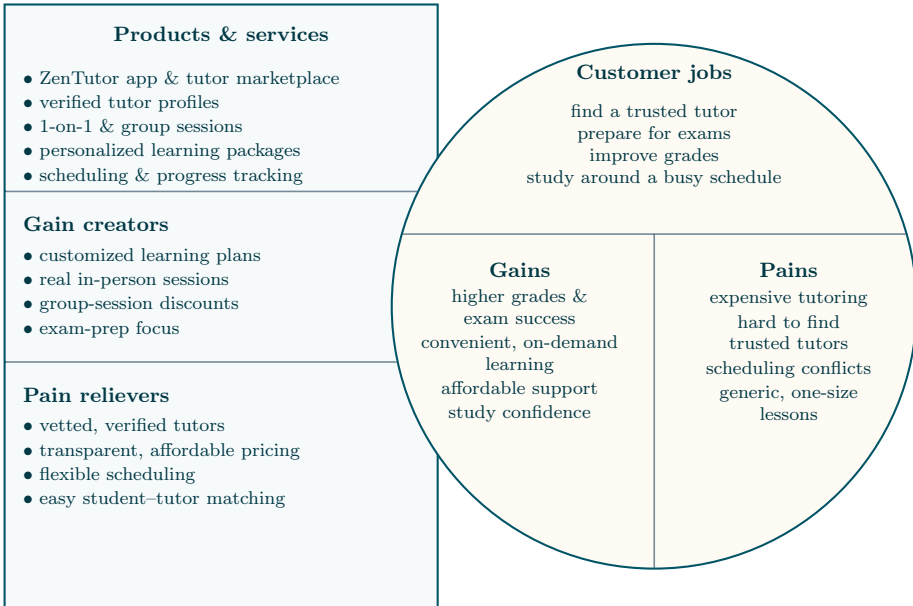


Figure 3.3: ZenTutor’s Value Proposition Canvas, filled in class: the offer (square) against the student-and-parent customer profile (circle). Student artifact, redrawn with permission.

3.5 Quantify the Value Proposition

“Saves time” is a claim; “gives a restaurant manager 6 hours back per week, worth roughly €90 at her wage” is a value proposition. Quantifying forces you to express the benefit in the customer’s own units — hours, euros, errors, anxiety reduced — and it is the single strongest preparation for pricing (Chapter 11) and for the pitch (Chapter 12) [7, 25]. If you cannot yet quantify, that is a finding: it means your discovery is not finished, and the right next task is more interviews, not more features.

Worked example

Three student VPCs, three lessons.

- *SportSpot* (sports booking platform): the customer job was not

“book a court” but “get fit with friends without organizing logistics”; the pains (no geolocated booking, no online payment) mapped one-to-one to pain relievers (geolocation finder, fast booking, history, referral discounts). Lesson: the job was social, the product was logistics — the canvas caught the difference.

- *Desi* (Serbian language learning): jobs included assimilation into society and comfortable communication; pains were forgetting words, limited practice, and lack of motivation; gain creators used consistency reinforcement, spaced repetition, and real-time feedback. Lesson: pains came straight from PMR verbatims, so relievers were evidence-based.
- *Fridge Mind* (AI fridge assistant): jobs (manage food, plan meals, reduce waste), pains (manual tracking, fear of expired food, no inspiration), with an AI layer as the gain creator (scanning, expiry notifications, personalized recipes). Lesson: AI entered as a *reliever of a named pain*, not as a feature looking for one — the healthy pattern Chapter 13 will generalize.

Guided steps

- Step 1. Profile.** Write the end-user profile and the ICP for your beachhead (Table 3.1 dimensions).
- Step 2. Persona.** Build one named persona from your PMR evidence — demographics, goals, frustrations, a day in the life.
- Step 3. Life-cycle.** Map the full life-cycle use case (awareness → advocacy); mark the stages where your persona currently *struggles most*.
- Step 4. Canvas.** Fill the VPC: jobs, pains, gains on the right; offer, relievers, creators on the left; check one-to-one fit.
- Step 5. Quantify.** Express the top benefit in the customer’s units, with the assumptions shown.

Tasks & exercises

Task 3.1 (individual). End-user profile + ICP for your venture. Deliverable: one page, evidence cited per line.

Task 3.2 (team). One persona, validated against at least three PMR interviews. Deliverable: persona sheet + the quotes behind each field.

Task 3.3 (individual). Full life-cycle use case with the two weakest stages marked and explained. Deliverable: arc diagram.

Task 3.4 (team). Completed Value Proposition Canvas with an explicit fit check (list every unmatched pain and every orphan feature). Deliverable: canvas + fit list.

Task 3.5 (individual). Quantified value proposition: one sentence + the assumptions table. Deliverable: half a page.

Template

Template 3a — Persona Canvas (B2C & B2B variants) (Appendix): profile fields, day-in-the-life strip, judging criteria.

Template 3b — Value Proposition Canvas (Appendix): the full-page square-and-circle canvas from Figure 3.2, with fit-check columns.

Systems Thinking Lab — The Concentric Rings

Fill the R2–R3 connection. You now know the customer (R2) precisely. On your map, write the persona's sharpest pain on ring R2, then write the matching pain reliever where R2 touches R3 (business model). The value proposition lives exactly on that boundary.

Then shift it. Replace your persona with the *runner-up* persona from your PMR, someone of a different age, context, or buying role. Which pains and gains change? Which gain creators survive the switch, and which die? Write down what this costs your product plan (R4).

Record. The persona you keep after this exercise is the one quoted in your pitch deck. The discarded one goes to the follow-on markets note for Chapter 14.

AI angle

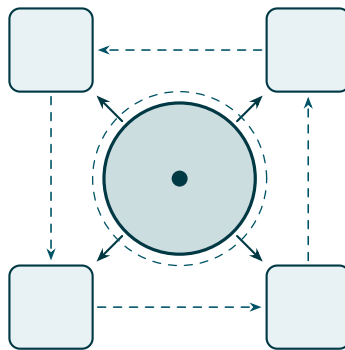
AI personalizes value propositions at a granularity no manual segmentation reaches: recommendation, adaptive content, individualized pricing experiments. In discovery, AI tools cluster interview verbatims by theme in minutes and draft persona candidates — but treat those as *hypotheses to verify against real quotes*, never as discovery itself. And note the Fridge Mind pattern: the strongest AI value propositions relieve a pain that already existed; the weakest describe what the model can do and stop there.

Chapter references

Osterwalder & Pigneur, *Business Model Generation*, Customer Insights [6]; Osterwalder et al., *Value Proposition Design* [25]; Aulet, Steps 3–8 [7]; StartSmart workshop, persona and use case [19]; Brown, *Change by Design* (design thinking) [24]; Blank & Dorf, customer discovery [26]; course lecture *Ideas* (target customer) [12].

PART II

Designing the Venture



In this part

With a validated opportunity in hand, the question becomes architectural: how will this venture actually work? This part designs the machine — the business model and its open-innovation boundary, the innovation strategy and process that keep it novel, the prototype and MVP that test it, and the competitive position that defends it. The practicum's logic here is ruthless: every design choice is a hypothesis, every hypothesis gets an experiment, and every experiment has a kill criterion.

Chapters in this part: 4 Business Models and Open Innovation · 5 Innovation Strategy and the Innovation Process · 6 From Idea to MVP: Prototyping and Experiments · 7 Competitive Positioning and Strategy

Business Models and Open Innovation

Learning outcomes

After this chapter you will be able to:

- explain what a business model is and design one with all nine building blocks;
- recognize and apply the five classic business model patterns;
- distinguish open from closed innovation and design deliberate knowledge flows across your venture's boundary;
- map your ecosystem of partners and identify where your model is (and should be) open;
- work the outer ring (R8) of your system map and trace ecosystem shocks inward.

4.1 What a Business Model Is

A business model is the logic by which an organization creates, delivers, and captures value [6]. Two questions anchor it: *Who is the customer?* and *What are their needs?* — but a successful model answers them in a way that is **scalable**: value flows to defined customers continuously, without you personally pushing each unit [20]. The product is what you build; the business model is how that building becomes a company. Students routinely perfect the first and improvise the second — and then wonder why a good product did not become a business.

4.2 The Nine Building Blocks

The Business Model Canvas organizes the logic into nine coupled blocks on one page (Figure 4.1) [6, 20]:

- **Customer segments** — for whom we create value (your Chapter 2–3 work lives here);
- **Value propositions** — the bundle of products and services that solves their problem;
- **Channels** — how the value reaches the customer (and the customer reaches you);
- **Customer relationships** — what kind of link we maintain (self-service, community, dedicated support...);
- **Revenue streams** — how the venture captures money: subscriptions, sales, commissions, freemium;
- **Key resources** — what we must have: people, data, IP, platform;
- **Key activities** — what we must do: build, operate, market, partner;
- **Key partnerships** — who we depend on and why;
- **Cost structure** — what the model burns to run.

The canvas is not nine boxes to fill but *one system to balance*: every arrow you draw between blocks is a hypothesis, and the weakest arrow — not the strongest box — determines the model.

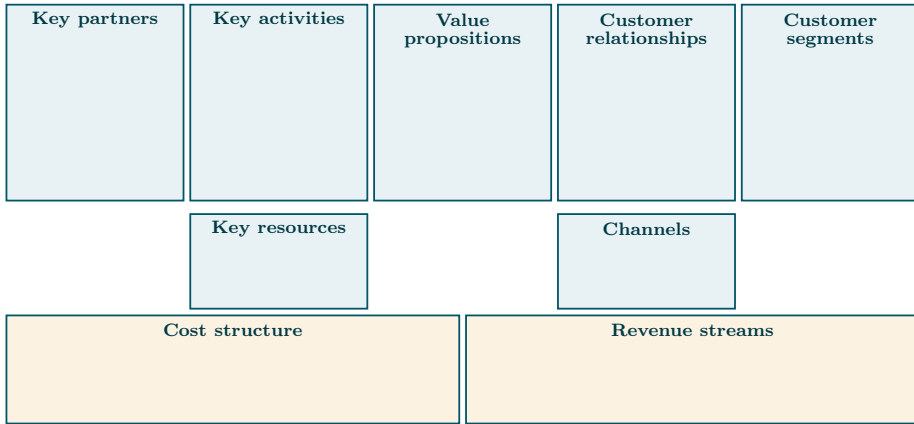


Figure 4.1: The Business Model Canvas: nine building blocks, one system. Fill it in this order: segments → value → channels → relationships → revenue → resources → activities → partners → costs.

A note on the blocks students underestimate. **Channels:** “we will be on social media” is not a channel strategy; ask where your persona actually is at 7pm on Friday, and whether you can afford to be there [20]. **Customer relationships:** retention is designed, not hoped for — an awarding system, a newsletter, a community are deliberate choices with costs. **Cost structure:** include the monetary value of your own time — *what would it cost to hire yourself?* — or your model is profitable only because the founders are unpaid [20].

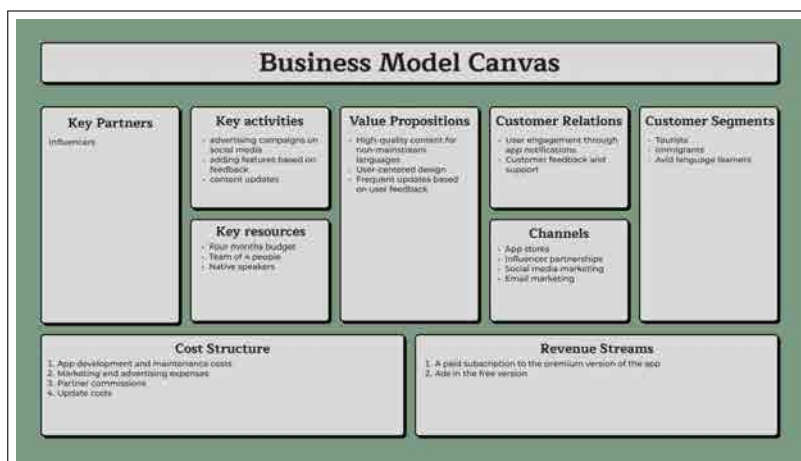


Figure 4.2: Student artifact: the *Fridge Mind* team’s Business Model Canvas (reproduced with permission). Compare it block by block with Figure 4.1: the value proposition is carried over from their VPC work (Chapter 3), key activities name the AI/ML pipeline explicitly, and the cost structure includes development tools – the canvas reads as one system, not nine separate lists.

Worked example

SMARTwear — the same nine blocks, a hardware accent. A second student team filled the canvas for smart glasses (blue-light protection, auto-adjusting lenses, AI support). Read it against *Fridge Mind*’s software canvas: *key partners* are dominated by trustworthy suppliers and distributors; *key resources* name a patent, a physical retail store, and skilled technicians next to the software team; *channels* split between the online store and in-store promotions; *revenue streams* add hardware habits a pure app never meets — product sales, Buy-Now-Pay-Later, and service fees after the two-year warranty expires. Same canvas, different physics: inventory, warranty and supply chain appear in the cost structure where *Fridge Mind* lists cloud and model training. When you fill yours, ask which *physics* your idea lives in — bits, atoms, or both.

4.3 Five Patterns Worth Stealing

Osterwalder documents recurring architectures behind successful models [6]:

- **Unbundling** — separate businesses with different economics (product innovation, customer relationships, infrastructure) that corporates merge and startups should not;
- **Long tail** — sell many niche items in small quantities each (platform economics makes the tail profitable);
- **Multi-sided platforms** — serve two or more interdependent groups who need each other (hosts and guests; drivers and riders); the hardest pattern to start, the strongest once spinning;
- **FREE** — one side subsidizes another: freemium, advertising, cross-subsidy; remember that “free” is a pricing tactic, not a business model, until you name who pays;
- **Open business models** — value created *with* partners: outside-in (their technology, data, channels in your product) and inside-out (your IP licensed into their products).

Recognizing which pattern you are in — before choosing it deliberately — prevents half of the pricing and channel mistakes we see in student plans.

4.4 Open Innovation: Designing the Permeable Boundary

For most of the twentieth century, innovation doctrine said: generate ideas inside, develop inside, protect, ship. Chesbrough named the alternative and showed it was already winning: **open innovation** — purposive inflows and outflows of knowledge across the firm’s boundary, because not all the smart people work for you and they never will [27, 28]. For a software startup this is not a strategy option but a description of reality: open-source code, cloud platforms, public APIs, pretrained models, app stores, developer communities. The question is whether your boundary is permeable *by design*.

Dodgson, Gann and Salter add the management view: innovation happens in *networks* — suppliers, customers, universities, competitors, regulators — and the firm’s task is to position itself in these networks and learn from them systematically [5]. That is exactly your outer ring, R8.

Three open-innovation moves are available to a student venture immediately:

1. **Outside-in sourcing:** build on existing components (models, datasets, APIs) instead of reinventing; your key activities shrink to what is genuinely yours.
2. **Coupled co-creation:** develop with a partner who has what you lack — a university lab, a distributor, a community (the BabyFM team used distributors precisely as an outside-in channel; see Chapter 2).
3. **Inside-out leverage:** license or expose what you build (an API, a dataset, a white-label version) so others pay to extend it.

Openness has a price: attention, alignment, and protection. What stays closed is decided in Chapter 9 (intellectual property) — openness without an IP decision is just leakage.

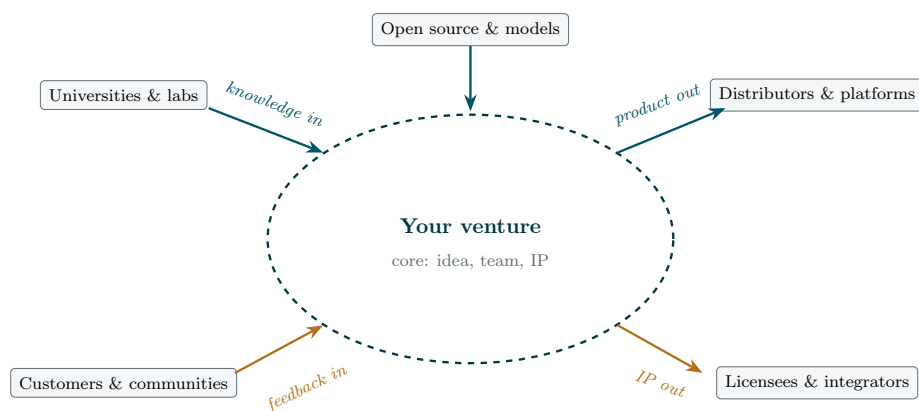


Figure 4.3: The permeable boundary: open innovation as managed flows, not a wall with holes. Decide deliberately what flows in, what flows out, and what stays in the core.

Worked example

An AI learning toy — reading a student BMC. A student team designed an AI voice-recognition toy for preschool children. Read their canvas as a system: *key partners* were NGOs and educational institutions — not decoration, but the channel and the credibility at once; *key resources* mixed the usual (developers, data scientists) with the unusual (a database of sounds and pictures, education experts); *revenue streams* combined sponsorship, donations, and subscription — three logics that must not cannibalize each other. And notice the *openness*: charity work and NGO partnerships are outside-in flows of trust and distribution that a closed model would have had to buy with marketing money. The team’s biggest unstated hypothesis sat between blocks: does the voice-recognition quality (key resource) actually sustain the “personalized storyline” (value proposition)? That arrow is what Chapter 6 would test first.

Guided steps

- Step 1. Draft.** Fill all nine blocks for your venture, in the order of Figure 4.1; use your persona and VPC from Chapter 3 for the right side.
- Step 2. Arrows.** Draw the five most load-bearing arrows between blocks; write the hypothesis on each.
- Step 3. Pattern.** Name the pattern(s) from Section 4.3 your model follows — or justify a hybrid.
- Step 4. Open it.** On the flow map (Figure 4.3), mark what you source outside-in, what you co-create, and what could go inside-out later.
- Step 5. Guard it.** Note which resources must stay closed; this seeds the IP inventory in Chapter 9.

Tasks & exercises

Task 4.1 (team). Complete Business Model Canvas, one page, no lorem ipsum bullets — each entry specific. Deliverable: canvas.

Task 4.2 (individual). Arrow hypotheses: the five load-bearing arrows, each with the metric that would prove or break it. Deliverable: arrow list.

Task 4.3 (individual). Pattern analysis: which of the five patterns fits, with one real-world comparable company. Deliverable: half a page.

Task 4.4 (team). Open Innovation Flow Map: outside-in, coupled, and inside-out flows with named potential partners. Deliverable: flow map.

Task 4.5 (team). Debate: one team argues for closing the model, one for opening it further; ten minutes each, then reconcile into a boundary decision. Deliverable: the decision + rationale.

Template

Template 4a — Business Model Canvas (Appendix): full-page nine-block canvas with the fill order printed.

Template 4b — Open Innovation Flow Map (Appendix): the boundary diagram from Figure 4.3 with empty ecosystem nodes and flow arrows to label.

Systems Thinking Lab — The Concentric Rings

Fill Ring R8 and R3. On your map, write the business model's one-sentence logic on R3, and your three most important ecosystem actors on R8.

Then shift R8. Your key partner (channel or data) terminates the cooperation tomorrow. Trace the shock inward: which arrows on your BMC break, what happens to R7 (costs), and what is your 30-day response? Write the cascade as a numbered chain — this is the systems habit in its purest form.

Record. Keep the cascade. It becomes a risk entry (with mitigation) in Chapter 10 and a resilience paragraph in your business plan.

AI angle

AI redraws every block of the canvas. As *key resource*: pretrained models and datasets substitute for assets you could never afford to build — the classic outside-in move. As *key activity*: AI compresses content production, support, even parts of coding, shrinking your cost structure. As *channel*: AI-mediated discovery (answer engines) is becoming a channel of its own. And as a *partnership* question: depending on one model provider is an R8 risk of exactly the kind you just rehearsed in the rings lab — plan a second source or an exit, in writing, now.

Chapter references

Osterwalder & Pigneur, *Business Model Generation* — canvas, patterns, open business models [6]; Chesbrough, *Open Innovation* [27]; Enkel, Gassmann & Chesbrough on open R&D [28]; Holgersson, Dahlander, Chesbrough & Bogers on open innovation in the age of AI [29]; Teece on business models and dynamic capabilities [30]; Dodgson, Gann & Salter — innovation networks and collaboration [5]; course lecture *Business Model Canvas* [20]; StartSmart workshop, business model step [19].

Innovation Strategy and the Innovation Process

Learning outcomes

After this chapter you will be able to:

- link an innovation effort to a business strategy and run a simple innovation audit;
- manage innovation as a process: idea management, the funnel, stage-gate and its Lean alternatives;
- explain how R&D and new product development are organized in technology ventures;
- position your product on the technology adoption life cycle and plan the crossing to the mainstream;
- define innovation KPIs and manage experiments as a portfolio.

5.1 Innovation Is a Strategy Before It Is a Process

Chapters 1–4 gave you an idea, a market, a customer, and a model. A fair question now is: *how much innovation does this venture actually need, and of what kind?* Not every successful startup is an R&D lab. Innovation is a means to a competitive position, not an end in itself — and Dodgson, Gann and Salter warn that tools and techniques (stage-gates, roadmaps, funnels) fail precisely when they are applied without a strategy to guide them [5].

Three strategic questions frame the chapter:

1. **Where do we innovate?** Product, process, business model, or experience — and which one carries our advantage (Chapter 1 types; Chapter 4 patterns)?
2. **How much novelty is enough?** Christensen’s warning cuts both ways: overshooting what customers can absorb is as fatal as under-shooting [31].
3. **What do we protect, what do we share?** The open/closed boundary from Chapter 4 is a strategic choice, renewed at every stage.

5.1.1 The innovation audit

A one-page audit keeps strategy honest. Score your venture 1–5 on: clarity of innovation goals; fit between innovation type and resources; balance of the idea portfolio (safe vs. risky); learning speed from experiments; openness of the boundary. Anything below 3 names this semester’s management priority [5, 8].

5.2 Managing the Process: Funnel, Stage-Gate, Lean

Ideas are abundant; attention is scarce. The **innovation funnel** (Figure 5.1) exists to kill options cheaply: many ideas enter, few receive resources, one or two reach launch. The classic managerial form is the **stage-gate** process: development is split into stages, and between them sit gates — explicit go/kill/hold decisions with criteria (Cooper’s canonical model) [32]. Stage-gate shines where mistakes are expensive: hardware, regulated products, big budgets.

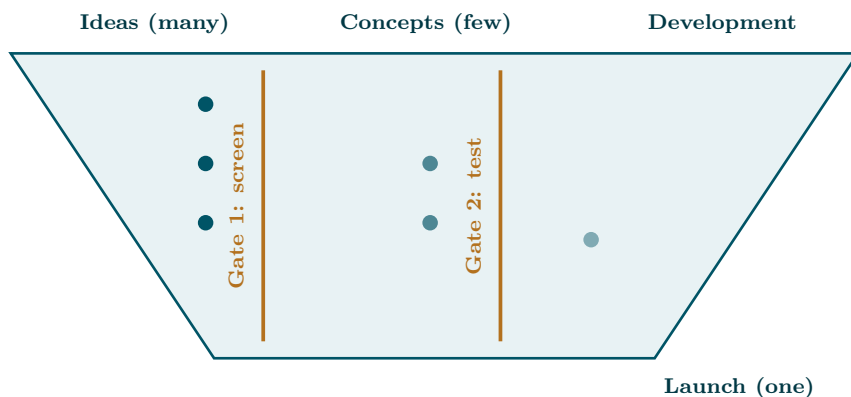
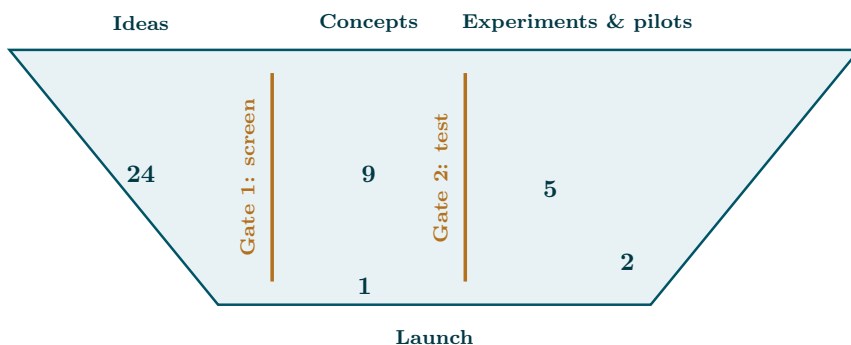


Figure 5.1: The innovation funnel: volume enters, evidence filters. Gates are decision points with criteria — not meetings with opinions.

Worked example

Fridge Mind — one semester through the funnel. The Fridge Mind team (a smart-fridge app for students and young families) kept a shared idea backlog in the fixed format *problem – who – evidence* and ran two written gates. Their semester, counted:



- **24 ideas in** — barcode scanning, expiry-date alerts, shared household shopping lists, recipe suggestions from fridge contents, fridge-camera integration, price tracking...
- **Gate 1 (screen: persona fit, evidence, effort)** kept **9 concepts**; the fridge camera died here — no evidence the persona would pay for hardware.

- **Gate 2 (test: an Experiment Card each, Chapter 6)** sent **5** to experiments; two passed their thresholds and became **2 pilots**: expiry alerts and the shared shopping list.
- **1 launched**: the expiry-alert engine, after the pilot showed weekly active use; the shopping list stays parked in the archive — with the reason written down, not deleted.

Software startups rarely need full stage-gate. Their alternative stack, which you have already met, is: **design thinking** for problem discovery [24], **Lean Startup** for build–measure–learn cycles [9], and **agile delivery** for execution (Chapter 10). But do not throw away the gate: even a three-person team benefits from one monthly moment of “are we still building the right thing?” with written criteria. That is a gate wearing a hoodie.

5.2.1 Idea management

A funnel only works if ideas enter it. Lightweight practice: a shared backlog where every team member drops ideas in a fixed format (problem – who – evidence), a weekly 20-minute triage, and a rule that killed ideas are archived with the reason, not deleted. The archive is an asset: markets change, and yesterday’s wrong idea is often next year’s right one [5, 33].

5.3 R&D and New Product Development (NPD)

In technology ventures the funnel’s development stage is NPD: turning a validated concept into a reliable product. Research on NPD performance converges on a few levers that matter far more than heroics [5, 32]:

- **upfront homework** — sharp product definition before build (your Chapters 2–3 outputs);
- **voice of the customer throughout** — not only at the start (your PMR rhythm);
- **small, empowered teams** with real ownership (Chapter 8);
- **spiral development** — build a little, show users, correct, repeat (the

MVP logic of Chapter 6);

- **ruthless gates** — stopping a weak project early is a success, not a failure [4].

Note the maturity question hiding inside NPD: is the technology itself ready? Chapter 6 introduces the **Technology Readiness Levels (TRL)** scale to answer exactly that for your MVP.

5.4 Diffusion: How Innovations Actually Spread

You can run the perfect process and still misread the market's timing. Rogers' **technology adoption life cycle** explains why: adopter groups differ fundamentally in motivation (Figure 5.2) [34]. *Innovators* (2.5%) try things because they are new; *early adopters* (13.5%) buy vision and advantage; the *early majority* (34%) buys proven practicality and references; the *late majority* (34%) buys under social pressure and standards; *laggards* (16%) buy last, if ever.

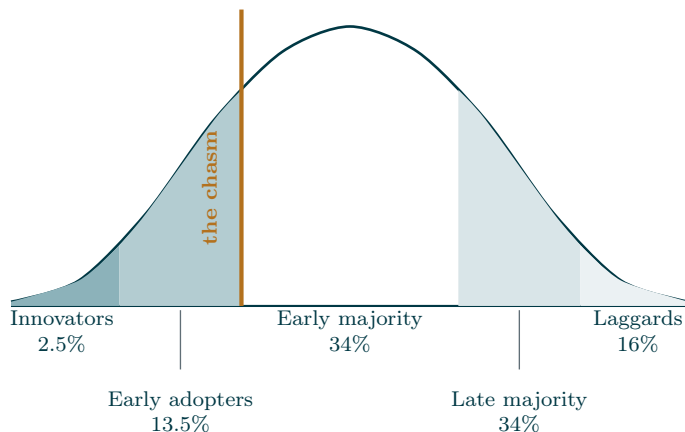


Figure 5.2: The technology adoption life cycle [34] with Moore's chasm [35]: between visionaries and pragmatists yawns a gap that consumes most products.

Moore's addition is the uncomfortable one: between early adopters and the early majority lies a **chasm** [35]. Visionaries forgive incompleteness; pragmatists demand a whole product, references, and support. Most

startups die in the chasm while celebrating early traction. The beachhead strategy from Chapter 2 is, in fact, the canonical chasm-crossing move: dominate one narrow segment completely, use it as the reference the pragmatists need, then expand to adjacent segments — the bowling-pin logic [7, 35].

5.5 Measuring Innovation

“We are being innovative” is not measurable; management needs instruments. Keep the set small and honest [8, 36]:

- **Input metrics** — ideas entering the funnel per month; experiment velocity (experiments started/finished);
- **Throughput metrics** — gate pass-through rates; time from idea to tested learning;
- **Output metrics** — validated learnings per month; percentage of revenue from offers younger than two years (later, when revenue exists);
- **Learning metrics** — pivot count with reasons; archived ideas reactivated.

Doerr’s OKR discipline applies cleanly here: one objective for the innovation effort, three measurable key results, reviewed in public on a fixed cadence [36]. Cheek’s tactics add the operational glue — owners per metric and recurring check-ins [8]. Treat your experiments as a *portfolio*: a couple of safe optimizations, a couple of bold bets, all logged, all with kill dates.

Worked example

Fridge Mind — a hardware funnel with real gates. The Fridge Mind device team’s plan shows stage-gate thinking in miniature, whether or not they named it: *blueprints* → *build hardware* → *debug* → *MVP device* → *test* → *stable version* → *final test* → *production*. Between the MVP and the stable version sits a gate in everything but name: only a tested MVP earns the right to production budget.

Notice also the parallel funnels — backend, device, and mobile app each had their own plan, which raises the integration question: who owns the gate where *all three* must pass together? In your venture, that gatekeeper role belongs to the team lead (Chapter 8) — and its criteria belong on paper, not in anyone’s head.

Guided steps

- Step 1. Audit.** Run the one-page innovation audit; record the two lowest scores.
- Step 2. Funnel.** Draw your innovation funnel: what enters, your two gates with written criteria, what survives.
- Step 3. Position.** Place your product on the adoption curve; write the evidence for the placement.
- Step 4. Plan the crossing.** Sketch your bowling-pin sequence: beachhead → which two adjacent pins?
- Step 5. Measure.** Define three innovation KPIs (one input, one throughput, one output) with owners and a review cadence.

Tasks & exercises

- Task 5.1 (individual).** Innovation audit of your venture + one paragraph on the lowest score. Deliverable: audit page.
- Task 5.2 (team).** Innovation Funnel Map with two gates and written go/kill criteria. Deliverable: funnel diagram.
- Task 5.3 (individual).** Adoption-curve positioning: where is your product, who are your first 10 customers by name/profile, what evidence supports it? Deliverable: half a page.
- Task 5.4 (team).** Bowling-pin plan: beachhead and two follow-on segments with the reference logic between them. Deliverable: one page.
- Task 5.5 (team).** Innovation KPI table: three KPIs with owner, cadence, and kill/scale thresholds. Deliverable: table.

Template

Template 5a — Innovation Funnel Map (Appendix): funnel with two editable gates and a criteria panel per gate.

Template 5b — Innovation KPI Table (Appendix): metric – type – owner – cadence – threshold, with the OKR header row [36].

Systems Thinking Lab — The Concentric Rings

See the engine. The innovation process is not a ring — it is what moves matter *between* rings: ideas leave R1, become product on R4, revenue on R7. On your map, draw two curved arrows marking where your funnel’s gates sit between rings.

Then shift it. Your Gate 2 gets stricter: the pass criterion doubles (twice the evidence required). What happens to experiment velocity (R6), to burn rate (R7), to team morale (R5)? Write one coupling per ring.

Record. Note which ring currently starves your funnel — ideas (too few enter), product (nothing exits), or learning (gates without criteria). That is your audit’s lowest score made visible.

AI angle

AI compresses the funnel itself. Ideation: models generate option volume on demand — your Gate 1 criteria must get stricter, not looser. Development: AI-assisted coding moves features from concept to demo in days, which shifts the bottleneck from *building* to *deciding what deserves to be built* — exactly the judgment this chapter trains. Diffusion: AI products currently enjoy innovator attention; the chasm question (“would a pragmatic office manager trust this?”) arrives earlier than founders expect. And measurement: use AI to mine your experiment archive for patterns, but keep the KPIs themselves boring, few, and human-owned.

Chapter references

Dodgson, Gann & Salter, Parts 2–3 (strategy, R&D, NPD) [5]; Cooper, *Winning at New Products* (stage-gate) [32]; Christensen, *The Innovator's Dilemma* [31]; Rogers, *Diffusion of Innovations* [34]; Moore, *Crossing the Chasm* [35]; Doerr, *Measure What Matters* [36]; Cheek, tactics on goals and metrics [8]; Brown [24]; Ries [9]; Sastry & Penn [4]; Maurya [33]; course lectures *Innovations and Startups* (both editions); Whittington, *Digital Innovation and Entrepreneurship* [37]; Papić, Gutić, Pantelić & Petrović on open innovation and managerial capabilities in AI-driven organizations (Sinteza 2025) [38]. Aulet, *Disciplined Entrepreneurship* (funnel discipline) [7].

From Idea to MVP: Prototyping and Experiments

Learning outcomes

After this chapter you will be able to:

- surface the assumptions hidden inside your idea and rank them by risk;
- choose the right prototype fidelity for the question you need answered;
- design an MVP that tests a hypothesis rather than demonstrates technology;
- position your technology on the TRL 1–9 scale and plan the next readiness step;
- write rigorous experiment cards with kill criteria, and run an experiment backlog.

6.1 Assumptions Before Products

Every business plan is a pile of guesses wearing a costume of certainty. Ries calls the biggest of them **leap-of-faith assumptions**: the value hypothesis (will anyone care?) and the growth hypothesis (will it spread?) [9]. Blank puts the same idea bluntly: no plan survives first contact with customers, so leave the building and test [26]. The discipline of this chapter is: *name your guesses, rank them, then buy evidence in ascending order of cost.*

An assumption matters in proportion to two things: how **impactful** it is if wrong, and how **little evidence** you currently have. Plot every assumption on those two axes and you get the working tool of this chapter — the **Assumption Map** (Figure 6.1). The top-left quadrant (high impact, no evidence) is where startups die and where your first experiments must go. The bottom-right (low impact, solid evidence) is where founders love to experiment because it is comfortable — resist it.

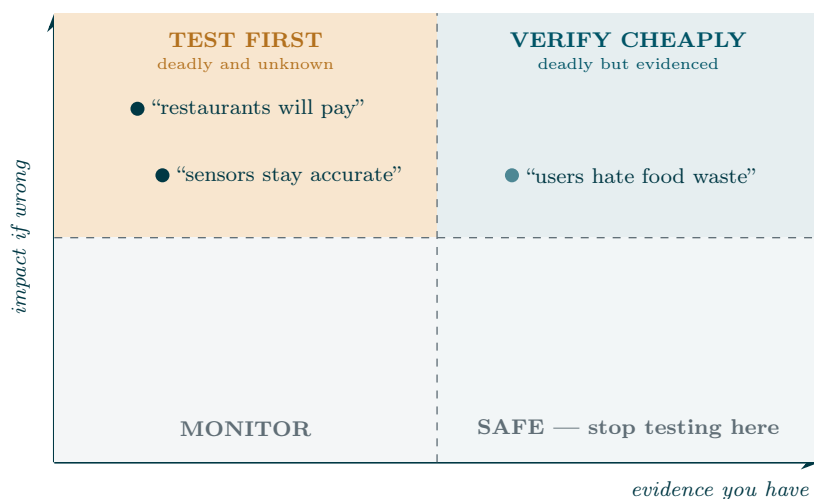


Figure 6.1: The Assumption Map: rank guesses by impact-if-wrong and by evidence in hand. Experiments start top-left and work their way down and right.

6.2 Prototyping: Ask Questions with Artefacts

A prototype is not a small product; it is a *question made physical*. The fidelity you choose should match the question, not your pride:

- **Paper / sketch** — is the flow understandable at all? (hours)
- **Wireframe / clickable mock-up** — can a user find the core action without help? (days)
- **Concierge** — deliver the service by hand to five users; learn what the service actually is (week)
- **Wizard of Oz** — the interface works, a human (or today, an AI)

does the magic backstage; tests demand before any real engineering [9]

- **Functional prototype** — the riskiest technical assumption in isolation, not the whole product.

Two rules keep prototypes honest. First, **one question per prototype** — an artifact that tests three things teaches nothing cleanly. Second, **test with the persona from Chapter 3**, not with friends: politeness is not data [25, 26].

6.3 The MVP: A Testing Machine, Not a Demo

The minimum viable product is the smallest artifact that produces *validated learning* about a leap-of-faith assumption [9]. “Minimum” refers to effort; “viable” refers to learning. An MVP that cannot falsify your hypothesis is just a small bad product.

Common MVP shapes for software ventures: landing page with sign-up counter (demand), explainer video (understanding + demand), single-feature app (core value), concierge/Oz service (value without scale), pre-sale or letter of intent (willingness to pay — the strongest currency). Disciplined Entrepreneurship’s version of the same ladder ends in the *high-level product specification* and the *MVBP* — minimum viable *business* product: enough product for a paying customer, nothing more [7]. Cheek’s execution lens adds: decide in advance what number ends the debate, and put an owner and a date on every test [8].

What an MVP is not: a beta with missing features, a tech demo for investors, or an excuse to skip design. Users judge the experience whole; sloppy MVPs measure sloppiness, not demand [33].

6.4 Technology Readiness Levels: How Real Is the Technology?

Software MVPs often hide a hardware, data-science, or model question underneath: *does the underlying technology actually work yet?* The

Technology Readiness Level scale (TRL 1–9), developed by NASA and adopted across EU research programs, is the shared language for that question [39]:

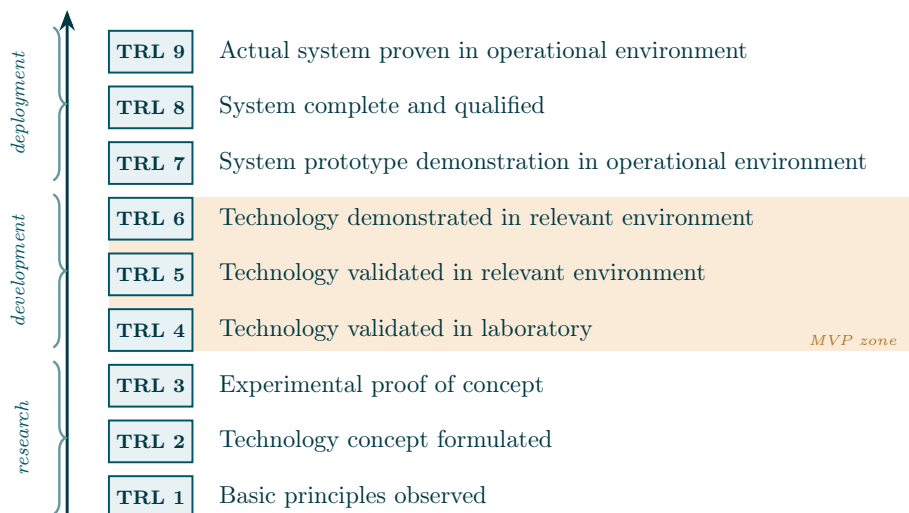


Figure 6.2: Technology Readiness Levels 1–9 [39]. An MVP usually lives at TRL 4–6: beyond the lab, not yet trustworthy in the real environment.

Place your venture’s hardest technical component on the ladder and write down the evidence for the placement (“we ran it on 200 real images from three cafeterias” is evidence; “it worked on my laptop” is TRL 4 at best). The gap between your current TRL and TRL 7–8 — where customers can rely on it — is your technical risk budget, and it feeds directly into the risk register of Chapter 10 and the milestones of Chapter 12.

6.5 Designing Experiments That End Debates

An experiment is only as good as its pre-registered decision rule. Use the **Experiment Card** (Template 6b) for every test:

1. **Hypothesis** — “We believe that *[segment]* will *[action]* because *[insight]*.”
2. **Test** — the artifact and the channel (landing page? Oz pilot?).

3. **Metric and threshold** — the number decided *before* running, with the kill/scale consequence attached (“ $\geq 8\%$ sign-up $\rightarrow$ proceed; $< 3\%$ $\rightarrow$ pivot the channel”).
4. **Timebox and owner** — experiments without end dates become hobbies [8, 36].

Run experiments as a visible backlog ordered by the Assumption Map, review weekly, and archive results with reasons — this is the same learning archive as the idea funnel of Chapter 5, now at product level. Maurya’s advice holds: speed of iteration beats quality of plan [33].

Experiment card #3 — Fridge Mind, expiry alerts (filled)

Hypothesis. We believe that *students sharing a flat will log at least five fridge items per week* because *thrown-away food is money they feel* — if logging takes under five seconds per item.

Test. Oz MVP: no scanning is built. Twelve testers from two student dormitories send a chat message per item; a teammate answers with expiry reminders by hand, pretending to be the app.

Metric & threshold. Pass if $\geq 60\%$ of testers still log five or more items in week two. Weak signal (iterate the message flow): 40–60%. Kill below 40%.

Kill / scale rule. Fail $\rightarrow$ drop auto-logging, pivot the backlog toward the shared shopping list (next card in the queue). Pass $\rightarrow$ commission the barcode-scan prototype and re-test with it.

Timebox & owner. Two weeks, weeks 7–8 of the semester; owner: the team’s product lead; weekly review in the Friday retro.

TRL before / after. Not a technology test — demand side. (The recognition sensor is tracked separately, TRL 4 $\rightarrow$ 6.)

Result & decision. Nine of twelve testers (75%) met the threshold in week two; average logging time was four seconds via chat shortcuts. *Decision: scale.* Note archived: three testers asked for “remind me *before* I shop, not after” — that sentence became Experiment Card #4.

Worked example

Fridge Mind — three MVPs in one venture. The Fridge Mind device plan (*blueprints* → *build* → *debug* → *MVP device* → *test*) is a **functional-prototype MVP**: it tests the technical assumption “sensors + camera can reliably recognize stored food” — on the TRL ladder, moving from lab validation toward demonstration in a relevant environment (TRL 4→6). But the venture needs two more MVPs its plan does not show: a **demand MVP** (would households actually pay — a landing page or pre-order test costs days, not months) and an **Oz MVP** for the app (a human sends “your milk expires tomorrow” messages by hand before any recommendation engine exists). Discussion: which assumption is currently in Fridge Mind’s top-left quadrant? Is the device build the cheapest way to test it — or is the team building the product it knows how to build rather than the evidence it needs?

Guided steps

- Step 1. Surface.** List the 10 assumptions inside your idea (value, growth, technical, channel, pricing).
- Step 2. Map.** Place all ten on the Assumption Map; circle the top-left quadrant.
- Step 3. Choose.** For the #1 assumption, pick the cheapest prototype/MVP shape that can falsify it.
- Step 4. Position.** Place your hardest technical component on the TRL ladder with written evidence.
- Step 5. Run.** Write two Experiment Cards with thresholds, kill criteria, owners, and dates.

Tasks & exercises

Task 6.1 (individual). Assumption inventory: 10 assumptions, each phrased falsifiably. Deliverable: list.

Task 6.2 (team). Assumption Map with all ten plotted and the top-left quadrant defended in five sentences. Deliverable: map.

Task 6.3 (team). MVP design: chosen shape, what it measures, why it is the *minimum*, why it is *viable*. Deliverable: one page.

Task 6.4 (individual). TRL positioning of your hardest component with the evidence and the next readiness step. Deliverable: half a page.

Task 6.5 (team). Two Experiment Cards with pre-registered thresholds and kill criteria. Deliverable: two cards.

Template

Template 6a — Assumption Map (Appendix): the 2×2 grid with plotting guide and quadrant rules.

Template 6b — Experiment Card (Appendix): hypothesis – test – metric/threshold – kill/scale rule – timebox – owner.

Systems Thinking Lab — The Concentric Rings

Fill. R4 is now your fullest ring: product definition, assumptions map, MVP shape, TRL position. Add it to your system map.

Then shift it. Your TRL assessment was too optimistic — the component is one level lower than you believed (e.g. 4 instead of 5). What moves? Experiment sequence and budget (R6), time-to-revenue (R7), the promise you can make to the persona (R2), possibly the partner you now need (R8). Trace four couplings.

Record. Mark on your map which ring currently carries the most unproven assumptions. That ring gets the next experiment, not the next feature.

AI angle

AI collapses prototype costs along the whole ladder: mock-ups, copy, landing pages, even functional front-ends appear in hours. Two consequences. First, the *Wizard of Oz is back*: an LLM behind the curtain lets you test services that would have cost a team a year — demand evidence before architecture. Second, a trap: when building is cheap, teams skip the Assumption Map and ship ten

features nobody measured. Cheap prototypes raise the value of the *question*, not of the artifact. Also place your AI component itself on the TRL ladder — a prompt that impressed you in a demo is TRL 3–4; evaluation on your real data is what moves it up.

Chapter references

Ries, *The Lean Startup* (leap-of-faith, MVP, validated learning) [9]; Blank & Dorf, *The Startup Owner's Manual* (get out of the building) [26]; Maurya, *Running Lean* (iteration speed, experiment discipline) [33]; Aulet, *Disciplined Entrepreneurship* (high-level product specification, MVBP) [7]; Osterwalder et al., *Value Proposition Design* (testing library) [25]; European Commission, TRL scale [39]; Cheek (owners and deadlines on tests) [8]; Doerr (timeboxed key results) [36]; course lectures *From Idea to MVP* and *Product Development*.

Competitive Positioning and Strategy

Learning outcomes

After this chapter you will be able to:

- define your venture's Core — the one thing you will do measurably better than anyone;
- map direct, indirect, and potential competitors and chart your competitive position;
- apply the Blue Ocean lens to escape feature-by-feature comparison;
- evaluate your business model against the forces of its environment;
- run a SWOT analysis that ends in decisions, and name your first 10 customers and follow-on markets.

7.1 Define Your Core

By now your venture has an idea, a market, a customer, a model, and a tested MVP. What it does not yet have is an answer to the hardest strategic question: *what will you be better at than anyone else on the planet?* Disciplined Entrepreneurship calls this your **Core** [7]. The Core is not your product, your technology, or a list of strengths — it is the single capability or asset that (a) your beachhead customers value deeply, (b) is hard for others to copy, and (c) you can build everything else around.

Typical Cores in software ventures: a proprietary dataset or data flywheel; a network effect; exceptional user experience in a neglected niche; a cost

structure competitors cannot match; deep domain know-how encoded in the product. “Great team” and “first mover” are not Cores — teams change, and fast followers eat pioneers [7, 9]. Notice the systems connection: the Core lives at the very center of your rings map, next to the idea itself. Everything you build on the outer rings should *compound* the Core; anything that does not is a distraction wearing a strategy costume.

A useful test: if a well-funded competitor copied your *features* tomorrow, what would still be yours in twelve months? That remainder is your Core candidate.

7.2 Know the Arena: Mapping Competitors

Students routinely claim “we have no competitors.” This is never true and always fatal: your real competitor is often *the customer’s current habit* — spreadsheets, pen and paper, doing nothing. Map three circles of competition:

- **Direct** — same problem, similar solution (other apps in your category);
- **Indirect** — same problem, different solution (agencies, Excel, pen-and-paper, WhatsApp groups);
- **Potential** — players who could enter fast (platforms, adjacent startups, the customer’s own IT department).

For each important competitor record: target segment, value proposition, pricing, strengths, and — most useful of all — *what their users complain about* in reviews and forums. Complaints are unmet needs with a name and an address; they feed straight back into your Value Proposition Canvas (Chapter 3) [25].

7.2.1 The competitive positioning chart

Once competitors are mapped, chart the position (Figure 7.1). Choose two axes that *your persona* cares about — never axes flattering to you. Price vs. quality is rarely the right pair for software; think “setup effort vs. depth of insight,” “generic vs. niche fit,” “self-serve vs. supported.” Plot

competitors honestly, then place your venture where the evidence from Chapters 2–6 says you can win [7].

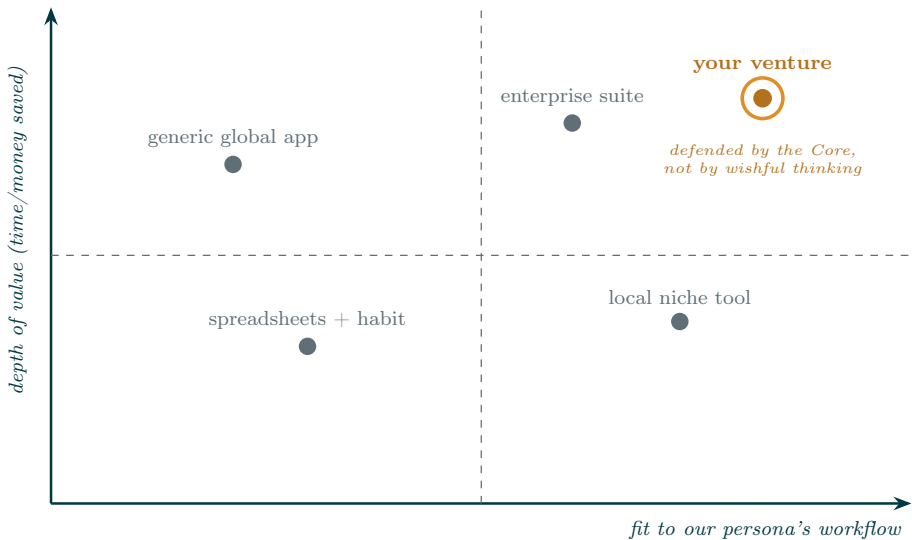


Figure 7.1: A competitive positioning chart. Axes must come from the persona’s priorities (Chapter 3); the top-right claim must survive the question “says who?”

The discipline is in the honesty: if your dot lands top-right mostly because you drew the axes, an investor will redraw them in ten seconds. Attach one piece of evidence to each axis claim (interview quote, pilot metric, review analysis) [7, 19].

7.3 Escape the Comparison: The Blue Ocean Lens

Competing head-on turns you into a feature list with a price. Kim and Mauborgne’s **Blue Ocean** strategy asks a different question: which factors of competition can you *eliminate, reduce, raise, and create* — the ERRC grid — so that comparison itself becomes awkward [40]? Cirque du Soleil did not build a better circus; it deleted animals and star performers, raised artistry and theme, and created a theater-grade experience — a new category between circus and theater.

For a student software venture, ERRC is a 30-minute exercise with outsized returns:

- **Eliminate** — which industry-standard feature or cost do customers not actually miss? (on-site installation, per-seat licenses. . .)
- **Reduce** — what is over-served? (feature count, setup complexity)
- **Raise** — what is under-served? (local-language support, niche data, onboarding care)
- **Create** — what has the industry never offered your persona? (a community, a guarantee, an integration)

Compare the result with your positioning chart: if ERRC changed nothing, your position is a red ocean and your Core is probably not load-bearing yet.

7.4 Your Business Model in Its Environment

No canvas survives contact with its context unchanged. Osterwalder and Pigneur suggest stress-testing the model against four environmental forces [6]: **market forces** (is the segment growing? switching costs?), **industry forces** (suppliers' power, substitutes, new entrants), **key trends** (technology, regulation, society — AI currently rewrites several rows at once), and **macro-economic forces** (budgets, employment, capital costs). One page, four headings, three bullets each: which force helps your model, which threatens it, and what early-warning signal you will watch. This is also where the outer ring (R8) of your map gets its systematic content.

7.5 From Position to Plan: First 10 Customers and Follow-on Markets

Positioning is only real when it names names. Two Disciplined Entrepreneurship steps close the chapter [7, 19]:

Identify your next 10 customers (DE Step 9): an actual list — names, roles, why each fits the persona, how you will reach them, and what you expect each to say. Then go ask. If 10 conversations do not

produce enthusiasm, the beachhead (Chapter 2) or the value proposition (Chapter 3) needs work; this list is the cheapest strategy audit in existence.

Follow-on markets (DE Step 14): after the beachhead, which adjacent segments open up? Estimate their TAM roughly and check that the Core transfers. This is the bowling-pin logic from Chapter 5 made concrete: beachhead first, pins second, and the sum of reachable markets is what makes the venture interesting rather than merely nice.

7.6 SWOT That Ends in Decisions

SWOT is the most used and most abused tool in student projects: four boxes, twelve adjectives, zero decisions. Done properly it is a *sorting* instrument, not a brainstorm: **strengths and weaknesses are internal** (you control them — team, Core, product, cash), **opportunities and threats are external** (you do not — market shifts, competitors, regulation). Every entry needs evidence and an owner, or it is decoration [5].

Then convert: strengths $\times$ opportunities = *attack* (where to push); weaknesses $\times$ opportunities = *build or borrow* (what to fix or partner for, Chapter 4); strengths $\times$ threats = *defend* (where the Core absorbs shocks); weaknesses $\times$ threats = *retreat or insure* — and every serious threat lands in the risk register you will formalize in Chapter 10.

Worked example

Four student SWOTs, four different lessons.

- **EatSmart** (healthy-eating app for students): the SWOT was strong on *opportunities* (canteen partnerships, student budgets) but the honest weakness — no nutrition expertise on the team — pointed to a coupled-open-innovation move: partnering with the faculty of nutrition rather than hiring.
- **Fridge Mind** (AI fridge assistant): the threat column carried the real news — big appliance makers could bundle the feature. The defense was not features but the Core candidate: the accumulated dataset of real household food-usage patterns. Notice how a

SWOT threat sharpened the Core definition from Section 7.1.

- **Creative Wave** (creative-industry platform): pairing SWOT with a simple risk table showed that their top two threats (slow two-sided adoption; dependence on a few anchor clients) were both *external*, which changed the plan from “add features” to “secure two lighthouse customers first.”
- **SMARTwear** (smart glasses): their matrix honestly listed *small market share* and *limited brand awareness* as weaknesses, against threats named *Huawei*, *Meta* and *Xiaomi*. Combined with a strength — adaptive lenses that darken in sunlight — the conversion wrote itself: do not out-advertise giants (a weakness × threat trap that only raises acquisition costs); instead *defend* the niche where the strength is decisive (users with impaired vision who need auto-adjusting lenses) and *borrow* reach through retail partners. Discussion: which of your SWOT entries would survive the demand for one piece of evidence each?

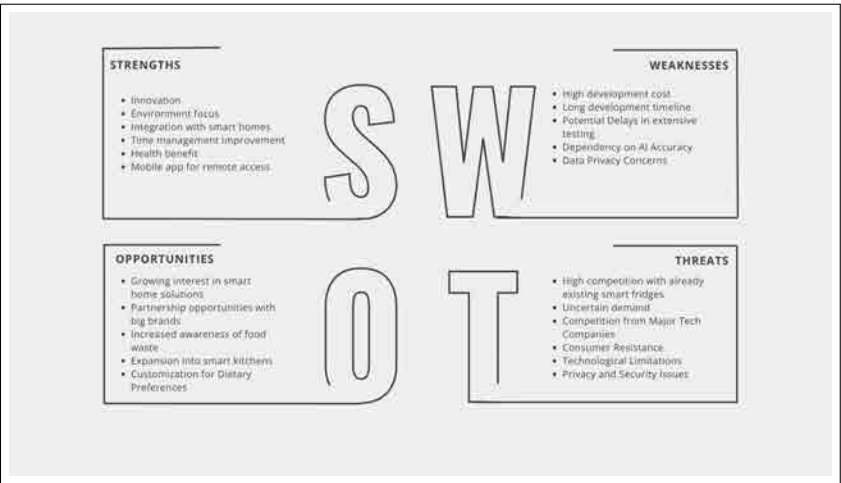


Figure 7.2: Student artifact: the *Fridge Mind* team’s SWOT slide (reproduced with permission). Two details are worth copying: entries are specific enough to act on (“AI accuracy concerns” is verifiable, “strong team” would not be), and the external items — accuracy, privacy, data handling — flow directly into the risk table the team built later (Chapter 10, Template 10a).

Guided steps

- Step 1. Core.** Write your Core in one sentence; test it against the twelve-month-copy question.
- Step 2. Map.** List 5–8 competitors across the three circles (direct, indirect, potential) with one user complaint each.
- Step 3. Chart.** Draw the positioning chart with persona-driven axes and evidence notes.
- Step 4. Escape.** Run ERRRC; adjust the value proposition or the chart if the ocean stays red.
- Step 5. Commit.** Name your first 10 customers and two follow-on markets; finish the SWOT with attack/defend/build/retreat actions.

Tasks & exercises

- Task 7.1 (individual).** Core statement + the copy-test answer in five sentences. Deliverable: half a page.
- Task 7.2 (team).** Competitor table: 5–8 competitors, three circles, one user complaint each with source. Deliverable: table.
- Task 7.3 (team).** Competitive positioning chart with axis justification and evidence. Deliverable: chart.
- Task 7.4 (individual).** ERRRC grid for your venture with one concrete change you will make because of it. Deliverable: grid + note.
- Task 7.5 (team).** SWOT with evidence column + converted actions (attack/defend/build/retreat); threats copied into the risk register starter list. Deliverable: SWOT matrix.

Template

- Template 7a — SWOT/TOWS Matrix** (Appendix): four evidence-anchored quadrants plus the four conversion fields.
- Template 7b — Competitive Positioning Chart** (Appendix): two-axis chart with axis-justification and evidence rows.

Systems Thinking Lab — The Concentric Rings

Fill. R8 gains its competitive content (the three circles, the chart) and R1 gains the Core next to the idea. Update the map.

Then shift it. A well-funded competitor launches a lookalike feature set next month, a hard shift on the outer ring. Which inner rings absorb the shock? Does your Core (R1) hold? Does your customer relationship (R2) slow the churn? Does your cost structure (R7) survive a price war? Does a partner (R8) become more valuable? Mark every ring that moves and write the defense it provides.

Record. If no inner ring absorbs the shock, your Core is not yet a Core. Decide what would make it one — that is a strategy task, not a marketing task.

AI angle

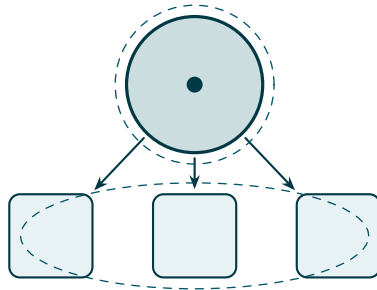
AI is the best competition analyst a student team can afford: it mines competitor reviews and forums for complaint patterns, drafts feature-comparison matrices, monitors launch announcements, and summarizes pricing pages in minutes — work that once cost a consultancy. Two warnings. First, AI summarizes what is *written*; the competitor's roadmap and your customers' switching costs are not written anywhere — interviews still rule. Second, the same analysis is available to every competitor about *you*: assume your public surface (site, app store page, docs) is being machine-read and keep your real differentiation where copying is slow — in the Core.

Chapter references

Aulet, *Disciplined Entrepreneurship* (Core, competitive position, next 10 customers, follow-on markets) [7]; StartSmart SEE workshop, Steps 9–12 and 14 [19]; Kim & Mauborgne, *Blue Ocean Strategy* (ERRC) [40]; Osterwalder & Pigneur, *Business Model Generation* (model environment) [6]; Osterwalder et al., *Value Proposition Design* [25]; Ries (fast-follower dynamics) [9]; Dodgson et al. (strategy tools in practice) [5]; course lecture *Opportunity Analysis* (competitor mapping) [13].

PART III

Building the Organization



In this part

A design does not execute itself; an organization does. This part is the management core of the practicum, and it starts from the classic foundations: management as the functions of planning, organizing, leading and controlling (POLC); effectiveness before efficiency — doing the right things before doing things right; and objectives as the contract between a team and its future [10]. You will assemble the founding team and write its charter, give the venture a legal shell and protect its intellectual property, and install the project and risk disciplines that keep delivery honest.

Chapters in this part: 8 Founding Team, Roles and Leadership · 9 Legal Structure and Intellectual Property · 10 Project and Risk Management for Startups

Founding Team, Roles and Leadership

Learning outcomes

After this chapter you will be able to:

- choose co-founders by complementarity and commitment, and structure the equity conversation with vesting;
- read your venture as an organizational-behavior system: individuals, group, culture, structure;
- map your team against Belbin's nine roles and close the dangerous gaps;
- run the basic communication and leadership rhythms of a technology team;
- recognize founder burnout early and design the team to survive a key person leaving.

8.1 Why Teams Beat Heroes

Investors have a saying: *bet on the team, not the idea* — and Chapter 1 gave the reason: ideas pivot, teams execute. A venture is, before anything else, a small group of people making hundreds of irreversible-feeling decisions under uncertainty, at speed, with no boss to escalate to. Everything in this chapter serves one goal: make that group resilient enough to survive its own learning curve.

The research is blunt about solo founders: slower progress, higher dropout,

harder fundraising [7]. But a bad co-founder is worse than none. Choose by two criteria, in this order: **commitment** (will this person still be here, working, in year three of no salary?) and **complementarity** (do they cover what you cannot: building, selling, or the domain?). Friendship is neither — it is only the glue, and glue alone does not carry load.

8.2 The Equity Conversation — Have It Now

Nothing kills student teams more reliably than unspoken equity assumptions. The mechanics are simple; the courage is the hard part.

- **Split by contribution and risk, not politeness.** Who brings the idea is worth less than who brings the next two years. Weight: past contribution, future commitment (full-time vs. evenings), cash invested, opportunity cost.
- **Vesting is non-negotiable.** Standard: four-year vesting with a one-year *cliff* — nobody owns anything until twelve months of work, then shares mature monthly. The person who leaves in month three keeps nothing; the team that stayed keeps the company. This one clause prevents the classic disaster: a departed founder with 33% and no obligations.
- **Write it down.** A one-page founders' agreement: roles, equity, vesting, decision rules, what happens when someone leaves, and how disputes get resolved. You are not planning to fail; you are making failure survivable.

8.3 The Human System: OB in Miniature

Organizational behavior sounds like a corporate subject; it is actually the physics of your five-person team. Robbins and Judge's levels map directly onto a startup [41]: the **individual** (motivation, skill, personality — why is your developer slower this month?), the **group** (norms, roles, conflict — who really decides?), the **culture** (what behavior gets praised here?), and the **structure** (who owns what, formally). In a small team all four levels are visible in one room — which means dysfunctions are visible early, if you dare to look.

Culture deserves a paragraph of its own because in a startup it is set once, by the founders, mostly by accident. The HBR Manager's Handbook puts it operationally: culture is what newcomers conclude they must do to succeed here [42]. You will write a culture either way; writing it *deliberately* is the management move. This is where Dalio's *Principles* is unexpectedly practical for students: his core idea — run the team on **written principles** and radical truth-telling, so decisions come from agreed logic rather than from mood or hierarchy [43] — scales down perfectly to five people. A one-page team charter (Template 8a) is your Principles, miniature edition: how we decide, how we disagree, how we communicate, what we do when someone is stuck.

And keep Wheatley's warning nearby: teams, like all living systems, *self-organize* — you cannot command behavior into existence, you can only set the conditions (identity, information, relationships) and then pay attention [3]. The retrospective is the attention instrument; skipping it is how small cracks become founder breakups.

8.4 Belbin's Nine Team Roles

Skills are what people *can* do; team roles are what they *tend* to do under pressure. Belbin's research identified nine behavioral roles that cover what a team needs (Figure 8.1): three **thinking** roles (Plant — the idea generator; Monitor Evaluator — the sober judge; Specialist — the depth provider), three **action** roles (Shaper — the pusher; Implementer — the organizer; Completer Finisher — the quality guard), and three **social** roles (Coordinator — the clarifier of goals; Teamworker — the diplomat; Resource Investigator — the networker) [44].

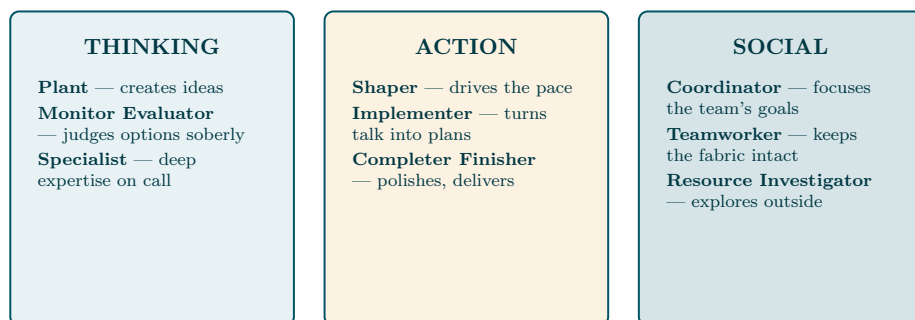


Figure 8.1: Belbin’s nine team roles in three families [44]. A student team of five must cover nine functions — expect double-hatting, and *name* it.

Two rules make this useful rather than horoscopic. First, **map gaps, not labels**: a team of four Plants and a Shaper ships nothing. The classic student-team gap is Completer Finisher — nobody loves the last 10%, which is why demos crash. Second, **allowable weaknesses come with each role** (Plants are chaotic, Shapers annoy people): the charter should name them, so the team stops reading role behavior as personality flaws [44].

8.5 Leading a Technology Team

Leadership style follows the work’s maturity, not the founder’s taste. Early chaos needs **directive** clarity (who does what by Friday); a maturing team needs **coaching** and then **delegation**; throughout, the *situational* question is “what does this person need from me for *this* task?” [41, 42]. In software teams two rhythms carry most of the load:

- **Weekly cycle**: one short planning moment (three priorities, not thirty), one demo of working software, one retrospective (what helped, what hurt, one change for next week). Cheek’s tactics add the owner-and-deadline discipline that keeps the cycle honest [8].
- **Decision rules**: agree *who decides what* — technical choices to the person who builds, product choices to the persona’s evidence, strategy to the whole team with the lead’s tie-break. Decision latency, more than coding speed, is what slows student teams.

Recent survey work on open-source software teams also links transformational leadership to stronger organisational commitment and innovation in AI-driven settings [45].

Senge's learning-organization point closes the loop: the team's real product is its own learning capacity — a team that reviews itself weekly compounds like interest [1].

8.6 Founder Burnout and Team Sustainability

Burnout is a systems failure, not a character flaw: chronic overload plus no visible progress plus isolation. The defenses are structural — realistic weekly scopes (your velocity is data from Chapter 6 experiments, not ambition), rotating the unglamorous roles, protecting one screen-free day, and *saying the hard thing early* in the retrospective instead of at the breaking point [3, 42]. Watch for the symptoms in each other: cynicism about the product, meetings skipped, code quality quietly dropping. A team that notices in week two fixes a workload; a team that notices in month six attends a breakup.

Worked example

How real student teams assembled. **Creative Wave** began from a member list with declared strengths (design, video, marketing, web) — then discovered the gaps only when the first client deadline approached: nobody owned *delivery management*, and the Belbin map would have shown it in fifteen minutes (three social/thinking roles, no Implementer). Their fix — naming a rotating “delivery owner” per project — is a charter clause, not a hire. **HotHeads** ran their five-person team on a shared Trello board with columns for backlog / this week / doing / done — a humble tool, but it made *work visible*, which is the single highest-leverage move in a student team: invisible work cannot be discussed, and undiscussed work becomes conflict. Discussion: on your team's board right now, which card has been in “doing” the longest — and what is the Belbin explanation?

Guided steps

- Step 1. Needs first.** List the six functions your venture needs this semester (build, sell, design, domain, ops, finance).
- Step 2. Map.** Do the Belbin self-assessment per member; place everyone on the roles map; circle the gaps.
- Step 3. Charter.** Draft the one-page team charter: decision rules, rhythms, disagreement protocol.
- Step 4. Equity.** Run the equity checklist: contributions, commitment, vesting, leaver clause.
- Step 5. Stress.** Walk the rings-lab scenario (key person leaves) and record the single point of failure you will fix first.

Tasks & exercises

- Task 8.1 (individual).** Your own Belbin profile: two preferred roles, one allowable weakness, one role you will consciously cover for the team. Deliverable: half a page.
- Task 8.2 (team).** Belbin Roles Map of the whole team with gaps marked and a cover plan per gap. Deliverable: map.
- Task 8.3 (team).** Team Charter: decision rules, weekly rhythm, disagreement protocol, burnout early-warning signals. Deliverable: one page, signed.
- Task 8.4 (team).** Equity split proposal with rationale and a vesting schedule (four years, one-year cliff). Deliverable: one page.
- Task 8.5 (individual).** Bus-factor analysis: if you left tomorrow, what breaks first? Deliverable: five sentences.

Template

- Template 8a — Team Charter** (Appendix): purpose, roles, decision rules, rhythms, conflict protocol, signatures.
- Template 8b — Belbin Roles Map** (Appendix): the nine-role grid with member placement and gap analysis rows.

Worked example

ZenTutor — a team charter excerpt and the Belbin map, filled in class. Four co-founders, one page of rules written *before* the first conflict:

- *Mission (one sentence):* every student in Serbia finds a trusted tutor in under ten minutes.
- *Decision rights:* product lead decides UX and backlog order; growth lead decides channels and campaigns; **pricing and tutor-vetting standards require the whole team.**
- *Delivery owner:* rotates each sprint (the Creative Wave fix, adopted on day one); the owner runs the Friday retro.
- *Conflict rule:* disagree in the meeting, commit within 48 hours, revisit only with new evidence — never with the same opinion louder.
- *Visibility rule:* the board (Template 10c) is updated daily; a card older than a week in *Doing* is discussed, not ignored.

Member	Strongest Belbin roles	What they own
A. (product)	Plant, Specialist	app, backlog, UX
B. (marketplace ops)	Coordinator, Teamworker	tutor vetting, support
C. (growth)	Resource Investigator, Shaper	partnerships, campaigns
D. (finance/ops)	Monitor-Evaluator	budget, metrics, legal
Gap	no Implementer / Completer-Finisher — mitigation: rotating delivery owner + definition-of-done checklist, not a hire the budget cannot carry	

Systems Thinking Lab — The Concentric Rings

Fill. R5 now has content: members, roles, charter, rhythms. Add it to your system map.

Then shift it. Your most loaded person — the one holding the most “doing” cards — leaves the team next week. Trace the cascade: delivery slips (R6), the MVP date moves (R4), the first-10-customers

meetings pause (R2), burn rate extends (R7), and a partner deadline may break (R8). Mark every ring that moves within one month.

Record. Find your single point of failure — the person without whom one ring stops. The fix is almost never “hire”: it is documentation, pairing, and rotating ownership. Write the fix into the charter.

AI angle

AI quietly changes team arithmetic. Tasks that justified a fifth or sixth member — first-draft code, copy, design variants, research summaries — are now AI-augmented, so the optimal student team is *smaller and more senior in judgment*. A new role appears: the **AI product owner**, who owns prompts, evaluation sets, model choice, and cost-per-feature — on the Belbin map this is usually a Monitor Evaluator with Specialist depth. Two cautions: AI output still needs a Completer Finisher (the last 10% got cheaper to start, not to finish), and letting AI mediate *team communication* (summaries of standups, auto-written feedback) erodes exactly the trust this chapter is building. Use AI for artifacts, keep humans for relationships.

Chapter references

Belbin, *Team Roles at Work* [44]; Robbins & Judge, *Organizational Behavior* [41]; HBR *Manager’s Handbook* (culture, styles, rhythms) [42]; Dalio, *Principles* (written decision logic) [43]; Wheatley, *Leadership and the New Science* (self-organization) [3]; Senge, *The Fifth Discipline* [1]; Aulet (founding teams, solo-founder risk) [7]; Cheek (owners, deadlines) [8]; Drucker, *Management* (objectives and responsibility) [10]; Skelton & Pais, *Team Topologies* (team structures for fast flow) [46]; Bock, *Work Rules!* (people practices from Google) [47]; Forsgren et al. on the SPACE dimensions of developer productivity [48]; course lectures on teams and leadership. Aleksić & Dakić on transformational leadership in open-source development [45].

Legal Structure and Intellectual Property

Learning outcomes

After this chapter you will be able to:

- choose a legal form for the venture with a justification, and time the registration correctly;
- explain limited liability — and when it does not protect you;
- take an inventory of your intellectual property across copyright, trademarks, patents, and trade secrets;
- handle the IP questions specific to software: code, open-source licenses, contractors, and AI-generated content;
- write a one-page protection plan that fits a student budget.

9.1 Legal Form Is a Strategy Decision, Not Paperwork

Founders treat registration as bureaucracy; it is actually four decisions in one. **Liability:** who pays when things go wrong — the company, or you personally? **Taxes:** which regime your revenue flows through. **Credibility:** nobody signs a serious B2B contract — and no investor invests — into a private individual. **Ownership:** the legal form is the shell that holds your equity split (Chapter 8) and your IP (this chapter). Choose the shell before you need it; needing it means a contract or an investment is already on the table.

The timing rule of thumb: *register when the first real obligation appears* — the first paying customer, the first employee, the first investor conversation, or the first partner NDA. Earlier than that you pay administrative costs for a venture that may still pivot; later than that you are personally on the hook for everything.

9.2 The Menu of Legal Forms

Under Serbian law (and most European systems follow the same logic), a student venture realistically chooses among these [49, 50]:

Form	Liability	Min. capital	Best for
Entrepreneur (<i>pre-duzetnik</i> , incl. flat-rate <i>paušalac</i>)	unlimited, personal	none	freelancing, first small revenues, validating before commitment
LLC (<i>d.o.o.</i>)	limited to contribution	RSD 100 (symbolic)	the default for startups: teams, investors, B2B contracts
Joint-stock (<i>a.d.</i>)	limited	RSD 3,000,000	regulated sectors, stock-exchange ambitions — not for students
Partnerships (<i>o.d./k.d.</i>)	personal / mixed	none	rare; professional practices
Branch / representative office	parent's	—	foreign companies entering the market

Table 9.1: Legal forms at a glance. Registration runs through the Serbian Business Registers Agency (APR); an LLC registration typically completes within about a week [50].

Three practical notes. First, **limited liability is not absolute**: mixing personal and company money, or using the company to defraud creditors, lets courts “pierce the veil” and reach your personal assets — clean bookkeeping is part of the protection. Second, the **paušalac route is a feature, not a loophole**: a predictable monthly lump sum (available below a turnover cap) lets a student validate real revenue with near-zero accounting overhead; when revenue or investors outgrow it, the natural graduation is the d.o.o. Third, corporate income tax in Serbia is a flat 15%, and qualifying IP income enjoys a significantly reduced effective rate (the “IP box”) — one more reason to take the IP half of this chapter seriously [50].

9.3 Intellectual Property: What You Actually Own

IP law protects four different things; confusing them is the classic student mistake [51, 52]:

- **Copyright** — protects *expression*: your code, texts, designs, pitch deck. It arises *automatically* the moment the work is created; no registration needed (though optional deposits help prove dates). It does *not* protect the idea, the algorithm, or the functionality.
- **Trademark (žig)** — protects *signs*: product name, logo. This one requires registration (national office, or EUIPO/international via Madrid) and a prior *search* — naming your product before checking the register is how teams get rebranded by a lawyer’s letter.
- **Patents and industrial designs** — protect *technical inventions* and *product appearance*. Patents are slow and expensive (years, real money), and in Europe *pure software “as such” is excluded* — only computer-implemented inventions with a genuine technical effect may qualify. For most student software, patents are the wrong tool; speed and secrecy beat them.
- **Trade secrets** — protect *what you do not disclose*: datasets, trained-model configurations, pricing logic. Protection here is behavioural: NDAs, access control, and need-to-know.

9.3.1 The IP traps that actually hit student teams

1. **The contractor trap.** Code written by a freelancer or a friend belongs to *them* unless a written contract assigns it to the venture. One unsigned contributor can block an investment round years later. The fix is a one-page IP-assignment clause for *everyone* who touches the product — including all team members (this belongs in the founders’ agreement from Chapter 8).
2. **The open-source trap.** Libraries under permissive licenses (MIT, Apache 2.0) are safe to build on; copyleft licenses (GPL) can force *your* code open if you distribute it. Keep a list of every dependency and its license — investors’ lawyers will ask.

3. **The university trap.** Built it in a lab, on university equipment, with a mentor? Check what your faculty’s rules say about student IP *before* the success, not during it.

9.4 The Protection Plan

A student-budget plan fits on one page: (1) IP inventory table — every asset, its type, its owner, its status (Template 9b); (2) IP-assignment clauses signed by all contributors; (3) trademark search for the product name, registration when first revenue lands; (4) trade-secret hygiene — private repositories, NDAs for pilot partners; (5) calendar reminder to revisit patents only if a hardware or deep-tech component emerges (your TRL ladder from Chapter 6 will tell you).

Worked example

Creative Wave — who owns an AI-generated design? The Creative Wave platform connects clients with designers who increasingly use generative AI in their workflow. Their compliance question came from a real client brief: *if the logo was AI-assisted, can the client register it as a trademark, and can anyone claim copyright on it?* The team’s working policy, after checking with the IP Office guidance: (a) pure AI output without meaningful human creative contribution is treated as *not copyrightable* — so it cannot be exclusively sold as “copyrighted work”; (b) designs with documented human authorship (briefs, sketches, selection, substantial editing) are treated as human works *with* AI assistance, and the platform keeps the process record as evidence; (c) every delivery contract states which of the two the client is buying, and the warranty differs accordingly. The lesson generalizes: in the AI era, **your process documentation is part of your IP** — the prompts, drafts, and edit history are the proof of human authorship [51, 52].

Guided steps

- Step 1. Decide.** Run the decision logic: personal liability acceptable? investors/B2B soon? → pick preduzetnik or d.o.o. with a one-line justification.
- Step 2. Time it.** Write the trigger event that will make you register (first invoice, first hire, first investor).
- Step 3. Inventory.** Fill the IP inventory: code, name, logo, designs, datasets, know-how — type, owner, status.
- Step 4. Search.** Do the trademark search for your product name in the national register; screenshot the result.
- Step 5. Protect.** Draft the one-page protection plan; list which dependencies are copyleft.

Tasks & exercises

- Task 9.1 (individual).** Legal form choice with justification: form, trigger for registration, one risk you personally carry until then. Deliverable: half a page.
- Task 9.2 (team).** IP inventory table covering at least six assets, each with type/owner/status. Deliverable: table.
- Task 9.3 (individual).** Trademark check: search the register for your product name and the two closest alternatives; report conflicts. Deliverable: screenshot + five sentences.
- Task 9.4 (team).** Dependency license audit: list your top ten libraries/models/datasets with license type; flag copyleft. Deliverable: list.
- Task 9.5 (team).** One-page protection plan including the IP-assignment status of every contributor. Deliverable: plan.

Template

- Template 9a — Legal Form Decision Table** (Appendix): the four-criteria comparison with a justification row.
- Template 9b — IP Inventory** (Appendix): asset – IP type – owner – protection status – action.

Worked example

Fridge Mind — the dependency license audit, filled. The team’s top dependencies, with the two traps this chapter warned about actually appearing in a real list:

Dependency	License	Action / flag
React Native (app shell)	MIT	safe; keep notice file
TensorFlow Lite (on-device model)	Apache 2.0	safe
ZXing (barcode scanning)	Apache 2.0	safe
PostgreSQL (database)	PostgreSQL License	safe
Food-101 image dataset	CC BY-NC	red flag: non-commercial only — cannot train the product model on it; replaced with own scans + licensed set
“RecipeGuru” scraping script	none stated	no license = all rights reserved — deleted; recipes via a licensed API instead
UI kit from a freelancer	contract	trap: contract was silent on IP — annex signed assigning rights to the company
University lab GPU time	university rules	checked: results stay ours; acknowledgment added

Two red flags found before investors’ lawyers could find them — that is the entire point of the exercise.

Systems Thinking Lab — The Concentric Rings

Fill. Your legal shell touches two rings directly: R7 (taxes, liability, investment-readiness) and R8 (who can contract with you, and what you may license in or out). Mark the chosen form on both.

Then shift it. A corporate partner loves your pilot and offers a paid contract — on condition that all IP from the joint work is assigned to *them*. Watch the rings: revenue appears on R7, but R1 (your Core) may walk out of the building, R4’s roadmap becomes theirs, and future R8 partners face a field you no longer own. Write the counter-offer your team would propose instead (license instead of assignment? field-of-use limits?).

Record. Note one sentence in your venture one-pager: “Our IP position is...” If you cannot finish the sentence, the inventory is not done.

AI angle

Two AI questions dominate student IP consultations. **Who owns AI-generated content?** Practice is converging on: meaningful human creative contribution is required for copyright — prompts alone are a weak claim; documented selection, editing, and composition are a strong one (keep the record, as Creative Wave did). **Can AI be an inventor?** Patent offices and courts (the DABUS cases) have answered: inventors must be human; AI-assisted inventions are patentable only through the human contribution. And a third, quieter risk: the *terms of service* of your AI tools — some business tiers grant the provider rights to process your data in ways a stealth-mode venture should not accept. Read the terms of the tier you actually pay for.

Chapter references

Course lecture *Legal Forms and Intellectual Property* [49]; WIPO, IP for business guides [51]; Intellectual Property Office of the Republic of Serbia, registers and guides [52]; Serbian Business Registers Agency, registration portal [50]; for the strategy context: Dodgson et al. on appropriability [5]; Aulet on investor expectations [7].

Project and Risk Management for Startups

Learning outcomes

After this chapter you will be able to:

- structure your venture’s work as a project: lifecycle, phases, milestones, deliverables;
- choose consciously between Waterfall, Agile, and the hybrid that student teams actually need;
- plan with WBS and Gantt, steer with Kanban, and reason about the critical path;
- run a real risk process: identify, prioritize by probability $\times$ impact, mitigate, review;
- protect your scarcest resource — founder time — with deliberate time management.

10.1 Work Deserves a Container

Everything so far in this practicum — research, prototypes, experiments, partnerships — is *work*: bounded efforts with a beginning, an end, and a deliverable. Project management is the discipline of giving that work a container: scope, sequence, dates, owners, and a way to notice trouble early. Student teams resist it (“we are agile, we don’t plan”) and then discover, around week eight, that nobody knows what “done” means, three people built the same feature, and the demo date was never written down. The point of this chapter is not bureaucracy; it is **making trouble**

visible while it is still cheap [22].

There is also a deeper reason. The rings model says the project ring (R6) is the *transmission belt* of the venture: it converts decisions made on every other ring into scheduled, owned work. A strategy that never reaches a Gantt chart is a wish; a plan that ignores the risk register is fiction.

10.2 Lifecycles: Project and Product

A **project lifecycle** is the generic arc: *initiate* (why, for whom, success criteria) → *plan* (scope, WBS, schedule, budget, risks) → *execute* (build, communicate, track) → *monitor & control* (in parallel: are we on scope/time/cost?) → *close* (deliver, document, retrospective). The closing phase is the one student teams skip, and it is the one that produces the learning [22].

Software adds its own loop, the **SDLC**: requirements → design → implementation → testing → deployment → maintenance. Two truths about the SDLC matter for you. First, the phases exist regardless of methodology — Agile does not abolish testing, it redistributes it. Second, the cost of fixing a defect grows by roughly an order of magnitude per phase it survives: a wrong requirement caught in a conversation costs minutes, caught in production it costs users [22]. That asymmetry is the economic argument for everything in Chapter 6: prototypes and experiments are defect-catching moved earlier. Continuous integration, containerisation and microservices can also help maintain compliance in software delivery across different industry contexts [53].

10.3 Waterfall, Agile, and the Student Hybrid

Waterfall runs the SDLC strictly in sequence: full requirements, then full design, then build, then test. It works when requirements are stable, the technology is known, and the cost of rework is high — hardware, regulated domains, contractual fixed scopes (the Fridge Mind device plan from Chapter 5 was a waterfall in miniature, and rightly so). Its failure mode is famous: you learn the requirements were wrong only at the end, when changing them costs the most.

Agile assumes requirements are discovered, not known: short iterations, working software every cycle, continuous user feedback, scope flexes instead of dates. Scrum gives you roles and ceremonies; Kanban gives you a board and work-in-progress limits. Agile works when learning speed is the bottleneck — which is the definition of a startup [9].

The honest answer for a student venture is a **hybrid**, and you have already seen its parts: *Agile inside, gates outside*. Inside the semester you iterate weekly (Chapter 8’s rhythm); at the few decision points that matter — MVP scope freeze, pilot start, launch — you hold a gate with written criteria (Chapter 5’s funnel). Choose per component: the mobile app is Agile, the hardware is Waterfall, the integration milestones are gates [22, 54]. Figure 10.1 puts the two logics side by side.

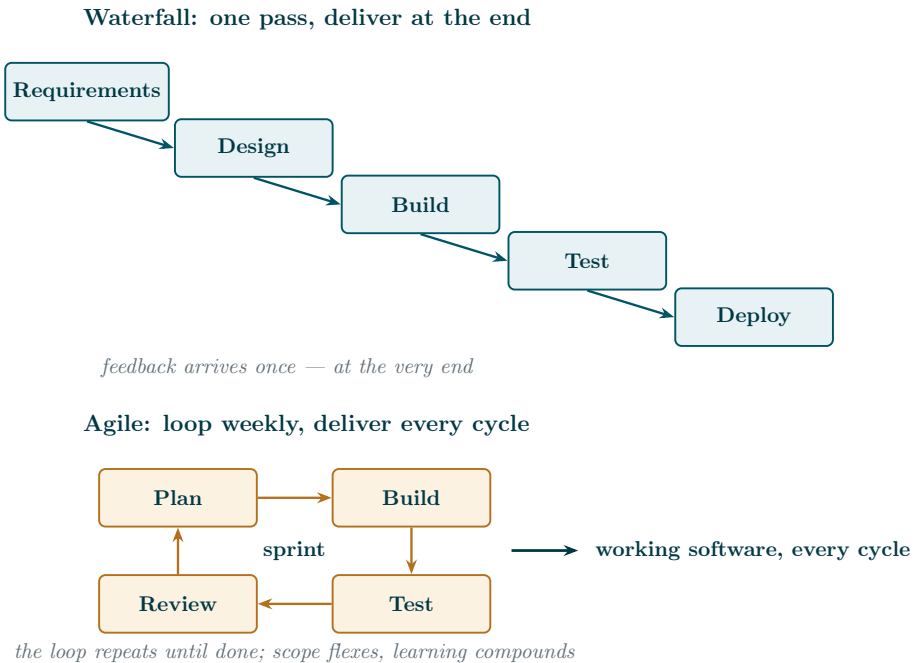


Figure 10.1: Two delivery logics. Waterfall moves each phase to completion once and learns late; Agile repeats a small plan–build–test–review loop and learns every week. The student hybrid uses the loop inside the semester and waterfall-style gates at the few irreversible decisions.

10.4 Planning Tools: WBS, Gantt, Critical Path, Kanban

10.4.1 WBS: decompose before you schedule

A **Work Breakdown Structure** decomposes the project into deliverable-oriented chunks, recursively, until each leaf is a task of one-to-three days with a single owner. Rules: decompose *deliverables*, not activities (“payment flow working in staging,” not “work on payments”); the 100% rule (the WBS covers everything — including testing, documentation, and the pitch); and if you cannot estimate a task, it is not decomposed enough. The WBS is also your best scope police: anything not in it is, by definition, a change request [22].

10.4.2 Gantt: sequence and time

The **Gantt chart** lays WBS tasks on a calendar: bars for tasks, diamonds for milestones, arrows for dependencies (Figure 10.2). Its superpower is not the pretty picture but the questions it forces: what depends on what, what runs in parallel, and — the one that sinks student plans — *where are the exams?* A Gantt that ignores the calendar reality of its team is decoration.

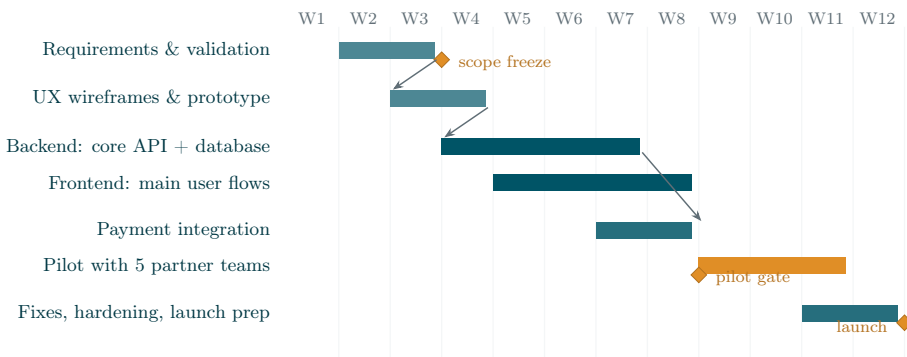


Figure 10.2: A semester Gantt for a software MVP: bars are tasks from the WBS, diamonds are milestones, arrows are dependencies. The critical path runs through the backend build, the pilot, and the hardening phase — a week lost there is a week lost everywhere.

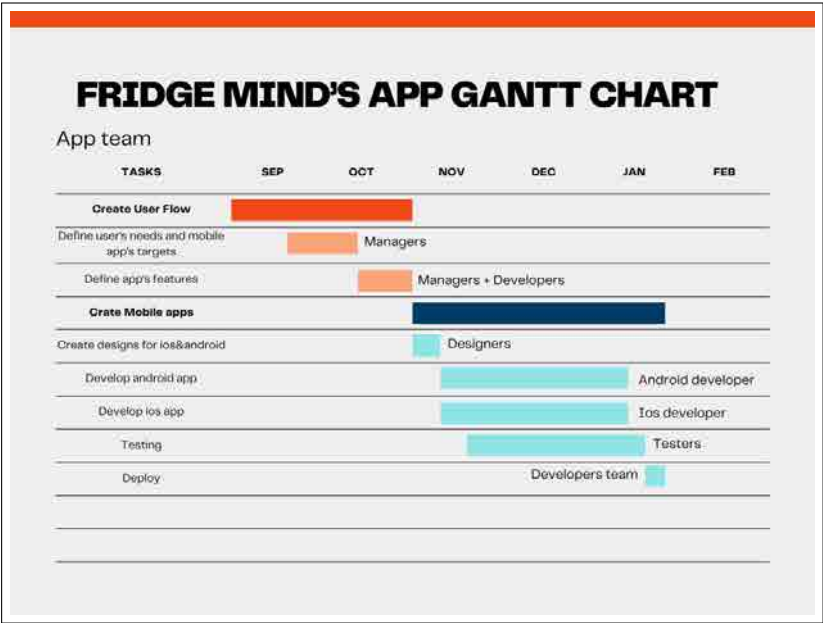


Figure 10.3: Student artifact: the *Fridge Mind* team’s own Gantt chart (reproduced with permission). Notice what a real student plan contains that a textbook diagram does not: named owners on every bar, a long pole (the device and app tracks running September to February), and the honest overlap between the mobile app and everything else.

10.4.3 Critical path: where delay is fatal

In any network of dependent tasks, one chain is the longest — the **critical path**. Delays on it move the end date one-for-one; delays off it may be absorbed by slack. You do not need the full CPM arithmetic to benefit: listing your tasks’ dependencies and asking “which chain, if everything slips a week, slips the launch?” already tells you where to put your best people and your daily attention [22]. In Figure 10.2 the critical path runs through the backend and the pilot — so the payment integration has slack, and the project lead should know why that matters.

10.4.4 Kanban: steer the week

Planning is monthly; steering is daily. The **Kanban board** (backlog / this week / doing / review / done) with two rules — every card has one owner,

and *doing* has a work-in-progress limit of one-to-two cards per person — turns the plan into visible flow. Tools are interchangeable (Trello, Asana, Jira, paper on a wall); the discipline is not [8]. The board’s health metric for a student team: *how old is the oldest card in “doing”?* Answer honestly in the weekly retro. Figure 10.4 shows the model; Figure 10.5 shows what it looks like in the wild.

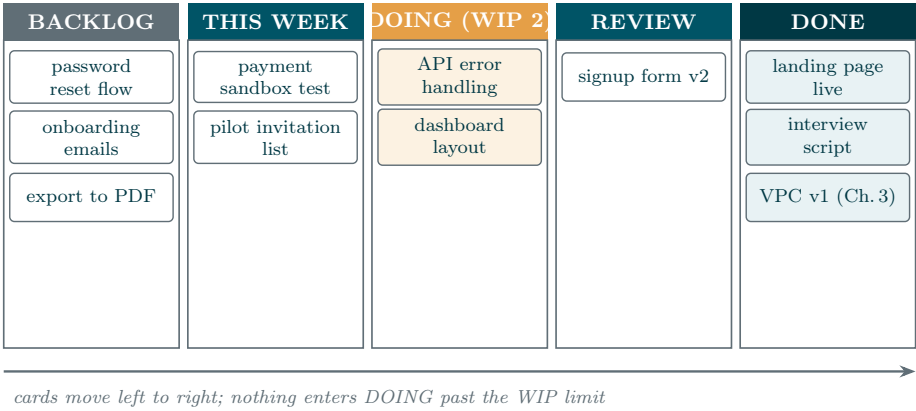


Figure 10.4: The Kanban board model: columns are stages of the weekly workflow, cards are tasks small enough to finish within the week, and the work-in-progress (WIP) limit on *doing* is what turns a to-do list into flow.

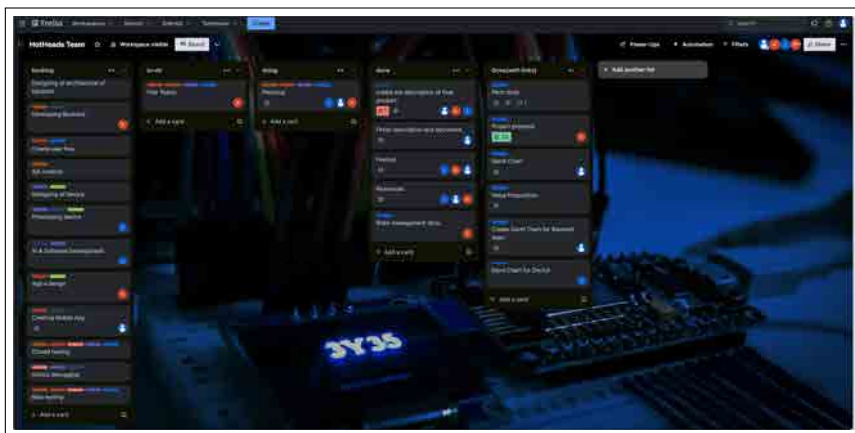


Figure 10.5: Student artifact: the *HotHeads* team’s Trello board (reproduced with permission). The columns mirror the lifecycle from Section 10.2 (planning → backlog → sprint → done), every card is small enough to finish within a week, and the “done” column is honest — it only counts work a teammate has reviewed.

10.5 Risk Management: Managing the Future Before It Manages You

A risk is an *uncertain event that would affect objectives* — note “would,” not “might inconvenience.” The process has four steps, and the deliverable is a living document, the **risk register** [22, 54]:

1. **Identify.** Work through the seven categories of the course risk template: *technology* (the TRL gaps from Chapter 6), *timing, milestones and results* (the schedule you just built), *employees and partners* (the bus factor from Chapter 8), *procurement* (anything you must buy or license to deliver), *budgetary* (runway, cost overruns), *market* (the persona does not pay; a competitor moves, Chapter 7), and *environmental and social* (regulation, sustainability expectations, community impact). For each category write two rows: the description of the risk and the mitigation actions — exactly the discipline of the Risk Management Table in the Appendix (Template 10a). Sources for candidates: your assumption map, your SWOT threats, your rings-lab shift scenarios — this practicum has been collecting your risks all along.

2. Prioritize. Score each risk for *probability* and *impact* (low/medium/high) and plot it on the risk matrix (Figure 10.6). The red zone gets mitigation plans; the green zone gets a line in the register and nothing more. Beware the two classic misreads: high-impact/low-probability risks that nobody owns (“a platform bans our category”) and high-probability/low-impact nuisances that eat meetings.

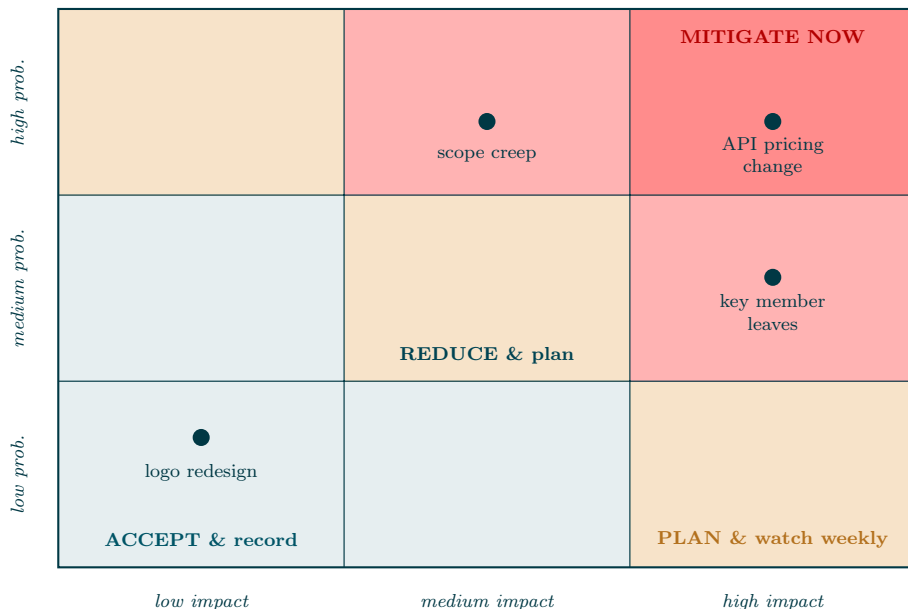


Figure 10.6: The risk matrix: probability against impact. Red-zone risks get named mitigation with owners and triggers; the register is reviewed in the weekly retro, not written once and framed.

3. Mitigate. Four strategies: *avoid* (change the plan so the risk cannot occur — drop the risky feature), *reduce* (prototypes, tests, spikes — your Chapter 6 experiments are risk-reduction machines), *transfer* (insurance, contracts, partnerships — Chapter 4’s open-innovation allies absorb risk for a share), *accept* (consciously, with a trigger: “if X happens by date Y, we do Z”). Every red-zone risk needs: strategy, owner, trigger, fallback. Without those four fields it is anxiety, not management.

4. Review. Risks decay and mutate; the register is a weekly five-minute agenda item. Add one practice from Chapter 5’s portfolio logic: the

pre-mortem — before the pilot, the team writes “it is three months later and we failed; here is why.” Research and practice agree this surfaces risks that polite brainstorming hides [4].

10.6 Time Management for Founders

For a student founder, time is scarcer than money and less renewable. Drucker’s discipline remains the foundation: know where your time actually goes (track it for a week — the gap between belief and data is always shocking), then consolidate it into blocks large enough to matter [10]. Tracking is painless with a free tool: in Clockify, for example, every work session is logged against a project, and the weekly dashboard then tells you the truth — how many hours the venture really got, and where the rest went (Figure 10.7). The tool supports the classic personal methods directly, so pick the one that fits your week: **Pomodoro** (a built-in timer for focused intervals followed by breaks), **time blocking** (a calendar view where your planned blocks sit next to the time you actually tracked, with Google/Outlook sync), **timesheets** (the one-minute weekly review of where the hours went), and **automatic tracking** with idle detection and reminders for those who always forget to press start. On top of that, four working rules:

- **Important vs. urgent.** Sort the week with the Eisenhower logic; the quadrant “important, not urgent” (experiments, the charter, the risk register) is where ventures are built, and it has no deadlines — so you must invent them.
- **Maker time is sacred.** Development and design need uninterrupted blocks of two-plus hours; a day sliced into meetings produces status reports, not software. Protect two meeting-free blocks per week per builder.
- **Deadlines are a tool, not an insult.** External dates (demo day, pilot start, a partner’s launch) convert infinite polishing into shipping. Choose them deliberately and *write them on the Gantt*.
- **Energy is the real currency.** Schedule demanding work into your personal peak hours and admin into the valleys; burnout prevention

from Chapter 8 applies here at the daily scale.

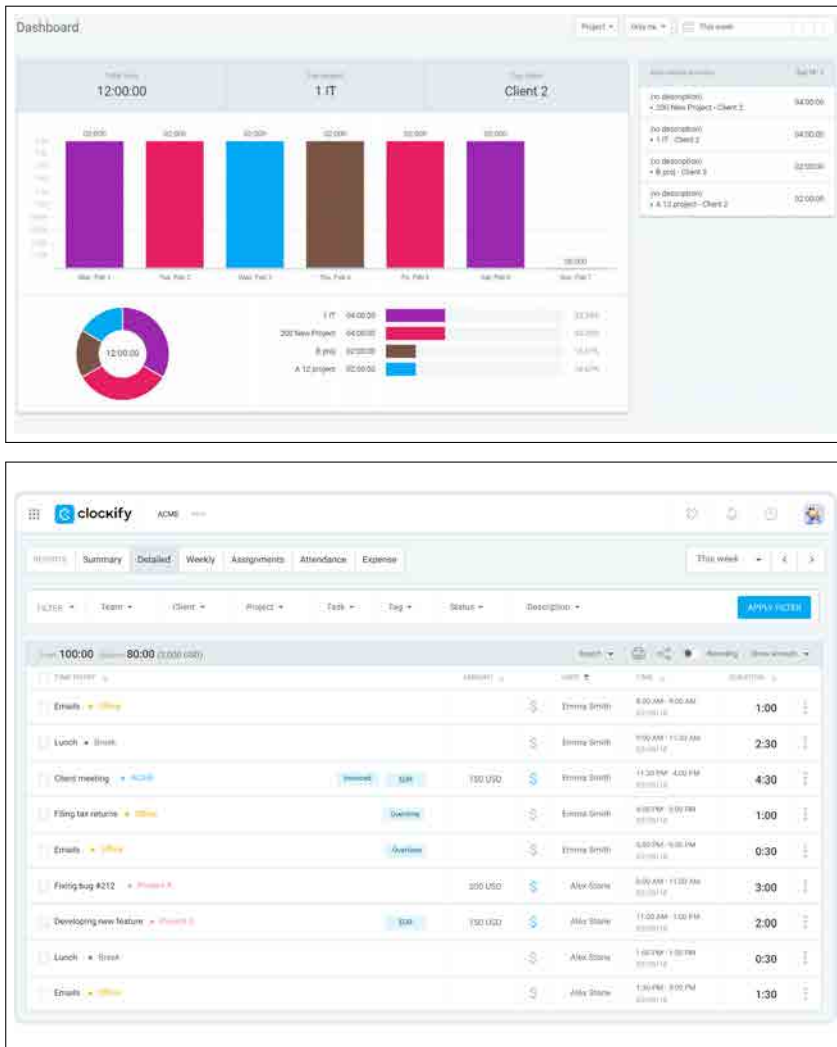


Figure 10.7: Time tracking in practice (Clockify; screenshots: clockify.me, used for illustration). *Top*: the weekly dashboard — total hours, the project that absorbed them, and the day-by-day shape of the week. *Bottom*: the detailed report — every entry logged against a project and a person. For a student team this answers two questions no memory can: did the venture get its planned hours this week, and which task is silently eating the budget?

Worked example

Three student teams, three planning personalities. Fridge Mind planned in true Gantt style (their actual chart is Figure 10.3) — parallel tracks for backend, device, and mobile app, with milestone-like stages (*MVP device*, *stable version*, *production*); its strength was sequencing, its unasked question was integration: which milestone requires *all three* tracks together, and who owns it? **HotHeads** steered with a Trello board (Figure 10.5) — excellent weekly flow, but without a Gantt above it the team could not see that two cards were secretly critical-path. **Creative Wave** exported their Asana timeline for a client project and discovered the plan’s hidden assumption: every task depended on one designer. Three tools, three lessons: *Gantt sees time*, *Kanban sees flow*, and *neither sees people* — the risk register is where the people-risks live. Discussion: which of the three failure modes is your team’s most likely — wrong sequence, invisible flow, or invisible dependency on a person?

Guided steps

- Step 1. Decompose.** Build the WBS of your MVP down to one-to-three-day tasks with owners.
- Step 2. Schedule.** Lay the WBS on a Gantt; mark dependencies, milestones, and the critical path; overlay the semester calendar.
- Step 3. Steer.** Set up the Kanban board with WIP limits and the weekly retro question (oldest card in “doing”).
- Step 4. Register.** Fill the risk register from your assumption map, SWOT threats, and rings-lab scenarios; score and plot on the matrix.
- Step 5. Mitigate.** For every red-zone risk: strategy, owner, trigger, fallback. Schedule the pre-mortem before your pilot.

Tasks & exercises

Task 10.1 (team). WBS of the MVP to leaf-task granularity, with owners. Deliverable: WBS tree.

Task 10.2 (team). Gantt chart with milestones, dependencies, and the critical path marked and defended in three sentences. Deliverable: chart.

Task 10.3 (team). Kanban board populated with real tasks, WIP limits, and a screenshot after one week of use. Deliverable: board + screenshot.

Task 10.4 (team). Risk register: at least eight risks across the five families, scored, plotted, with mitigation for the red zone. Deliverable: register + matrix.

Task 10.5 (individual). One-week time audit vs. plan: where did your hours actually go, and what one structural change follows? Deliverable: one page.

Template

Template 10a — Risk Management Table (Appendix): the course format — seven risk categories (technology; timing, milestones & results; employees & partners; procurement; budgetary; market; environmental & social), each with a *description* and an *actions* row, rated HIGH / MEDIUM / LOW, extended with owner, trigger, and fallback columns.

Template 10b — WBS/Gantt Skeleton (Appendix): WBS tree grid and a twelve-week Gantt frame with milestone row.

Template 10c — Kanban Board (Appendix): the plain board from Figure 10.4 as a fill-in sheet — Backlog / To do / Doing / Review / Done columns with WIP limits and a blocked-cards field.

Systems Thinking Lab — The Concentric Rings

Fill. R6 now carries the plan: WBS, Gantt, board, register. Connect it visibly to the rings it serves — every bar should trace to a ring.

Then shift it. The pilot milestone slips four weeks (partner's

schedule changed). Follow the delay through the system: budget burn extends (R7 — how many weeks of runway remain?), the marketing moment you announced moves (R2/R8 — who must be told?), team morale dips as exams arrive (R5), and the follow-on market window may close (R8). For each affected ring, write the *absorber*: what in that ring cushions the shock (slack, savings, a waiting list, an honest conversation)?

Record. Which ring in your venture has the *least* slack? That is where a four-week slip becomes a crisis — and it deserves a line in the risk register with a mitigation, not a hope.

AI angle

AI is quietly becoming the project office of small teams: it drafts the WBS from a product description, proposes estimates from similar tasks, turns meeting notes into cards, and flags plans whose dependencies do not add up. Estimation is the real test case: AI estimates inherit the same optimism bias as humans, plus confident formatting — so treat them as a *second opinion* to be argued with, and calibrate against your own velocity after two sprints. Risk identification benefits most: ask the model to enumerate failure modes for your specific plan, then score them yourself — generation is cheap, judgment is yours. One boundary to hold: the retro. A team that lets AI summarize “how the week went” outsources exactly the conversation that keeps the team a team.

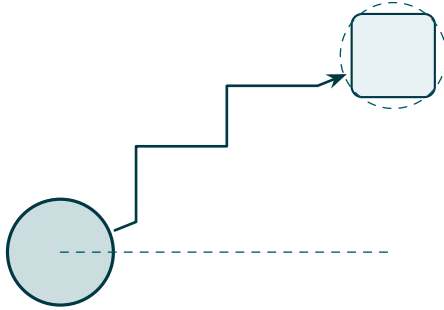
Chapter references

Luckey & Phillips, *Software Project Management for Dummies* (lifecycle, WBS, CPM, risk) [22]; Siciliano, *True Business Success* (planning discipline) [54]; Drucker, *Management* (time and effectiveness) [10]; Ries (Agile rationale) [9]; Cheek (board discipline, owners, deadlines) [8]; Sastry & Penn (pre-mortem practice) [4]; course lectures on project management, SDLC, risk, and time management; Forsgren, Humble & Kim, *Accelerate* (delivery performance as a measurable capability) [55]; Prasetyo et al.

on AI in open-innovation project management [56]. Dakić on software compliance with CI/CD, dynamic microservices and containers [53].

PART IV

From Plan to Growth



In this part

A design becomes a venture only when someone funds it, buys it, and repeats the purchase. This final part turns your system into numbers and momentum: the budget, unit economics and the business plan that assemble everything into one document; the fundraising ask and the pitch that carries it; the specifics of building with AI at the core; and the long game — growth, internationalization, and the exit that returns value to everyone who believed early. The deliverables of this part are the capstone itself: the business plan, the pitch deck, and your completed rings map.

Chapters in this part: 11 Financials and the Business Plan · 12 Fundraising and Pitching · 13 Building Ventures in the Age of AI · 14 Growth, Internationalization and Exit

Financials and the Business Plan

Learning outcomes

By the end of this chapter you will be able to:

- build a project budget from your WBS: fixed costs, variable costs, and a contingency reserve;
- read the three financial statements and explain why cash flow, not profit, kills startups;
- compute COGS and gross margin for a software product, and defend them to a skeptic;
- calculate LTV, CAC and the LTV:CAC ratio, and judge whether the unit economics work;
- find your break-even point and your runway, and turn both into decisions;
- assemble the full business plan — the capstone document of this practicum.

11.1 Money Is a Design Variable

Everything you have built so far — the persona, the canvas, the MVP, the team charter, the Gantt — has a price tag and a revenue logic attached. This chapter makes both explicit. The financial ring (R7) is where the whole system is stress-tested: if the numbers cannot work on paper, with generous assumptions and unlimited time to fix them, they will not work in the market.

Two warnings before we start. First, **financial projections for a**

pre-revenue startup are not forecasts — they are hypotheses in spreadsheet form, and their main value is forcing you to make assumptions explicit enough to test. Second, **you do not need to become an accountant**. You need a working vocabulary — budget, cash flow, COGS, LTV, CAC, break-even, runway — and the discipline to connect every number to a decision [54]. That is financial management for non-financial managers, and it is a core skill for any technical founder [57]. Explainable AI is also being applied to payroll analytics, giving Serbian SMEs a transparent way to detect anomalies in cost data [58].

11.2 The Project Budget: From WBS to Money

A **project budget** is a financial plan that estimates the resources your project requires — expected costs on one side, expected revenues (or funding) on the other [57]. You already did the hard part in Chapter 10: the budget is the WBS with prices on it.

Three cost categories, three different management logics [57]:

- **Fixed costs** stay constant regardless of progress: salaries, rent, the annual domain fee. They are easy to plan and dangerous to accumulate — every fixed cost raises your break-even point for as long as it exists.
- **Variable costs** scale with activity: cloud and API usage, payment-gateway fees, contractors hired per task. In software they often hide inside “free tiers” that expire exactly when your pilot succeeds.
- **Contingency reserve** is the honest line: buffer funds for the expenses you *cannot* list — the API that changes pricing, the laptop that dies, the feature that takes three weeks instead of one. A student rule of thumb: 10–15% of the total. A plan without contingency is not optimistic; it is incomplete.

The steps mirror the planning you already know: scope and objectives (Chapter 6) → deliverables and tasks (the WBS) → cost estimate per task → costs placed on the timeline (the Gantt) → monitoring and adjustment as the project runs [57]. The timeline step matters more than students expect: the same total, distributed differently across months, can mean a

solvent venture or a dead one. That is why the budget and the Gantt are two views of one plan.

11.3 Three Statements, One Survival Question

Accounting has three standard reports; a founder needs them for three different questions [54, 57]:

- The **income statement** (P&L) shows revenues, expenses and profit over a period. Question answered: *is the model profitable — eventually?*
- The **balance sheet** is a snapshot of assets, liabilities and equity at a moment. Question answered: *what do we own and owe, right now?*
- The **cash flow statement** tracks cash in and out — operating, investing, financing. Question answered: *can we pay our obligations next month?*

For a startup the third question dominates. Profit is an opinion shaped by accounting rules; cash is a fact. A venture can be “profitable” on paper and dead by Friday, because customers pay in 60 days and salaries do not wait. This is why investors and experienced founders track two numbers weekly: **burn rate** (net cash out per month) and **runway** (cash on hand $\div$ burn rate = months of life left). Every decision in Part IV of this practicum — hiring, spending on ads, buying equipment — should be checked against runway before it is checked against anything else.

11.4 COGS and Gross Margin in Software

Cost of Goods Sold (COGS) is the direct cost of producing what you sell. In a software product this includes developer salaries allocated to the product, hosting and server costs, third-party licenses and APIs, maintenance and support, and infrastructure [57]. Revenue minus COGS is **gross profit**; divided by revenue it is the **gross margin** — the single most important structural number in your model, because it determines how much of each new euro of revenue is available to pay for everything else (marketing, salaries, growth).

Software is famous for high gross margins — the marginal copy costs almost nothing — but two traps await student ventures. First, **AI features have real marginal costs**: every user request to a large-model API is a metered expense, so an “AI inside” product can behave more like a consultancy than like classic software. Price accordingly, and watch the cost per active user per month, not just the total API bill (Chapter 13 returns to this). Second, **support is COGS too**: if every new customer needs two hours of hand-holding, that is a variable cost of sale, not free goodwill.

11.5 Unit Economics: LTV, CAC, and the 3:1 Rule

Zoom in from the whole company to one customer. Two numbers decide whether growth helps you or kills you [57]:

Customer Lifetime Value (LTV) is the total revenue you expect from one customer over the whole relationship:

$$\text{LTV} = \text{average revenue per customer per period} \times \text{customer lifetime in periods}.$$

A language-learning app charging \$10 per month, with an average subscriber staying 24 months, has $\text{LTV} = 10 \times 24 = \240 . (For precision you would discount and use gross margin instead of revenue — but the simple version already separates viable models from hopeless ones.)

Customer Acquisition Cost (CAC) is everything you spend to win one customer — marketing, sales, onboarding:

$$\text{CAC} = \frac{\text{total sales and marketing costs}}{\text{new customers acquired}}.$$

Spending \$5,000 in a month to acquire 100 subscribers gives $\text{CAC} = \$50$.

The verdict comes from the ratio: $\text{LTV} : \text{CAC} = 240 : 50 = 4.8 : 1$ — healthy, because the benchmark is roughly **3:1**: each euro of acquisition spend should return about three over the customer’s lifetime [57]. Below 3:1 growth burns money; far above it (say 8:1) you are probably under-

investing in growth. A second, stricter test is the **CAC payback period**: how many months of gross margin from the new customer until the CAC is recovered. Investors like payback under 12 months; student ventures with semester horizons should aim for “within the pilot.”

Notice what this pair of numbers does to your strategy conversations. “Should we spend on Instagram ads?” becomes arithmetic: expected CAC from that channel versus the LTV of the persona it reaches. Chapter 2’s TAM–SAM–SOM told you the market is big enough; LTV–CAC tells you whether reaching it is worth the price.

11.6 Break-Even and Runway: The Two Dates That Matter

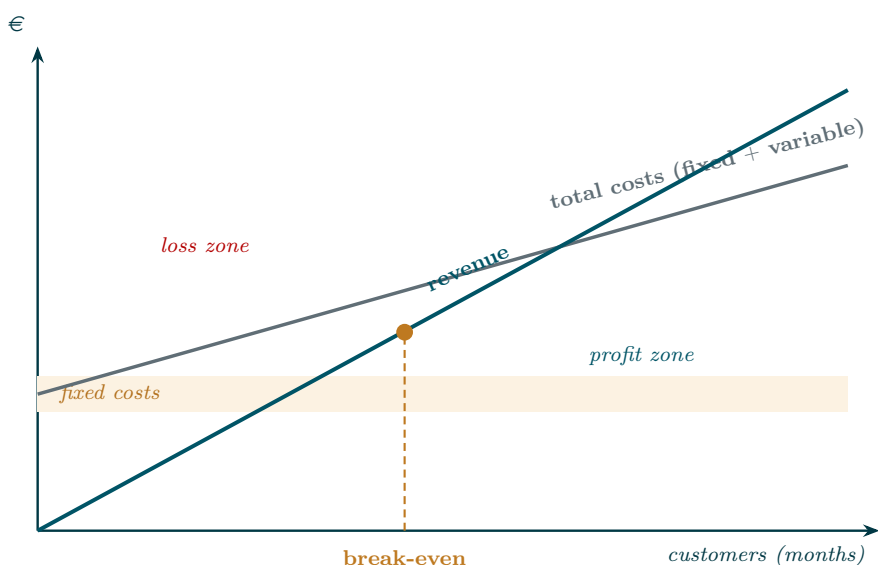


Figure 11.1: Break-even logic: revenue starts at zero, total costs start at the fixed-cost level and grow with each customer; the crossing point is the number of customers (or months) at which the venture stops losing money. Everything in Chapters 2–7 is ultimately an attempt to move this point to the left.

The **break-even point** is where cumulative revenue catches up with total costs (Figure 11.1). In units: break-even customers = fixed costs $\div$ gross margin per customer. The number is less important than its direction: every decision you make either moves break-even closer (lower fixed costs, higher price, higher margin) or pushes it away. Put the date on your Gantt next to the milestones from Chapter 10: *pilot gate*, *launch*, *break-even*. Now the plan has a financial spine.

Runway converts the same logic into time. If your budget shows €18,000 of costs before first revenue and you have €12,000 (savings, a grant, a university award), you do not have a funding gap of €6,000 — you have a *time* gap: the venture must either reach break-even sooner, spend slower, or raise the difference. Chapter 12 is about the third option; this chapter's job is to make the first two honest.

11.7 ROI: The Investor's Question, Asked of Yourself

Return on investment measures the efficiency of money spent: benefits relative to costs [57]. In software and data projects the tangible benefits (new revenue, reduced costs) are only half the story — the other half is intangible: better decisions from better data, a brand that attracts talent, a dataset that compounds (your Core candidate from Chapter 7). The discipline is to name the benefit class explicitly instead of waving at “value”: is this expense buying revenue, buying speed, buying learning, or buying risk reduction? A prototype experiment buys learning; a marketing campaign buys revenue; a contingency reserve buys resilience. A budget where every line names what it buys is a budget you can defend — to an investor, and to yourself at 2am.

The same lens applies to the venture as a whole. An investor's ROI question — *if I give you X, what comes back and when?* — is exactly the question your financial projections must answer. Which brings us to the document that assembles all of it.

11.8 The Business Plan: Assembling the System

The **business plan** is the capstone deliverable of this practicum — the document where every ring of your system map appears, translated into a coherent story with numbers. Its sections should by now feel like a table of contents of your own work [7, 8, 57]:

1. **Executive summary** — one page: problem, solution, market, model, team, the ask. Written last, read first.
2. **Problem and opportunity** — your Chapter 2 evidence: PMR findings, market size (TAM–SAM–SOM), why now.
3. **Solution and product** — Chapters 3 and 6: value proposition, VPC fit, MVP scope, TRL status.
4. **Business model** — Chapter 4: the nine blocks, revenue streams, pricing logic.
5. **Market and competition** — Chapter 7: positioning chart, competitors, your Core and moat.
6. **Go-to-market** — beachhead, first 10 customers, channels, and the CAC assumptions per channel.
7. **Team and organization** — Chapter 8: founders, roles, gaps you will hire for; Chapter 9: legal form and IP status.
8. **Execution plan** — Chapter 10: WBS, Gantt, milestones, and the risk register summary.
9. **Financials** — this chapter: budget, projections for 3 years, unit economics, break-even, funding need and use of funds.
10. **Appendices** — the artifacts: canvases, persona, experiments log, letters of intent if you have them.

Three rules make the difference between a plan that gets read and one that gets skimmed. **Numbers tell a story**: the projections must be derivable from the assumptions on the earlier pages — if page 2 says 5,000 students in the beachhead and page 9 assumes 4,000 paying users in month 6, the reader will catch it. **Show the assumptions**, not just the results: a table of ten key assumptions beats ten pages of charts. And **write the executive summary last but make it stand alone** —

many readers will read only that page plus the financials.

Do not confuse the plan with the **investment deck**: the deck is the plan's theatrical sibling — ten slides, five minutes, designed to earn a meeting, not to document a system. Chapter 12 builds it; your course template lists its components: title, problem, solution, market opportunity, business model, go-to-market, competitive analysis, traction, financial projections, team, the investment ask, and a closing slide [57].

Worked example

Fridge Mind's first honest numbers. The team priced the device-plus-app bundle at €89 with a €4.90 monthly subscription for the AI recipe engine. The budget from their WBS came to €14,200 before launch: developer time valued at junior-market rates (the biggest fixed cost), cloud hosting, the model API, and a 12% contingency line they initially wanted to skip — until the payment-integration task slipped two weeks and ate half of it, exactly as Chapter 10's risk register had predicted. Unit economics: COGS per active user (hosting + API + support) was €1.60/month, so subscription gross margin was €3.30. Estimated CAC from campus channels was €9; average subscriber lifetime 14 months gave $LTV \approx €69$ (subscription margin + device margin), so $LTV : CAC \approx 7.7 : 1$ — suspiciously high, which told them they were under-investing in growth and that the CAC estimate came from friendly early adopters, not the real market. Break-even: 270 subscribers covered the remaining fixed costs; the Gantt said month 7. Discussion: which of these numbers would you attack first in a due-diligence meeting — and which one did the team itself mark as “assumption, not data”?

Guided steps

- Step 1. Budget.** Take your WBS from Chapter 10 and price every task: fixed, variable, contingency (10–15%). Deliverable: one budget table on the Gantt timeline.
- Step 2. Unit economics.** Compute COGS per user, gross margin, LTV, CAC and the ratio. Mark every input as *data* or

assumption.

Step 3. Break-even. Compute the break-even customers and the break-even month; put both on the Gantt.

Step 4. Runway. $\text{Cash available} \div \text{monthly burn} = \text{months of life}$. Write the date the venture dies without revenue or funding — and what would move it six months out.

Step 5. Assemble. Draft the business plan skeleton (ten sections above); for each, link the chapter artifact that feeds it.

Tasks & exercises

Task 11.1 (individual). Project budget for your venture: fixed / variable / contingency, on a monthly timeline. Deliverable: one page + the three largest cost drivers named.

Task 11.2 (individual). Unit-economics sheet: COGS per user, gross margin, LTV, CAC, ratio, payback. Every input tagged *data* or *assumption*. Deliverable: one page.

Task 11.3 (team). Break-even and runway: the two dates, placed on your Gantt; one paragraph on what moves each date in your favor. Deliverable: updated Gantt + paragraph.

Task 11.4 (team, capstone milestone). Business plan v1: all ten sections drafted, with the executive summary written last. Deliverable: 10–15 pages. This is the document the capstone jury reads.

AI angle

AI angle. LLMs are genuinely useful sparring partners for financial modeling: they will build you a projection spreadsheet structure, stress-test assumptions (“what if churn doubles?”), and explain any accounting term at 2am. Two traps are specific enough to name. First, **confident arithmetic errors**: always recompute every number an AI gives you — a language model does not know it made a mistake, and a wrong LTV formula copied into your plan is your credibility, not its. Second, **benchmarks from nowhere**: “typical CAC for SaaS

is \$200” is a hallucination genre; use AI to structure the question, then find the number in citable data. The deeper shift for your model: AI API costs are a *variable COGS that scales with success* — simulate your unit economics at 10× and 100× users, not just at launch volume.

Systems Thinking Lab — The Concentric Rings

Systems Thinking Lab — the financials ring (R7). *Fill:* on your rings map, write the four numbers that define R7 for your venture: monthly burn, gross margin per customer, LTV:CAC, break-even month. *Shift:* your main distribution channel doubles its prices overnight (the app store raises fees; the ad platform reprices). Walk the rings: CAC rises → ratio falls below 3:1 → do you raise prices (R3 business model shifts; persona tolerance from R2), cut a feature to lower COGS (R4), slow hiring (R5), or renegotiate the milestone plan (R6)? Write one sentence per affected ring. *Reflect:* which ring absorbed the shock best — and what does that tell you about where your venture is fragile?

Template

Template 11a — Project Budget Sheet (Appendix): tasks × months grid, with fixed/variable/contingency rows and a running cash line.

Template 11b — Unit Economics Sheet (Appendix): COGS per user, margin, LTV, CAC, ratio, payback — with the data/assumption tag on every input.

Template 13 — Business Plan (Appendix): the ten-section skeleton with page limits and the artifact checklist per section — the capstone format.

Worked example

ZenTutor — the first six months of the budget, filled. Months 1–6 of the EUR 60 K ask from their business plan (Appendix E); the remaining ≈ EUR 29.7 K covers months 7–14 at a higher marketing

cadence. Founders are unpaid — their time is the hidden line, and it is stated, not hidden.

Budget line (EUR)	M1	M2	M3	M4	M5	M6	Total
Development, tools & cloud (fixed)	1700	1700	1700	1700	1700	1700	10 200
Legal & admin (fixed)	430	430	430	430	430	430	2 580
Tutor vetting & support (variable, EUR 8/tutor)	160	320	640	960	1280	1600	4 960
Marketing (variable, unlocked at pilot gate)	0	0	800	2000	3000	4000	9 800
Contingency (≈10%)	230	250	360	510	640	770	2 760
Monthly total	2 520	2 700	3 930	5 600	7 050	8 500	30 300

Two habits worth copying: marketing is *zero until the pilot gate passes* (the budget obeys the evidence gates, Chapter 5), and the variable lines carry their driver in the row label — EUR 8 per onboarded tutor is an assumption anyone can challenge.

Chapter references

Course lectures: *Financial Management for Non-Financial Managers* (budgeting, statements, COGS, LTV/CAC, ROI, investment deck) [57]. Foundations: Siciliano [54] on finance for non-financial managers; Aulet [7] on unit economics in disciplined entrepreneurship; Cheek [8] on the tactics of pitching numbers; Osterwalder & Pigneur [6] on revenue and cost blocks; Harris, *The Business Value of Software*, on how software creates and measures economic value [59]; Avlijaš & Avlijaš, *Preduzetništvo*, chapters on financial planning [15]; Hisrich, Peters & Shepherd on the financial section of the business plan [23]. Milojković, Kljajić & Dakić on explainable AI for payroll analytics [58].

Fundraising and Pitching

Learning outcomes

By the end of this chapter you will be able to:

- choose the right funding source for your stage — bootstrapping, grants, angels, VC — and explain what each expects in return;
- reason about equity, valuation and dilution with a simple cap table;
- formulate the investment ask and defend the use of funds from Chapter 11's budget;
- build a twelve-slide pitch deck and a 60-second elevator pitch;
- deliver the pitch as a performance and survive the Q&A;
- anticipate what due diligence will check — before it checks you.

12.1 Money Has a Voice

Chapter 11 ended with a funding gap converted into time. This chapter is about closing it — and about the uncomfortable truth that **external money is never just money**. Every funding source arrives with expectations, a clock, and a say in your decisions. Taking investment is selling a part of your venture's future; the craft is to sell as little as possible, at the best possible terms, to people whose expectations match your plan. That is why this chapter sits on the boundary of two rings: the financials ring (R7) that prices the deal, and the outer ecosystem ring (R8) where the funders live [7, 8].

12.2 The Funding Ladder

Sources form a ladder ordered by how much proof they demand [54, 60]:

- **Bootstrapping** — founders' savings, revenue, unpaid founder time. Cost: speed. Benefit: total control and a cap table that stays clean. Most student ventures live here longer than they expect, and that is not a failure — bootstrapped traction is the strongest possible argument in every later conversation.
- **Grants and awards** — university competitions, national innovation-fund early-stage programs, EU instruments. Non-dilutive: nobody takes equity. Cost: paperwork, reporting, and timelines you do not control. For a student team this is usually the best first external money: it validates the project publicly and leaves 100% of the company with the founders.
- **Business angels** — individuals investing their own money at the earliest stages. They decide fast, invest partly in *you*, and often bring mentoring and contacts. They expect equity and a believable path to an exit (Chapter 14).
- **Venture capital** — funds investing other people's money against a mandate: they must return the fund, so they need ventures that can plausibly grow 10× or more. VC money is rocket fuel with a steering wheel attached: board seats, growth targets, and a timeline. If your ambition is a profitable niche product, VC is the wrong fuel — and that is a legitimate choice, not a lack of ambition.
- **Loans** — rarely available or wise for pre-revenue teams; debt demands repayment regardless of traction. Mention for completeness; usually the wrong tool before product–market fit.

The matching principle: *raise from the rung that matches your evidence*. A validated VPC and a pilot (Chapters 3 and 6) unlock grants; repeatable sales with healthy unit economics (Chapter 11) unlock angels; a proven growth machine unlocks VC. Raising above your evidence wastes months and burns reputation; raising below it starves a working machine.

12.3 Equity, Valuation, Dilution — Founder Arithmetic

When an investor gives you money for equity, three numbers define the deal. **Pre-money valuation** is what the company is deemed worth before the investment; **post-money** is pre-money plus the new cash; the investor's share is $\text{investment} \div \text{post-money}$. Example: €400,000 pre-money, €100,000 invested $\rightarrow$ post-money €500,000, investor gets 20%, founders are **diluted** from 100% to 80%.

Dilution is not theft — 80% of a funded venture can be worth far more than 100% of a starving one. But it compounds: each round sells another slice, so the discipline from Chapter 8 (equity splits, vesting) extends outward: keep the cap table simple, write everything down (handshake equity is a lawsuit in waiting), and never give away the slice you will need to stay motivated in year three. A useful heuristic for a first round: selling 10–20% keeps everyone aligned; selling 40% in the first round usually means the valuation conversation went badly.

How is a pre-revenue company valued at all? Honestly: by negotiation anchored in comparables (what similar-stage startups raised at), the strength of the team, traction, and the size of the market story. There is no formula — which is precisely why Chapter 11's numbers matter: they are your arguments.

12.4 The Ask and the Use of Funds

Every pitch ends with **the ask**: a specific amount, for a specific runway, to reach specific milestones. “We are raising €150,000 for 18 months of runway to reach 1,000 paying users and break-even on the campus channel” is an ask; “we are looking for investment” is a wish. The amount must come from Chapter 11's budget — the use of funds is literally your budget's lines: 60% product development, 25% go-to-market, 15% operations [57]. Investors do not fund your costs; they fund the *milestones* those costs buy — so phrase the use of funds as milestone purchases, not expense categories, whenever you speak.

12.5 The Pitch Deck: Twelve Slides, One Story

The **pitch deck** is the business plan’s theatrical sibling: a dozen slides, five to ten minutes, designed to earn the next meeting — not to close the deal on the spot. The components, as your course materials define them [57]:

1. **Title** — name, one-liner, contact; clean and on-brand.
2. **Problem** — the pain, in the customer’s words (your PMR verbatims from Chapter 2).
3. **Solution** — the product as the answer; one screenshot beats three paragraphs.
4. **Market opportunity** — TAM–SAM–SOM and the “why now.”
5. **Business model** — how you make money; pricing; the revenue streams from Chapter 4.
6. **Go-to-market** — beachhead, channels, first 10 customers.
7. **Competition** — the positioning chart from Chapter 7; your Core as the moat.
8. **Traction** — pilots, LOIs, users, growth rate; anything real.
9. **Financial projections** — three years; unit economics; break-even date.
10. **Team** — why *you* are the people to do this; gaps you will hire.
11. **The ask** — amount, runway, milestones bought.
12. **Closing** — call to action and contact; end on the vision, not on a “thank you” slide.

Underneath the slide list sits a **narrative arc**, and the classic student template names it well: problem → solution → *utopia*. First make the audience feel the problem, then show the mechanism that removes it, then paint the world where it is gone — and only then prove you can get there (market, model, traction, team). Slides 2–3 sell the wound and the cure; slides 4–10 sell the evidence; slide 12 returns to the utopia. Facts convince, but the arc is what they remember.

Three craft rules. **One message per slide** — a slide is a billboard, not

a document; anything the audience must read, they are reading instead of listening to you. **Show, don't tell:** the demo screenshot, the customer quote, the real graph of pilot users — artifacts beat adjectives. **Design is a signal:** a deck that respects typography tells the investor you will respect their money. You do not need to be a designer; you need to be consistent [8].

12.6 The Elevator Pitch: Sixty Seconds of Truth

Before the deck comes the sentence. The **elevator pitch** answers, in under a minute and in plain language, the questions your course format specifies [57]: *What is the company? What problem does it solve? What is the solution? Who are the users? What are the advantages? What is the added value — what differentiates you from the competition?*

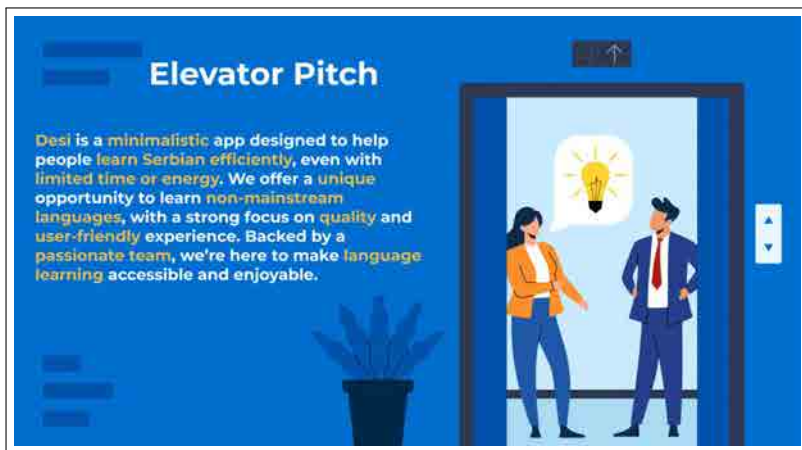


Figure 12.1: Student artifact: the *Desi* team’s elevator-pitch slide (reproduced with permission). Read it against the checklist: company and solution in the first sentence, users explicit (“people learning Serbian”), differentiation claimed (“non-mainstream languages”). What would you tighten? The added value is asserted but not proven — one number (“learners finish 3× more lessons than with textbook apps”) would do more than three adjectives.

The test of an elevator pitch is not whether it sounds impressive but whether a listener can repeat it to someone else. If your grandmother

cannot relay what you do after hearing it twice, it is not finished. Write it, say it aloud ten times, then cut every word that does not carry information — the version that survives is the one you will use at every demo day, every chance meeting, and the first thirty seconds of every investor pitch (Figure 12.1).

12.7 Pitching Is a Performance

Content is half the pitch; the other half is delivery [8]. Working rules:

- **Rehearse aloud, standing, with the clicker in hand** — at least five full runs, one of them recorded. Watching your own recording is painful and worth three rehearsals.
- **Open with the problem, not your name.** The first sentence should make the room lean forward: a customer story, a shocking number, a demo. Credentials can wait until slide 10.
- **One voice, not four.** In a team pitch, one founder leads; handovers are choreographed, not improvised. The person not presenting watches the room, not the screen.
- **Time is respect.** If the slot is five minutes, finish at 4:45. Running over signals that your milestones will also run over.
- **The Q&A is the real pitch.** Investors use questions to test how you think, not what you know. Answer directly, admit unknowns (“we don’t know yet; here is how we’ll find out” is a strength, not a weakness), and never bluff — rooms full of experts detect bluffing instantly, and one detected bluff discredits every true thing you said.

12.8 Due Diligence: What They Will Check

If the pitch earns a second meeting, the tone changes from theater to audit. Expect verification of: the team (references, the cap table, IP ownership — Chapter 9’s traps resurface here: who signed what, who owns the code), the numbers (do the projections derive from the assumptions? is the LTV:CAC built on data or hope?), the market (they will call customers — including, possibly, the ones you listed), and the legal shell (registration,

contracts, open-source licenses). The preparation is boring and decisive: a shared folder with the business plan, the budget, the artifacts from your appendix, and the contracts — everything a skeptic needs, organized so they find it in minutes. Teams that prepare this folder discover its first benefit themselves: it shows where the venture is still made of promises instead of documents [8, 54].

Worked example

Creative Wave’s first angel conversation — annotated. The team opened with a client story (a festival organizer who lost sponsors because reporting took three weeks) — problem first, name later. Their deck ran eleven slides; the strongest was traction: two signed pilot clients, one paid invoice. The ask was €60,000 for 12 months to reach ten recurring clients. The angel’s hardest question was not about the market but about the team: “What happens if your lead developer graduates and leaves?” Their answer — vesting from Chapter 8, documented code, a named successor in the partner network — turned the risk question into evidence of management maturity. They did not get a check that day; they got a second meeting and a list of five things to verify. That is what winning a first pitch looks like. Discussion: which of the five verification items would your team struggle to produce tomorrow?

Guided steps

- Step 1. Elevator pitch.** Draft the 60-second version answering the six course questions; test it on someone outside the field and ask them to repeat it back.
- Step 2. Deck skeleton.** Twelve slides, one message each; every claim paired with the artifact or number that proves it.
- Step 3. The ask.** Amount, runway, milestones — derived line by line from the Chapter 11 budget.
- Step 4. Rehearse.** Five runs, one recorded; Q&A simulation with the ten hardest questions written down in advance.
- Step 5. Diligence folder.** Assemble the documents a skeptic would

ask for; note every gap as a task.

Tasks & exercises

Task 12.1 (individual). Written elevator pitch (max 120 words) + recorded delivery. Deliverable: text + one-minute audio/video.

Task 12.2 (team). Funding-source map: which rung of the ladder matches your evidence today, and what proof would unlock the next rung? Deliverable: half a page.

Task 12.3 (team). The ask: amount, runway, milestones, use of funds — every number traceable to the budget. Deliverable: one slide.

Task 12.4 (team, capstone milestone). Pitch deck v1 (twelve slides) + five-minute recorded pitch. Deliverable: deck (PDF) + recording. This is the second document the capstone jury sees.

AI angle

AI angle. Generative tools can produce a slick deck in minutes — and investors can tell. The tell is genericness: an AI that has never met your customers writes “revolutionary platform” where you should have a verbatim quote. Use AI where it is genuinely strong: *structure* (slide outlines, speaker notes), *adversarial rehearsal* (“you are a skeptical angel; attack my unit economics” is an excellent prompt), *design drafts* you then finish by hand, and *translation* for international decks. Never use it for: market numbers you cannot cite, traction claims (that is not optimism, it is fraud), or customer quotes you did not collect. The pitch’s currency is credibility; AI should raise the quality of what is true, never the plausibility of what is not.

Systems Thinking Lab — The Concentric Rings

Systems Thinking Lab — financials meets the ecosystem (R7 + R8). *Fill:* on your rings map, add the funding line to R7: source, amount, runway it buys, milestones it unlocks. On R8, mark the investors/grantors as ecosystem actors with their own goals.

Shift: your lead investor withdraws three weeks before signing. Walk the rings: runway (R7) shrinks → the milestone plan (R6) must re-sequence → does the team (R5) take salary cuts or lose a member? → does the product scope (R4) shrink to protect the launch? Write one sentence per ring. *Reflect:* what would make your venture *less* dependent on any single actor in the outer ring — and why is that a design question, not luck?

Template

Template 12a — The Ask / Use of Funds (Appendix): amount → runway → milestones table, tied to the budget lines.

Template 14 — Pitch Deck (Appendix): the twelve-slide skeleton with per-slide checks — the capstone deck format, matching the course investment-deck components.

Chapter references

Course lectures: *Financial Management for Non-Financial Managers* (investment deck components, elevator pitch format) [57] and *Entrepreneurship Development Support* (funding sources and support programs) [60]. Foundations: Aulet [7] on the funding logic of disciplined ventures; Cheek [8] on pitch tactics and investor conversations; Siciliano [54] on valuation and financial statements for non-financial founders; Hisrich, Peters & Shepherd [23] on sources of capital and the investor's perspective.

Building Ventures in the Age of AI

Learning outcomes

By the end of this chapter you will be able to:

- distinguish AI as a feature, an engine, and a workforce — and explain why only two of the three can be a business;
- choose deliberately between renting (API), adapting (fine-tuning) and building (own models);
- write a data strategy: what data you need, what rights you have, and how the flywheel compounds;
- price AI products against their real variable costs — the cost-per-request discipline from Chapter 11;
- apply responsible-AI basics: bias, privacy, transparency, accountability, and the EU AI Act’s risk tiers;
- use AI inside the founding workflow without outsourcing the founder’s judgment.

13.1 AI Changes the Raw Material, Not the Discipline

Every previous chapter of this practicum applies to an AI venture without exception — and that is the chapter’s first lesson. Persona before product, evidence before adjectives, unit economics before growth: the disciplines do not change because the technology is fashionable. What changes is the *raw material*: software can now perform tasks that previously required human judgment — reading images, drafting text, predicting,

recommending, conversing. A capability that powerful is a gift and a trap. The gift: one student team can now build what needed a department five years ago. The trap: when everyone can build it, **the technology stops being the advantage**, and the advantage moves to the things this book has been teaching all along — proprietary data, distribution, trust, and a named customer pain [8, 31, 61].

The healthy pattern appeared back in Chapter 3: Fridge Mind’s AI entered as a *reliever of a named pain* (forgetting expiry dates), not as a feature looking for one. Hold every AI idea to that standard: if removing the letters “AI” from your one-liner leaves a coherent value proposition, you have a venture; if it leaves nothing, you have a demo.

13.2 Three Ways AI Enters a Venture

- **AI as a feature** — intelligence relieves a specific pain inside a product that would exist without it (expiry scanning in a fridge app, smart suggestions in a booking platform). Lowest risk, fastest to ship, easiest to copy. This is where most student ventures should start.
- **AI as the engine** — the product *is* the intelligence: a model, a dataset, a prediction that did not exist before (the household food-usage dataset Fridge Mind identified as its Core candidate in Chapter 7). Defensible if the data is proprietary; expensive and slow to prove.
- **AI as the workforce** — AI does your back office: support drafts, code scaffolding, market scans, test generation. This is not your product; it is your productivity. It belongs in the budget as reduced costs, and in Chapter 8’s team design as changed roles.

Write down which of the three yours is. Teams that cannot answer usually have a feature pretending to be an engine — with an engine’s burn rate. The strategic stakes of the choice are well documented: firms that compete on AI as an *engine* — algorithms plus networks plus data feedback loops — play by different rules than product companies [62], and the business-model consequences of AI capabilities have their own research stream [63, 64].

13.3 Build, Borrow, or Blend

The central technical-business decision of an AI venture is how much of the intelligence you own [51]:

	Rent (API)	Adapt (fine-tune)	Build (own model)
Cost to start	lowest	medium	highest
Cost at scale	grows per request	medium, predictable	lowest marginal, high fixed
Control & privacy	lowest (data leaves you)	medium	highest
Speed to MVP	days	weeks	months
Defensibility	none — anyone rents the same brain	some — your data shapes it	strongest, if data is yours
Typical choice	student the MVP	post-pilot, if margins demand it	rarely before seed funding

Table 13.1: The build–borrow–blend decision for AI capability.

The practicum’s default is unglamorous and correct: **rent for the MVP** (Chapters 5–6 want speed of learning, not elegance), **adapt when unit economics demand it** (Chapter 11’s margin math tells you when the API bill becomes the biggest COGS line), and **build only what becomes your Core**. Watch the trap from Chapter 6: none of this changes the TRL logic — a rented model still has to survive *your* pilot, in *your* environment, with *your* users’ messy inputs.

13.4 The Data Strategy: Your Real Asset

In an AI venture the durable asset is rarely the model — models commoditize — but the **data**: its exclusivity, quality, and compounding structure. A data strategy answers four questions [25, 51]:

1. **What data do we need** to deliver the value proposition — and what does the product naturally *generate*? The second is the design opportunity: instrument the product so that normal use produces training signal (explicit feedback, corrections, usage patterns).
2. **What rights do we have?** User consent in the terms of service,

licenses for third-party data, and the unresolved questions around AI training rights from Chapter 9. “We scraped it” is not a right; it is a liability with a delay fuse.

3. **How good is it?** Representativeness before volume: a small dataset that covers your real customer distribution beats a large one that does not — and bias in training data becomes bias in your product, which is a market defect before it is an ethical one.
4. **Does it compound?** The **data flywheel**: usage generates data, data improves the model, the better model attracts more usage. A flywheel is a strategy, not a slogan — it works only if each loop is instrumented and the improvement is visible to the user. If your data does not make the product measurably better every month, it is exhaust, not fuel.

This is where Chapter 7’s Core analysis gets its sharpest AI form: a proprietary, consented, compounding dataset is one of the few moats a student AI venture can realistically dig.

13.5 The Economics of AI Products

Chapter 11 warned that AI features have real marginal costs; here is the discipline in full. Every user request to a model is metered: tokens in, tokens out, euros out. Three numbers belong on your dashboard from day one: **cost per active user per month**, **cost per key action** (one scan, one generated recipe, one answered query), and the **margin after AI costs** — not the software-margin fantasy, the real one. Design levers that move them: smaller models for easy tasks (route only the hard 10% to the expensive model), caching repeated queries, batching, and honest limits in the free tier. Pricing must follow the same logic: per-seat pricing breaks when an AI feature’s cost scales with *usage*, so usage-based or tiered pricing is usually the truthful model for AI-heavy products. Run the Chapter 11 simulation at 10× and 100× users — an AI venture whose margins *worsen* with success is a very elegant way to go bankrupt growing.

13.6 Responsible AI: License to Operate

Responsible AI is not decoration for the ethics slide; it is your **license to operate** — in procurement checklists, in app-store reviews, and increasingly in law. The working minimum:

- **Bias and fairness** — test the model on the distribution of your real users, not on demo data; document known failure groups. Fridge Mind’s own SWOT (Figure 7.2) listed “AI accuracy concerns” as a threat — the mature response is a measured accuracy number and a fallback, not a reassurance.
- **Privacy** — personal data sent to a third-party model is data processing; consent, minimization, and retention rules apply (GDPR in the EU market, Chapter 9). Design rule: send the model the least data that can do the job.
- **Transparency** — users should know when they are talking to a model and what it is for; generated content should be marked where deception would otherwise be possible.
- **Accountability** — name the human who answers when the AI is wrong. “The algorithm decided” is not an answer a customer, a court, or a jury accepts.

The **EU AI Act** formalizes much of this on a risk-based ladder [65]: a small set of *prohibited* practices (manipulation, social scoring), *high-risk* uses (hiring, credit, education access, critical infrastructure) that carry documentation, testing and oversight obligations, *limited-risk* uses with transparency duties (chatbots must disclose they are chatbots), and *minimal-risk* uses with no new obligations. Most student apps land in the last two tiers — but check honestly: an app that screens job applicants or grades students is high-risk territory, with homework attached. Obligations phase in over 2025–2027, so “we’ll deal with it later” expires soon.

13.7 AI Inside the Founding Workflow

The same technology changes *how you found*, not just what you build. The honest map, consistent with every AI angle box in this book: AI is strong at **drafting** (plans, decks, contracts to be reviewed by humans), **sparring** (adversarial questions, scenario generation, “attack my assumptions”), **scaffolding** (boilerplate code, tests, data cleaning), and **scanning** (competitor monitoring, literature, regulation). It is weak at — and must not be trusted with — **customer truth** (only interviews and behavior provide it), **commitments** (prices, deadlines, promises to partners), and **verification** (an AI cannot check its own arithmetic or its own citations; Chapter 11’s trap). The founder’s judgment is the one function with no API.

Worked example

Fridge Mind’s accuracy crisis, handled like adults. During the pilot, the expiry-scanning feature misread dates on curved packaging — accuracy dropped from the demo’s 96% to 81% in real kitchens, exactly the threat their SWOT had named (Chapter 7). The team’s response is worth copying. First, they *measured and published internally*: per-category accuracy, worst categories named. Second, they *degraded gracefully*: low-confidence scans ask the user to confirm (human-in-the-loop), so the error cost a tap, not a spoiled dinner. Third, they *turned the fix into the moat*: every confirmation became labeled training data — the flywheel spun *because* the model was imperfect. Fourth, they *priced the truth*: the API cost per scan went into the COGS line, and the subscription tier was sized to cover it at 10× volume. Discussion: which of the four moves is hardest for your product — and what does your current “accuracy, honestly measured” number look like?

Guided steps

Step 1. Classify. Feature, engine, or workforce — one sentence, in writing.

Step 2. Choose. Rent / adapt / build (Table 13.1); write the trigger

condition that would move you one column right.

Step 3. Data strategy. Answer the four questions (need, rights, quality, compounding); deliverable: one page.

Step 4. Cost the AI. Cost per active user and per key action; margin after AI costs; the 10× and 100× simulation.

Step 5. Responsible-AI pass. Bias test plan, privacy minimization, transparency note, named accountable human; locate your product on the AI Act ladder.

Tasks & exercises

Task 13.1 (individual). Remove “AI” from your one-liner: rewrite the value proposition without it, then state precisely what intelligence adds. Deliverable: two sentences + one paragraph.

Task 13.2 (team). Data strategy sheet: need / rights / quality / flywheel, with consent wording sketched. Deliverable: one page (Template 15a).

Task 13.3 (team). AI cost model: cost per key action, per active user, at 1×/10×/100× volume; pricing implication in one sentence. Deliverable: half a page, tied to Template 11b.

Task 13.4 (team). Responsible-AI checklist: bias, privacy, transparency, accountability, AI Act tier — each with one concrete action and an owner. Deliverable: one page (Template 15b).

AI angle

AI angle — the meta edition. This chapter *is* the AI angle, so the box goes one level up: use AI to audit your AI venture. Three prompts that earn their tokens: (1) “List ten ways this product could harm a user if the model is confidently wrong, ranked by severity” — the start of your risk-tier analysis; (2) “Which parts of our data strategy would fail a GDPR review, and why?” — then verify the answer with a human who actually knows; (3) “Argue that our moat is not defensible” — if you cannot answer the argument, neither can the investor who will make it. And the standing rule, worth

repeating in the AI chapter itself: every number an AI gives you is a hypothesis until you find the source.

Systems Thinking Lab — The Concentric Rings

Systems Thinking Lab — product meets business model (R4 + R3). *Fill:* on R4, mark which product elements are AI and how each is sourced (rent/adapt/build); on R3, write the cost-per-key-action next to the revenue stream it serves. *Shift:* your model provider deprecates the version you built on — 90 days to migrate, and the replacement is 30% more expensive per call. Walk the rings: R4 (migration sprint displaces which roadmap items?), R3 (margin falls below target — reprice, reroute to smaller models, or cap the free tier?), R7 (runway impact), R2 (do customers even notice?). One sentence per ring. *Reflect:* what would it take for this shift to be a non-event — and is that worth building now?

Template

Template 15a — Data Strategy Sheet (Appendix): the four questions — need, rights, quality, compounding — with consent and provenance rows.

Template 15b — Responsible AI Checklist (Appendix): bias / privacy / transparency / accountability / AI Act tier, each with action, owner, and evidence column.

Worked example

Fridge Mind — the Responsible AI checklist, filled for the expiry-scan feature. Same rows as Template 15b, with the team's actual answers:

Area	Action	Owner	Evidence
Bias & fairness	test scans across cuisines and packaging types, not just the team's own fridge	ML lead	accuracy report per food category
Privacy & minimization	photos processed on-device; only labels and dates leave the phone; 90-day retention	backend lead	data-flow diagram
Transparency	in-app note: "suggested dates are estimates"; AI-labeled suggestions	product lead	UX copy, store listing
Accountability	product lead answers when the model is wrong; the 96%→81% accuracy crisis (this chapter) triggered the human-fallback path	product lead	incident log, retro notes
AI Act tier	minimal risk — no safety, health or rights impact; documented why	CEO	tier memo [65]

Cost check (ties to Template 11b): EUR 0.012 per scan, EUR 0.34 per active user per month — inside the pricing model only because caching and the small-model route do the heavy lifting.

Chapter references

Foundations: Christensen [31] on why technology alone is not the advantage; Cheek [8] on AI-era execution tactics; Osterwalder & Pigneur [25] on value propositions with intelligent products; WIPO guidance on IP and data in AI ventures [51]; Regulation (EU) 2024/1689 (the AI Act) for the risk-tier framework [65]. The AI-and-strategy shelf: Iansiti & Lakhani, *Competing in the Age of AI* (Harvard) [62]; Davenport & Ronanki, *AI for the real world* (HBR) [66]; Ransbotham et al., *The cultural benefits of AI* (MIT Sloan Management Review & BCG) [67]; McAfee & Brynjolfsson on the big-data management revolution [68]; Vial’s review of digital transformation [69]; Kane, Phillips, Copulsky & Andrus, *The Technology Fallacy* — people, not technology, as the key to digital transformation [61]. The AI-and-business-model shelf: Sjödin et al. on AI capabilities and business model innovation [63]; Jorzik et al.’s systematic review [64];

Climent, Haftor & Staniewski on AI-enabled business models [70]; Roberts & Candi on AI in innovation management [71]; Kuzior, Sira & Brożek on AI in open innovation [72]; Kassa & Worku on AI and organizational performance [73]; Papić on AI adoption stages in technology-intensive SMEs [74]. Course linkage: the AI angle boxes of Chapters 1–12 collect the per-topic AI guidance this chapter generalizes.

Growth, Internationalization and Exit

Learning outcomes

By the end of this chapter you will be able to:

- identify your growth engine — sticky, viral, or paid — and the metrics that prove it is turning;
- sequence expansion with the bowling-alley logic and prepare the organization for scale;
- screen international markets and choose an entry mode appropriate to a software venture;
- name the exit strategies — family transfer, employee ownership, sale, management buyout, termination — and their trade-offs;
- run a sale process from preparation to transition, and face failure with a turnaround discipline;
- complete the practicum: assemble the final business plan, pitch deck, rings map, and toolkit.

14.1 The Long Game

The practicum ends where the venture begins. Thirteen chapters built the machine; this one asks what the machine is *for* over years: how it grows without breaking, where in the world it should grow, and how the founders eventually harvest the value — or close the venture with dignity and data. Entrepreneurs spend enormous energy starting ventures and remarkably little planning the exit; many even read the sale of a company

as failure [75]. It is not. An exit is a design decision made on day one — because the growth path you choose determines which exits are even possible.

14.2 Growth: Scale What Repeats

Growth is not more of everything; it is **more of what repeatably works**. Before scaling, name your engine [9]:

- **Sticky engine** — growth from retention: customers stay and pay longer than it cost to acquire them (Chapter 11’s LTV:CAC is its gauge). Metric: churn / retention cohorts.
- **Viral engine** — growth from usage that recruits: each user brings users (invites, shared outputs, network effects). Metric: the viral coefficient — how many new users one user generates; above 1.0 the loop feeds itself.
- **Paid engine** — growth bought through channels whose CAC is provably below LTV. Metric: CAC payback by channel.

Most real ventures run one primary engine with another as support; claiming all three usually means none is measured. The second discipline is sequencing — Moore’s **bowling alley**: knock down one niche (your beachhead from Chapter 2), then the adjacent niche that the first makes easier, then the next; crossing to the mainstream without the alley is how good products die in the chasm between early adopters and the pragmatic majority [34, 35]. Chapter 5’s adoption curve is the map; the bowling alley is the route.

Finally, growth is an organizational event, not only a market one. What breaks first is never the server — it is the coordination: hiring ahead of the pain (Chapter 8’s charter scales into roles and processes), delegation with written principles (Dalio’s lesson from Chapter 8), and management by objectives so the founder stops being the bottleneck [10, 36]. Plan the organization’s next version with the same honesty you plan the product’s.

14.3 Internationalization: The World Is the Follow-On Market

For a software venture from a small domestic market, internationalization is not ambition — it is arithmetic: the home market caps the TAM, and the product already speaks a portable language (code). The questions are *when*, *where*, and *how* [60].

When: after the engine repeats at home — internationalizing a broken funnel just burns money in more time zones. **Where:** screen markets on five criteria: market size and growth (your TAM method from Chapter 2, applied per country), linguistic and cultural distance (localization cost), willingness-to-pay and price levels, competition intensity, and regulatory load (GDPR for EU entry; the AI Act tiers from Chapter 13 if your product is AI). **How** — the entry modes for software, lightest first:

1. **Digital-first export** — sell from home through app stores, marketplaces, and the web; localize the product, not the company. Default first move.
2. **Partners and resellers** — a local partner carries sales and support for a share; classic open-innovation logic (Chapter 4) applied to geography.
3. **Local presence** — a subsidiary or hired country team, only when volume justifies fixed costs.

Support exists and should be used deliberately: export-promotion and innovation-support programs, EU instruments for digital SMEs, diaspora networks as beachhead channels abroad [60]. And one localization warning: translation is the smallest part of localization — payment habits, support expectations, and trust signals differ more than vocabulary.

14.4 Exit: Designing the End at the Beginning

An **exit strategy** answers: how do the owners eventually convert this venture into value, freedom, or both? The menu, with its trade-offs [75]:

- **Transfer to a family member** — the classic succession, and statistically fragile: about half of such transfers fail from the first to the second generation, and only around 14% of firms survive to the third. If this is the plan, write an *inheritance plan*: the owner's role during transition, family dynamics, the successor's early involvement in operations, a fixed finalization date — and the honest check of whether the successor wants and can run the business at all.
- **Transfer to employees** — sell to the people who built it: a *shareholding plan* spreads the purchase over two to three years (borrowed money buys the owner's shares into employees' accounts, repaid over time) and doubles as a unique incentive; the process must be open and involve all employees, or it breeds departures instead of loyalty.
- **Sale of the company** — the standard startup exit. Options: *direct sale* to an outside buyer; the employee shareholding plan above; or a *management buyout* (MBO), where loyal key managers buy the company, typically financed through banks against a profitable operation and a detailed financial plan. For tech ventures add two industry-specific variants: the *acqui-hire* (bought for the team) and the *strategic acquisition* (bought for the product, the dataset — your Chapter 7 Core — or the customer base). IPOs exist mostly in slide decks; plan for acquisition.
- **Termination** — closing the venture. Not a strategy anyone wants, but a strategy everyone should have: an orderly wind-down protects customers, employees, and the founders' reputation — which is the only asset you carry into the next venture for free.

Which exit fits is decided by what you built: a profitable niche product suits employee or family transfer; a compounding dataset or a defensible platform suits strategic sale. State your intended exit in the business plan — investors choose *you* partly by whether your exit story matches their fund's math (Chapter 12).

14.5 The Sale Process, Briefly

If sale is the path, the process has a choreography [75]:

1. **Preparation** — clean books, contracts and IP assignments in order (Chapter 9’s homework pays off here), no dependence on the founder that cannot be documented away; NDAs signed before any sensitive information moves; the management team strong enough that the buyer is acquiring a company, not rescuing one.
2. **The initial meeting and offer evaluation** — examine the buyer’s reputation, credibility, management style and integrity as carefully as they examine your numbers; you are choosing who inherits your team.
3. **Negotiation** — not a single session but a stream running through every phase, with new information and analysis at each step; price is one term among many (earn-outs, retention, your role after).
4. **Due diligence** — the mirror image of Chapter 12: the buyer now audits commercial, financial and legal reality, while lawyers draft the sale contracts.
5. **Transition** — buyers commonly ask the seller to stay for a period to smooth the handover; negotiate its length, role, and compensation explicitly.

14.6 Failure and Turnaround

Many new ventures fail — especially in downturns — and a practicum that pretends otherwise would be lying to you [75]. If it happens to yours, the sequence is human before financial: talk it through with family; ask for external help — professionals, friends, business associates; and have the courage to step away from a venture that will obviously keep draining resources, because closing early is cheaper than closing late. If there is something worth saving, the turnaround discipline is three provisions: *aggressive, engaged management*; *a plan* (where are we now, where are we going); and *action* (how do we reach the goal) — with everyone contributing to cost cuts, efficiency, and the sales push. Notice that this is Chapter 10’s project management and Chapter 11’s runway math, running in reverse at high speed. And keep the artifacts: a failed venture with a written risk register, experiments log and post-mortem is called *experience*; one without them is just a loss.

Worked example

Fridge Mind, five years out — three plausible futures. *Future A (sticky engine, international):* campus retention proves the model (churn under 4%/month), the bowling alley runs campus → city households → the region; entry into the EU market goes digital-first with localized recipes and GDPR-compliant data handling; a European appliance maker acquires them for the dataset — the Core from Chapter 7 pays off exactly as designed. *Future B (steady niche):* growth plateaus at a profitable size; the founders decline VC pressure, and five years later sell to their own early employees through a shareholding plan — smaller outcome, intact culture. *Future C (the honest one):* the device economics never close; after two failed pivots the team runs the failure sequence, winds down with customers migrated gracefully, and publishes the post-mortem — two founders are hired on the strength of it. Discussion: which future does your venture's current evidence actually point to — and which ring would you change this month to shift the odds?

Guided steps

- Step 1. Engine.** Name your primary growth engine and its one metric; compute that metric from your pilot data or state the assumption.
- Step 2. Alley.** Write the bowling-alley sequence: three niches in order, with the entry evidence for each.
- Step 3. Screen.** Score two candidate foreign markets on the five criteria; pick an entry mode and justify it in three sentences.
- Step 4. Exit.** Choose the intended exit strategy; write what must be true (numbers, dates) for it to be available.
- Step 5. Assemble.** Final capstone: business plan final, pitch deck final, rings map complete, all 17 templates filled.

Tasks & exercises

Task 14.1 (individual). Growth-engine memo: engine, metric, current value, next experiment to improve it. Deliverable: half a page.

Task 14.2 (team). Internationalization screen: two markets $\times$ five criteria + entry mode. Deliverable: one page (Template 16a).

Task 14.3 (team). Exit plan: chosen strategy, preconditions, and the one thing to fix now to keep that exit open. Deliverable: half a page (Template 16b).

Task 14.4 (team, final capstone). The complete practicum package: final business plan, final pitch deck + recorded pitch, the rings map with all eight rings filled, and the management toolkit (Templates 1–16 complete). Deliverable: the full portfolio — this is what the jury grades.

AI angle

AI angle. AI compresses two of this chapter’s hardest tasks. For *internationalization*: market scans, competitor maps per country, and first-pass localization (UI, support macros, onboarding) are now afternoon tasks — but have a native speaker review anything customers will read, because near-perfect translation with one wrong cultural signal is worse than no translation. For *growth*: AI is genuinely strong at analyzing funnel data and proposing experiments; keep the experiment discipline from Chapter 6 (hypothesis, metric, kill criterion) — an AI that generates fifty growth ideas is useful only if you still test them one at a time. And in the sale process: never let an AI draft your data room answers unsupervised; due diligence is the one place where a confident hallucination becomes a legal problem.

Systems Thinking Lab — The Concentric Rings

Systems Thinking Lab — the whole map (all rings). *Fill*: complete the rings map — every ring now carries evidence, not aspirations. *Shift*: a strategic buyer offers to acquire your venture

at a fair price — but only if you abandon the second follow-on market and integrate into their product line within a year. Walk all eight rings: which rings does the offer strengthen (R7 financials, R8 ecosystem), which does it amputate (R1 the idea's ambition? R5 the team's roles?), and which become someone else's decision (R3, R4)? *Reflect*: after fourteen chapters — is your answer to the offer driven by the numbers, by the mission sentence from the Introduction, or by the risk register? There is no wrong answer; there is only an unexamined one. Write the examined one. This is the last lab; the map is yours now.

Template

Template 16a — Growth & Internationalization Sheet (Appendix): engine + metric, bowling-alley sequence, market screen grid, entry mode.

Template 16b — Exit Plan (Appendix): strategy choice, preconditions, sale-process checklist, failure/turnaround sequence.

Worked example

ZenTutor — screening the second market, filled. Belgrade works; the engine repeats. Two candidates scored on this chapter's five criteria (1 = bad, 5 = good):

Criterion	Zagreb (HR)	Sofia (BG)
Market size & growth (TAM method, Ch. 2)	3	3
Linguistic & cultural distance (localization cost)	5	3
Willingness to pay & price levels	4	2
Competition intensity	3	4
Regulatory load (EU/GDPR, consumer rules)	4 (EU member)	4 (EU member)
Total	19	16

Decision: Zagreb, **digital-first export** mode — the app is localized, the company is not; tutor supply is seeded through one partner academy (light partner logic from Chapter 4), and a country hire is explicitly deferred until monthly bookings there cover it. Sofia stays in the archive with the reason written down: weaker willingness-to-pay would force a price tier the unit economics (Chapter 11) does not yet support.

Chapter references

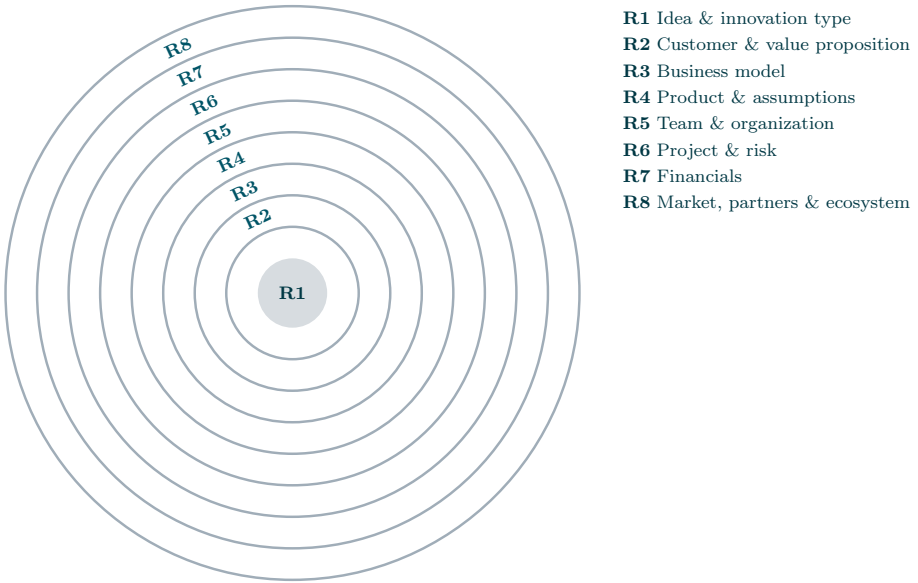
Course lectures: *Entrepreneur Exiting Venture* (exit strategies, sale process, failure and turnaround) [75] and *Entrepreneurship Development Support* (support programs, internationalization) [60]. Foundations: Ries [9] on growth engines; Moore [35] and Rogers [34] on adoption, the chasm and the bowling alley; Drucker [10] on management at scale; Doerr [36] on objectives that survive growth; Fitzgerald, Stol, Minör & Cosmo, *Scaling a Software Business*, on the digitalization journey beyond the first product [76]; van Cann, Jansen & Brinkkemper, *Software Business Start-up Memories*, on the decisions that separated surviving software ventures from dead ones [77].

The Management Toolkit: Templates

This appendix collects every template referenced in the fourteen chapters, in fill-in form. Print them, copy them, redraw them on whiteboards — they are working instruments, not illustrations. Each template carries a pointer to the chapter that teaches it and, where one exists, to a filled student example in the main text. The final sheet is the Concentric Rings workbook page: by the end of the practicum it should hold your whole venture on one page.

The Concentric Rings Workbook Sheet

The practicum's master sheet. Fill a ring when its chapter is done; revisit it in every Systems Thinking Lab. By Chapter 14 all eight rings should carry evidence, not aspirations.



Evidence per ring — one line each: what do we *know*, and how do we know it?

.....

.....

.....

.....

.....

.....

.....

.....

Last shift tested — what moved, and which other rings moved with it?

.....

.....

.....

Template 1 — Idea Validation Matrix

Chapter 1. Score every idea 1–5 per dimension, multiply by the weights, rank the portfolio. Weights shown are the practicum default; change them only with a written reason.

Dimension	Weight	Idea A	Idea B	Idea C
Personal motivation & founder fit	×3			
Evidence of a real problem (PMR)	×3			
Market size & growth	×2			
Feasibility with our skills & time	×2			
Differentiation / “why now?”	×2			
Weighted total				

Idea A (one line)
.....

Idea B (one line)
.....

Idea C (one line)
.....

The winning idea survives because
.....
.....

First pivot trigger we would accept

.....

Template 2a — Market Research Worksheet & TAM–SAM–SOM

Chapter 2. One worksheet per research question; the sizing table turns findings into numbers. Every cell cites a source or is marked assumption.

Research goal (what decision will this inform?)

.....

Sources used (reports, registers, interviews with dates)

.....

.....

Key findings (with source per line)

.....

.....

.....

Level	Definition used	Calculation	Value
TAM — total market			
SAM — serviceable			
SOM — obtainable (beachhead)			

Bottom-up check (count $\times$ price, per segment)

.....

.....

“Why now?” — the trend that makes this timely

.....

Template 2b — Option Prioritization Matrix (Opportunity × Feasibility)

Chapter 2. The course rubric: score each option 0–3–6–9–12 per factor using the anchored statements below (course scaling table), then plot. Adapted from a University of Cambridge IfM workshop format [21].

Opportunity factors	Feasibility factors	Anchors (0 / 6 / 12)
Market attractiveness	Differentiation	none / some features better / several clearly better
Customer importance	Sustainability of advantage (IPR)	easy to copy / continuous effort / protected by IPR
Impact on key customer relations	Technical challenge	not demonstrated / partly prototyped / fully tested
Market growth & timing	Market knowledge	unsupported / sized ±50% / known ±5%, tested
	Technical capability	new team needed / acquirable in 3 months / in-house
	Fit to sales & distribution	new channel / some changes / existing channels fit
	Finance	source unknown / justifiable / within budget
	Strategic & organisational backing	opposition / persuading needed / full support



Options plotted and their totals

.....

.....

Template 3a — Persona Canvas (B2C & B2B variants)

Chapter 3. Build the persona from PMR evidence only — every field should be traceable to an interview quote. A filled B2C/B2B example pair appears as Figure 3.1.

Name & one-line story

.....

B2C fields	B2B fields
Who (age, life stage)	Industry segment
Location	Public or private
Income & purchase habits	Employees, revenue, turnover
Hobbies & goals	Growth rate, current state
Challenges & frustrations	Decision maker
Where to find them	Situational factors

A day in the life (the worst Tuesday)

.....

.....

.....

Judging criteria — how they will evaluate us

.....

.....

Evidence: three interview quotes behind the fields

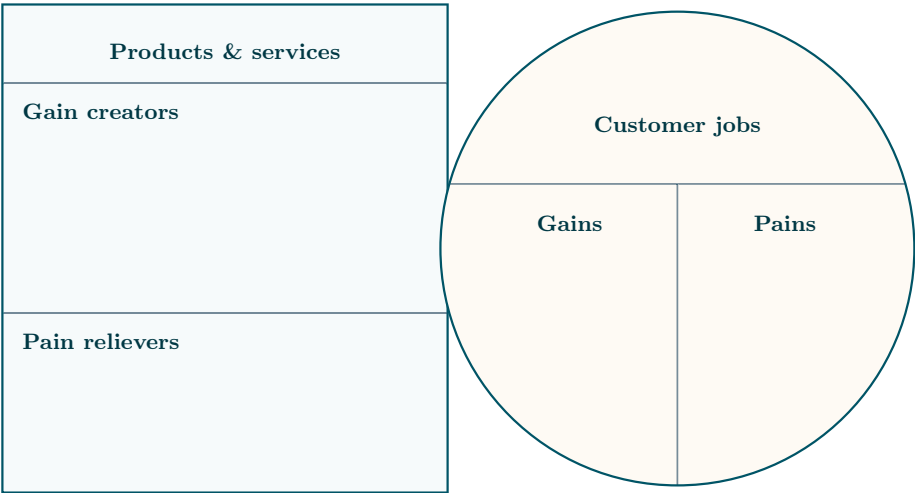
.....

.....

.....

Template 3b — Value Proposition Canvas

Chapter 3. The canvas from Figure 3.2, full page. Fill the circle (customer) first, then the square (offer); check one-to-one fit. A filled student example is discussed in Chapter 3.



Fit check: every serious pain has a reliever?

.....

Fit check: every important gain has a creator?

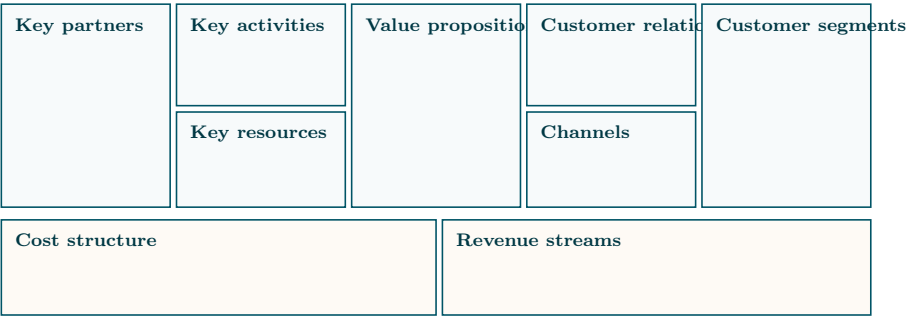
.....

Quantified top benefit (customer's units, assumptions shown)

.....

Template 4a — Business Model Canvas

Chapter 4. Fill in this order: segments → value → channels → relationships → revenue → resources → activities → partners → costs. A filled student example appears as Figure 4.2.



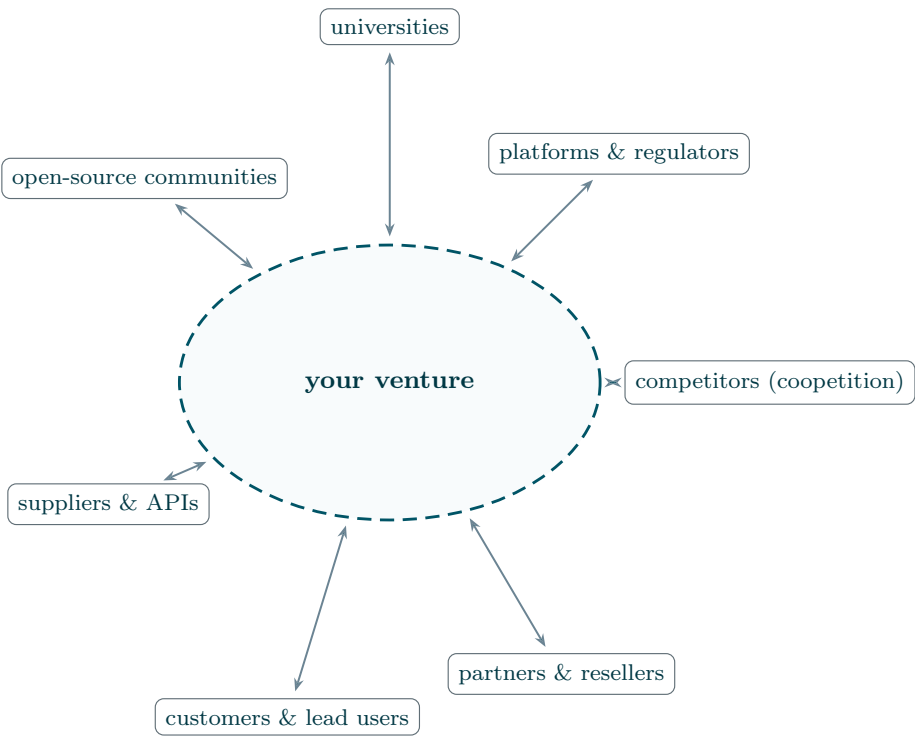
The block we are least sure about, and the test for it

.....

.....

Template 4b — Open Innovation Flow Map

Chapter 4. Draw your venture’s boundary, place the ecosystem actors, then label every arrow with what flows (knowledge, data, IP, revenue) and its direction. Base diagram: Figure 4.1.



Outside-in flows we deliberately use

.....

.....

Inside-out flows we deliberately release (and what we get back) ...

.....

.....

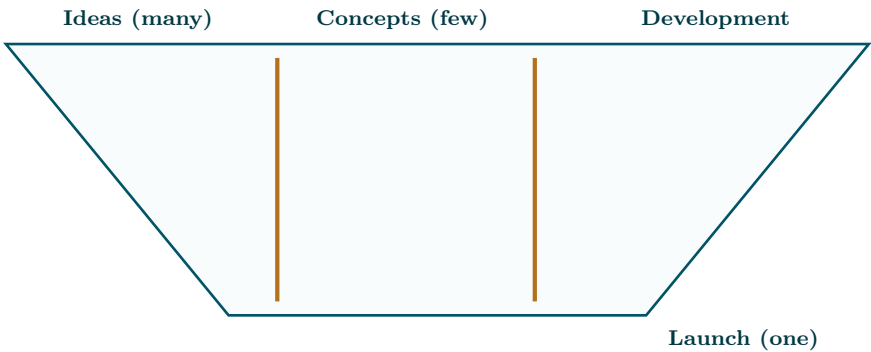
Coupled partners (joint development) and the agreement status ...

.....

.....

Template 5a — Innovation Funnel Map

Chapter 5. Your funnel with two editable gates; write the criteria per gate before ideas arrive. Kill criteria are a kindness.



Gate 1 (screen) criteria — go/kill with numbers

.....

.....

Gate 2 (test) criteria — go/kill with numbers

.....

.....

Ideas currently in the funnel, and their stage

.....

.....

.....

Template 5b — Innovation KPI Table

Chapter 5. Every metric gets a type, an owner, a cadence and a kill/scale threshold — the OKR discipline applied to innovation [36].

Metric	Type (input / throughput / output)	Owner	Cadence	Kill / scale threshold
--------	------------------------------------	-------	---------	------------------------

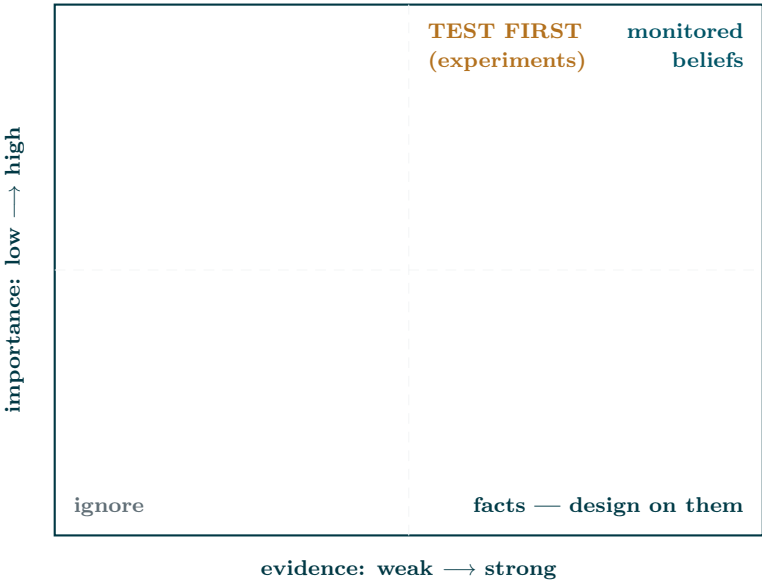
Objective these KPIs serve (one sentence, OKR style)

.....

.....

Template 6a — Assumption Map

Chapter 6. Plot every assumption by importance and evidence. Top-right quadrant (important, unproven) is where experiments come from — Figure 6.1 shows the filled logic.



Top three test-first assumptions

.....

.....

.....

The one we are afraid to test, and why

.....

.....

Template 6b — Experiment Card

Chapter 6. One card per experiment. No card, no experiment — opinions are not tests.
Pair with the TRL sheet: state TRL before and expected TRL after.

Experiment card #

Hypothesis (we believe that ...)

.....

Test (the smallest thing that could prove us wrong)

.....

Metric & threshold (pass if ...)

.....

Kill / scale rule (if fail → ...; if pass → ...)

.....

Timebox & owner

.....

TRL before / expected after

.....

Result & decision (fill after running)

.....

Template 7a — SWOT / TOWS Matrix

Chapter 7. Course format: every entry needs evidence; risks must be visible from the SWOT (they feed Template 10a). A filled student example appears as Figure 7.2.

Strengths (internal) + evidence	Weaknesses (internal) + evidence
Opportunities (external) + evidence	Threats (external) + evidence

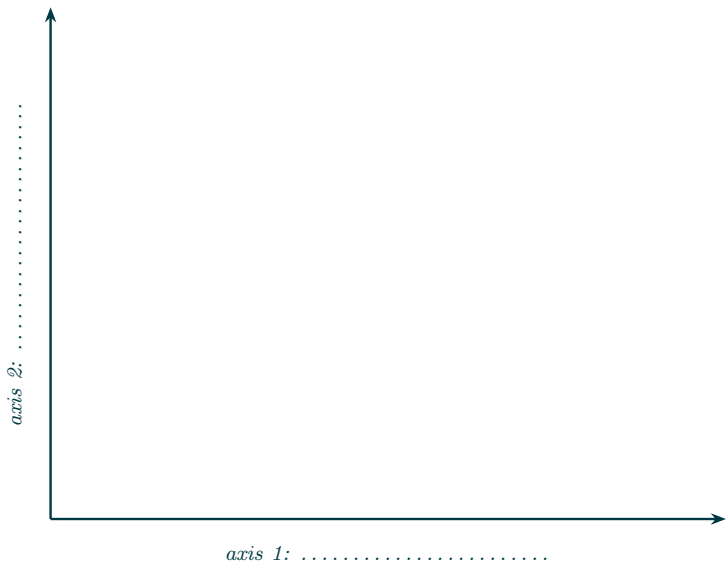
Attack (strength × opportunity)
.....
Build or borrow (weakness × opportunity)
.....
Defend (strength × threat)
.....
Retreat or insure (weakness × threat)
.....

Threats copied into the risk register (Template 10a)

.....

Template 7b — Competitive Positioning Chart

Chapter 7. Choose the two axes your customers actually care about; justify each axis in one line; plot with evidence.



Why these two axes (customer evidence)

.....

.....

Competitors plotted, with the evidence per position

.....

.....

.....

Our intended position — and the Core that defends it

.....

.....

Template 8a — Team Charter

Chapter 8. The team’s constitution — your “Principles, miniature edition” [43]. Sign it; revisit it monthly.

Purpose (why this team exists, one sentence)

.....

Members, roles & decision rights

.....

.....

.....

Decision rules (what needs consensus, what one owner decides) ...

.....

.....

Rhythms (weekly meeting, retro, demo — days and times)

.....

.....

Conflict protocol (how we disagree, who arbitrates, the 48-hour rule)

.....

.....

Equity & vesting summary (4 years / 1-year cliff; what happens on exit)

.....

.....

Signatures & date

.....

Template 8b — Belbin Roles Map

Chapter 8. Place each member under their natural roles (max two), then find the gaps [44]. The nine roles appear in Figure 8.1.

Thinking roles	Action roles	People roles
Plant	Shaper	Coordinator
Monitor Evaluator	Implementer	Team Worker
Specialist	Completer Finisher	Resource Investigator

Member → roles (who covers what)
.....
.....
.....

Gaps (roles nobody covers) and the mitigation
.....
.....

Bus-factor check: what breaks if each member leaves?
.....
.....

Template 9a — Legal Form Decision Table

Chapter 9. Score the four realistic forms for your situation; the justification row is the deliverable. Serbian specifics (capital, tax) are in Table 9.1; check current figures with the APR [50].

Criterion	Preduzetnik	d.o.o.	a.d.	Other
Setup cost & speed				
Liability protection				
Tax fit for our case				
Investor readiness				
Our justification				
Chosen form & why				
Registration steps and dates (APR)				

Template 9b — IP Inventory

Chapter 9. List everything protectable, its type, owner, status and next action. The three student traps apply: contractor code, open-source obligations, university involvement [51, 52].

Asset	IP type	Owner (on pa- per)	Status	Next action
-------	---------	-----------------------	--------	-------------

Trap check: contractor agreements signed? open-source licenses listed?
university rules checked?

.....

.....

Template 10a — Risk Management Table

Chapter 10. The course format from the exercises: seven categories, each with a description and an actions row, rated HIGH / MEDIUM / LOW — extended with owner, trigger and fallback. Risks must be traceable from your SWOT threats (Template 7a).

Category	Description of risk	Actions	H/M/L	Owner	Trigger / fallback
Technology					
Timing, milestones & results					
Employees & partners					
Procurement					
Budgetary					
Market					
Environmental & social					

Red-zone risks discussed in this week’s retro (date + decision)

.....

.....

Template 10b — WBS / Gantt Skeleton

Chapter 10. Decompose first (WBS: 100% rule, single owner per leaf), then schedule.
A filled student Gantt appears as Figure 10.3.

Deliverable-oriented chunks (WBS level 1)

.....
.....

Level-2 tasks with owners (WBS level 2)

.....
.....
.....

	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12
<i>task</i>												

Milestones (diamonds): scope freeze / pilot gate / launch — dates

.....

.....

Critical path (the chain where a week lost is a week lost everywhere)

.....

.....

Backlog	To do	Doing	Review	Done
card: verb + deliverable + owner				
WIP limit: -	WIP limit: 5	WIP limit: 3	WIP limit: 2	WIP limit: -

.....

.....

Template 11a — Project Budget Sheet

Chapter 11. The WBS with prices on it: fixed, variable, contingency (10–15%), on a monthly timeline, with the running cash line [57].

Line	M1	M2	M3	M4	M5	M6
Fixed costs (salaries, rent)						
Variable costs (cloud, APIs, contractors)						
Contingency reserve						
Revenue / funding in						
Running cash						
Three largest cost drivers						
.....						
.....						
Burn rate / runway (cash ÷ burn = months)						
.....						
Break-even month (on the Gantt)						
.....						

Template 11b — Unit Economics Sheet

Chapter 11. Tag every input data or assumption. The verdict is the ratio: target $\geq$ 3:1, payback within 12 months (or within the pilot for a semester venture) [57].

Price and revenue model

.....

COGS per user per month (hosting, APIs, support) — data/assumption

.....

Gross margin per user per month

.....

Average customer lifetime (months) — data/assumption

.....

LTV = margin $\times$ lifetime

.....

CAC per channel (costs $\div$ new customers) — data/assumption

.....

LTV : CAC and CAC payback (months)

.....

The simulation: these numbers at 10× and 100× users

.....

.....

Template 12a — The Ask / Use of Funds

Chapter 12. Every euro traceable to the budget (Template 11a); phrase the use as milestones bought, not expense categories.

The ask (amount, one sentence with runway and milestones)

.....

Use of funds	Amount / %	Milestone it buys (with date)
--------------	------------	-------------------------------

Funding-source fit: which rung of the ladder, and the evidence that unlocks it

.....

.....

Template 13 — Business Plan Skeleton

Chapter 11. The capstone document. Page limits are guidance for the student edition; every section lists the artifact that feeds it.

Section	Pages	Feeds from (chapter / template)
1. Executive summary	1	written last; stands alone
2. Problem & opportunity	1–2	Ch. 2 PMR, TAM–SAM–SOM (2a)
3. Solution & product	1–2	Ch. 3 VPC (3b), Ch. 6 MVP/TRL (6a, 6b)
4. Business model	1–2	Ch. 4 BMC (4a), OI flows (4b)
5. Market & competition	1–2	Ch. 7 positioning (7b), SWOT (7a)
6. Go-to-market	1	beachhead, first 10 customers, CAC
7. Team & organization	1	Ch. 8 charter (8a, 8b), Ch. 9 legal/IP (9a, 9b)
8. Execution plan	1–2	Ch. 10 WBS/Gantt (10b), risks (10a)
9. Financials	2–3	Ch. 11 budget (11a), unit economics (11b), the ask (12a)
10. Appendices	as needed	artifacts, experiments log, LOIs

Assumptions table: the ten inputs everything derives from

.....

.....

Template 14 — Pitch Deck Skeleton

Chapter 12. Twelve slides, one message each, arc: problem → solution → utopia → evidence → the ask. Matches the course investment-deck components [57].

Slide	One message is ...	Check
1. Title	name + one-liner, on-brand	readable in 3 s
2. Problem	pain in the customer’s words	PMR verbatim present
3. Solution	product as the answer	one screenshot
4. Market opportunity	TAM–SAM–SOM + why now	numbers cited
5. Business model	how we make money	pricing visible
6. Go-to-market	beachhead, channels, first 10	CAC per channel
7. Competition	positioning chart	Core as moat
8. Traction	pilots, LOIs, growth	only real things
9. Financials	3-year projections, break-even	derivable from plan
10. Team	why us; gaps to hire	photos, roles
11. The ask	amount, runway, milestones	ties to budget
12. Closing	vision + call to action	contact on screen

Elevator pitch (60 s version, from Task 12.1)

.....

.....

Template 15a — Data Strategy Sheet

Chapter 13. In an AI venture the durable asset is the data: exclusivity, quality, compounding.

Data we need to deliver the value proposition

.....

.....

Data the product naturally generates (instrumented how?)

.....

.....

Rights: consent wording, third-party licenses, provenance

.....

.....

Quality: how representative is it of real users? Known gaps

.....

.....

The flywheel: usage → data → better model → usage — where each loop is measured

.....

.....

Template 15b — Responsible AI Checklist

Chapter 13. Your license to operate. Locate the product on the EU AI Act ladder and attach evidence per row [65].

Area	Action	Owner	Evidence
Bias & fairness	test on the real user distribution; document failure groups		
Privacy & minimization	least data to the model; consent; retention		
Transparency	users know it is AI; generated content marked		
Accountability	named human answers when the AI is wrong		
AI Act tier	prohibited / high / limited / minimal — and why		

Cost per key action and per active user (ties to Template 11b)

.....

Template 16a — Growth & Internationalization Sheet

Chapter 14. Scale what repeats: one primary engine, one metric; then the bowling alley; then the world.

Primary growth engine (sticky / viral / paid) and its metric

.....

Current value of that metric (data or assumption)

.....

Bowling-alley sequence: niche 1 → 2 → 3, with entry evidence

.....

.....

Criterion	Market A	Market B
Size & growth		
Language & cultural distance		
Willingness to pay		
Competition intensity		
Regulatory load (GDPR, AI Act)		

Entry mode (digital-first / partners / local presence) — three-sentence justification

.....

.....

Template 16b — Exit Plan

Chapter 14. Design the end at the beginning: the intended exit decides what you build and who funds it [75].

Intended exit strategy (family / employees / direct sale / MBO / wind-down)

.....

What must be true for it (numbers, dates, preconditions)

.....

.....

Who the plausible buyers/successors are, and why they would want us

.....

.....

Sale-process readiness: books, contracts, IP assignments, NDAs, transition role

.....

.....

Failure sequence (family → external help → orderly close) — and what we would publish

.....

.....

*When this page is filled, the toolkit is complete: seventeen
instruments,
one venture, one map. — End of the practicum templates.*

Capstone Grading Rubrics

The capstone package (Task 14.4) is graded with the three rubrics below: the **business plan** (50%), the **pitch deck with delivery** (30%), and the **rings map with the toolkit** (20%). Levels: **4** = exceeds expectations, **3** = meets them fully, **2** = partial, **1** = missing or superficial. Each criterion's score is level $\times$ weight.

Rubric B1 — Business Plan (50%)

Criterion	Weight	Level 4 looks like
Evidence base	3	every claim traceable to PMR, pilots or cited data; assumptions tagged
Customer & value logic	3	persona, VPC fit and quantified benefit consistent throughout
Business & financial model	3	projections derivable from stated assumptions; unit economics honest
Execution plan	2	WBS/Gantt realistic; risks with owners, triggers, fallbacks
Coherence of the system	2	no contradictions between sections; rings logic visible
Writing & format	1	summary stands alone; professional layout; within page limits

Rubric B2 — Pitch Deck & Delivery (30%)

Criterion	Weight	Level 4 looks like
Narrative arc	3	problem → solution → utopia; room leans forward in 30 s
Slide craft	2	one message per slide; artifacts, not adjectives
Evidence & numbers	3	traction real; the ask traceable to the budget
Delivery	2	rehearsed, timed, one voice; credible Q&A

Rubric B3 — Rings Map & Toolkit (20%)

Criterion	Weight	Level 4 looks like
Completeness	2	all eight rings filled; all templates present
Systems thinking	3	shifts traced across rings correctly; labs show real reasoning
Evidence discipline	3	entries carry evidence or honest “assumption” tags

Weights are multiplied by levels (1–4). Suggested grade bands: 90%+ = excellent, 75–89% = very good, 60–74% = good, below 60% = revise and resubmit. Instructors may adjust weights per course (PRBS, ESE, IM, AI in Entrepreneurship) — announce changes in week one.

Glossary

A/B testing Controlled comparison of two variants to measure which performs better on a chosen metric..

Agile Delivery approach built on short iterations, working software every cycle, and scope that flexes instead of dates..

Angel investor Individual investing personal money at the earliest stage, usually for equity plus mentoring..

API Application programming interface: a defined set of endpoints and protocols that let software components talk to each other..

Assumption map 2×2 grid plotting beliefs by importance and evidence; the top-right quadrant feeds the experiment queue..

Beachhead market The first, deliberately narrow segment you dominate before expanding (Moore's bowling alley starts here)..

Belbin roles Nine team roles in three families (thinking, action, people) used to map coverage and gaps in a team..

BMC Business Model Canvas: nine blocks describing how the venture creates, delivers and captures value..

Break-even point The customer count or month where cumulative revenue catches total costs..

Burn rate Net cash out per month; with cash on hand it defines runway..

Bus factor How many team members can disappear before the venture stops; a risk, not a joke..

CAC Customer acquisition cost — sales and marketing spend divided by new customers won..

Cap table The table of who owns what; keep it simple and in writing..

Chasm The gap between early adopters and the pragmatic majority where unsequenced products die (Moore)..

Chatbot A conversational software agent that interprets user messages and returns relevant replies, often powered by a large language model..

CI/CD Continuous integration and continuous delivery/deployment: automated build, test and release pipelines..

COGS Cost of goods sold; in software: developers, hosting, third-party licenses, support, infrastructure..

Contingency reserve Budget line for expenses you cannot yet name; 10–15% in student projects..

Core What you can do that others cannot easily copy — the anchor of competitive strategy (DE step 10)..

Critical path The longest chain of dependent tasks; a week lost there moves the launch a week..

Data flywheel Usage generates data, data improves the model, the better model attracts usage — if instrumented..

Data-driven Making decisions based on measured evidence rather than opinion or assumptions..

Design thinking Empathize–define–ideate–prototype–test; the discovery discipline behind Chapters 3–6..

DevOps Culture and practice that unites development and operations through automation, shared responsibility and fast feedback..

Dilution The reduction of founders' ownership when new equity is issued; not theft, but compounding..

Due diligence The buyer's or investor's audit of commercial, financial and legal reality..

Elevator pitch The 60-second version: company, problem, solution, users, advantages, added value..

Experiment card Hypothesis, test, metric/threshold, kill/scale rule, timebox, owner — no card, no experiment..

Explainable AI AI systems whose outputs can be understood and traced back to their inputs by humans..

Gantt chart WBS tasks laid on a calendar: bars, milestone diamonds, dependency arrows..

Go-to-market The plan and tactics for launching a product to its first customers..

Gross margin Revenue minus COGS, divided by revenue; decides how much of each euro can fund growth..

ICP Ideal customer profile — screenable criteria, different for B2C (person) and B2B (firm)..

Kanban Board with columns as workflow stages and a work-in-progress limit; steering, not planning..

KPI Key performance indicator: a focused metric that tracks progress toward a strategic goal..

LTV Customer lifetime value — revenue (better: margin) per customer over the whole relationship..

Machine learning Algorithms that improve their performance on a task by learning from data rather than being explicitly programmed..

MBO Management buyout: loyal key managers buy the company, usually with bank financing..

MVP Minimum viable product: the smallest build that tests the riskiest assumption, TRL 4–6 territory..

Neuromarketing Application of biometric and neuroscience methods to measure consumer response to marketing stimuli..

OKR Objectives and key results; objectives with measurable outcomes, owned and time-bound (Doerr)..

Open innovation Deliberate flows of knowledge across the firm's boundary: outside-in, inside-out, coupled..

Open-source Software whose source code is publicly available to use, study, modify and distribute..

Persona One concrete, named customer with a story — the reference object for product decisions..

Pilot A small, controlled trial of a product or process with real users before full rollout..

Pivot A structured change to one or more business-model elements based on validated learning..

PMR Primary market research: interviews and observation of real customers, verbatim evidence..

Pre-money / post-money Valuation before / after the investment; investor share = investment $\div$ post-money..

Proof of concept An early demonstrator that shows a key technical or business idea is feasible..

Risk register The living list of risks with description, actions, rating, owner, trigger, fallback (Template 10a)..

Runway Cash on hand divided by burn rate — months of life left..

Scalability The ability of a product or process to handle growing demand without proportional cost growth..

Stage-gate Innovation process with explicit go/kill decision points and written criteria (Cooper)..

Stakeholder Any person or group with an interest in or influence on the venture's outcome..

SWOT/TOWS Sorting instrument: internal strengths/weaknesses, external opportunities/threats, converted into attack/build/defend/retreat actions..

TAM–SAM–SOM Total, serviceable and obtainable market; SOM is the beachhead you can actually win..

Technology stack The set of languages, frameworks, tools and services used to build and run a product..

TRL Technology readiness level, 1–9; the MVP zone sits at 4–6..

Unit economics The revenue and cost associated with one customer, one transaction or one unit sold..

Value chain The sequence of activities that add value to a product or service from raw inputs to the customer..

Vesting Earning equity over time (typically 4 years, 1-year cliff); protects the team from early departures..

VPC Value Proposition Canvas: offer (products, pain relievers, gain creators) fitted to the customer (jobs, pains, gains)..

WBS Work breakdown structure: deliverable-oriented decomposition, 100% rule, one owner per leaf..

WIP limit Work-in-progress limit — the Kanban rule that turns a to-do list into flow..

Startup Gallery: Six Ventures to Dissect

Six real ventures from the regional ecosystem — including one founded by the author of this practicum — that the chapters keep referring to. Each profile names where the venture plugs into the practicum and ends with the question we actually use in class. Treat them as sparring partners: read the profile, then argue with it using the templates from Appendix A.

BabyFM — one technology, two markets

BabyFM is an AI-driven healthtech startup founded by Tamara Papić, developing systems for continuous temperature monitoring. A single core technology went to two very different customers at once: baby monitoring (medical and non-medical variants) and pet monitoring, through seven possible routes — veterinary clinics, distributors in the USA, Qatar and the UAE, pharmacies and private clinics in Serbia. With a small team and limited resources, “all seven” was never a strategy; the Option Prioritization Matrix (Opportunity $\times$ Feasibility, anchored 0–3–6–9–12 scores) forced the ranking that a spreadsheet of opinions never could [21].

Where it plugs in: Chapter 2 (option prioritization, beachhead logic); rings R2 (customer) and R8 (channels).

Discussion: when does one technology justify two markets — and what evidence would tell you to drop the second one?

UrbiGo — the smart garden

UrbiGo is a Belgrade hardware startup founded by Anja Varničić, building a “smart garden” for apartments: sensors that monitor plant health, lights that replace the sun, and a mobile app that coaches urban residents — people with no land, little time and less gardening knowledge — into growing herbs and mini vegetables at home. Recognized by the Innovation Fund in 2018 and developed further in the Katapult accelerator, with the European market (including an Amazon launch path) as the target [78].

Where it plugs in: Chapters 6 and 10 (hardware+software MVPs and their timelines); rings R4 (product) and R8 (going beyond a home market).

Discussion: what does an “MVP” even mean when your product has atoms, not just bits — and which assumption would you test before ordering the first enclosure?

Strawberry energy — the bench that is not the business

Founded in 2011 by Miloš Milisavljević, then an engineering student in Belgrade, Strawberry energy invented the solar-powered public charger (the Strawberry Tree) and then the Smart Bench: free device charging, Wi-Fi and environmental sensors in public spaces. The benches now stand in more than 90 cities across 30-plus countries on five continents [79]. The lesson is in the revenue model: citizens pay nothing; cities and sponsors pay for branding, connectivity and data — the bench is the medium, not the product. The company also crowdfunded a growth round, turning its user community into investors.

Where it plugs in: Chapter 4 (multi-sided and FREE patterns), Chapter 12 (crowdfunding as a funding rung); rings R3 (business model) and R7 (funding).

Discussion: who is the customer when the users never pay — and which ring carries the answer?

Anora — the smart glove for the blind

Anoris Technology, founded by Nikola Krstić, developed Anora: a wearable assistant that helps blind and partially sighted people orient in space (vibration that intensifies near obstacles), recognize colors and banknotes, tell the time, and send a location to guardians via a panic button. Supported by the Innovation Fund's Early Development Programme with EU funds, the team tested with hundreds of end users and pursued a deliberate model: the glove reaches end users free of charge, paid for by responsible businesses, aid organizations and government institutions [80].

Where it plugs in: Chapter 3 (persona co-design with the Association of the Blind), Chapter 12 (grant funding as evidence-matched capital); rings R2 (customer) and R7 (financials).

Discussion: user, payer and decision-maker are three different actors here — map all three onto the B2B profile of Chapter 3 before you read our answer.

Inclusio — access as a service

Inclusio, founded by Bojan Martinović — visually impaired himself — built a system that lets people with disabilities signal their arrival in front of a public facility (bank, shop, post office) while informing employees about the specific assistance needed. The idea dates to 2017, the first prototype was tested with a focus group of people with disabilities in 2018, and the finished app reached its first institutional partners in 2023–2024, supported by the Innovation Fund, the Katapult accelerator and the EBRD Star Venture programme [81].

Where it plugs in: Chapter 3 (designing *with* a sensitive user group, not *for* them), Chapter 14 (partnership-driven scaling); rings R2 (customer) and R8 (ecosystem).

Discussion: the founder's lived experience was the original insight — what does that replace, and what can it never replace, in your market research?

CodeKido — security products from Niš

CodeKido is a cybersecurity company based in Niš that develops its own security products, founded and led by Gorana Radovanović, founder and CEO of CodeKido Engineering, named Engineer of the Year 2026 — a useful reminder that a venture from this region does not have to choose between being a local services shop and a Silicon Valley fantasy. Selling security means selling trust: the buyer is a professional, the sales cycle is long, and the product is expected to survive a level of scrutiny that most student MVPs never face.

Where it plugs in: Chapter 3 (a value proposition built on trust rather than features), Chapter 9 (compliance and IP as entry tickets to the market), Chapter 14 (selling a product from Niš to the world); rings R2 (customer) and R7 (financials — surviving long procurement cycles).

Discussion: in B2B security, the end user and the person who signs the contract are rarely the same — which ring does that complicate the most, and how do your discovery experiments change when the buyer never touches the product?

Using the gallery

For any profile, run one template against it: a VPC for UrbiGo, a BMC for Strawberry energy, a risk table for Anora, a funding ladder for Inclusio. One page, one template, defended in ten minutes — that is a standard exam warmup in this course.

A Student Business Plan, Condensed: ZenTutor

This appendix shows what the capstone document (Template 13) looks like when a real student team fills it. The venture is ZenTutor, whose artifacts you have already met in Chapters 1–4; the numbers are the team’s working estimates from Chapter 2, kept deliberately rough — a business plan is a set of assumptions with consequences, not a prophecy. Each section header names the chapter and template that feed it. Use it as a floor, not a ceiling: your capstone should be this complete, and more honest.

1. Executive summary

written last; stands alone

ZenTutor is a tutoring marketplace and mobile app connecting students aged 12–18 and their parents with verified tutors for exam preparation, easy scheduling and group sessions. Private tutoring in Serbia is a recurring, painful, poorly organized spend: finding a trusted tutor runs on word of mouth, scheduling runs on messages, and prices are opaque. ZenTutor charges a commission per booked session, sells subscription learning packages and group-session seats, and licenses the platform to schools. Year-one goal: 900 active families in four cities, EUR 300 K marketplace revenue, and one signed B2B pilot with a private academy. The ask: EUR 60,000 of pre-seed funding for fourteen months of runway (Section 9).

2. Problem & opportunity

Ch. 2, Template 2a

Three pains, all verified in the team's PMR interviews: (a) finding a *trusted* tutor is luck, not process; (b) scheduling across school, sport and tutor calendars is chaos; (c) quality tutoring is expensive, with no package or group logic. Market sizing (assumptions in Chapter 2): TAM $\approx$ EUR 50 M/year of private tutoring spend in Serbia; SAM $\approx$ EUR 9.5 M (urban, mobile-first, exam-prep subjects); SOM $\approx$ EUR 300 K within three years — 3% of SAM, cross-checked bottom-up: 900 families $\times$ EUR 28/month $\times$ 12.

3. Solution & product

Ch. 3 (Template 3b), Ch. 6 (6a, 6b)

The app holds verified tutor profiles, 1-on-1 and group sessions, personalized learning packages, and scheduling with progress tracking. The Value Proposition Canvas (Figure 3.3) maps every reliever to a named pain — most importantly, *vetted, verified tutors* against *hard to find trusted tutors*. MVP: marketplace with manual onboarding and chat-based scheduling (no AI matching in v1); the riskiest assumption — *parents will book a tutor they have never met based on in-app verification* — is tested first with a concierge pilot in one city.

4. Business model

Ch. 4, Template 4a

Four revenue streams: commission on each booked session (18%), subscription learning packages, group-session fees, and B2B licenses for schools and academies. Key partners: verified independent tutors, schools and universities, payment processing, education influencers. Key costs: platform development and maintenance, tutor payouts, marketing, support. The canvas (the team's BMC from June 2025) reads as one system: the same verified-tutor network is simultaneously a key resource, the core of the value proposition, and the main onboarding cost.

5. Market & competition

Ch. 7, Templates 7a, 7b

Competitors: generic tutoring platforms (global, impersonal, online-only), local agencies (trusted, but offline and slow), and the grey market of classified ads (cheap, unverified). ZenTutor’s positioning: *verified like an agency, convenient like a platform, priced like a group*. The SWOT’s decisive pair: strength — local, in-person network effects per city; threat — a global platform copying the group-session feature; defense — school partnerships and tutor exclusivity clauses in launch cities.

6. Go-to-market

Ch. 2 (beachhead), Ch. 11 (CAC)

Beachhead: Belgrade, math and physics for final-year high-school students (exam pressure is highest, parents decide fastest). First 10 customers: recruited through the team’s own school networks and one partner academy. Acquisition channels: parent-focused social campaigns, school partnerships, referral discounts (“bring a classmate, both get a group session”). Target CAC below EUR 25 per active family against a first-year value per family of roughly EUR 110 of marketplace revenue.

7. Team & organization

Ch. 8 (8a, 8b), Ch. 9 (9a, 9b)

Four co-founders: product (backend), marketplace operations (tutor vetting), growth (partnerships and social), and finance/ops. The charter names decision rights and a rotating delivery owner per sprint — the lesson borrowed from Creative Wave’s near-miss (Chapter 8). Belbin gap acknowledged: no strong Implementer; mitigation is the weekly retro with a written task board, not a hire the team cannot afford. Legal: DOO (LLC), founders’ agreement with four-year vesting; tutor contracts include a non-circumvention clause; the brand is filed for trademark.

8. Execution plan

Ch. 10 (10a, 10b, 10c)

Twelve-week runway to pilot: weeks 1–3 concierge pilot with 30 families (manual everything); weeks 4–8 marketplace MVP (profiles, booking, payments); weeks 9–12 group sessions and the first school pilot. Milestones:

W3 pilot gate ($\geq 60\%$ repeat bookings or rethink), W8 MVP live, W12 first B2B letter of intent. Top risks from the register: tutor supply cold-start (mitigation: campus ambassador program), payment-provider onboarding delay (fallback: bank transfer + manual reconciliation), scope creep toward AI matching (deferred to v2 by founder vote).

9. Financials

Ch. 11 (11a, 11b), Ch. 12 (12a)

Line	Year-1 working number
Marketplace revenue	EUR 300 K gross bookings $\rightarrow \approx$ EUR 54 K commission (18%) + packages
COGS	tutor payouts (from bookings), payment fees $\approx 3\%$
Operating budget	EUR 60 K: development 40%, marketing 30%, vetting & support 20%, legal/admin 10%
Unit economics	CAC target < EUR 25; revenue per active family year 1 $\approx$ EUR 110
Break-even	≈ 550 active families sustained on commission + packages
The ask	EUR 60,000 for 14 months: MVP, vetting pipeline, two city launches

Three statements exist in the team’s spreadsheet; the plan shows the logic, the spreadsheet shows the arithmetic. Burn is front-loaded into development; marketing spend unlocks only after the W3 pilot gate — the budget follows the evidence gates of Chapter 5.

10. Appendices

Artifact checklist: filled Templates 1, 2a, 2b, 3a, 3b, 4a, 7a, 8a, 10a–c, 11a, 11b, 12a; experiment log (concierge pilot results); the rings map with evidence per ring; LOI draft from the partner academy.

How to read this plan

Notice what is *not* here: no invented precision (“EUR 300,000” is a target with shown assumptions, not a forecast with decimals), no feature list without a pain, no hiring plan the budget cannot carry. The plan is the rings map turned into prose — every section cites the artifact that proves it. That is the standard for your capstone.

The Rings Volvelle: A Paper Instrument

The Concentric Rings model also works as a physical instrument — a *volvelle*, the same idea as a wooden perpetual calendar: discs rotate against each other, and a fixed window reads whatever lines up. This appendix gives you a cut-out version of the eight rings. Each disc carries the elements of one ring printed around its rim; stacked on a split pin, the discs turn independently, so a shift in one ring visibly changes its alignment with all the others.

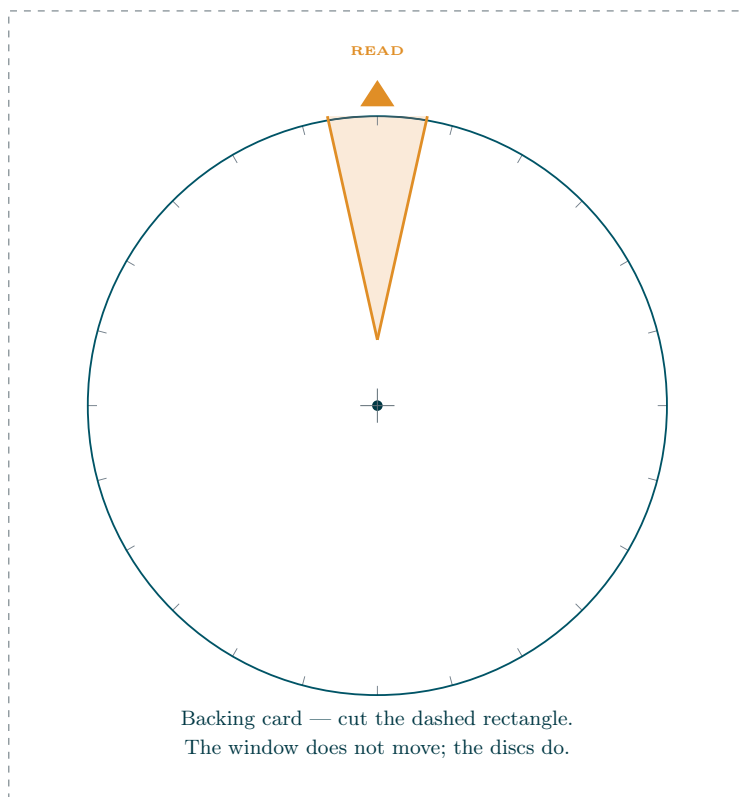
Guided steps

Assembly

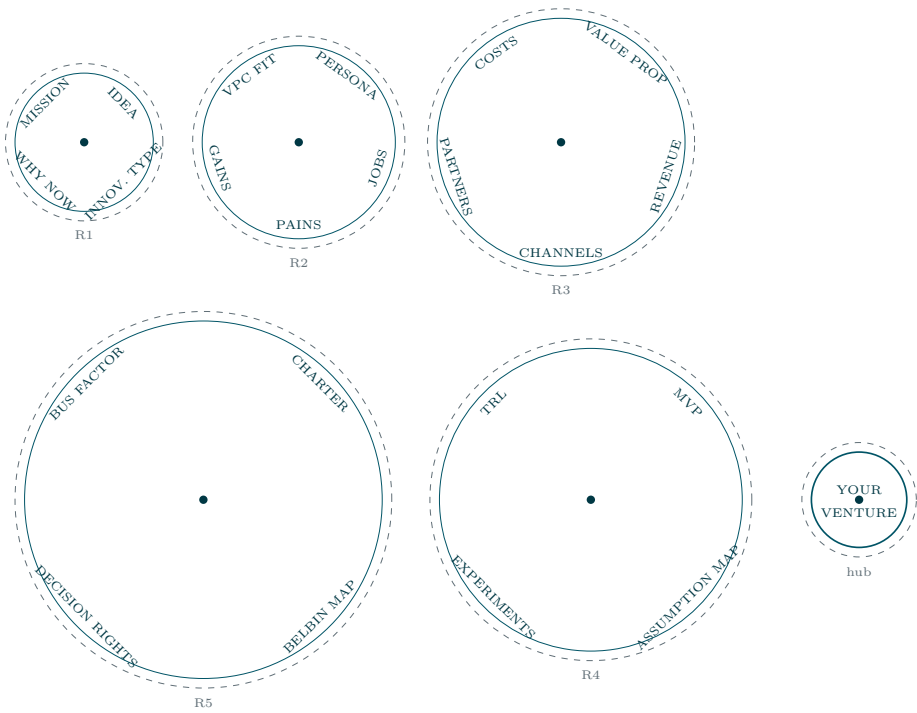
1. Cut out the backing card (page 210) and the eight ring discs plus the hub disc (pages 211–212), following the dashed circles.
2. Pierce the center mark on every disc and on the backing card.
3. Stack the discs on the backing card in order: R8 (largest) at the bottom, R1 (smallest) on top, hub last. Push a split pin (brad) through the center and open its legs loosely, so the discs still turn.
4. Rotate each disc until the element you are currently working on sits inside the amber window at the top. Read the whole column: whatever is in the window is one vertical slice through your venture.

Systems Thinking Lab — The Concentric Rings

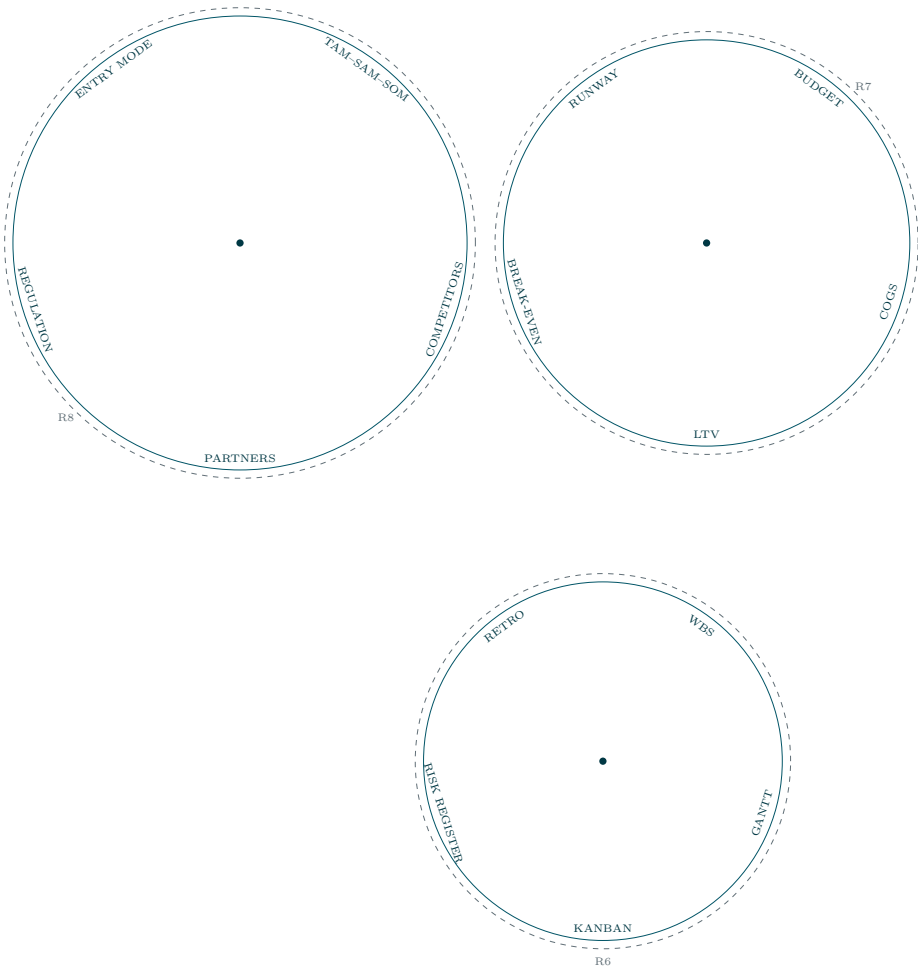
Using the volvelle in the lab Fire any shift event from the Systems Thinking Labs in this book (a competitor launches, a channel doubles its prices, a key member leaves. . .). Turn the affected ring to the element under pressure, then walk the column in the window: which elements above and below it must now move? Physically rotating the discs makes the ripple effect hard to ignore — that is exactly why the instrument exists.



Pin hole at the center. The amber window reads the aligned column: one element per ring, R1 in the middle to R8 at the rim.



Discs R1–R5 and the hub. Write your venture’s name on the hub before pinning. Stack order: R8 at the bottom, R1 on top, hub last.



Discs R6–R8. When all eight are pinned, every rotation of a single disc re-reads the whole venture — one column in the window at a time.

Declaration on the Use of Artificial Intelligence

In the preparation of this practicum, AI-assisted tools were used for language editing, consistency checking of the manuscript, and assistance in drafting the code that draws the TikZ figures and templates. Every chapter, example, calculation, and citation was reviewed, verified, and approved by the authors, who takes full responsibility for the content of this book.

AI did not create any tasks for the students. Every student item that is replicated in this book is a working example, and the appendices are real team deliverables used with the teams' written permission, that have only been redesigned as necessary for legibility.

The same standard applies to all readers and students that are reading this book. Throughout this practicum you will meet the *AI angle* boxes, which show where AI tools genuinely help (market scans, persona drafts, experiment copy, code sketches) and where they must not decide for you (problem-solution fit, evidence, numbers you sign). When you use AI tools in your own deliverables, you are expected to follow your institution's academic integrity rules: disclose which tools you used and for what, verify every fact and figure they produce, and never present generated text as evidence. In this practicum, an AI output is a draft — a validated learning card, a filled template, or a defended number is the deliverable.

Bibliography

- [1] P. M. Senge, *The Fifth Discipline: The Art and Practice of the Learning Organization*. New York, NY, USA: Doubleday, 2006.
- [2] D. H. Meadows, *Thinking in Systems: A Primer*. White River Junction, VT, USA: Chelsea Green Publishing, 2008.
- [3] M. J. Wheatley, *Leadership and the New Science: Discovering Order in a Chaotic World*. San Francisco, CA, USA: Berrett-Koehler, 2006.
- [4] A. Sastry and K. Penn, *Fail Better: Design Smart Mistakes and Succeed Sooner*. Harvard Business Review Press, 2014.
- [5] M. Dodgson, D. M. Gann, and A. Salter, *The Management of Technological Innovation: Strategy and Practice*. Oxford, UK: Oxford University Press, 2008.
- [6] A. Osterwalder and Y. Pigneur, *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. Hoboken, NJ, USA: Wiley, 2010.
- [7] B. Aulet, *Disciplined Entrepreneurship: 24 Steps to a Successful Launch*. Hoboken, NJ, USA: Wiley, 2013.
- [8] P. Cheek, *Disciplined Entrepreneurship Startup Tactics: 15 Tactics to Turn Your Business Plan into a Business*. Hoboken, NJ, USA: Wiley, 2024.
- [9] E. Ries, *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. New York, NY, USA: Crown Business, 2011.
- [10] P. F. Drucker, *Management: Tasks, Responsibilities, Practices*. New Brunswick, NJ, USA: Transaction Publishers, 2007.

- [11] S. Altman *et al.*, “How to start a startup (stanford cs183b lecture series),” <http://startupclass.samaltman.com/>, 2014.
- [12] T. Papić, “Ideas and how to start a startup,” Course lecture, Entrepreneurship in Software Engineering, Singidunum University, 2022.
- [13] T. Papić, “Opportunity analysis,” Course lecture, Entrepreneurship in Software Engineering, Singidunum University, 2022.
- [14] T. Papić, “Entrepreneurship in software engineering — course introduction,” Course lecture, Singidunum University, 2022.
- [15] R. Avlijaš and G. Avlijaš, *Preduzetništvo*. Beograd: Univerzitet Singidunum, 2021.
- [16] S. Kraus, C. Palmer, N. Kailer, F. L. Kallinger, and J. Spitzer, “Digital entrepreneurship: A research agenda on new business models for the twenty-first century,” *International Journal of Entrepreneurial Behavior & Research*, vol. 27, no. 2, pp. 336–361, 2021.
- [17] T. N. Duening, R. A. Hisrich, and M. A. Lechter, *Technology Entrepreneurship: Taking Innovation to the Marketplace*, 3rd ed. Amsterdam: Elsevier, 2020.
- [18] T. Papić, “Business plan,” Course lecture, Entrepreneurship in Software Engineering, Singidunum University, 2022.
- [19] V. Papakonstantinou, “Disciplined entrepreneurship workshop (adapted from mit “new enterprises”, b. aulet),” StartSmart SEE workshop materials, 2025.
- [20] T. Papić, “Business model canvas,” Course lecture, Entrepreneurship in Software Engineering, Singidunum University, 2022.
- [21] N. Sullivan, “Technology innovation workshop materials,” Institute for Manufacturing (IfM), University of Cambridge, 2025.
- [22] T. Luckey and J. Phillips, *Software Project Management For Dummies*. Hoboken, NJ, USA: Wiley, 2006.

- [23] R. D. Hisrich, M. P. Peters, and D. A. Shepherd, *Entrepreneurship*, 10th ed. New York: McGraw-Hill Education, 2016.
- [24] T. Brown, *Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation*. HarperBusiness, 2009.
- [25] A. Osterwalder, Y. Pigneur, G. Bernarda, and A. Smith, *Value Proposition Design: How to Create Products and Services Customers Want*. Hoboken, NJ, USA: Wiley, 2014.
- [26] S. Blank and B. Dorf, *The Startup Owner's Manual: The Step-by-Step Guide for Building a Great Company*. K&S Ranch, 2012.
- [27] H. Chesbrough, *Open Innovation: The New Imperative for Creating and Profiting from Technology*. Boston, MA, USA: Harvard Business School Press, 2003.
- [28] E. Enkel, O. Gassmann, and H. Chesbrough, "Open R&D and open innovation: Exploring the phenomenon," *R&D Management*, vol. 39, no. 4, pp. 311–316, 2009.
- [29] M. Holgersson, L. Dahlander, H. Chesbrough, and M. L. A. M. Bogers, "Open innovation in the age of AI," *California Management Review*, vol. 67, no. 1, pp. 5–20, 2024.
- [30] D. J. Teece, "Business models and dynamic capabilities," *Long Range Planning*, vol. 51, no. 1, pp. 40–49, 2018.
- [31] C. M. Christensen, *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail*. Boston, MA, USA: Harvard Business School Press, 1997.
- [32] R. G. Cooper, *Winning at New Products: Accelerating the Process from Idea to Launch*, 3rd ed. Perseus Publishing, 2001.
- [33] A. Maurya, *Running Lean: Iterate from Plan A to a Plan That Works*. O'Reilly, 2012.
- [34] E. M. Rogers, *Diffusion of Innovations*, 5th ed. Free Press, 2003.
- [35] G. A. Moore, *Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers*. HarperBusiness, 1991.

- [36] J. Doerr, *Measure What Matters: OKRs — The Simple Idea that Drives 10x Growth*. Portfolio/Penguin, 2018.
- [37] R. Whittington, *Digital Innovation and Entrepreneurship*. Cambridge: Cambridge University Press, 2018.
- [38] T. Papić, B. Gutić, N. Pantelić, and N. Petrović, “The role of open innovation in enhancing managerial,” in *Proceedings of the International Scientific Conference — Sinteza 2025*, Belgrade, Serbia, 2025, pp. 396–401. [Online]. Available: <https://doi.org/10.15308/sinteza-2025-396-401>
- [39] European Commission, “Technology readiness levels (trl): General annex g, horizon 2020 work programme,” 2014.
- [40] W. C. Kim and R. Mauborgne, *Blue Ocean Strategy: How to Create Uncontested Market Space and Make the Competition Irrelevant*. Boston, MA, USA: Harvard Business School Press, 2005.
- [41] S. P. Robbins and T. A. Judge, *Organizational Behavior*. Upper Saddle River, NJ, USA: Prentice Hall, 2012.
- [42] Harvard Business Review, *The Harvard Business Review Manager’s Handbook: The 17 Skills Leaders Need to Stand Out*. Boston, MA, USA: Harvard Business Review Press, 2017.
- [43] R. Dalio, *Principles: Life and Work*. Simon & Schuster, 2017.
- [44] R. M. Belbin, *Team Roles at Work*, 2nd ed. Abingdon, UK: Routledge, 2010.
- [45] M. Aleksić and P. Dakić, “The role of transformational leadership in open-source software development: Insights from survey data on enhancing organizational commitment and innovation in ai-driven business environments,” *SCIENCE International Journal*, vol. 4, no. 4, pp. 107–112, 2025. [Online]. Available: <https://doi.org/10.35120/sciencej0404107a>
- [46] M. Skelton and M. Pais, *Team Topologies: Organizing Business and Technology Teams for Fast Flow*. Portland, OR: IT Revolution Press, 2019.

- [47] L. Bock, *Work Rules! Insights from Inside Google that Will Transform How You Live and Lead*. New York: Twelve, 2015.
- [48] N. Forsgren, M.-A. Storey, C. Maddila, T. Zimmermann, B. Houck, and J. Butler, “The SPACE of developer productivity,” *ACM Queue*, vol. 19, no. 1, pp. 20–48, 2021.
- [49] T. Papić, “Legal forms and intellectual property,” Course lecture, Entrepreneurship in Software Engineering, Singidunum University, 2022.
- [50] Serbian Business Registers Agency, “Business registration portal (e-registration),” <https://www.apr.gov.rs>, 2025.
- [51] World Intellectual Property Organization, “Intellectual property for business — guides and tools,” <https://www.wipo.int>, 2025.
- [52] Intellectual Property Office of the Republic of Serbia, “Zavod za intelektualnu svojinu — registers and guides,” <https://www.zis.gov.rs>, 2025.
- [53] P. Dakić, “Software compliance in various industries using ci/cd, dynamic microservices, and containers,” *Open Computer Science*, vol. 14, no. 1, 2024. [Online]. Available: <https://doi.org/10.1515/comp-2024-0013>
- [54] G. Siciliano, *Finance for Non-Financial Managers*. New York, NY, USA: McGraw-Hill, 2003.
- [55] N. Forsgren, J. Humble, and G. Kim, *Accelerate: The Science of Lean Software and DevOps*. Portland, OR: IT Revolution Press, 2018.
- [56] M. L. Prasetyo, R. A. Peranginangin, N. Martinovic, M. Ichsan, and H. Wicaksono, “Artificial intelligence in open innovation project management: A systematic literature review on technologies, applications, and integration requirements,” *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 11, no. 1, p. 100445, 2025.
- [57] T. Papić, “Financial management for non-financial managers (course

- lecture),” 2025, singidunum University, Management for Software and Data Engineering.
- [58] S. Milojković, Ž. Kljajić, and P. Dakić, “Z score and validoai: An explainable ai framework for payroll analytics with statistical anomaly detection in serbian small and medium enterprises,” *EMC Review – Časopis Za Ekonomiju – APEIRON*, vol. 31, no. 2, 2025. [Online]. Available: <https://doi.org/10.7251/EMC2502379M>
 - [59] M. D. S. Harris, *The Business Value of Software*. Boca Raton: CRC Press / Auerbach Publications, 2018.
 - [60] T. Papić, “Entrepreneurship development support (course lecture),” 2025, singidunum University, Management for Software and Data Engineering.
 - [61] G. C. Kane, A. N. Phillips, J. R. Copulsky, and G. R. Andrus, *The Technology Fallacy: How People Are the Real Key to Digital Transformation*. Cambridge, MA: MIT Press, 2019.
 - [62] M. Iansiti and K. R. Lakhani, *Competing in the Age of AI: Strategy and Leadership when Algorithms and Networks Run the World*. Boston: Harvard Business Review Press, 2020.
 - [63] D. Sjödin, V. Parida, M. Palmié, and J. Wincent, “How AI capabilities enable business model innovation: Scaling AI through co-evolutionary processes and feedback loops,” *Journal of Business Research*, vol. 134, pp. 574–587, 2021.
 - [64] P. Jorzik, S. P. Klein, D. K. Kanbach, and S. Kraus, “AI-driven business model innovation: A systematic review and research agenda,” *Journal of Business Research*, vol. 182, 2024.
 - [65] European Parliament and Council, “Regulation (eu) 2024/1689 laying down harmonised rules on artificial intelligence (ai act),” 2024, official Journal of the European Union, L series, 12 July 2024.
 - [66] T. H. Davenport and R. Ronanki, “Artificial intelligence for the real world,” *Harvard Business Review*, vol. 96, no. 1, pp. 108–116, 2018.
 - [67] S. Ransbotham, S. Khodabandeh, D. Kiron, B. LaFountain, and

- M. Chung, “The cultural benefits of artificial intelligence in the enterprise,” MIT Sloan Management Review & Boston Consulting Group research report, 2021.
- [68] A. McAfee and E. Brynjolfsson, “Big data: The management revolution,” *Harvard Business Review*, vol. 90, no. 10, pp. 60–68, 2012.
- [69] G. Vial, “Understanding digital transformation: A review and a research agenda,” *The Journal of Strategic Information Systems*, vol. 28, no. 2, pp. 118–144, 2019.
- [70] R. Costa Climent, D. M. Haftor, and M. W. Staniewski, “AI-enabled business models for competitive advantage,” *Journal of Innovation & Knowledge*, vol. 9, no. 3, p. 100532, 2024.
- [71] D. L. Roberts and M. Candi, “Artificial intelligence and innovation management: Charting the evolving landscape,” *Technovation*, vol. 136, 2024.
- [72] A. Kuzior, M. Sira, and P. Brożek, “Use of artificial intelligence in terms of open innovation process and management,” *Sustainability*, vol. 15, no. 9, p. 7205, 2023.
- [73] B. Y. Kassa and E. K. Worku, “The impact of artificial intelligence on organizational performance: The mediating role of employee productivity,” *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 11, no. 1, p. 100474, 2025.
- [74] T. Papić, “Artificial intelligence as a strategic tool for sme growth: From process automation to knowledge intelligence,” *SCIENCE International Journal*, vol. 4, no. 4, pp. 113–118, 2025. [Online]. Available: <https://doi.org/10.35120/sciencej0404113p>
- [75] T. Papić, “Entrepreneur exiting venture (course lecture),” 2025, singidunum University, Entrepreneurship in Software and Data Engineering.
- [76] B. Fitzgerald, K.-J. Stol, S. Minör, and H. Cosmo, *Scaling a Software Business: The Digitalization Journey*. Cham: Springer, 2017.
- [77] R. van Cann, S. Jansen, and S. Brinkkemper, *Software Business*

- Start-up Memories: Key Decisions in Success Stories.* Basingstoke: Palgrave Macmillan, 2013.
- [78] Innovation Fund Serbia, “Urbigo: A smart garden in your home,” <https://www.inovacionifond.rs/en/news/urbigo-a-smart-garden-in-your-home>, 2022.
- [79] Strawberry energy, “Why us? smart and solar powered urban furniture,” <https://strawberryye.com/why-us/>, 2025.
- [80] Innovation Fund Serbia, “Anora’s smart glove changes the lives of blind people,” <http://www.inovacionifond.rs/en/news/anoras-smart-glove-changes-the-lives-of-blind-people>, 2021.
- [81] Innovation Fund Serbia, “Inclusio: For a world without barriers,” <https://www.inovacionifond.rs/en/news/inclusio-for-a-world-without-barriers>, 2025.
- [82] I. A. Vajs, G. S. Kvascev, T. M. Papić, and M. M. Janković, “Eye-tracking image encoding: Autoencoders for the crossing of language boundaries in developmental dyslexia detection,” *IEEE Access*, vol. 11, pp. 3024–3033, 2023. [Online]. Available: <https://doi.org/10.1109/ACCESS.2023.3234438>
- [83] I. Vajs, T. Papić, V. Ković, A. M. Savić, and M. M. Janković, “Accessible dyslexia detection with real-time reading feedback through robust interpretable eye-tracking features,” *Brain Sciences*, vol. 13, no. 3, p. 405, 2023. [Online]. Available: <https://doi.org/10.3390/brainsci13030405>
- [84] I. Vajs, V. Ković, T. Papić, A. M. Savić, and M. M. Janković, “Spatiotemporal eye-tracking feature set for improved recognition of dyslexic reading patterns in children,” *Sensors*, vol. 22, no. 13, p. 4900, 2022. [Online]. Available: <https://doi.org/10.3390/s22134900>
- [85] T. Jakovljević, M. M. Janković, A. M. Savić, I. Soldatović, I. Mačužić, T. Jere Jakulin, and G. Papa, “The effect of colour on reading performance in children, measured by a sensor hub: From the perspective of gender,” *PLOS*

- ONE*, vol. 16, no. 6, p. e0252622, 2021. [Online]. Available: <https://doi.org/10.1371/journal.pone.0252622>
- [86] T. Jakovljević, M. M. Janković, A. M. Savić, I. Soldatović, P. Todorović, T. Jere Jakulin, and G. Papa, “The sensor hub for detecting the developmental characteristics in reading in children on a white versus coloured background,” *Sensors*, vol. 21, no. 2, p. 406, 2021. [Online]. Available: <https://doi.org/10.3390/s21020406>
- [87] T. Papić and T. Jere Jakulin, “System thinking for business model innovation,” in *Proceedings of the 14th International Quality Conference*, 2023.
- [88] T. Papić, P. Dakić, and B. Gutić, “Leveraging open innovation and systems thinking to enhance the success of ai-driven healthtech ventures: A case study of babyfm in south-east europe,” in *Advances in ICT Research in the Balkans*. Springer, 2026, pp. 333–346. [Online]. Available: https://doi.org/10.1007/978-3-032-24513-7_28
- [89] T. Papić, P. Dakić, and B. Gutić, “Software quality and compliance in intelligent health monitoring systems: A case study of baby fm,” in *Proceedings of the Twelfth Workshop on Software Quality Analysis, Monitoring, Improvement, and Applications (SQAMIA 2025)*, ser. CEUR Workshop Proceedings, vol. 4077, Maribor, Slovenia, 2025. [Online]. Available: <https://ceur-ws.org/Vol-4077/paper7.pdf>
- [90] T. Papić, A. Mihajlović, and J. Gajić, “Advanced technologies as a framework for sustainable marketing campaigns (ai application in neuromarketing),” in *Proceedings of the International Scientific Conference — Sinteza 2023*. Belgrade, Serbia: Singidunum University, 2023, pp. 180–184. [Online]. Available: <https://doi.org/10.15308/sinteza-2023-180-184>
- [91] A. Mihajlović, J. Gajić, and T. Papić, “Analysis of the cost-effectiveness of the university instagram marketing campaign using a/b testing,” in *Proceedings of the International Scientific Conference — Sinteza 2024*. Belgrade, Serbia: Singidunum University, 2024, pp. 266–273. [Online]. Available: <https://doi.org/10.15308/sinteza-2024-266-273>

- [92] P. Dakić, T. Heričko, Ž. Kljajić, and V. Todorović, “The economics of ai-powered call center development using chatgpt for the needs of an automotive retail business,” in *Proceedings of the International Scientific Conference — Sinteza 2025*, Belgrade, Serbia, 2025, pp. 382–388. [Online]. Available: <https://doi.org/10.15308/sinteza-2025-382-388>
- [93] S. Milojković, P. Dakić, T. Heričko, M. Aleksić, and J. Lang, “Msq methodology in ai-driven analysis of assets and liabilities: A case study of the validoai system,” in *Proceedings of the Twelfth Workshop on Software Quality Analysis, Monitoring, Improvement, and Applications (SQAMIA 2025)*, ser. CEUR Workshop Proceedings, vol. 4077, Maribor, Slovenia, 2025. [Online]. Available: <https://ceur-ws.org/Vol-4077/paper13.pdf>
- [94] F. I. Ali, T. E. Ali, W. Aljamal, P. Dakić, and A. D. Zoltan, “Ai-powered quiz generation for learning management systems: A full-stack implementation for canvas lms,” in *Proceedings of the Third Workshop on Co-Creating New Ways of Information Systems Education (NWISEd 2025)*, ser. CEUR Workshop Proceedings, vol. 4076, Maribor, Slovenia, 2025. [Online]. Available: <https://ceur-ws.org/Vol-4076/short3.pdf>
- [95] F. Cák and P. Dakić, “Configuration tool for ci/cd pipelines and react web apps,” in *2024 14th International Conference on Advanced Computer Information Technologies (ACIT)*. České Budějovice, Czech Republic: IEEE, 2024, pp. 586–591. [Online]. Available: <https://doi.org/10.1109/ACIT62333.2024.10712482>
- [96] P. Dakić, “Importance of knowledge management for ci/cd and security in autonomous vehicles systems,” *JITA – Journal of Information Technology and Applications (Banja Luka) – APEIRON*, vol. 24, no. 1, 2024. [Online]. Available: <https://doi.org/10.7251/jit2401007d>
- [97] T. E. Ali, F. I. Ali, P. Dakić, and A. D. Zoltan, “Trends, prospects, challenges, and security in the healthcare internet of things,” *Computing*, vol. 107, no. 1, p. 28, 2024. [Online]. Available: <https://doi.org/10.1007/s00607-024-01352-4>

- [98] P. Dakić, M. Zivkovic, L. Jovanovic, N. Bacanin, M. Antonijevic, J. Kaljevic, and V. Simic, "Intrusion detection using metaheuristic optimization within iot/iiot systems and software of autonomous vehicles," *Scientific Reports*, vol. 14, no. 1, p. 22884, 2024. [Online]. Available: <https://doi.org/10.1038/s41598-024-73932-5>
- [99] M. D. Haiderzai, P. Dakić, I. Stupavský, M. Aleksić, and V. Todorović, "Pattern shared vision refinement for enhancing collaboration and decision-making in government software projects," *Electronics*, vol. 14, no. 2, p. 334, 2025. [Online]. Available: <https://doi.org/10.3390/electronics14020334>
- [100] I. Stupavský, P. Dakić, and V. Vranić, "The impact of fake news on traveling and antisocial behavior in online communities: Overview," *Applied Sciences*, vol. 13, no. 21, p. 11719, 2023. [Online]. Available: <https://doi.org/10.3390/app132111719>

Author Biographies



doc. dr Tamara Papić, PhD.

ORCID: 0000-0002-9666-7824

Tamara Papić is a docent at the Faculty of Technical Sciences, Singidunum University, Belgrade. She holds a PhD in Sensor Technologies from the Jožef Stefan International Postgraduate School, Ljubljana, Slovenia. Her current research interests include eye tracking and biometric sensor technologies for developmental dyslexia detection [82–86], neuromarketing, artificial intelligence for business and healthcare applications, and systems thinking for business model innovation [87].

Her recent publications include work on open innovation and systems thinking for AI-driven HealthTech ventures [88], artificial intelligence as a strategic tool for SME growth [74], software quality and compliance in intelligent health monitoring [89], neuromarketing and sustainable marketing campaigns [90], and cost-effectiveness analysis of university marketing campaigns [91].



doc. dr Mgr. Pavle Dakić, PhD.

ORCID: 0000-0003-3538-6284

Pavle Dakić is a docent at the Faculty of Informatics and Computing, Singidunum University, Belgrade, and a Researcher at the Faculty of Informatics and Information Technologies, Slovak University of Technology. He has a PhD in Electrical Engineering and Computing from Singidunum University, Belgrade, and a second PhD degree in Applied Informatics from the Faculty of Informatics and Information Technologies, Slovak University of Technology. His current research interests include computer networks, AI security, high-performance systems (HPC), Internet of Things (IoT), software development and testing, SEO/SEM, CI/CD, encryption, data security, and business applications of telecommunications technologies.

His recent publications cover AI-powered business tools (including ChatGPT call-centre economics [92], explainable AI for payroll and asset analytics [58, 93], and LMS quiz generation [94]), CI/CD and software-quality engineering [53, 95], IoT and autonomous-vehicle software compliance [96–98], and leadership in open-source and AI-driven software development [45, 99, 100].

CIP – Katalogizacija u publikaciji
Narodna biblioteka Srbije, Beograd

334.722(075.8)(0.034.2)
001.895(075.8)(0.034.2)

PAPIĆ, Tamara, 1989-

Entrepreneurship, innovation management and business skills in the it sector : a step-by-step innovation practicum / Tamara Papić, Pavle Dakić. - Belgrade : Singidunum University, 2026 (Belgrade : Singidunum University). - 1 elektronski optički disk (CD-ROM) : tekst, slika ; 12 cm

Slike autora. - Tiraž 10. - Author biographies. - Bibliografija.

ISBN 978-86-7912-869-0

1. Dakić, Pavle, 1990- [аутор]
а) Предузетништво б) Иновације

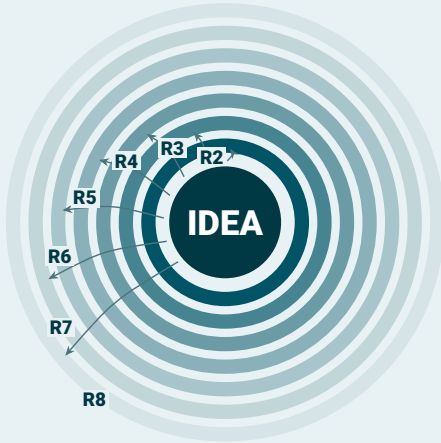
COBISS.SR-ID 200349193

Copyright © 2026 Singidunum University, Belgrade and its authors. All rights reserved. No part of this work covered by the copyright herein may be reproduced, transmitted, stored or used in any form or by any means graphic, electronic, or mechanical, including but not limited to photocopying, recording, scanning, digitizing, taping, Web distribution, information networks, or information storage and retrieval systems, without the prior written permission of the publisher.

Tamara Papić | Pavle Dakić

Entrepreneurship, Innovation Management and Business Skills in the IT Sector

A Step-by-Step Innovation Practicum



- R1** Idea & innovation type
- R2** Customer & value proposition
- R3** Business model
- R4** Product & assumptions
- R5** Team & organization
- R6** Project & risk
- R7** Financials
- R8** Market, partners & ecosystem

The Concentric Rings model visualizes the venture as a nested system. A change in any ring — idea, customer, value, or the venture itself — propagates through every other part of the system, so decisions must be made with the whole in mind.

From a raw idea to a defended business plan — this practicum takes IT students and technical founders through every stage of building a venture: recognizing opportunity, researching the market, designing a defensible business model, assembling and leading a founding team, and turning a plan into funded, growing traction. Each chapter pairs concise theory with guided steps, worked examples, tasks, and an *AI angle* that shows how modern tools accelerate the entrepreneurial process. Organized around a system of concentric rings — idea, customer, value, and the venture that carries them — the book equips readers to test assumptions with evidence, design with discipline, and pitch with confidence.

ISBN 978-86-7912-869-0

