# Managing DIGITAL

## Concepts and Practices

Charles T. Betz

THE Open GROUP

# Managing Digital

## *Concepts and Practices*

Charles T. Betz

# Table of Contents

Front Matter 1
Praise for this book 1
The Open Group Press 2
Copyright 3
Epigraph 5
Dedication 5
Foreword 6
Preface by original author 7
Introduction for instructors and trainers 10
The IT industry and the rise of digital 10
A process of emergence 14
Labs 17
Introduction for the student 19
This book’s structure 21
Emergence means formalization 23
Assumptions about the reader 24
Part I — Founder 25
1. Digital Value 27
1.1. Introduction 27
1.1.1. Chapter outline 27
1.1.2. Learning objectives for this chapter 27
1.2. What is digital value? 28
1.2.1. A digital value scenario 28
1.2.2. Various forms of digital value 30
1.3. Defining digital 31
1.3.1. What is IT, anyways? 31
1.3.2. IT and digital transformation 32
1.3.3. Defining “IT” 33
1.3.4. IT versus "Digital" 34
1.4. Digital services, systems, and applications 35
1.4.1. Inside a digital service 35
1.4.2. What versus how 37
1.5. The digital service lifecycle 39
1.6. Defining consumer, customer, and sponsor 41
1.7. Understanding digital context 42
1.7.1. Market-facing, supporting, back office 43
1.7.2. Diffusion theory and other approaches 43
1.7.3. Business discovery approaches 45

Business model canvas 46
Business case analysis 47
Lean Startup 48
1.8. Conclusion 49
1.8.1. Discussion questions 50
1.8.2. Research & practice 50
1.8.3. Further reading 50
2. Digital infrastructure 51
2.1. Introduction 51
2.1.1. Chapter summary 51
2.1.2. Learning objectives 52
2.2. Infrastructure overview 52
2.2.1. What is infrastructure? 53
2.2.2. Basic IT infrastructure concepts 54
2.3. Choosing infrastructure 58
2.4. From “physical” compute to cloud 60
2.4.1. Virtualization 61
2.4.2. Why is virtualization important? 63
2.4.3. Virtualization, managed services, and cloud 64
2.4.4. Containers and looking ahead 67
2.5. Infrastructure as code 67
2.5.1. Why infrastructure as code? 68
2.5.2. A simple infrastructure as code example 69
2.6. Configuration management: the basics 72
2.6.1. What is version control? 72
Source control 73
The “commit” concept 77
2.6.2. Package management 78
2.6.3. Deployment management 79
Deployment basics 80
Imperative and declarative 80
2.7. Topics in IT infrastructure 81
2.7.1. Configuration management, version control, and metadata 81
2.8. Conclusion 82
2.8.1. Discussion questions 83
2.8.2. Research & practice 83
2.8.3. Further reading 83
3. Application Delivery 85
3.1. Introduction 85
3.1.1. Chapter outline 85
3.1.2. Learning objectives 86

3.2. Basics of applications and their development 86
3.2.1. Defining “application” 86
3.2.2. History of applications and application software 86
3.2.3. Applications and infrastructure: the old way 87
3.2.4. Applications and infrastructure today 88
3.3. From waterfall to Agile 89
3.4. The DevOps challenge 94
3.5. Describing system intent 97
3.6. Test-driven development and refactoring 98
3.6.1. Test-driven development 99
3.6.2. Refactoring 101
3.7. Continuous integration 102
3.7.1. Version control, again: branching and merging 102
3.7.2. Build choreography 106
3.8. Releasing software 107
3.8.1. Continuous deployment 107
3.8.2. The concept of “release” 108
3.9. Application development topics 109
3.9.1. Application architecture 109
3.9.2. Applications and project management 109
3.10. Conclusion 110
3.10.1. Discussion questions 110
3.10.2. Research & practice 111
3.10.3. Further reading 111
Part I Conclusion 112
Part II — Team 113
Special section: Systems thinking and feedback 115
A brief introduction to feedback 116
What does systems thinking have to do with IT? 117
Reinforcing feedback: the special case investors want 119
Open *versus* closed-loop systems 120
OODA 121
The DevOps consensus as systems thinking 122
4. Product Management 124
4.1. Introduction 124
4.1.1. Chapter 4 outline 124
4.1.2. Chapter 4 learning objectives 125
4.2. Why product management? 125
4.2.1. The product vision 125
4.2.2. Defining product management 127
4.2.3. Process, project, and product management 128

4.2.4. Productization as a strategy at Amazon . . . 130
4.3. Organizing the product team . . . 131
4.3.1. The concept of collaboration . . . 132
4.3.2. Lean UX . . . 134
4.3.3. Scrum . . . 135
4.3.4. More on product team roles . . . 137
4.4. Product discovery . . . 138
4.4.1. Formalizing product discovery . . . 140
4.4.2. Product discovery techniques . . . 143
Jobs to be Done . . . 143
Impact mapping . . . 144
The Business Analysis Body of Knowledge . . . 145
4.5. Product design . . . 146
4.5.1. Design thinking . . . 147
4.5.2. Hypothesis testing . . . 148
4.5.3. Usability and interaction . . . 148
4.5.4. Parable: The Flower and the Cog . . . 149
4.5.5. Product discovery *versus* design . . . 150
4.6. Product planning . . . 151
4.6.1. Product roadmapping and release planning . . . 151
4.6.2. Backlog, estimation, and prioritization . . . 152
4.7. Conclusion . . . 154
4.7.1. Discussion questions . . . 154
4.7.2. Research & practice . . . 155
4.7.3. Further reading . . . 155
5. Work Management . . . 157
5.1. Introduction . . . 157
5.1.1. Chapter 5 outline . . . 158
5.1.2. Chapter 5 learning objectives . . . 158
5.2. Task management . . . 159
5.3. Learning from manufacturing . . . 162
5.3.1. Kanban and its Lean origins . . . 162
5.3.2. The Theory of Constraints . . . 164
5.3.3. Queues and limiting work in process . . . 165
5.3.4. Multi-tasking . . . 169
5.3.5. Scrum, Kanban, or both? . . . 170
5.4. The shared mental model of the work to be done . . . 171
5.4.1. Visualization . . . 171
5.4.2. Andon, and the Andon cord . . . 172
5.4.3. Definition of done . . . 173
5.4.4. Time and space shifting . . . 173

5.5. Lean product development and Don Reinertsen 174

5.5.1. Product development *versus* production 174

5.5.2. Cost of delay 177

5.6. The service desk and related tools 181

5.7. Towards process management 183

5.7.1. Process basics 184

5.7.2. The Checklist Manifesto 185

5.7.3. Case management 186

5.8. Conclusion 188

5.8.1. Discussion questions 188

5.8.2. Research & practice 188

5.8.3. Further reading 188

6. Operations Management 190

6.1. Introduction 190

6.1.1. Chapter 6 outline 191

6.1.2. Chapter 6 learning objectives 191

6.2. An overview of operations management 192

6.2.1. Defining operations 192

6.2.2. The concept of “service level” 195

6.3. Monitoring 195

6.3.1. Monitoring techniques 195

6.3.2. Designing operations into products 199

6.3.3. Aggregation and operations centers 200

6.3.4. Understanding business impact 202

6.3.5. Capacity and performance management 203

6.4. Operational response 204

6.4.1. Communication channels 205

6.4.2. Operational process emergence 206

6.4.3. Post-mortems, blamelessness, and operational demand 208

6.5. Configuration management and operations 209

6.5.1. State and configuration 209

6.5.2. Environments and the fierce god of “Production” 210

6.5.3. “Development is production” 212

6.6. Designing for operations and scale 212

6.6.1. The CAP principle 213

6.6.2. The AKF scaling cube 215

6.7. Advanced topics in operations 218

6.7.1. Drills, game days, and Chaos Monkeys 218

6.7.2. Site reliability engineering at Google 219

6.8. Conclusion 219

6.8.1. Discussion questions 220

6.8.2. Research & practice 220
6.8.3. Further reading 220
6.8.4. Articles/posts 220
6.8.5. Videos 221
Part II Conclusion 222
Part III — Team of Teams 223
Special section: Scaling the organization and its work 226
The two dimensions of demand management 226
Adding a third dimension with Cynefin 229
The Betz organizational scaling cube 231
Demand, supply, and execution 232
Part III chapter structure 235
The delivery models 236
7. Coordination 239
7.1. Introduction 239
7.1.1. Chapter overview 240
7.1.2. Chapter learning objectives 240
7.2. Defining coordination 241
7.2.1. Example: Scaling one product 241
7.2.2. A deeper look at dependencies 242
7.2.3. Organizational tools and techniques 243
Structure 246
Synchronization 246
Boundary spanning 247
7.2.4. Coordination effectiveness 249
7.3. Coordination, execution, and the delivery models 250
7.3.1. Product management release trains 251
7.3.2. Project management as coordination 252
7.3.3. Decision rights 253
7.3.4. Process management as coordination 254
7.3.5. Projects and processes 255
7.4. Why process management? 255
7.4.1. Process conception 258
7.4.2. Process execution 259
7.4.3. Measuring process 260
7.4.4. Process improvement 261
7.4.5. The disadvantages of process 262
The pitfall of process “silos” 263
Process proliferation 263
7.5. Process control and continuous improvement 264
7.5.1. History of continuous improvement 266

7.5.2. Frederick Taylor and efficiency . . . 267
7.5.3. W.E. Deming and variation . . . 268
7.5.4. Lean product development and cost of delay . . . 270
7.5.5. Scrum and empirical process control . . . 270
7.6. Conclusion . . . 271
7.6.1. Discussion questions . . . 271
7.6.2. Research & practice . . . 272
7.6.3. Further reading . . . 272
8. Investment and Planning . . . 274
8.1. Introduction . . . 274
8.1.1. Chapter 8 outline . . . 275
8.1.2. Chapter 8 learning objectives . . . 276
8.2. IT financial management . . . 276
8.2.1. Historic IT financial practices . . . 277
Annual budgeting and project funding . . . 278
Cost accounting and chargeback . . . 279
8.2.2. Next generation IT finance . . . 281
Beyond budgeting . . . 282
Internal “venture” funding . . . 283
Options as a portfolio strategy . . . 284
Lean product development . . . 285
Lean accounting . . . 285
Value stream orientation . . . 286
Internal market economics . . . 287
Service brokerage . . . 288
8.3. IT sourcing and contract management . . . 288
8.3.1. Basic concerns . . . 288
8.3.2. Outsourcing and cloudsourcing . . . 291
8.3.3. Software licensing . . . 292
8.3.4. The role of industry analysts . . . 293
8.3.5. Software development and contracts . . . 294
8.4. Structuring the investment . . . 297
8.4.1. Features *versus* components . . . 298
8.4.2. Epics and new products . . . 299
8.5. Larger-scale planning and estimating . . . 301
8.5.1. Why plan? . . . 301
8.5.2. Planning larger efforts . . . 303
Accountability . . . 303
Coordination . . . 303
Risk management . . . 303
8.6. Why project management? . . . 304

8.6.1. A traditional IT project 305
The decline of the “traditional” IT project 308
8.6.2. How is a project different from simple “work management"? 310
8.6.3. The “iron triangle” 311
8.6.4. Project practices 312
Scope management 313
Project risk management 314
Project assignment 314
Governing outsourced work 314
8.6.5. The future of project management 315
8.7. Topics 316
8.7.1. Critical chain 316
8.7.2. The Agile project frameworks 317
8.8. Conclusion 317
8.8.1. Discussion questions 317
8.8.2. Research & practice 317
8.8.3. Further reading 318
9. Organization and culture 320
9.1. Introduction 320
9.1.1. Outline 320
9.1.2. Learning objectives 321
9.2. IT versus product organization 321
9.3. Defining the organization 326
9.3.1. Team persistence 328
9.4. Product and function 328
9.4.1. Waterfall and functional organization 329
9.4.2. The continuum of organizational forms 331
9.4.3. Scaling the product organization 334
9.4.4. From functions to components to shared services 334
9.5. Final thoughts on organization forms 335
9.6. IT human resource management 335
9.6.1. Basic concerns 335
9.6.2. Hiring 336
9.6.3. Process as skill 337
9.6.4. Allocation and tracking people’s time 338
9.6.5. Accountability and performance 340
9.7. Why culture matters 342
9.7.1. Motivation 343
9.7.2. Schneider and Westrum 343
9.7.3. Toyota Kata 346
9.8. Industry frameworks 347

9.8.1. Defining frameworks 347
9.8.2. Observations on the frameworks 348
The problem of statistical process control 348
Local optimization temptation 349
Lack of execution model 350
Secondary artifacts, compounded by batch orientation 350
Confusion of process definition 351
Conclusion 352
Discussion questions 352
Research & practice 353
Further reading 353
Part III conclusion 355
Part IV — Enterprise 356
Special section: The IT lifecycles 357
10. Governance, Risk, Security, and Compliance 360
10.1. Introduction 360
10.1.1. Chapter 10 outline 361
10.1.2. Chapter 10 learning objectives 361
10.2. Governance 362
10.2.1. What is governance? 362
A governance example 362
Some theory of goverance 363
COSO and control 365
10.2.2. Analyzing governance 368
Governance context 368
Governance and the emergence model 369
10.3. Enablers 373
10.3.1. An introduction to enablers 373
Principles, policies, and frameworks 375
People, skills, and competencies 375
Culture, ethics and behavior 375
Organizational structures 375
Processes 375
Information 375
Services, infrastructure, and applications 376
10.3.2. One-way policy begins 376
10.3.3. Mission, principle, strategy, and policy 378
10.3.4. Standards, frameworks, methods, and the innovation cycle 380
10.4. Risk management 382
10.4.1. Risk management fundamentals 382
Risk identification 384

Risk assessment 385
Risk response 385
Controls 386
Business continuity 387
10.4.2. Compliance 388
10.5. Assurance and audit 389
10.5.1. Assurance 389
Three party foundation 391
Types of assurance 393
Assurance and risk management 394
Non-audit assurance examples 395
10.5.2. Audit 398
External versus internal audit 400
Audit practices 400
10.6. Security 401
10.6.1. Information classification 404
10.6.2. Security engineering 404
Consumer versus sponsor perspective 405
Security architecture and engineering 405
Security and the systems lifecycle 406
Sourcing and security 407
10.6.3. Security operations 407
Prevention 408
Detection 408
Response 409
Forensics 409
Relationship to other processes 409
10.6.4. Security and assurance 410
10.7. Digital governance 410
10.7.1. The failings of IT governance 411
The new digital operating model 412
Project versus operations as operating model 412
CIO as order-taker 413
The fallacies of “rigor” and repeatability 414
10.7.2. Digital effectiveness 415
10.7.3. Digital efficiency 415
Consolidate the pipelines 415
Reduce internal service friction 415
Manage the process portfolio 416
Treat governance as demand 416
Leverage the digital pipeline 417

10.7.4. Digital risk management 417

Cost of delay as risk 417

Team dynamics as risk 418

Sourcing risk 418

10.7.5. Automating digital governance 419

Digital exhaust 419

Additional automation 422

10.8. Conclusion 423

10.8.1. Discussion questions 423

10.8.2. Research & practice 423

10.8.3. Further reading 424

11. Enterprise Information Management 425

11.1. Introduction 425

11.2. Chapter 11 outline 426

11.3. Information and value 427

11.3.1. The origins of digital information 428

11.3.2. The measurable value of information 429

11.3.3. Information, efficiency, and effectiveness 431

11.4. Data management basics 436

11.4.1. The importance of context 436

11.4.2. Data management and the DMBOK 437

The Data Management Body of Knowledge 437

11.4.3. Data architecture and development 438

Data and process 438

The ontology problem 438

Data modeling 439

Database administration 441

Patterns and reference architectures 443

Section conclusion 443

11.5. Towards enterprise information management 444

11.5.1. Advanced data management 444

Data integration and the "system of record” 444

Reference data management 447

Commercial data 447

Data quality 448

11.5.2. Enterprise records management 449

11.5.3. Data governance 451

Information-related risks 451

E-discovery and cyberlaw 452

11.6. Analytics 452

11.6.1. Analytics in context 452

11.6.2. Data warehousing and business intelligence . . . 455
11.7. Agile information management . . . 458
11.7.1. Software versus data . . . 458
11.7.2. Next generation practices in information management . . . 460
Cross-functional teams . . . 460
Domain-driven design . . . 460
Generic structures and inferred schemas . . . 461
Append-only to the rescue? . . . 463
Test data . . . 463
11.8. Information management topics . . . 464
11.8.1. Social, mobile, analytics, and cloud . . . 464
11.8.2. Big data . . . 465
11.8.3. Managing the information of digital delivery . . . 466
11.9. Conclusion . . . 467
11.9.1. Discussion questions . . . 467
11.9.2. Research & practice . . . 468
11.9.3. Further reading . . . 468
12. Architecture and Portfolio . . . 469
12.1. Introduction . . . 469
12.1.1. Chapter 12 outline . . . 470
12.1.2. Chapter 12 learning objectives . . . 470
12.2. Why architecture? . . . 471
12.2.1. Defining Enterprise Architecture . . . 473
12.2.2. Architecture organization . . . 475
Architecture as staff function . . . 475
Enterprise Architecture and the operating model . . . 478
Peer organizations . . . 480
12.2.3. The value of Enterprise Architecture . . . 482
Reducing cost of delay . . . 484
Technical debt revisited . . . 485
Scaling the enterprise mental model . . . 485
12.3. Architecture practices . . . 485
12.3.1. From the author: What do architects do, actually? . . . 486
12.3.2. Architecture and governance . . . 487
12.3.3. Architecture as a management program . . . 489
12.3.4. Modeling and visualization . . . 491
Human visual processing . . . 493
Visualization in digital systems . . . 494
Limitations of visualization . . . 496
12.3.5. Repositories and knowledge management . . . 499
Catalogs, diagrams, matrices . . . 500

Architecture data management 502
An economic view 504
12.3.6. The “rationalization” quest 506
Application rationalization 507
Data and information 507
Technology rationalization case #1 508
Technology rationalization case #2 508
Service or technology rationalization? 508
12.4. Architecture domains 509
12.4.1. Architecture perspectives 510
Data architecture 510
Process architecture 511
Capability architecture 512
12.4.2. Architecture layers 512
Business Architecture 512
Application architecture 513
Technical architecture 514
12.4.3. Other forms of architecture 514
Solutions architecture 514
Software architecture 515
Information architecture (alternate usage) 515
12.5. Agile and architecture 515
12.5.1. The hubris of Architecture 516
12.5.2. The hubris of Agile 517
The limitations of cost of delay 518
Documentation 518
Sourcing and technology standards 519
Architecture as emergent 519
12.5.3. Towards reconciliation 520
Why: Creating the context 520
What: The architecture of architecture, or the digital pipeline itself 520
How: Execution 521
Architecture Kata 523
Evaluating architecture outcomes 526
12.6. Portfolio analysis 527
12.6.1. Application value analysis 529
12.6.2. Application rationalization 529
12.7. Architecture standards 530
12.7.1. Business Architecture 530
12.7.2. The TOGAF® standard 530
12.7.3. The ArchiMate® Enterprise Architecture modeling language 530

12.7.4. Industry verticals . . . 530
12.7.5. Governmental frameworks . . . 531
12.8. Conclusion . . . 531
12.8.1. Discussion questions . . . 531
12.8.2. Research & practice . . . 531
12.8.3. Further reading . . . 532
Part IV and book conclusion . . . 533
Appendices and back matter . . . 534
The major frameworks . . . 535
CMMI . . . 535
CObIT . . . 536
ITIL . . . 538
PMBOK . . . 540
The TOGAF® standard . . . 542
Other frameworks . . . 542
Project management . . . 543
Project basics . . . 543
Process management and modeling . . . 546
Process and related terms . . . 546
Process modeling . . . 552
IGOE (Input/Guide/Output/Enabler) . . . 554
Ordering, synchronization, and conditionality . . . 555
Swimlanes . . . 557
A final caution on technique . . . 559
References . . . 560
Glossary . . . 573
Backlog and release notes . . . 574
Index . . . 575
Colophon . . . 595
Author biography . . . 596

# Front Matter

## Praise for this book

Managing Digital *is a perfect fit for my Management Information Systems class to introduce students to the fast-paced world of IT Infrastructure that they will be dealing with shortly upon graduation. This book uses multiple perspectives (Founder, Team Leader, VP, C-level executive) to demonstrate to the student not only how a business grows, but how they need to continually grow their skill set. The use of hands-on exercises encouraged by the format of this book complements my teaching style that allows students to learn by doing, failing, and doing again. An additional benefit is that this book begins with a focus on the startup mentality which I will use in my Business Innovation class.*

**Prof. Pat Paulson, Winona State University**

*Charles has produced the ultimate roadmap for the 21st century organization. A concise and powerful tool, this book will help anyone assess where they are on their journey today, where they may have gone wrong in the past, and where they should be driving towards today!*

**Ben Rockwood, Director IT & Operations, Chef**

*Investments in “digital” are critical for organizations and the economy as a whole. Delivering digital effectively benefits both individuals and communities. This is the time to re-visit and integrate industry guidance and reach a consensus on how digital and IT professionals can best approach their responsibilities. We need deliverance from the dark ages of IT. "Managing Digital" is comprehensive, well written, and enlightening. Charles T. Betz has nailed it once again.*

**Mark Smalley, The IT Paradigmologist**

*Managing Digital is a large and significant contribution to computing and IT education. No longer will graduates require on-the-job training after graduation to be current in the workforce. This book merges the requirements of academia with the needs of the modern digital business. Discussions of Lean, DevOps, the domain of the digital business, and more, are all active topics in the marketplace and now they are part of an educational curricula as well.*

**Chris Little, Consultant**

# The Open Group Press

*About The Open Group Press*

The Open Group Press is an imprint of The Open Group for advancing knowledge of information technology by publishing works from individual authors within The Open Group membership that are relevant to advancing The Open Group mission of Boundaryless Information Flow™. The key focus of The Open Group Press is to publish high-quality monographs, as well as introductory technology books intended for the general public, and act as a complement to The Open Group Standards, Guides, and White Papers. The views and opinions expressed in this book are those of the author, and do not necessarily reflect the consensus position of The Open Group members or staff.

*About The Open Group*

The Open Group is a global consortium that enables the achievement of business objectives through technology standards. Our diverse membership of more than 600 organizations includes customers, systems and solutions suppliers, tool vendors, integrators, academics, and consultants across multiple industries.

**The Open Group Vision**

*Boundaryless Information Flow* achieved through global interoperability in a secure, reliable, and timely manner

**The Open Group Mission**

The mission of The Open Group is to drive the creation of Boundaryless Information Flow achieved by:

- Working with customers to capture, understand, and address current and emerging requirements, establish policies, and share best practices
- Working with suppliers, consortia, and standards bodies to develop consensus and facilitate interoperability, to evolve and integrate specifications and open source technologies
- Developing and operating the industry’s premier certification service and encouraging procurement of certified products

Further information on The Open Group is available at *www.opengroup.org*.

# Copyright

Managing Digital: Concepts and Practices

Published by The Open Group Press

Any publication of The Open Group may include technical inaccuracies or typographical errors. Changes may be periodically made to these publications; these changes will be incorporated in new editions of these publications. The Open Group may make improvements and/or changes in the products and/or the programs described in these publications at any time without notice.

Should any viewer of this document respond with information including feedback data, such as questions, comments, suggestions, or the like regarding the content of this document, such information shall be deemed to be non-confidential and The Open Group shall have no obligation of any kind with respect to such information and shall be free to reproduce, use, disclose, and distribute the information to others without limitation. Further, The Open Group shall be free to use any ideas, concepts, know-how, or techniques contained in such information for any purpose whatsoever including but not limited to developing, manufacturing, and marketing products incorporating such information.

If you did not obtain this copy through The Open Group, it may not be the latest version. For your convenience, the latest version of this publication may be downloaded at *The Open Group Library*.

DRAFT: Built with *asciidoctor*, version 1.5.7.1.

Backend: pdf

Version 1.0 Build date: 2018-09-11 23:03:19 BST

Epigraph citations: [*92*], [*1*], [*280*]

# Epigraph

*… human beings are inherently storytellers who have a natural capacity to recognize the coherence and fidelity of stories they tell and experience … we experience and comprehend life as a series of ongoing narratives, as conflicts, characters, beginnings, middles, and ends.*

Walter R. Fisher

*Always design a thing by considering it in its next larger context -— a chair in a room, a room in a house, a house in an environment, an environment in a city plan.*

Eliel Saarinen

*Scaling simply refers, in its most elemental form, to how a system responds when its size changes. What happens to a city or a company if its size is doubled?*

Geoffrey West

# Dedication

*To all students, past, present, and future*

# Foreword

The label "Digital" is applied to a lot of management and technology guidance these days, and risks becoming just another in a long line of technology buzzwords. Isn’t "Digital" just one more step in the long continuum of business exploiting emerging technology? Or, have we crossed some "tipping point" causing a fundamental shift in how we manage such transformation?

My answer would be that there is something different happening this time — digital delivery is becoming core, not context. The convergence of an abundance of computing resources, improving software development management, combined with a change in market focus from the supplier to the customer is changing the way we view Enterprise Architecture and IT management, and identifying the need to develop a digital workforce. The defining characteristics of a digital enterprise are becoming clear:

- Products or services that are either delivered fully digitally (e.g., digital media or online banking), or where physical products and services are obtained by the customer by digital means (e.g., online car sharing services)
- A "digital-first" culture, where the business models, plans, architectures, and implementation strategies start from an assumption of digital delivery
- A workforce who is digitally savvy enough to execute a digital-first approach

*About This Book*

This book, "Managing Digital: Concepts and Practices", is intended to guide a practitioner through the journey of building a digital-first viewpoint and the skills needed to thrive in the digital-first world. As such, this book is a bit of an experiment for The Open Group; it isn’t structured as a traditional standard or guide. Instead, it is structured to show the key issues and skills needed at each stage of the digital journey, starting with the basics of a small digital project, eventually building to the concerns of a large enterprise. So, feel free to digest this book in stages — the section *Introduction for the student* is a good guide.

Finally, the electronic publication of this book will be an experiment for The Open Group in another way, in that it is our intent for this book to be a living document updated through crowdsourced content and refreshed periodically in order to keep pace with the rapid evolution of digital business. In parallel with the publication of this book The Open Group Digital Practitioners Work Group will be developing standards and best practices for Digital Practitioners; once those normative standards are published, it is expected that this book will be revised to reflect those.

So, please let us know what you think, and please join our community as a contributor.

David Lounsbury
Chief Technical Officer
The Open Group

# Preface by original author

I wrote my first book, *Architecture and Patterns for IT Service Management, Resource Planning and Governance: Making Shoes for the Cobbler's Children* in 2006, with a second edition in 2011. I presented the second edition at the national SEI  Saturn conference in Minneapolis in 2013, where I was approached by Dr. Bhabani Misra, the head of the Graduate Programs in Software at the University of St. Thomas in St. Paul. Minnesota. Dr. Misra asked me to teach a class called "IT Infrastructure Management," later "IT Delivery," which was to cover not just technical topics but also process and governance.

The course (which has run every semester since January 2013) has been developed during an extraordinary period for IT and digital management. Even in 2013, the trend towards a new style of IT delivery, based on Agile and DevOps practices was notable and accelerating. At this writing, these approaches seem to have “crossed the chasm” in the words of Geoffrey Moore, and are becoming the dominant models for delivering IT value. As this book describes, there are good reasons for this historical shift, and yet its speed and reach are still disorienting.

For three semesters I assigned my first book (*Architecture and Patterns for IT: Service Management, Resource Planning, and Governance*) as a required text for the class. However, I did not write this as a textbook, and its limitations became clear. While I gave considerable attention to Lean and Agile in writing the book, it has a strongly architectural approach, coming at the IT management problem as a series of *views on a model*. I do not recommend this as a pedagogical approach for a survey class. It also had a thoroughly enterprise perspective, and I began to question whether this was ideal for new students. Further thought led to the idea of the emergence model (detailed in the Introduction).

I proposed the idea of a third edition to my publisher —– one that would pivot the existing material towards something more useful in class. They agreed to this and I started the rewrite. However, by the time I was halfway done with the first draft, I had a completely new book. The previous work was analytic and technical, while the new book was more conversational, attempting to engage students of a wide background.

A number of factors converged:

- My view that the “medium is the message,” which extends to the choice of authoring approach, toolchain, and publisher
- A desire to freely share at least a rough version of the book, both for marketing purposes and in the interests of giving back to the global IT community
- A desire to be able to rapidly update the book with as little friction as possible
- A practical realization that the book might get more uptake globally if it were available, at least in some form, as open-sourced intellectual property
- The fact that I had already started to *publish my labs on GitHub* and had, in fact, developed a workable continuous delivery (“DevOps”) toolchain based on virtualization for classroom use.

Ultimately, the idea of starting my own publishing company, and managing my own product, appeared both desirable and practical. So I decided to self-publish, via LeanPub. Further developments, including my ongoing participation in the IT4IT™ Reference Architecture, a standard of The Open Group, led me to sign the work over to The Open Group, where it will in part serve as a basis for the new Digital Practitioner Body of Knowledge.

Many assisted with and/or contributed to this work before its transition to The Open Group:

- Stephen Fralippolippi
- Roger K. Williams
- Jason Baker
- Mark Kennaley
- Glen Alleman
- Jeff Sussna
- Nicole Forsgren
- Richard Barton
- Evan Leybourn
- Chris Little
- Jabe Bloom
- Lorin Hochstein
- Gene Kim
- Murray Cantor
- Rob England
- Firasat Khan
- Mary Mosman
- David Bahn
- Amos Olagunju
- Svetlana Gluhova
- Mary Lebens
- Justin Opatrny
- Grant Spencer
- Halbana Tarmizi
- Will Goddard
- Terry Brown
- Francisco Piniero
- Pat Paulson and his students
- Majid Iqbal
- Mark Smalley
- Ben Rockwood

- Mark Mzyk
- Michael Fulton
- Sriram Narayam
- John Spangler
- Tom Limoncelli
- Kate Campise
- Dmitry and Alina Kirsanov

Special thanks to Dr. Bhabani Misra for asking me to teach at the University of St. Thomas and providing direction at key points, including challenging me to add a practical component to the course, and to Dave Lounsbury and The Open Group for seeing the merit in this work.

*Charles Betz, January 2018*

# Introduction for instructors and trainers

Welcome to *Managing Digital: Concepts and Practices*. So, what exactly *is* this book?

- It is the first general, survey-level text on IT management with a specific Agile, Lean IT, and DevOps orientation
- It has a unique and innovative learning progression based on the concept of organizational evolution and scaling
- Because it is written with continuous integration and print-on-demand techniques, it can be continually updated to reflect current industry trends

The book is intended for both academic and industry training purposes. There has been too much of a gap between academic theory and the day-to-day practices of managing digital products. Industry guidance has over the years become fragmented into many overlapping and sometimes conflicting bodies of knowledge, frameworks, and industry standards. The emergence of Agile and DevOps as dominant delivery forms has thrown this already fractured ecosystem of industry guidance into chaos. Organizations and individuals with longstanding investments in guidance such as ITIL (*aka* the IT Infrastructure Library). [1: Technically, ITIL is now just the abbreviation, but we spell it out here for reference.] and the Project Management Body of Knowledge (*aka* PMBOK) are re-assessing these commitments. This book seeks to provide guidance for both new entrants into the digital workforce and experienced practitioners seeking to update their understanding on how all the various themes and components of IT management fit together in the new world.

Digital investments are critical for modern organizations and the economy as a whole. Participating in their delivery (i.e., working to create and manage them for value) can provide prosperity for both individuals and communities. Now is the time to re-assess and synthesize the bodies of knowledge and developing industry consensus on how digital and IT professionals can and should approach their responsibilities.

## The IT industry and the rise of digital

> Now agile methodologies -— which involve new values, principles, practices, and benefits and are a radical alternative to command-and-control-style management -— are spreading across a broad range of industries and functions and even into the C-suite. [*223*]
>
> — Darrell Rigby et al., Harvard Business Review

Consider the following two industry reports.

In September 2015, Minneapolis-based Target Corporation laid off 275 workers with IT skillsets such as business analysis and project management, while simultaneously hiring workers with newer “Agile” skills. As quoted by a local news site, Target stated:

*“As a part of our transition to an Agile technology development and support model, we conducted a comprehensive review of our current structure and capabilities … we are eliminating approximately*

*275 positions and closing an additional 35 open positions. The majority of the impact was across our technology teams and was primarily focused on areas such as business analysis and project management."* [*149*]

Jim Fowler, Chief Information Officer at General Electric, says:

*“When I am in business meetings, I hear people talk about digital as a function or a role. It is not. Digital is a capability that needs to exist in every job. Twenty years ago, we broke e-commerce out into its own organization, and today e-commerce is just a part of the way we work. That’s where digital and IT are headed; IT will be no longer be a distinct function, it will just be the way we work. … [W]e’ve moved to a flatter organizational model with “teams of teams” who are focused on outcomes. These are co-located groups of people who own a small, minimal viable product deliverable that they can produce in 90 days. The team focuses on one piece of work that they will own through its complete lifecycle … in [the “back-office”] model, the CIO controls infrastructure, the network, storage, and makes the PCs run. The CIOs who choose to play that role will not be relevant for long.”* [*122*]

Modern Management Information Systems (MIS) courses and textbooks, especially at the undergraduate, survey level, seek to orient *all* students (whether IT/MIS specialists or not) to the role and function of information systems and their possibilities and value in the modern enterprise. This book, by contrast, intends to prepare the student for a career as a digital professional, in those industries that offer digital products *per se* as well as industries that rely on digital technology instrumentally for delivering all kinds of products. A central theme of the book is that IT, considered as a component, represents an increasing proportion of *all* industrial products (both consumer and business-facing). This trend towards IT’s increase is a key characteristic of Digital Transformation.

Current MIS survey texts have some common characteristics:

- They tend to focus on the largest organizations and their applications of computing. This can lead to puzzling topic choices; for example, in one current MIS text I reviewed, one of the first sections is dedicated to the problem of enterprise IT asset management -— a narrow topic for the earlier sections of a survey course and increasingly irrelevant in the age of the cloud.
- Because their coverage area is termed "information systems," many start with extensive and detailed coverage of database management at an enterprise scale, often focused on the classical relational database -— which is now just one choice among many in a world of NoSQL and microservices.
- They do not (and this is a primary failing) cover Agile and its associated digital ecosystems well, if at all. Brief mentions of Agile may appear in sections on project management, but in general there is a lack of awareness of the essential role of Agile and related methods in accelerating digital transformation.
- Their coverage of cloud infrastructure can also be limited, even with new editions coming out every year. Topics like infrastructure as code go unaddressed.
- Finally, current texts often uncritically accept and cite “best practice” IT frameworks such as CMMI, ITIL, PMBOK, and CObIT. New digital organizations do not, in general, use such guidance, and there is much controversy in the industry as to the value and future of these frameworks. This book strives to provide a clear, detailed, and well-supported overview of these issues.

IT, or the digital function, has had a history of being under-managed and poorly understood relative to peer functions in the enterprise. It struggles with a reputation for expensive inflexibility and Dilbert-esque dysfunction, and has generated a fair amount of mistrust with its business sponsors. The DevOps and Agile movements promise transformation but are encountering an entrenched legacy of

- Enterprise Architecture
- Program and project management
- Business process management
- IT service management practices
- IT governance concerns

Understanding and engaging with the challenges of this legacy are ongoing themes throughout this introductory text. Some of the more radical voices in the Agile movement sometimes give the impression that the legacy can be simply swept away. Mike Burrows notes that, in terms of the systems thinking at the core of Agile philosophy, such a move might be ill-advised:

*“Some will tell you that when things are this bad, you throw it all away and start again. It’s ironic: The same people who would champion incremental and evolutionary approaches to product development seem only too eager to recommend disruptive and revolutionary changes in people-based systems -— in which the outcomes are so much less certain”* [*45 p. 827*].

IT management at scale within an organization is a complex system. The IT workforce, its collective experience, and its ongoing development (through education and training) is another complex system orders of magnitude larger. Complex systems do not respond well to dramatic perturbations. They are best changed incrementally, with careful monitoring of the consequences of each small change. (This is part of the systems theory foundation underlying the Agile movement). This is why the book covers topics such as:

- Investment, sourcing, and people
- Project and process management
- Governance, risk, security, and compliance
- Enterprise information management
- Enterprise Architecture and portfolio management

While these practices, and their associated approaches and policies, have caused friction with digital and Agile practitioners, they all have their reasons for existing. The goal of this book is to understand their interaction with the new digital approaches, but in order to do this we must first understand them on their own terms. It does no good to develop a critique based on misconceptions or exaggerations about what (for example) process management or governance is all about. Instead, we try to break these large and sometimes controversial topics into smaller, more specific topics:

- Work and effort
- Ordering of tasks
- Task dependencies

- Coordination
- Investment
- Cost of delay
- Planned *versus* unplanned work
- Estimation *versus* commitment
- Value stream *versus* skill alignment
- Repeatability
- Defined *versus* empirical process control
- Synchronization and cadence
- Resource demand
- Shared mental models
- Technical debt
- Risk, etc.

By examining IT management in these more clinical terms, we can develop a responsible critique of current industry best practices that will benefit students as they embark on their careers.

**NOTE**

A key choice in the book’s evolution was to NOT include dedicated chapters on “Project Management” and “Process Management.” Instead, more general chapter titles of “Coordination” and “Investment and Planning” were chosen. Rationale for the decision is given in those chapters and in Part III generally. Similarly, there is no one section covering “IT service management” *per se*; its significant concerns are seen throughout Chapters 4 through 10.

### A process of emergence

> Joseph Campbell popularized the notion of an archetypal journey that recurs in the mythologies and religions of cultures around the world. From Moses and the burning bush to Luke Skywalker meeting Obi wan Kenobi, the journey always begins with a hero who hears a calling to a quest …
>
> The hero’s journey is an apt way to think of startups. All new companies and new products begin with an almost mythological vision –— a hope of what could be, with a goal few others can see …
>
> Most entrepreneurs feel their journey is unique. Yet what Campbell perceived about the mythological hero’s journey is true of startups as well: However dissimilar the stories may be in detail, their outline is always the same. [*31*]
>
> — Steve Blank, The Four Steps to Epiphany

One of the most important and distinguishing features of this book is its emergence model. In keeping with the entrepreneurial spirit of works like Ries’ *The Lean Startup*, the book adopts a progressive, evolutionary approach. The student’s journey through it reflects a process of emergence. Such processes are often associated with founding and scaling a startup. There are many helpful books on this topic, such as the following:

- *Nail It Then Scale It* by Furr and Ahlstrom [*102*]
- *Scaling Up* by Harnish [*117*]
- *Startup CEO* by Blumberg [*33*]
- *The Lean Startup* by Ries [*222*]
- *Hello, Startup* by Brikman [*35*]

The emergence model and overall book structure is discussed in depth in the main introduction. Here, for the instructor, are some notes on the thought process.

A central problem in conceiving the book was the question of overall learning progression, or narrative. As noted in the Preface, it was first developed to support a required semester-long survey class on IT management at the University of St. Thomas in St. Paul, Minnesota, in the largest software engineering program in the country.

Currently, two primary narratives or learning progressions are used to in teaching computing:

- The “stack”
- The “lifecycle”

The stack is how the most rigorous topics are taught. Algebra is the foundation for trigonometry, trigonometry is required for calculus, for example. Logic is needed for discrete math, required for automata and compilers, and so forth. The stack is also how technology is described: physical, logical, and conceptual layers, for example, or layered abstractions in networking protocols.

The systems lifecycle, on the other hand, is how we tend to structure industry guidance. We plan and design, we build, we run. Guidance such as CObIT and ITIL show lifecycle influences, as do software engineering programs in colleges.

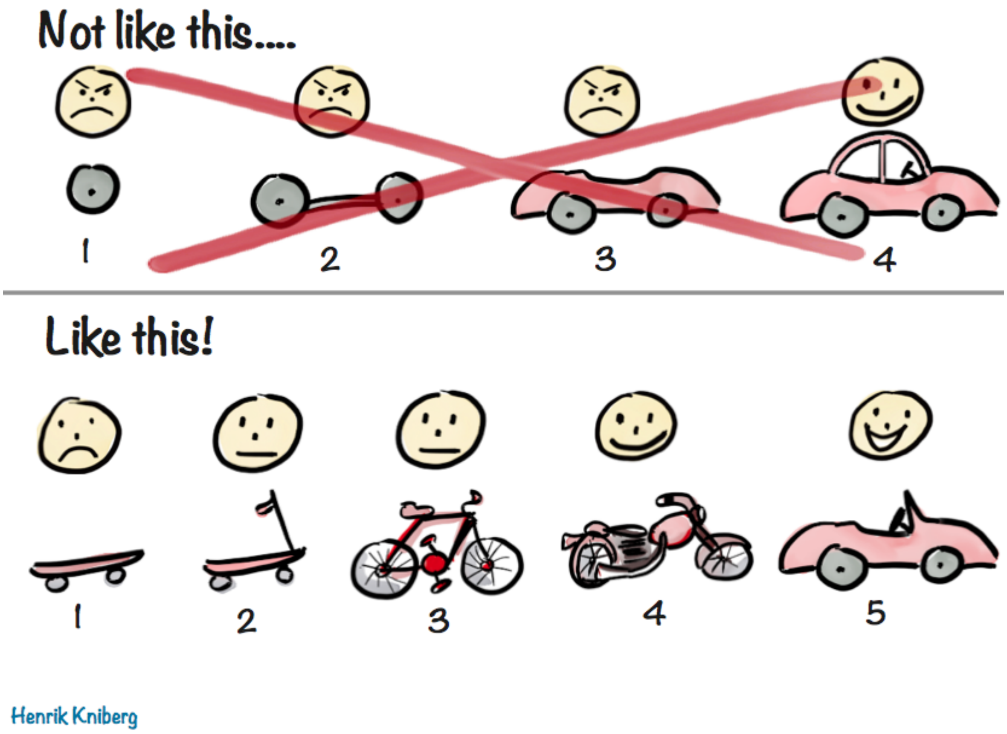

*Figure 1. Systems evolve and scale iteratively*

However, both the stack and the lifecycle have limitations:

- The stack can fall into reductionism, or what venture capitalist Anshu Sharma calls the “stack fallacy,” the “mistaken belief that it is trivial to build the layer above yours” [*244*]. A different form of the stack fallacy is seen when practitioners assume that systems can easily be decomposed through layers (business, data, application, technology).
- The lifecycle narrative is far too prone to promoting ***waterfall thinking***, anathema to the current Agile and *Lean Product Development* approaches redefining the digital profession.

Instead, this book’s emergence narrative draws on systems theory, in particular John Gall’s idea that “*A complex system that works is invariably found to have evolved from a simple system that worked. A complex system designed from scratch never works and cannot be patched up to make it work. You have to start over, beginning with a working simple system*” [*103*]. Henrik Kniberg created a compelling visual image showing how systems scale: through ongoing, iterative re-design and elaboration of fundamental characteristics (*Systems evolve and scale iteratively* [2: Image credit Henrik Kniberg *http://blog.crisp.se/author/henrikkniberg*, used by permission]).

What if we treated the student’s understanding as such a systems scaling problem? What would be the simplest possible thing that could work? How would we iteratively evolve their understanding, based on practical topics? Scaling seems to be orthogonal to the other narratives (*Three narrative dimensions*).

As we’ll cover in the main introduction, reading books on organizational scaling inspired the idea that growth does not happen smoothly; instead organizations tend to cluster at certain scales and struggle to grow to the next scale. Hence the overall structure of the book:

- Founder
- Team
- Team of Teams
- Enterprise

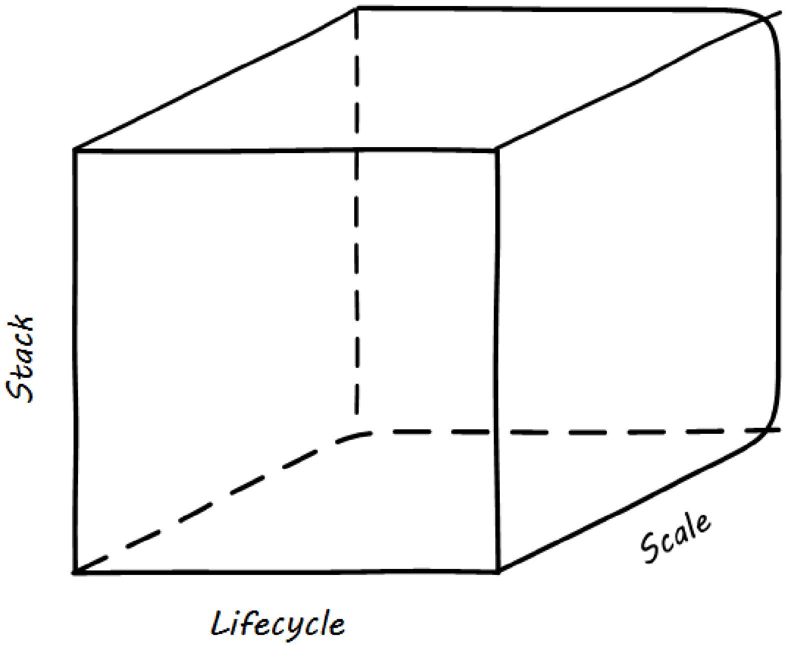

*Figure 2. Three narrative dimensions*

A key focus of the book is explaining what practices are formalized at which level of growth. The thought experiment is, “what would I turn my attention to next as my IT-based concerns scale up?” For example, the book currently proposes that work management (implying rudimentary workflow, e.g., Kanban) correctly comes before formalized project and/or process management, which in turn tend to emerge before enterprise governance practices (e.g., formalized risk management).

Note that this would be a testable and falsifiable hypothesis if empirical research were done to take inventory of and characterize organization scaling patterns. If we found, for example, that a majority of organizations formalize governance, risk, security, and compliance practices before formalizing product management, that would indicate that those chapters should be re-ordered. In my experience, small/medium businesses may have formal product management but Governance, Risk, and Compliance, (GRC) are still *tacit*, not formalized. *This does not mean that GRC is not a concern*, but they have not yet instituted formal policy management, internal audit, or controls.

The presence of product management at an early stage in the book (Chapter 4) is intended to provoke thought and debate. Product management is poorly addressed in most current college computing curricula as well as the *de facto* industry standards (e.g., the TOGAF® ADM, PMBOK, and ITIL). Yet formalizing it is one of the earliest concerns for a startup, and the imperatives of the product vision drive all that comes after. Evidence to this effect is seen (as of 2015) at the University of California at Berkeley I-School, which has replaced its Project Management course with

*Lean/Agile Product Management*, taught currently by Jez Humble, author of *Continuous Delivery, Lean Enterprise*, and co-author of *The DevOps Handbook*.

This book however is not a complete dismissal of older models of IT delivery. Wherever possible, new approaches are presented relative to what has gone before. The specifics of “what’s different” are identified, in the interest of de-mystifying what can be fraught and quasi-religious topics. (Why is a Scrum standup or a Kanban board effective, in terms of human factors?)

**The emergence model can also be understood as an individual’s progression within a larger enterprise.** Even starting from day one at a Fortune 100 corporation, I believe a person’s understanding still progresses from individual, to team, to team of teams, to enterprise. Of course, the person may cease evolving their understanding at any of these stages, with corresponding implications for their career.

This book does not cover specific technologies in any depth. Many examples are used, but they are carefully framed to not require previous expertise. This is about broader, longer lifecycle trends.

There is benefit to restricting the chapters to 12, as a typical semester runs 14 weeks, and the book then fits well, with one chapter per class and allowing for an introductory session and final exam. Of course, a two-semester series, with two weeks per chapter, would also work well. Each half of the book is also a logical unit. The chapters have been re-ordered and refactored many times, and based on further research may evolve further.

## Labs

With three chapters in each section, the book can be covered in one intense semester at a chapter a week, although expanding it to a two-semester treatment would allow for more in-depth coverage and increased lab exposure. This required new thinking. How could students learn IT management at scale in a lab setting? A hands-on component is essential, as IT management discussions can be abstract and meaningless to many students. (“Incidents are different from problems!”)

Ten years ago, the best that would have been possible would be paper case studies, perhaps augmented with spreadsheets. But new options are now available. The power of modern computers (even lightweight laptops) coupled with the widespread availability of open-source software makes it possible to expose students to industrial computing in a meaningful, experiential way. There is great utility in the use of lightweight, pre-configured virtualization technologies such as Vagrant, VirtualBox, and Docker.

The initial course used a central server, but even that is not necessary. The class can be taught with zero computing budget, assuming that each team of students has access to a modern laptop (recommend 8 gigabytes of RAM and 250 gigabyte drive) and a fast Internet connection. Initial lab versions used free and open-source versions of Chef, Jenkins, Nagios, JUnit, Ant, and other tools, evolving to Node.js, microservices, Docker, and Kubernetes. See the *dm-academy resources on GitHub* for the current status.

Some may question the inclusion of command-line experience, but without some common technical platform, it is hard to provide a meaningful, hands-on experience in the first half of the course. Currently, digital professionals require hands-on technology skills; barriers to developing them being much lower than in years past. The initial course assumed that the students are at least willing to learn computing techniques, with no prerequisites beyond that. Not even a programming language is required; the Java currently used as a sample is minimal.

Truly beginning students will have to work at the Linux tutorials, but all they need to master is basic command line navigation; this is possible with a diverse student body, some with no previous computing experience. The labs for the second half of the course mostly use games, experiential paper-based classroom exercises, GUI-based software, databases, and office productivity tools.

The emergence model is intended as the learning progression for either traditional semester-based education, or industry training. Labs tie into the emergence model and encourage the students in hands-on exploration of fundamentals, as if they were starting their own business or initiative. As they progress, they remain grounded in the basics of applied technology. Even the most highly scaled topics, such as Enterprise Architecture and portfolio management, become more real when the students are reminded that the systems in question are simply larger and more numerous versions of the examples they see in the lab.

# Introduction for the student

This is a *survey* text, intended for the advanced undergraduate or graduate student interested in the general field of applied Information Technology (IT) management. It is also intended for the mid-career professional seeking to update their understanding of IT management's evolution, especially in light of the impact of Agile, Lean, and DevOps.

The book is grounded in basic computing fundamentals but *does not require any particular technical skills to understand*. You do **not** need to have taken any courses in networking, security, or specific programming languages to understand this book. However, you occasionally will be presented with light material on such topics, including fragments of programming languages and pseudocode, and you will need to be willing to invest the time and effort to understand.

This book makes frequent reference to digital startups -— early stage companies bringing new products to market that are primarily delivered as some form of computer-based service. Whether or not you intend to pursue such endeavors, *the startup journey is a powerful frame for your learning*. Large IT organizations in enterprises sometimes gain a reputation for losing sight of business value. IT seems to be acquired and operated for its own sake. Statements like "we need to align IT with the business!" are too often heard.

A digital startup exposes with great clarity the linkage between IT and "the business." The success or failure of the company itself depends on the adept and responsive creation and deployment of the software-based systems. Market revenues arrive, or do not, based on digital product strategy and the priorities chosen. Features the market doesn't need? You won't have the money to stay in business. Great features, but your product is unstable and unreliable? Your customers will go to the competition.

The lessons that digital entrepreneurs have learned through this trial by fire shed great light on IT's value to the business. Thinking about a startup allows us to consider the most fundamental principles as a sort of microcosm, a small laboratory model of the same problems that the largest enterprises face.

Verne Harnish, in the book *Scaling Up* [*117 pp. 25-26*], describes how companies tend to cluster at certain levels of scale. (See *Organizations cluster at certain sizes (concept by Harnish)* [3: similar to [*117 p. 25*]). Of 28 million U.S. firms, the majority of firms (96%) never grow beyond a founder; a small percentage emerge as a viable team of 8-12, and even smaller numbers make it to the stable plateaus of 40-70 and 350-500. The "scaling crisis" is the challenge of moving from one major level to the next. (Harnish uses the more poetic term "Valley of Death.") This scaling model, and the needs that emerge as companies grow through these different stages, is the basis for this book's learning progression.

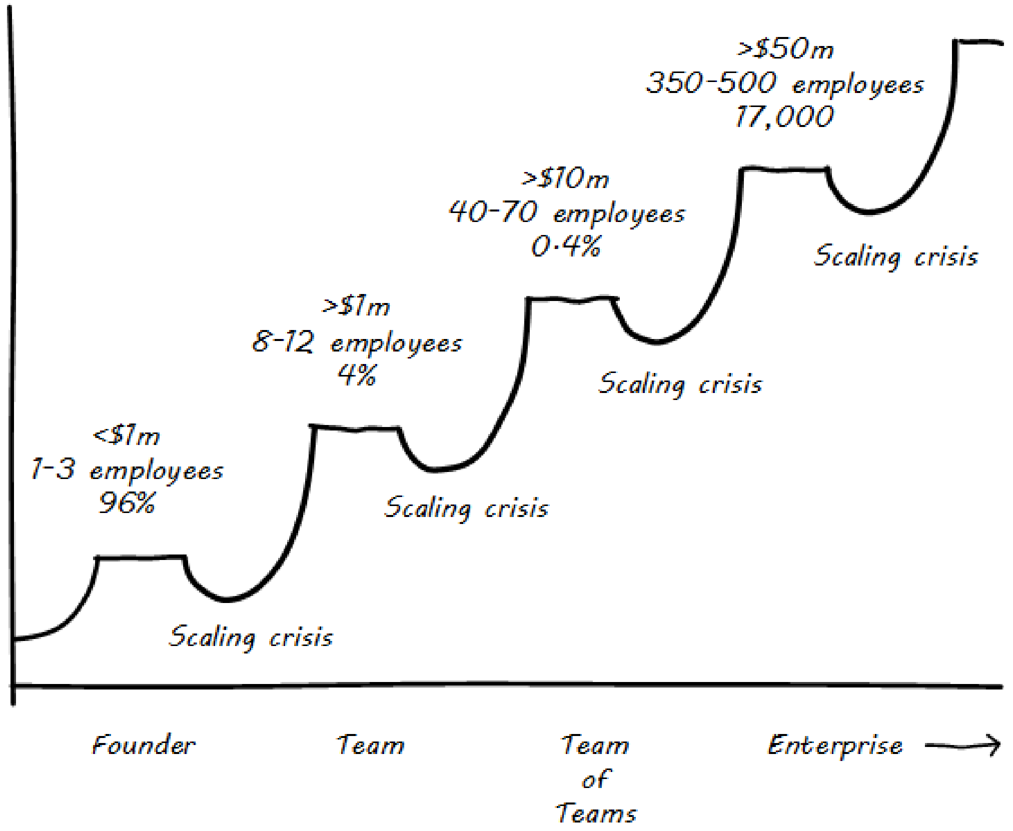

*Figure 3. Organizations cluster at certain sizes (concept by Harnish)*

However, this is not a textbook (or course) on entrepreneurship. It remains IT-centric. **And, the book is also intended to be relevant to students entering directly into large, established enterprises.** In fact, it prepares the student for working in all stages of growth because it progresses through these four contexts:

- Individual (founder)
- Team
- Small company (team of teams)
- Enterprise

Whether your journey begins in a startup or within a larger, established organization, you will (hopefully) become aware as you progress of a broadening context:

- Other team members
- Customers
- Suppliers
- Sponsors
- Necessary non-IT capabilities (finance, legal, HR, sales, marketing, etc.).
- Channel partners
- Senior executives and funders
- Auditors and regulators

Part of maturing in your career is understanding how all these relationships figure into your own