Fast Career Tack

Curated contents for software engineers.

10x Software Engineer

Learning Path

FABIO CICERCHIA

10x Software Engineer

Curated contents for software engineers.

Fabio Cicerchia

This book is for sale at http://leanpub.com/10xse

This version was published on 2021-02-10

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2019 - 2021 Fabio Cicerchia

Contents

Introduction

Hello, I am Fabio Cicerchia, a Passionate Solutions Architect and Application Developer with 15+ years of experience. Always enjoying creating quality web applications and web portals using cutting-edge technologies.

Working in several positions, from Software Developer to Frontend/Backend Developer, from Sysadmin to Team Leader, allowed me to work on each layer of a web application, covering the whole life-cycle from initial requirements gathering to design, planning, coding, testing, documentation, deployment, and maintenance.

I've started to move my first steps in the programming world, at the "late" age of 15 years old…

And I decided to undertake that kind of career without asking myself too many questions.

I've started, actually, developing software during my years spent in high school (obviously in the computer science specialisation) and then decided to focus totally on the web. I continue nowadays to spend my spare time learning new things (from methodologies to new technologies) keeping myself up-to-date or at least trying to do it.

Because, especially in this field, who hesitates is lost. Let's acknowledge this.

My career started as a freelancer, and then settled down in various companies, in various roles (still as a developer), in various industries such as e-commerce, marketing, web agency, analytics.

I continued moving up the ladder, switching to management and reaching the "peak" as CTO and then going back to the technical track.

Now I find myself years later from those my first steps, with HTML and VB, and still, remember the day in which I've opened the

website HTML.it[1] to start to learn the rudiments of what brought me where I am now.
A lot of satisfaction and few regrets.
Therefore I've decided to sum up all my 15+ years of experience about the things I've learned, tips and suggestions collected over time, with the hope that might be useful to you.
Consider this ebook as a starting point and not as a list of things to be blindly followed. Bear in mind that lots of things may be subjective, but I tried to be as much objective and unbiased as possible.

'Nuff said, happy reading!

Contacts

- Web Site[2]
- Twitter: @fabiocicerchia[3]
- LinkedIn[4]

[1] http://www.html.it
[2] https://fabiocicerchia.it
[3] https://twitter.com/fabiocicerchia
[4] http://linkedin.com/in/fabiocicerchia/

Preface

Audience

The ebooks will cover all the stages of a developer career, from the beginning to the advanced roles, the main core is focusing on improving your career when you've already started.
Therefore this is mainly for junior and mid-level programmers, but it has good points and useful information even for experienced developers as well.

The technical skillset as a software developer is in high demand since technology is a huge part of our lives and no company can afford to survive without IT. Companies do not need just software developers, they need software engineers with soft skills and breadth of knowledge.

Competency Levels

In the book will be used 3 of the levels defined by the Dreyfus model:

- Advanced Beginner
- Competence
- Proficient

> The **Dreyfus model of skill acquisition** is a model of how learners acquire skills through formal instruction and practising, used in the fields of education and operations research. Brothers Stuart and Hubert Dreyfus

> proposed the model in 1980 in an 18-page report on their research at the University of California, Berkeley, Operations Research Center for the United States Air Force Office of Scientific Research. The model proposes that a student passes through five distinct stages and was originally determined as: novice, competence, proficiency, expertise, and mastery.
>
> - https://en.wikipedia.org/wiki/Dreyfus_model_of_skill_acquisition

The book won't cover the *Novice* level, because it is required to have some knowledge about programming.
Also, it won't be covered about the *Expert* level, because the book will provide enough knowledge to be competent in many areas of the IT field, but giving the knowledge to be expert in all of them won't be realistically possible.

Contents of This Book

In each chapter, I provide a list of useful contents (articles, videos and/or books) to be actioned during the week. It is recommended to follow the order provided as the schedule will allow to build up the required knowledge.
I do not take credit about those external resources, but I relate and agree with. In this, you can find tips and suggestion extracted from my own career and from what I learned so far.
This is not a bible nor a reference manual, I strongly recommend you to do not follow any suggestion blindly without understanding the reasons behind what you'll read.
I hope while reading and maybe assimilating (un)consciously a few concepts, you'll find a way to improve yourself.
That's the aim of this book!

Disclaimer: Copyright & Legal & IP

Most of the data contained are of public domain, available on Google Search. The ebook, the website https://10xse.academy/ and the course itself is built on the 10x SE Learning Path: scouting, gathering, categorisation, structuring outline, maintenance, architecture & infrastructure, and so on. The links provided to the external content, and the external content itself, belongs to the content's author(s) and they are provided as-is: more details on T&C[5]. Should you wish to have your website removed can be done in the removal page[6].

[5] https://10xse.academy/wpautoterms/terms-and-conditions/
[6] https://10xse.academy/website-removal

Disclaimer: 10x Myth

I do believe in the 10x, in the sense that one developer could achieve x-times more of others developers in certain circumstances, for example, previous knowledge of the domain, previous knowledge of the language/framework, knowledge of the project, some good time management skills.

There's so much hype about the "myth" of the 10x engineer. Please do read more about it:

- The origins of the 10x developer[7]
- What makes a 10x programmer/software engineer?[8]
- The 10x Programmer: Is Individual Productivity Overrated?[9]
- The "10X Engineer" Has Officially Become a Meme[10]
- How To Become A 10X Engineer[11]
- Youâ€™ll never be a 10x Developer[12]

[7] https://medium.com/ingeniouslysimple/the-origins-of-the-10x-developer-2e0177ecef60
[8] https://www.quora.com/What-makes-a-10x-programmer-software-engineer/answer/Edward-De-Jong
[9] https://thenewstack.io/the-veracity-and-relevance-of-the-10x-programmer/
[10] https://www.7pace.com/blog/10x-engineers
[11] https://blog.codegiant.io/how-to-become-a-10x-engineer-492fa3f57101
[12] https://medium.com/dev-genius/youll-never-be-a-10x-developer-3312c1f003ed

Disclaimer: External Links and Reading Time

All the displayed reading time are just an approximation based on average reading times, most people read around 250/300 wpm, so as a baseline I've used 250 words per minute, and an average page length of 450 words per page. This is pretty standard relaxed time, to read one page it'll take into consideration 1 minute and 48 seconds, not too slow not too fast. If you're a faster reader you can devour all those links in less time than suggested, if you're slower than that who cares, as long as you can get the concepts :)

You might be wondering why to create a book on external content, can't I create my own?! Sure, butâ€¦ Those external contents are there for a reason, to be shared. In this way, you'll know many Developers or Architects or CTOs who are sharing their knowledge, rather than just share only my own. You could bookmark those blogs, subscribe to their feeds, read the other books, check the related news or similar books, get the most up to date content from their latest video on YouTube, or attend to a conference where they have a talk. Coming from different roles, different backgrounds, different styles. Cross-contamination of ideas and skills, it's great to learn from others and grow faster than expected.

Since the whole book is based on a curated collection of external resources, despite the effort taken to make sure all the links are always available, it might happen that some of them are not available any more.Â
Do not worry about it! I've got you covered, so does the Internet Archive.Â
By using the Wayback Machine[13] you can access to a snapshot of the page and read the stored content. If you want to know more about the web archiving functionality there's a Help Center[14] section full of details.

[13] https://archive.org/web

How to Use this Book

- Always start from the lowest level, even if you're more experienced.
  You'll never know you'll find something interesting.
- Use the Checklists to mark your progress.
  Each item has a checkbox (â–) in front of, used it with a pencil to record what you have read/studied.

10xSE Academy

The ebook is the foundation of the learning path, so I've decided to improve it and add an online course with assignments, exercises, book reviews, grades, newsletter, peer reviews and much more.

I can offer two special discounts just for the ebook readers:

- **14.99 EUR (instead of 47.88 - save 32.89)** for **1 year subscription access** for all courses and materials, even future ones.
- **24.99 EUR** for **lifetime access** for all courses and materials, even future ones.

To redeem the offer just go to https://10xse.academy/redeem-ebook and enter the code `Z98DQ8A0S3O`.

[14] https://help.archive.org/hc/en-us/categories/360000553851-The-Wayback-Machine

Level: Advanced Beginner

Level Definition

> The novice evolves by figuring out the mistakes in his work. The newly, “promoted” advanced beginner dwells into the world of troubleshooting. Unfortunately, the hasty mindset is not lost, and the individual still aims to acquire results fast, in this case gain knowledge and information. An example would be when a coder with years of experience starts learning a new program language, he could be a master in PHP but an advanced beginner in Python. Scrolling through the documentation will not lead to productive results.
>
> - https://www.360pmo.com/the-five-dreyfus-model-stages/

Duration: 13 weeks (∼3 months)Â
Average per week: ∼11 hours

Schedule

- Week 1: Build Up Dictionary
- Week 2: Ground Rules & Manifestos
- Week 3: SDLC
- Week 4: Algorithms
- Week 5: OOP
- Week 6: Design Patterns
- Week 7: Design Patterns
- Week 8: Asynchronous Programming
- Week 9: Testing
- Week 10: Refactoring
- Week 11: Refactoring
- Week 12: DB
- Week 13: Extra Resources

Week 1: Build Up Dictionary

0-9

2FA - Two Factor Authentication

The 2FA is an additional layer of security you can set up to keep your account secure. It requires a unique one-time use code.

A

A/B Testing

A/B testing is an experimentation method by comparing two (or more) variants of a webpage or app against each other to determine which one performs better.

AAA - Authentication Authorization Accounting

AAA refers to Authentication, Authorization and Accounting.

ACID - Atomicity Consistency Isolation Durability

An ACID database system guarantees that transactions are processed reliably, following the 4 properties: Atomicity, Consistency, Isolation, Durability.

Active Record

Active Record pattern is a pattern used to stores object data in the database. There's a direct relationship with the database schema and the basic CRUD operations.

Agile

Agile is an iterative approach to project management that helps to deliver value to the customers faster.

Algorithm

A logical approach which is a well-defined list of steps that allows a computer to solve a problem.

Antipattern

Antipatterns are apparently appropriate solutions to problems, but in reality, are ineffective or results in unexpected consequences.

API - Application Programming Interface

An API is an interface that let your service communicate with external services without them knowing the implementation details.

B

Big Data

Big data is a term that describes a large volume of data, that can be analysed for better decision making insights.

Blockchain

Blockchain is a shared, distributed and immutable ledger that facilitates the process of recording transactions.

Bug

A bug is an error, flaw or fault in a computer program that causes an incorrect or unexpected result.

C

Cache

A cache is a data storage layer which allows you to quickly serve previously retrieved or computed data.

CDN - Content Delivery Network

A CDN is a geographically distributed platform that helps reducing delays in web page content loading by reducing the physical distance between the server and the user.

Chatbots

A chatbot is an artificial intelligence software that can simulate a conversation with a user via messaging applications.

CLS - Cumulative Layout Shift

CLS is a metric for measuring visual stability of the web page, it helps quantify how often users experience unexpected layout shifts.

Code Review

A code review is the process (manual or automatic) of checking the source code for mistakes and improve the overall quality.

Container

A container is a set of processes that are isolated from the rest of the system. They are portable and consistent through different environments.

CORS - Cross-Origin Resource Sharing

CORS is a mechanism that grants the browsers access to resources outside the scope of the current origin.

CRUD - Create Read Update Delete

CRUD are the four basic functions that models should be able to do to implement persistent storage.

D

Database Normalization

Database Normalization is a technique for eliminating data redundancy.

Deadlock

A deadlock occurs when two threads are holding the locked variable that the other thread wants, nothing occurs, and the threads remain deadlocked.

Design Patterns

A design pattern is a general repeatable solution to a common problem in software design.

DOMContentLoaded

The DOMContentLoaded event is triggered when the HTML has been completely loaded and parsed, without waiting for assets to load.

E

EAV - Entity Attribute Value

Entity-attribute-value is a data model used to store a variable number of entity attributes in a table's space-efficient manner.

Environment Variable

An environment variable is a named, global, shared variable that contains data used by one or more applications.

F

FID - First Input Delay

FID is a metric for measuring the time from when a user first interacts with a page.

FCP - First Contentful Paint

FCP is a metric for measuring the time from when the page starts loading to when any part of the page's content is rendered on the screen.

G

GDPR - General Data Protection Regulation

The GDPR is a privacy and security law, drafted by the European Union, it defines obligations to worldwide organizations in case they're involved with data related to people in the EU.

GTD - Getting Things Done

Getting Things Done is a time management method, it is based on recording tasks and activities and breaking them down in actionable work.

H

Hydration

Most ORMs are performing a hydration process when converting database results into objects, it usually involves reading on-the-fly a record (or additional fields) and then make them available in the record object.

I

Idempotence

An HTTP method is idempotent if an identical request can be made more than once without any side-effects leaving the server in the same state.

Information Architecture

Information Architecture focuses on organizing and effectively structuring content.

Invariant

An invariant is a property of the program state that is always conceptually true.

IoT - Internet of Things

The Internet of Things describes the network of physical objects with embedded technologies to connect and exchange data with other devices and systems over the internet.

K

Kanban

Kanban is a framework used for agile software development, where work items are visually available for the team members to know the state of every piece of work at any time.

Key-Value

A key-value database is a type of non-relational database that uses simple key-value pairs to store data.

L

LCP - Largest Contentful Paint

LCP is a metric reporting the render time of the largest image or text block visible within the viewport.

Lean

Lean is a management philosophy inspired by Toyota practices to minimise risk and waste while maximizing customer value.

Legacy

A legacy system is an old system still in use, that usually drives the business, that is is out of date or in need of replacement.

M

Machine Learning

Machine learning is a data analysis method that allows systems to learn from data, identify patterns and make decisions with minimal human intervention.

MFA - Multi-Factor Authentication

MFA may use three or more checks to verify and authenticate customer identity.

Mobile-First

A mobile-first approach involves designing a website starting with the mobile version, with your mobile users in mind, and then adapt to larger screens.

Mockup

Mockups are essentially wireframes with an added surface layer that communicates the visual design to suggests what the final design will look like.

Murphy's Law

If something can go wrong, it will.

MVP - Minimum Viable Product

An MVP is a version of a new product which allows collecting assumptions about customers with the least effort.

N

NoSQL - Not Only SQL

NoSQL databases are non-tabular (document, key-value, wide-column, and graph) and store data differently than relational tables.

O

OOP - Object Oriented Programming

OOP is a programming model that organizes software design around objects, rather than functions and logic, that has unique attributes and behaviour.

ORM - Object Relational Mapping

ORM allows writing SQL queries using the object-oriented paradigm of a programming language.

OWASP - Open Web Application Security Project

The OWASP is a non-profit foundation that works to improve the security of software.

P

Pair Programming

Pair programming is a collaborative way of developing code in pair and involves a driver and a navigator.

PoC - Proof of Concept

A Proof of Concept is a demonstration to verify that certain concepts of a project or product are feasible and worthy enough to justify the expenses needed to support them.

PWA - Progressive Web App

PWA is a web app built to look and feel an actual native app.

Q

QA - Quality Assurance

Quality Assurance is a process for preventing mistakes and defects when delivering products or services to customers.

R

Race Condition

A race condition occurs when two threads write a shared variable at the same time.

RDBMS - Relational Database Management System

Relational Database Management System is an advanced version of a DBMS, stores the data in the form of tables, rows and columns.

RegEx

A regular expression allows you to create patterns that help match, locate, and manage text.

Remote Work

Remote work allows professionals to work outside of a traditional office environment.

REST - REpresentational State Transfer

REST is an architectural style for distributed APIs.

RPC - Remote Procedure Call

A RPC is a protocol that let you communicate with external services without knowing the implementation details.

Reverse Engineering

Reverse Engineering is a process of acquiring the knowledge of how a software works by recovering the design and specifications of a product from an analysis of its code.

Rubber Duck

Rubber duck debugging is a method of debugging code, by forcing one to explain it, line-by-line, to the duck.

S

Scrum

Scrum is a framework that allows people to address complex problems while delivering products.

SEO - Search Engine Optimization

SEO is the practice of increasing a website traffic quantity/quality through organic search engine results.

Spikes

A spike is a user story for which the team cannot estimate the effort needed, so it will be executed in a time-boxed exploration to learn about the issue or the possible solutions.

SOAP - Simple Object Access Protocol

SOAP is an XML-based protocol for accessing web services over HTTP.

SOLID

SOLID is an acronym for the five object-oriented design principles to develop a software that is easy to maintain and extend: Single-responsibility principle, Open-closed principle, Liskov substitution principle, Interface segregation principle, Dependency Inversion Principle.

SPA - Single Page Application

A SPA is a web application that dynamically rewrites the current web page with new data from the webserver based on user interactions.

SQL - Structured Query Language

SQL is a programming language used in a relational database.

SSL/TLS

SSL/TLS works by binding websites to a cryptographic key pair via certificates.

T

TBT - Total Blocking Time

TBT is a metric for measuring the total amount of time between FCP and TTI where the main thread was blocked for long enough to prevent input responsiveness.

Testing

Testing is the activity (manual or automated) of checking whether the actual results match the expected results and to ensure that the software system is free from defects.

TTFB - Time To First Byte

TTFB is the amount of time it takes to receive the first byte of an HTTP request from the webserver.

TTI - Time to Interactive

TTI is a metric for measuring the time from when the page starts loading to when its main sub-resources have loaded and it is capable of reliably responding to user input quickly.

V

VCS - Version Control System

Version control systems are tools that help to manage changes over time.

W

Waterfall

The waterfall model is a strictly linear and sequential of phases, where each phase depends on the deliverables of the previous one.

Web Vitals

Web Vitals are metrics developed by Google to measure quality signals for delivering great user experience.

Wireframe

A wireframe is a schematic model that is useful to communicate about the structure of the software or website.

WSDL - Web Services Description Language

WSDL is an XML format for describing services end-points and message formats.

X

XP - Extreme Programming

Extreme Programming is a software development methodology to improve software quality and responsiveness to changing customer requirements.

Week 2: Ground Rules & Manifestos

Time needed: 20 hours

Coding Standards

1. Semantic Versioning 2.0.0[15]
   Semantic Versioning 2.0.0 | Semantic Versioning
2. Arlo's Commit Notation[16]
   GitHub - RefactoringCombos/ArlosCommitNotation: A notation for small commits messages that show the risk involved in each step
3. Awesome Guidelines[17]
   GitHub - Kristories/awesome-guidelines: A curated list of high quality coding style conventions and standards.
4. HTTP headers for the responsible developer[18]
   HTTP headers for the responsible developer - Twilio
5. Pragmatic Programming Cheat Sheet[19]
   Pragmatic Programming Cheat Sheet by marconlsantos - Download free from Cheatography - Cheatography.com: Cheat Sheets For Every Occasion

[15] https://semver.org/
[16] https://github.com/RefactoringCombos/ArlosCommitNotation
[17] https://github.com/Kristories/awesome-guidelines
[18] https://www.twilio.com/blog/a-http-headers-for-the-responsible-developer
[19] https://cheatography.com/marconlsantos/cheat-sheets/pragmatic-programming/

Etiquette

1. Netiquette[20]
   Netiquette : Florida Atlantic University
2. 50 Amazing Office Etiquette Tips to Transform Your Company Culture[21]
   50 Amazing Office Etiquette Tips to Transform Your Company Culture - Small Business Trends
3. Dev etiquettes that you must not ignore[22]
   Dev etiquettes that you must not ignore | by Madhav Bahl | codeburst
4. Seven principles of pair programming etiquette[23]
   Seven principles of pair programming etiquette | by Juntao Qiu | ITNEXT
5. Developer Etiquette â€“ Code Review and Pull Request Comments[24]
   Pull Request Etiquette - A set of simple rules for your code review Â· Erik Zaadi

Job

1. 6 Ways To Get Noticed At Work[25]
   6 Ways To Get Noticed At Work - Business Insider Business Insider logo Close icon Loading Menu icon Search icon Business Insider logo Account icon Account icon Business Life News Reviews Search icon Insider logo Close icon Business Life News All Account icon World globe Facebook Icon Twitter icon LinkedIn icon YouTube icon Instagram icon Business

[20] https://www.fau.edu/oit/student/netiquette.php
[21] https://smallbiztrends.com/2017/06/office-etiquette.html
[22] https://codeburst.io/dev-etiquettes-that-you-must-not-ignore-619e1bb490b8
[23] https://itnext.io/seven-principles-of-pair-programming-etiquette-74a2b3b233b0
[24] https://erikzaadi.com/2019/09/29/pull-request-etiquette-a-set-of-simple-rules-for-your-code-review/
[25] https://www.businessinsider.com/6-ways-to-get-noticed-at-work-2013-8

Insider logo Close icon Chevron icon Chevron icon Facebook Icon Email icon Link icon Twitter icon LinkedIn icon Fliboard icon More icon Close icon Loading Close icon

2. How to Make Yourself Indispensable at Work[26]
   How to Make Yourself Indispensable at Work
3. Seven Ways to Be a Good Employee and Make Your Boss Happy[27]
   Seven Ways to Be a Good Employee and Make Your Boss Happy
4. The Top 10 Signs That You Are An Impostor At Work[28]
   The Top 10 Signs That You Are An Impostor At Work
5. Internal Developer Training: Doing It Right[29]
   Internal Developer Training: Doing It Right â€" Smashing Magazine Clear Search Back to top
6. 6 Stupid Mistakes Smart Developers Should Make[30]
   6 Stupid Mistakes Smart Developers Should Make - SitePoint SitePoint
7. Five Career Mistakes That Might Be Holding You Back[31]
   Five Career Mistakes That Might Be Holding You Back

Manifestos

1. Manifesto for Agile Software Development[32]
   Manifesto for Agile Software Development

[26] https://lifehacker.com/how-to-make-yourself-indispensable-at-work-1113590784
[27] https://lifehacker.com/seven-ways-to-be-a-good-employee-and-make-your-boss-hap-1622335033
[28] https://www.forbes.com/sites/kathycaprino/2013/08/14/the-top-10-signs-that-you-are-an-impostor-at-work
[29] https://www.smashingmagazine.com/2014/09/internal-developer-training-doing-it-right/
[30] https://www.sitepoint.com/6-stupid-mistakes-smart-developers-should-make/
[31] https://lifehacker.com/five-career-mistakes-that-might-be-holding-you-back-1596535994
[32] https://agilemanifesto.org/

2. Manifesto for Software Craftsmanship[33]
   Manifesto for Software Craftsmanship
3. Refactoring Manifesto[34]
   Refactoring Manifesto - Because the world needs better code
4. Software disenchantment[35]
   Software disenchantment @ tonsky.me
5. The Reactive Manifesto[36]
   The Reactive Manifesto

Roadmaps

1. Developer Roadmaps[37]
   Developer Roadmaps

Roles

1. Job Titles & Levels: What Every Software Engineer Needs to Know[38]
   Job Titles & Levels: What Every Software Engineer Needs to Know â€" Holloway
2. How to Find Your Career Path[39]
   How to Find Your Career Path
3. The Taxonomy of Terrible Programmers[40]
   The Taxonomy of Terrible Programmers â€“ Aaronontheweb

[33] http://manifesto.softwarecraftsmanship.org/
[34] https://refactoringmanifesto.org/
[35] https://tonsky.me/blog/disenchantment/
[36] https://www.reactivemanifesto.org/
[37] https://roadmap.sh/
[38] https://www.holloway.com/s/trh-job-titles-levels-fundamentals-for-software-engineering
[39] https://lifehacker.com/top-10-ways-to-find-your-career-path-1628537579
[40] http://www.aaronstannard.com/the-taxonomy-of-terrible-programmers/

4. The full-stack employee[41]
   The full-stack employee. Defining a new class of hybrid worker. | by Chris Messina | Chris Messina | Medium

Skills

1. Evergreen Skills for Software Developers[42]
   GitHub - romenrg/evergreen-skills-developers: List of evergreen skills, based on software development best practices & cross-framework principles, that should serve as a fair assessment of skilled software engineers / developers

VCS

1. 6 Version Control Systems Reviewed[43]
   6 Version Control Systems Reviewed â€" Smashing Magazine Clear Search Back to top
2. CS Visualized: Useful Git Commands[44]

Books

1. The Mythical Man-Month[45]

[41] https://medium.com/chris-messina/the-full-stack-employee-ed0db089f0a1
[42] https://github.com/romenrg/evergreen-skills-developers
[43] https://www.smashingmagazine.com/2008/09/the-top-7-open-source-version-control-systems/
[44] https://dev.to/lydiahallie/cs-visualized-useful-git-commands-37p1
[45] https://www.amazon.com/Mythical-Man-Month-Essays-Software-Engineering/dp/0201835959

2. Computer Science Distilled[46]
   Computer Science Distilled

[46] https://sourcemaking.com/computer-science-distilled

Notes

(free space)

Notes

(free space)

Week 3: SDLC

Time needed: 13 hours

Agile

1. Explaining Agile[47]
   Explaining Agile
2. Agile Patterns[48]
   Agile Patterns - Dzone Refcardz

Time Management

1. The Pomodoro TechniqueÂ®[49]
   The Pomodoro TechniqueÂ® - proudly developed by Francesco Cirillo | Cirillo Consulting GmbH
2. Types of Procrastination (And How To Fix Procrastination And Start Doing)[50]

[47] https://www.forbes.com/sites/stevedenning/2016/09/08/explaining-agile/
[48] https://dzone.com/refcardz/agile-patterns
[49] https://francescocirillo.com/pages/pomodoro-technique
[50] https://www.lifehack.org/articles/productivity/types-procrastination-and-how-you-can-fix-them.html

XP

1. Essential XP: Emergent Design[51]
   Essential XP: Emergent Design
2. BeckDesignRules[52]
   BeckDesignRules

Mix

1. Zero trust architecture design principles[53]
   GitHub - ukncsc/zero-trust-architecture: Principles to help you design and deploy a zero trust architecture
2. Principles of secure development & deployment[54]
   GitHub - ukncsc/secure-development-and-deployment: NCSC Guidance for secure development and deployment

---

Books

1. Agile Estimating and Planning[55]
2. Agile Patterns[56]
   Agile Patterns - Dzone Refcardz

[51] https://ronjeffries.com/xprog/classics/expemergentdesign/
[52] https://martinfowler.com/bliki/BeckDesignRules.html
[53] https://github.com/ukncsc/zero-trust-architecture
[54] https://github.com/ukncsc/secure-development-and-deployment
[55] https://www.amazon.com/Agile-Estimating-Planning-Mike-Cohn/dp/0131479415
[56] https://dzone.com/refcardz/agile-patterns

Notes

(free space)

Notes

(free space)

Week 4: Algorithms

Time needed: 2 hours

Articles

1. Algorithms {fundamental techniques}[57]
   Algorithms - Wikibooks, open books for an open world
2. Algorithms[58]
   Algorithms - GeeksforGeeks
3. IDEA â€“ nonverbal algorithm assembly instructions[59]
   IDEA â€“ nonverbal algorithm assembly instructions
4. Why do students fail in Algorithms and Data Structure Interviews for Top Companies? | by Shubham Gautam | Medium[60]
   Why do students fail in Algorithms and Data Structure Interviews for Top Companies? | by Shubham Gautam | Medium
5. 14 Patterns to Ace Any Coding Interview Question | Hacker Noon[61]
   14 Patterns to Ace Any Coding Interview Question | Hacker Noon

[57] https://en.wikibooks.org/wiki/Algorithms
[58] https://www.geeksforgeeks.org/fundamentals-of-algorithms/
[59] https://idea-instructions.com/
[60] https://medium.com/@shubhamkumargautam/why-do-students-fail-in-algorithms-and-data-structure-interviews-for-top-companies-4fcca7ce7580
[61] https://hackernoon.com/14-patterns-to-ace-any-coding-interview-question-c5bb3357f6ed

6. Data Structures 101: Graphs â€" A Visual Introduction for Beginners | by Estefania Cassingena Navone | freeCodeCamp.org | Medium[62]
   Data Structures 101: Graphs â€" A Visual Introduction for Beginners | by Estefania Cassingena Navone | freeCodeCamp.org | Medium
7. Sorting Algorithms - LAMFO[63]
   Sorting Algorithms - LAMFO
8. Big-O Algorithm Complexity Cheat Sheet (Know Thy Complexities!) @ericdrowell[64]
   Big-O Algorithm Complexity Cheat Sheet (Know Thy Complexities!) @ericdrowell
9. Time Complexity of Algorithmsâ€" Big O Notation Explained In Plain English | by Yong Cui | The Startup | Medium[65]
   Time Complexity of Algorithmsâ€" Big O Notation Explained In Plain English | by Yong Cui | The Startup | Medium

---

Videos

1. MIT 6.006 Introduction to Algorithms, Fall 2011[66]

[62] https://medium.com/free-code-camp/data-structures-101-graphs-a-visual-introduction-for-beginners-6d88f36ec768
[63] https://lamfo-unb.github.io/2019/04/21/Sorting-algorithms/
[64] https://www.bigocheatsheet.com/
[65] https://medium.com/swlh/time-complexity-of-algorithms-big-o-notation-explained-in-plain-english-e12a11dc4a4f
[66] https://www.youtube.com/playlist?list=PLUl4u3cNGP61Oq3tWYp6V_F-5jb5L2iHb

Notes

(free space)

Notes

(free space)

Week 5: OOP

Time needed: 15 hours

Basics

1. Objects Should Be Immutable[67]
   Objects Should Be Immutable
2. Getters/Setters. Evil. Period.[68]
   Getters/Setters. Evil. Period.
3. Seven Virtues of a Good Object[69]
   Seven Virtues of a Good Object
4. Why NULL is Bad?[70]
   Why NULL is Bad?

Calisthenics

1. Object Calisthenics[71]
   Object Calisthenics | William Durand

[67] https://www.yegor256.com/2014/06/09/objects-should-be-immutable.html
[68] https://www.yegor256.com/2014/09/16/getters-and-setters-are-evil.html
[69] https://www.yegor256.com/2014/11/20/seven-virtues-of-good-object.html
[70] https://www.yegor256.com/2014/05/13/why-null-is-bad.html
[71] https://williamdurand.fr/2013/06/03/object-calisthenics/

Clean Code

1. Clean Code Cheat Sheet[72]

Cohesion & Coupling

1. High Cohesion, Loose Coupling[73]
   High Cohesion, Loose Coupling â€“ A Sleek Geek Blog

SOLID

1. SOLID, GRASP, and Other Basic Principles of Object-Oriented Design[74]
   SOLID, GRASP, and Other Basic Principles of Object-Oriented Design - DZone Web Dev

---

Videos

1. Joshua Thijssen: Paradoxes and theorems every developer should know Video @ DPC2017[75] | Slides[76] 43:28

[72] https://www.bbv.ch/wp-content/uploads/2020/02/200-bbv-Software-Testing-Clean-Code-Cheat-Sheet.pdf
[73] https://thebojan.ninja/2015/04/08/high-cohesion-loose-coupling/
[74] https://dzone.com/articles/solid-grasp-and-other-basic-principles-of-object-o
[75] https://www.youtube.com/watch?v=JBUIIQnVfBQ
[76] https://speakerdeck.com/jaytaph/paradoxes-and-theorems-every-developer-should-know-3

Books

1. Algorithms in a Nutshell[77]
   Amazon.com: Algorithms in a Nutshell: A Practical Guide (9781491948927): Heineman, George T., Pollice, Gary, Selkow, Stanley: Books
2. Code Review Patterns and Anti-Patterns[78]
   Code Review Patterns and Anti-Patterns - Dzone Refcardz
3. InfoQ eMag: Technical Debt and Software Craftsmanship[79]
   InfoQ eMag: Technical Debt and Software Craftsmanship

[77] https://www.amazon.com/Algorithms-Nutshell-Desktop-Quick-Reference/dp/1491948922
[78] https://dzone.com/refcardz/code-review-patterns-and-anti-patterns
[79] https://www.infoq.com/minibooks/emag-technical-debt/

Notes

(free space)

Notes

(free space)

Week 6: Design Patterns

Time needed: 15 hours

Creational

1. Creational patterns[80]
   Creational patterns
2. Abstract Factory[81]
   Abstract Factory Design Pattern
3. Builder[82]
   Builder Design Pattern
4. Factory Method[83]
   Factory Method Design Pattern
5. Object Pool[84]
   Object Pool Design Pattern
6. Prototype[85]
   Prototype Design Pattern
7. Singleton[86]
   Singleton Design Pattern

[80] https://sourcemaking.com/design_patterns/creational_patterns
[81] https://sourcemaking.com/design_patterns/abstract_factory
[82] https://sourcemaking.com/design_patterns/builder
[83] https://sourcemaking.com/design_patterns/factory_method
[84] https://sourcemaking.com/design_patterns/object_pool
[85] https://sourcemaking.com/design_patterns/prototype
[86] https://sourcemaking.com/design_patterns/singleton

Structural

1. Structural patterns[87]
   Structural patterns
2. Adapter[88]
   Adapter Design Pattern
3. Bridge[89]
   Bridge Design Pattern
4. Composite[90]
   Composite Design Pattern
5. Decorator[91]
   Decorator Design Pattern
6. Facade[92]
   Facade Design Pattern
7. Flyweight[93]
   Flyweight Design Pattern
8. Private Class Data[94]
   Private Class Data
9. Proxy[95]
   Proxy Design Pattern

Behavioral

1. Behavioral patterns[96]
   Behavioral patterns

[87] https://sourcemaking.com/design_patterns/structural_patterns
[88] https://sourcemaking.com/design_patterns/adapter
[89] https://sourcemaking.com/design_patterns/bridge
[90] https://sourcemaking.com/design_patterns/composite
[91] https://sourcemaking.com/design_patterns/decorator
[92] https://sourcemaking.com/design_patterns/facade
[93] https://sourcemaking.com/design_patterns/flyweight
[94] https://sourcemaking.com/design_patterns/private_class_data
[95] https://sourcemaking.com/design_patterns/proxy
[96] https://sourcemaking.com/design_patterns/behavioral_patterns

2. Chain of Responsibility[97]
   Chain of Responsibility
3. Command[98]
   Command Design Pattern
4. Interpreter[99]
   Interpreter Design Pattern
5. Iterator[100]
   Iterator Design Pattern
6. Mediator[101]
   Mediator Design Pattern
7. Memento[102]
   Memento Design Pattern
8. Null Object[103]
   Null Object Design Pattern
9. Observer[104]
   Observer Design Pattern
10. State[105]
    State Design Pattern
11. Strategy[106]
    Strategy Design Pattern
12. Template Method[107]
    Template Method Design Pattern
13. Visitor[108]
    Visitor Design Pattern

[97] https://sourcemaking.com/design_patterns/chain_of_responsibility
[98] https://sourcemaking.com/design_patterns/command
[99] https://sourcemaking.com/design_patterns/interpreter
[100] https://sourcemaking.com/design_patterns/iterator
[101] https://sourcemaking.com/design_patterns/mediator
[102] https://sourcemaking.com/design_patterns/memento
[103] https://sourcemaking.com/design_patterns/null_object
[104] https://sourcemaking.com/design_patterns/observer
[105] https://sourcemaking.com/design_patterns/state
[106] https://sourcemaking.com/design_patterns/strategy
[107] https://sourcemaking.com/design_patterns/template_method
[108] https://sourcemaking.com/design_patterns/visitor

Books

1. Clean Code[109]

[109] https://www.amazon.com/Clean-Code-Handbook-Software-Craftsmanship/dp/0132350882

Notes

(free space)

Notes

(free space)

Week 7: Design Patterns

Time needed: 13 hours

UML

1. A Comprehensive Guide to 14 Types of UML Diagram[110]

Books

1. Design Patterns[111]

[110] https://warren2lynch.medium.com/a-comprehensive-guide-to-14-types-of-uml-diagram-affcc688377e

[111] https://www.amazon.com/Design-Patterns-Elements-Reusable-Object-Oriented/dp/0201633612

Notes

(free space)

Notes

(free space)

Week 8: Asynchronous Programming

Time needed: 2 hours

Basics

1. General asynchronous programming concepts[112]
   General asynchronous programming concepts - Learn web development | MDN
2. Async/Await - Best Practices in Asynchronous Programming | Microsoft Docs[113]
   Async/Await - Best Practices in Asynchronous Programming | Microsoft Docs
3. When to Use (and Not to Use) Asynchronous Programming: 20 Pros Reveal the Best Use Cases - DZone DevOps[114]
   When to Use (and Not to Use) Asynchronous Programming: 20 Pros Reveal the Best Use Cases - DZone DevOps
4. Asynchronous and Parallel Programming in C# .NET | by Thanh Le | Medium[115]

[112] https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous/Concepts

[113] https://docs.microsoft.com/en-us/archive/msdn-magazine/2013/march/async-await-best-practices-in-asynchronous-programming

[114] https://dzone.com/articles/when-to-use-and-not-to-use-asynchronous-programmin

[115] https://medium.com/@letienthanh0212/asynchronous-and-parallel-programming-in-c-net-1e0f14e1db80

Asynchronous and Parallel Programming in C# .NET | by Thanh Le | Medium

5. Reactive in practice: Concurrency, parallelism, asynchrony â€“ IBM Developer[116]
   Reactive in practice: Concurrency, parallelism, asynchrony â€“ IBM Developer Close Favorite this Thumbs up Show more icon Show more icon Show more icon Show more icon Show more icon Show more icon Show more icon Show more icon Show more icon Show more icon Show more icon Show more icon Show more icon Share this on Facebook Share this on Twitter Share this on LinkedIn Share this on WeChat Arrow down Arrow up Close Modal

Messages

1. Messaging patterns[117]
   Messaging patterns - Cloud Design Patterns | Microsoft Docs

[116] https://developer.ibm.com/languages/java/tutorials/reactive-in-practice-4/
[117] https://docs.microsoft.com/en-us/azure/architecture/patterns/category/messaging

Notes

(free space)

Notes

(free space)

Week 9: Testing

Time needed: 14 hours

Arrange Act Assert

1. 3A â€“ Arrange, Act, Assert[118]
   3A - Arrange, Act, Assert - XP123

Bottlenecks

1. Big List Of 20 Common Bottlenecks[119]
   Big List of 20 Common Bottlenecks - High Scalability -

Mocks & Stubs

1. Mocks Aren't Stubs[120]
   Mocks Aren't Stubs
2. TestDouble[121]
   TestDouble

[118] https://xp123.com/articles/3a-arrange-act-assert/
[119] http://highscalability.com/blog/2012/5/16/big-list-of-20-common-bottlenecks.html
[120] https://martinfowler.com/articles/mocksArentStubs.html
[121] https://martinfowler.com/bliki/TestDouble.html

Mutation

1. Mutation Testing in Software Testing: Mutant Score & Analysis Example[122]
   Mutation Testing in Software Testing: Mutant Score & Analysis Example

Test Pyramid

1. The Practical Test Pyramid[123]
   The Practical Test Pyramid
2. Types Of Software Testing: Different Testing Types With Details[124]
   Types of Software Testing: Different Testing Types with Details

Test Driven Development

1. Introduction to Test Driven Development (TDD)[125]
   Introduction to Test Driven Development (TDD)
2. test && commit || revert[126]
   test && commit || revert. As part of Limbo on the Cheap, weâ€¦ | by Kent Beck | Medium

Mix

1. Testing Microservices: an Overview of 12 Useful Techniques

[122] https://www.guru99.com/mutation-testing.html
[123] https://martinfowler.com/articles/practical-test-pyramid.html
[124] https://www.softwaretestinghelp.com/types-of-software-testing/
[125] http://agiledata.org/essays/tdd.html
[126] https://medium.com/@kentbeck_7670/test-commit-revert-870bbd756864

- Part 1[127]
  Testing Microservices: an Overview of 12 Useful Techniques - Part 1
2. Testing Microservices: Examining the Tradeoffs of Twelve Techniques - Part 2[128]
   Testing Microservices: Examining the Tradeoffs of Twelve Techniques - Part 2

Videos

1. Kevlin Henney: Enterprise Programming Tricks For Clean Code Video[129] | Slides[130]
2. Sander Hoogendoorn: How Thinking Small is Changing Software Development Big Time Video @ GOTO 2019[131] | Slides[132]
3. Mixed Paradigms: The Method to Madness. Venkat Subramaniam, Agile developer, Inc[133]
   Mixed Paradigms: The Method to Madness. Venkat Subramaniam, Agile developer, Inc - YouTube

Books

1. Test Driven Development: By Example[134]

[127] https://www.infoq.com/articles/twelve-testing-techniques-microservices-intro/
[128] https://www.infoq.com/articles/twelve-testing-techniques-microservices-tradeoffs/
[129] https://www.youtube.com/watch?v=dC9vdQkU-xI
[130] https://www.slideshare.net/Kevlin/clean-coders-hate-what-happens-to-your-code-when-you-use-these-enterprise-programming-tricks-77305014
[131] https://www.youtube.com/watch?v=YCQMiFF9QXM
[132] https://www.slideshare.net/aahoogendoorn/its-a-small-world-after-all-how-thinking-small-changes-software-big-time
[133] https://www.youtube.com/watch?v=QYBRifsWHD0
[134] https://www.amazon.com/Test-Driven-Development-By-Example/dp/0321146530

Notes

(free space)

Notes

(free space)

Week 10: Refactoring

Time needed: 16 hours

Articles

1. The Art of Enbugging[135]
2. 9 Anti-Patterns Every Programmer Should Be Aware Of[136]
   9 Anti-Patterns Every Programmer Should Be Aware Of
3. Provable Refactorings[137]
   GitHub - digdeeproots/provable-refactorings: A collection of refactoring recipes that are provably safe. They never accidentally introduce nor fix a bug, including one that you don't know exists. They maintain all behavior, including unknown or unspecified behavior. To accomplish this, each recipe is concrete and language-specific.
4. The Most Dangerous Word In Software Development[138]
   The Most Dangerous Word In Software Development â€“ A List Apart
5. Understanding Fake Agile[139]
   Understanding Fake Agile

[135] https://www2.ccs.neu.edu/research/demeter/related-work/pragmatic-programmer/jan_03_enbug.pdf

[136] https://sahandsaba.com/nine-anti-patterns-every-programmer-should-be-aware-of-with-examples.html

[137] https://github.com/digdeeproots/provable-refactorings

[138] https://alistapart.com/blog/post/the-most-dangerous-word-in-software-development/

[139] https://www.forbes.com/sites/stevedenning/2019/05/23/understanding-fake-agile

6. Why Do Managers Hate Agile?[140]
   Why Do Managers Hate Agile?

Code Smells

Bloaters

1. Long Method[141]
   Long Method
2. Large Class[142]
   Large Class
3. Primitive Obsession[143]
   Primitive Obsession
4. Long Parameter List[144]
   Long Parameter List
5. Data Clumps[145]
   Data Clumps”

Object-Orientation Abusers

1. Switch Statements[146]
   Switch Statements
2. Temporary Field[147]
   Temporary Field
3. Refused Bequest[148]
   Refused Bequest

[140] https://www.forbes.com/sites/stevedenning/2015/01/26/why-do-managers-hate-agile/
[141] https://sourcemaking.com/refactoring/smells/long-method
[142] https://sourcemaking.com/refactoring/smells/large-class
[143] https://sourcemaking.com/refactoring/smells/primitive-obsession
[144] https://sourcemaking.com/refactoring/smells/long-parameter-list
[145] https://sourcemaking.com/refactoring/smells/data-clumps
[146] https://sourcemaking.com/refactoring/smells/switch-statements
[147] https://sourcemaking.com/refactoring/smells/temporary-field
[148] https://sourcemaking.com/refactoring/smells/refused-bequest

4. Alternative Classes with Different Interfaces[149]
   Alternative Classes with Different Interfaces”

Change Preventers

1. Divergent Change[150]
   Divergent Change
2. Shotgun Surgery[151]
   Shotgun Surgery
3. Parallel Inheritance Hierarchies[152]
   Parallel Inheritance Hierarchies”

Dispensables

1. Comments[153]
   Comments
2. Duplicate Code[154]
   Duplicate Code
3. Lazy Class[155]
   Lazy Class
4. Data Class[156]
   Data Class
5. Dead Code[157]
   Dead Code
6. Speculative Generality[158]
   Speculative Generality”

[149] https://sourcemaking.com/refactoring/smells/alternative-classes-with-different-interfaces
[150] https://sourcemaking.com/refactoring/smells/divergent-change
[151] https://sourcemaking.com/refactoring/smells/shotgun-surgery
[152] https://sourcemaking.com/refactoring/smells/parallel-inheritance-hierarchies
[153] https://sourcemaking.com/refactoring/smells/comments
[154] https://sourcemaking.com/refactoring/smells/duplicate-code
[155] https://sourcemaking.com/refactoring/smells/lazy-class
[156] https://sourcemaking.com/refactoring/smells/data-class
[157] https://sourcemaking.com/refactoring/smells/dead-code
[158] https://sourcemaking.com/refactoring/smells/speculative-generality

Couplers

1. Feature Envy[159]
   Feature Envy
2. Inappropriate Intimacy[160]
   Inappropriate Intimacy
3. Message Chains[161]
   Message Chains
4. Middle Man[162]
   Middle Man”

Other Smells

1. Incomplete Library Class[163]
   Incomplete Library Class

Refactoring Techniques

Composing Methods

1. Extract Method[164]
   Extract Method
2. Inline Method[165]
   Inline Method
3. Extract Variable[166]
   Extract Variable

[159] https://sourcemaking.com/refactoring/smells/feature-envy
[160] https://sourcemaking.com/refactoring/smells/inappropriate-intimacy
[161] https://sourcemaking.com/refactoring/smells/message-chains
[162] https://sourcemaking.com/refactoring/smells/middle-man
[163] https://sourcemaking.com/refactoring/smells/incomplete-library-class
[164] https://sourcemaking.com/refactoring/extract-method
[165] https://sourcemaking.com/refactoring/inline-method
[166] https://sourcemaking.com/refactoring/extract-variable

4. Inline Temp[167]
   Inline Temp
5. Replace Temp with Query[168]
   Replace Temp with Query
6. Split Temporary Variable[169]
   Split Temporary Variable
7. Remove Assignments to Parameters[170]
   Remove Assignments to Parameters
8. Replace Method with Method Object[171]
   Replace Method with Method Object
9. Substitute Algorithm[172]
   Substitute Algorithm"

Moving Features between Objects

1. Move Method[173]
   Move Method
2. Move Field[174]
   Move Field
3. Extract Class[175]
   Extract Class
4. Inline Class[176]
   Inline Class
5. Hide Delegate[177]
   Hide Delegate

[167] https://sourcemaking.com/refactoring/inline-temp
[168] https://sourcemaking.com/refactoring/replace-temp-with-query
[169] https://sourcemaking.com/refactoring/split-temporary-variable
[170] https://sourcemaking.com/refactoring/remove-assignments-to-parameters
[171] https://sourcemaking.com/refactoring/replace-method-with-method-object
[172] https://sourcemaking.com/refactoring/substitute-algorithm
[173] https://sourcemaking.com/refactoring/move-method
[174] https://sourcemaking.com/refactoring/move-field
[175] https://sourcemaking.com/refactoring/extract-class
[176] https://sourcemaking.com/refactoring/inline-class
[177] https://sourcemaking.com/refactoring/hide-delegate

6. Remove Middle Man[178]
   Remove Middle Man
7. Introduce Foreign Method[179]
   Introduce Foreign Method
8. Introduce Local Extension[180]
   Introduce Local Extension”

Organizing Data

1. Self Encapsulate Field[181]
   Self Encapsulate Field
2. Replace Data Value with Object[182]
   Replace Data Value with Object
3. Change Value to Reference[183]
   Change Value to Reference
4. Change Reference to Value[184]
   Change Reference to Value
5. Replace Array with Object[185]
   Replace Array with Object
6. Duplicate Observed Data[186]
   Duplicate Observed Data
7. Change Unidirectional Association to Bidirectional[187]
   Change Unidirectional Association to Bidirectional
8. Change Bidirectional Association to Unidirectional[188]
   Change Bidirectional Association to Unidirectional

[178] https://sourcemaking.com/refactoring/remove-middle-man
[179] https://sourcemaking.com/refactoring/introduce-foreign-method
[180] https://sourcemaking.com/refactoring/introduce-local-extension
[181] https://sourcemaking.com/refactoring/self-encapsulate-field
[182] https://sourcemaking.com/refactoring/replace-data-value-with-object
[183] https://sourcemaking.com/refactoring/change-value-to-reference
[184] https://sourcemaking.com/refactoring/change-reference-to-value
[185] https://sourcemaking.com/refactoring/replace-array-with-object
[186] https://sourcemaking.com/refactoring/duplicate-observed-data
[187] https://sourcemaking.com/refactoring/change-unidirectional-association-to-bidirectional
[188] https://sourcemaking.com/refactoring/change-bidirectional-association-to-unidirectional

9. Replace Magic Number with Symbolic Constant[189]
   Replace Magic Number with Symbolic Constant
10. Encapsulate Field[190]
    Encapsulate Field
11. Encapsulate Collection[191]
    Encapsulate Collection
12. Replace Type Code with Class[192]
    Replace Type Code with Class
13. Replace Type Code with Subclasses[193]
    Replace Type Code with Subclasses
14. Replace Type Code with State/Strategy[194]
    Replace Type Code with State/Strategy
15. Replace Subclass with Fields[195]
    Replace Subclass with Fields"

Simplifying Conditional Expressions

1. Decompose Conditional[196]
   Decompose Conditional
2. Consolidate Conditional Expression[197]
   Consolidate Conditional Expression
3. Consolidate Duplicate Conditional Fragments[198]
   Consolidate Duplicate Conditional Fragments
4. Remove Control Flag[199]
   Remove Control Flag

[189] https://sourcemaking.com/refactoring/replace-magic-number-with-symbolic-constant
[190] https://sourcemaking.com/refactoring/encapsulate-field
[191] https://sourcemaking.com/refactoring/encapsulate-collection
[192] https://sourcemaking.com/refactoring/replace-type-code-with-class
[193] https://sourcemaking.com/refactoring/replace-type-code-with-subclasses
[194] https://sourcemaking.com/refactoring/replace-type-code-with-state-strategy
[195] https://sourcemaking.com/refactoring/replace-subclass-with-fields
[196] https://sourcemaking.com/refactoring/decompose-conditional
[197] https://sourcemaking.com/refactoring/consolidate-conditional-expression
[198] https://sourcemaking.com/refactoring/consolidate-duplicate-conditional-fragments
[199] https://sourcemaking.com/refactoring/remove-control-flag

5. Replace Nested Conditional with Guard Clauses[200]
   Replace Nested Conditional with Guard Clauses
6. Replace Conditional with Polymorphism[201]
   Replace Conditional with Polymorphism
7. Introduce Null Object[202]
   Introduce Null Object
8. Introduce Assertion[203]
   Introduce Assertion"

Simplifying Method Calls

1. Rename Method[204]
   Rename Method
2. Add Parameter[205]
   Add Parameter
3. Remove Parameter[206]
   Remove Parameter
4. Separate Query from Modifier[207]
   Separate Query from Modifier
5. Parameterize Method[208]
   Parameterize Method
6. Replace Parameter with Explicit Methods[209]
   Replace Parameter with Explicit Methods
7. Preserve Whole Object[210]
   Preserve Whole Object

[200] https://sourcemaking.com/refactoring/replace-nested-conditional-with-guard-clauses
[201] https://sourcemaking.com/refactoring/replace-conditional-with-polymorphism
[202] https://sourcemaking.com/refactoring/introduce-null-object
[203] https://sourcemaking.com/refactoring/introduce-assertion
[204] https://sourcemaking.com/refactoring/rename-method
[205] https://sourcemaking.com/refactoring/add-parameter
[206] https://sourcemaking.com/refactoring/remove-parameter
[207] https://sourcemaking.com/refactoring/separate-query-from-modifier
[208] https://sourcemaking.com/refactoring/parameterize-method
[209] https://sourcemaking.com/refactoring/replace-parameter-with-explicit-methods
[210] https://sourcemaking.com/refactoring/preserve-whole-object

8. Replace Parameter with Method Call[211]
   Replace Parameter with Method Call
9. Introduce Parameter Object[212]
   Introduce Parameter Object
10. Remove Setting Method[213]
    Remove Setting Method
11. Hide Method[214]
    Hide Method
12. Replace Constructor with Factory Method[215]
    Replace Constructor with Factory Method
13. Replace Error Code with Exception[216]
    Replace Error Code with Exception
14. Replace Exception with Test[217]
    Replace Exception with Test”

Dealing with Generalisation

1. Pull Up Field[218]
   Pull Up Field
2. Pull Up Method[219]
   Pull Up Method
3. Pull Up Constructor Body[220]
   Pull Up Constructor Body
4. Push Down Method[221]
   Push Down Method

[211] https://sourcemaking.com/refactoring/replace-parameter-with-method-call
[212] https://sourcemaking.com/refactoring/introduce-parameter-object
[213] https://sourcemaking.com/refactoring/remove-setting-method
[214] https://sourcemaking.com/refactoring/hide-method
[215] https://sourcemaking.com/refactoring/replace-constructor-with-factory-method
[216] https://sourcemaking.com/refactoring/replace-error-code-with-exception
[217] https://sourcemaking.com/refactoring/replace-exception-with-test
[218] https://sourcemaking.com/refactoring/pull-up-field
[219] https://sourcemaking.com/refactoring/pull-up-method
[220] https://sourcemaking.com/refactoring/pull-up-constructor-body
[221] https://sourcemaking.com/refactoring/push-down-method

5. Push Down Field[222]
   Push Down Field
6. Extract Subclass[223]
   Extract Subclass
7. Extract Superclass[224]
   Extract Superclass
8. Extract Interface[225]
   Extract Interface
9. Collapse Hierarchy[226]
   Collapse Hierarchy
10. Form Template Method[227]
    Form Template Method
11. Replace Inheritance with Delegation[228]
    Replace Inheritance with Delegation
12. Replace Delegation with Inheritance[229]
    Replace Delegation with Inheritance

Books

1. Refactoring to Patterns[230]
   Refactoring to Patterns: Kerievsky, Joshua: 0785342213355: Amazon.com: Books

[222] https://sourcemaking.com/refactoring/push-down-field
[223] https://sourcemaking.com/refactoring/extract-subclass
[224] https://sourcemaking.com/refactoring/extract-superclass
[225] https://sourcemaking.com/refactoring/extract-interface
[226] https://sourcemaking.com/refactoring/collapse-hierarchy
[227] https://sourcemaking.com/refactoring/form-template-method
[228] https://sourcemaking.com/refactoring/replace-inheritance-with-delegation
[229] https://sourcemaking.com/refactoring/replace-delegation-with-inheritance
[230] https://www.amazon.com/Refactoring-Patterns-Joshua-Kerievsky/dp/0321213351

Notes

(free space)

Notes

(free space)

Week 11: Refactoring

Time needed: 14 hours

Software Development AntiPatterns

1. The Blob[231]
   The Blob
2. Continuous Obsolescence[232]
   Continuous Obsolescence
3. Lava Flow[233]
   Lava Flow
4. Ambiguous Viewpoint[234]
   Ambiguous Viewpoint
5. Functional Decomposition[235]
   Functional Decomposition
6. Poltergeists[236]
   Poltergeists
7. Boat Anchor[237]
   Boat Anchor

[231] https://sourcemaking.com/antipatterns/the-blob
[232] https://sourcemaking.com/antipatterns/continuous-obsolescence
[233] https://sourcemaking.com/antipatterns/lava-flow
[234] https://sourcemaking.com/antipatterns/ambiguous-viewpoint
[235] https://sourcemaking.com/antipatterns/functional-decomposition
[236] https://sourcemaking.com/antipatterns/poltergeists
[237] https://sourcemaking.com/antipatterns/boat-anchor

8. Golden Hammer[238]
   Golden Hammer
9. Dead End[239]
   Dead End
10. Spaghetti Code[240]
    Spaghetti Code
11. Input Kludge[241]
    Input Kludge
12. Walking through a Minefield[242]
    Walking through a Minefield
13. Cut-And-Paste Programming[243]
    Cut-And-Paste Programming
14. Mushroom Management[244]
    Mushroom Management

Software Architecture AntiPatterns

1. Autogenerated Stovepipe[245]
   Autogenerated Stovepipe
2. Stovepipe Enterprise[246]
   Stovepipe Enterprise
3. Jumble[247]
   Jumble
4. Stovepipe System[248]
   Stovepipe System

[238] https://sourcemaking.com/antipatterns/golden-hammer
[239] https://sourcemaking.com/antipatterns/dead-end
[240] https://sourcemaking.com/antipatterns/spaghetti-code
[241] https://sourcemaking.com/antipatterns/input-kludge
[242] https://sourcemaking.com/antipatterns/walking-through-minefield
[243] https://sourcemaking.com/antipatterns/cut-and-paste-programming
[244] https://sourcemaking.com/antipatterns/mushroom-management
[245] https://sourcemaking.com/antipatterns/autogenerated-stovepipe
[246] https://sourcemaking.com/antipatterns/stovepipe-enterprise
[247] https://sourcemaking.com/antipatterns/jumble
[248] https://sourcemaking.com/antipatterns/stovepipe-system

5. Cover Your Assets[249]
   Cover Your Assets
6. Vendor Lock-In[250]
   Vendor Lock-In
7. Wolf Ticket[251]
   Wolf Ticket
8. Architecture By Implication[252]
   Architecture By Implication
9. Warm Bodies[253]
   Warm Bodies
10. Design By Committee[254]
    Design By Committee
11. Swiss Army Knife[255]
    Swiss Army Knife
12. Reinvent The Wheel[256]
    Reinvent The Wheel
13. The Grand Old Duke of York[257]
    The Grand Old Duke of York

Project Management AntiPatterns

1. Blowhard Jamboree[258]
   Blowhard Jamboree
2. Analysis Paralysis[259]
   Analysis Paralysis

[249] https://sourcemaking.com/antipatterns/cover-your-assets
[250] https://sourcemaking.com/antipatterns/vendor-lock-in
[251] https://sourcemaking.com/antipatterns/wolf-ticket
[252] https://sourcemaking.com/antipatterns/architecture-by-implication
[253] https://sourcemaking.com/antipatterns/warm-bodies
[254] https://sourcemaking.com/antipatterns/design-by-committee
[255] https://sourcemaking.com/antipatterns/swiss-army-knife
[256] https://sourcemaking.com/antipatterns/reinvent-the-wheel
[257] https://sourcemaking.com/antipatterns/the-grand-old-duke-of-york
[258] https://sourcemaking.com/antipatterns/blowhard-jamboree
[259] https://sourcemaking.com/antipatterns/analysis-paralysis

3. Viewgraph Engineering[260]
   Viewgraph Engineering
4. Death By Planning[261]
   Death By Planning
5. Fear of Success[262]
   Fear of Success
6. Corncob[263]
   Corncob
7. Intellectual Violence[264]
   Intellectual Violence
8. Irrational Management[265]
   Irrational Management
9. Smoke and Mirrors[266]
   Smoke and Mirrors
10. Project Mismanagement[267]
    Project Mismanagement
11. Throw It over the Wall[268]
    Throw It over the Wall
12. Fire Drill[269]
    Fire Drill
13. The Feud[270]
    The Feud
14. E-mail Is Dangerous[271]
    E-mail Is Dangerous

---

[260] https://sourcemaking.com/antipatterns/viewgraph-engineering
[261] https://sourcemaking.com/antipatterns/death-by-planning
[262] https://sourcemaking.com/antipatterns/fear-of-success
[263] https://sourcemaking.com/antipatterns/corncob
[264] https://sourcemaking.com/antipatterns/intellectual-violence
[265] https://sourcemaking.com/antipatterns/irrational-management
[266] https://sourcemaking.com/antipatterns/smoke-and-mirrors
[267] https://sourcemaking.com/antipatterns/project-mismanagement
[268] https://sourcemaking.com/antipatterns/throw-it-over-the-wall
[269] https://sourcemaking.com/antipatterns/fire-drill
[270] https://sourcemaking.com/antipatterns/the-feud
[271] https://sourcemaking.com/antipatterns/e-mail-is-dangerous

Books

1. Refactoring[272]

[272] https://www.amazon.com/Refactoring-Improving-Design-Existing-Code/dp/0134757599

Notes

(free space)

Notes

(free space)

Week 12: DB

Time needed: 8 hours

ACID properties

1. A Primer on ACID Transactions: The Basics Every Cloud App Developer Must Know[273]
   A Primer on ACID Transactions: The Basics Every Cloud App Developer Must Know - The Distributed SQL Blog hamburger-white

Consistency

1. Eventual Consistency vs Strong Consistency | by Vivek Kumar Singh | System Design Blog | Medium[274]
   Eventual Consistency vs Strong Consistency | by Vivek Kumar Singh | System Design Blog | Medium

[273] https://blog.yugabyte.com/a-primer-on-acid-transactions/

[274] https://medium.com/system-design-blog/eventual-consistency-vs-strong-consistency-b4de1f92534d

Joins

1. A Visual Explanation of SQL Joins[275]
   A Visual Explanation of SQL Joins

Normalisation

1. Normalization of Database[276]
   1NF, 2NF, 3NF and BCNF in Database Normalization | Studytonight

Sharding

1. Five sharding data models and which is right[277]
   Five sharding data models and which is right
2. Four Data Sharding Strategies We Analyzed in Building a Distributed SQL Database[278]
   Four Data Sharding Strategies We Analyzed in Building a Distributed SQL Database - The Distributed SQL Blog hamburger-white
3. Understanding Database Sharding[279]
   Understanding Database Sharding | DigitalOcean DigitalOcean home DigitalOcean Homepage

[275] https://blog.codinghorror.com/a-visual-explanation-of-sql-joins/
[276] https://www.studytonight.com/dbms/database-normalization.php
[277] https://www.citusdata.com/blog/2017/08/28/five-data-models-for-sharding/
[278] https://blog.yugabyte.com/four-data-sharding-strategies-we-analyzed-in-building-a-distributed-sql-database/
[279] https://www.digitalocean.com/community/tutorials/understanding-database-sharding

Mix

1. Understanding Window Functions[280]
   Understanding Window Functions
2. Why Order By With Limit and Offset is Slow - Faster Pagination in Mysql[281]
   Why Order By With Limit and Offset is Slow - Faster Pagination in Mysql
3. Managing Hierarchical Data in MySQL Using the Adjacency List Model[282]
   Managing Hierarchical Data in MySQL Using the Adjacency List Model
4. 101 Tips to MySQL Tuning and Optimization[283]
   101 Tips to MySQL Tuning and Optimization
5. A Review of Graph Databases[284]
   A Review of Graph Databases

Books

1. 97 Things Every SQL Developer Should Know[285]
   97 Things Every SQL Developer Should Know: Beaulieu, Alan: 9780596804336: Amazon.com: Books

[280] https://tapoueh.org/blog/2013/08/understanding-window-functions/
[281] https://www.eversql.com/faster-pagination-in-mysql-why-order-by-with-limit-and-offset-is-slow/
[282] https://www.mysqltutorial.org/mysql-adjacency-list-tree/
[283] https://www.monitis.com/blog/101-tips-to-mysql-tuning-and-optimization/
[284] https://nebula-graph.io/posts/review-on-graph-databases/
[285] https://www.amazon.com/Things-Every-Developer-Should-Know/dp/0596804334

Notes

(free space)

Notes

(free space)

Week 13: Extra Resources

Time needed: 6 hours

RegEx

1. Debuggex: Online visual regex tester. JavaScript, Python, and PCRE.[286]
   Debuggex: Online visual regex tester. JavaScript, Python, and PCRE.
2. RegExr: Learn, Build, & Test RegEx[287]

[286] https://www.debuggex.com/
[287] https://regexr.com/

Notes

(free space)

Notes

(free space)

VCS

1. Learn Git Branching[288]
   Learn Git Branching
2. Git Command Explorer[289]
   Git Explorer
3. GitHub Cheat Sheet[290]
   GitHub - tiimgreen/github-cheat-sheet: A list of cool features of Git and GitHub.

---

[288] https://learngitbranching.js.org/
[289] https://gitexplorer.com/
[290] https://github.com/tiimgreen/github-cheat-sheet

Notes

(free space)

Notes

(free space)

SEO

1. On-Page SEO Checklist for 2020[291]
   The On-Page SEO Checklist for 2021
2. UI Testing Best Practices[292]
   GitHub - NoriSte/ui-testing-best-practices: The largest UI testing best practices list (last update: January 2021)
3. A Breakdown of HTML Usage Across ~8 Million Pages (& What It Means for Modern SEO)[293]
   A Breakdown of HTML Usage Across ~8 Million Pages (& What It Means for Modern SEO) - Moz Moz Search Resources Menu icon-close Search Moz
4. 34 Ways To Improve SEO Rankings in 2020[294]
   34 Ways To Improve SEO Rankings in 2021
5. The Complete 51-Point SEO Checklist For 2020[295]
   The Complete 51-Point SEO Checklist For 2021 [Updated]
6. The Complete SEO Checklist For 2020[296]
   The Complete SEO Checklist For 2021
7. The SEMrush Website Migration Checklist[297]
   Website Migration Checklist: Everything You Need to Know
8. The Ultimate SEO Checklist for 2020[298]
   The Ultimate SEO Checklist for 2021 (66 Checks + PDF Download)
9. I Used The Web For A Day On A 50 MB Budget[299]
   I Used The Web For A Day On A 50 MB Budget â€" Smashing Magazine Clear Search Back to top

[291] https://www.gotchseo.com/on-page-seo/
[292] https://github.com/NoriSte/ui-testing-best-practices
[293] https://moz.com/blog/a-breakdown-of-html-usage-across-8-million-pages
[294] https://www.quicksprout.com/ways-to-improve-seo-ranking/
[295] https://www.clickminded.com/seo-checklist/
[296] https://backlinko.com/seo-checklist
[297] https://www.semrush.com/blog/website-migration-checklist/
[298] https://www.reliablesoft.net/seo-checklist/
[299] https://www.smashingmagazine.com/2019/07/web-on-50mb-budget/

10. Building the most inaccessible site possible with a perfect Lighthouse score[300]
    Building the most inaccessible site possible with a perfect Lighthouse score - Manuel MatuzoviÄ‡
11. The On-Page SEO Cheat Sheet[301]
    The On-Page SEO Cheat Sheet
12. How to Create a Site Structure That Will Enhance SEO[302]
    How to Create a Site Structure That Will Enhance SEO

[300] https://www.matuzo.at/blog/building-the-most-inaccessible-site-possible-with-a-perfect-lighthouse-score/
[301] https://neilpatel.com/2015/07/07/the-on-page-seo-cheat-sheet/
[302] https://neilpatel.com/blog/site-structure-enhance-seo/

Notes

(free space)

Notes

(free space)

Books

1. Git Patterns and Anti-Patterns Scaling from Workgroup to Enterprise[303]
   Git Patterns and Anti-Patterns - Dzone Refcardz
2. Regular Expressions A Look at Characters, Types, Operators, and More[304]
   Regular Expressions - Dzone Refcardz”

[303] https://dzone.com/refcardz/git-patterns-and-anti-patterns
[304] https://dzone.com/refcardz/regular-expressions

Level: Intermediate

Level Definition

> We are in the middle ground of the model. An individual falls into this category when he is fully capable of troubleshooting and solving problems on their own, as well as planning their future actions while avoiding previous mistakes. The practitioner will still experience trouble when it comes to pinpointing the exact details to focus on. The IT sphere works mainly in teams in order to smoothen out these processes.
>
> - https://www.360pmo.com/the-five-dreyfus-model-stages/

Duration: 9 weeks (∼2 months)Â
Average per week: ∼15 hours

Schedule

- Week 14: Build Up Dictionary
- Week 15: DDD, Functional & CQRS/ES
- Week 16: DDD, Functional & CQRS/ES

- Week 17: Networking
- Week 18: DevOps
- Week 19: Security
- Week 20: Architecture
- Week 21: Architecture
- Week 22: Jobs

Week 14: Build Up Dictionary

A

APM - Application Performance Management

> APMs are solutions to monitor applications to ensure performance and availability by generally collecting the application and server metrics to alert you when crossing thresholds.

B

Backend For Frontend

> A Backend for Frontend pattern is tightly coupled to a specific user experience and helps to create backends for client-facing mobile or web apps.

Blue-Green Deployment

> Blue-green deployment is a technique that reduces risk and downtime by running two environments and redirecting traffic to the latest one.

Brook's Law

Adding manpower to a late software project makes it later.

Burnout

Burnout is a state of mental exhaustion caused by excessive and prolonged stress.

C

Canary

Canary release is a technique used for rolling out new changes to a small subset of users before making it available to everyone, generally used to control and minimise the impact.

CAP Theorem

The CAP theorem states that a distributed system can deliver only two of three desired characteristics at the same moment: Consistency, Availability, Partition tolerance.

CI/CD - Continuous Integration / Continuous Deployment

A CI/CD pipeline automates your software delivery process, by building code, running tests and deploying the application.

CIDR - Classless Inter-Domain Routing

A CIDR is a set of IP standards used for creating unique identifiers for network devices.

Cohesion

Cohesion refers to what a module can do: Low would mean that it is too broad, High means that is very focused.

Contravariance

A property is contravariant if it reverses the ordering of types, which orders types from more generic to more specific.

Coupling

Coupling refers to how dependent two modules towards each other: Low would mean that changing something major should not affect the other module, High means that is difficult to change without side-effects.

Covariance

A property is covariant if it preserves the ordering of types, which orders types from more specific to more generic.

CQRS - Command Query Responsibility Segregation

CQRS it is a pattern used for splitting the read model (query) from the write model (command).

CSP - Content Security Policy

Content Security Policy an extra layer of security implemented in the browser that helps to detect and mitigate certain types of attacks (like XSS).

D

Data Lake

A data lake is a centralized repository that allows storing structured/unstructured data at scale.

Data Mesh

Data Mesh is a layer that abstracts the complexities of connecting, managing and supporting access to data and allows the data to be available and discoverable for the applications that needed access to it.

Data Warehouse

A data warehouse is a central repository of information that can be analyzed and aggregated.

DaaS - Data as a Service

DaaS can be considered as a subset of SaaS as it builds on the concept that its data product can be provided to the user on demand.

DBaaS - Database as a Service

DBaaS is a cloud service that lets users access and uses a cloud database system without having to manage it directly.

DevOps

DevOps is a mix of cultural practices and tools that increases an organization's ability to deliver applications and services.

DDD - Domain Driven Design

Domain-Driven Design is an approach to complex software where there's a focus on the core domain, via collaboration (and by speaking a ubiquitous language) with domain practitioners there will be a clear understanding of the data models and context in which they're applied.

E

Emergent Design

Emergent design is the ability to adapt and evolve a software design to new concepts or changes.

Event Sourcing

Event Sourcing ensures that all changes to application state are stored as a sequence of events, they can be used to reconstruct past states and also cope with retroactive changes.

Event Storming

Event Storming is a workshop-based method to find out details of the domain of a software program.

F

FaaS - Function as a Service

Function as a Service is based on functions that can be triggered by events, most of the times it's called serverless architecture.

Feature Toggle

A feature toggle is a mechanism that allows functionality to be turned enabled/disabled via a flag.

Forward Secrecy

(Perfect) Forward Secrecy means that the keys used to encrypt and decrypt information will be changed frequently, so if the latest key is compromised, it exposes only a small portion of the user's data.

Functional Programming

Functional programming is a way of thinking about software development by following the following concepts: pure functions, recursion, referential transparency, first-class & higher-order function, immutable variables.

G

GRASP - General Responsibility Assignment Software Patterns

GRASP consists of a series of guidelines to solve common problems in object-oriented design: controller, creator, indirection, information expert, low coupling, high cohesion, polymorphism, protected variations, and pure fabrication.

H

Hofstadter's Law

It always takes longer than you expect, even when you take into account Hofstadter's Law.

HSTS - HTTP Strict Transport Security

HSTS is a technology that secures HTTPS web servers against downgrade attacks.

I

IaaS - Infrastructure as a Service

Infrastructure as a Service is offered by cloud vendors and provides access to computing resources such as servers, storage and networking.

IaC - Infrastructure as Code

Infrastructure as Code is the management of infrastructure in a descriptive and versioned manner.

K

Kerchkhoff's Principle

In cryptography, a system should be secure even if everything about the system, except for a small piece of information - the key - is public knowledge.

Knuth's optimization principle

Premature optimization is the root of all evil.

KPI - Key Performance Indicators

The KPIs are the critical key indicators of progress toward an intended goal.

L

Lambda

A lambda is a small anonymous function that can be passed as an argument.

Linus's Law

Given enough eyeballs, all bugs are shallow.

M

Microservices

Microservices are an architectural approach to building applications as a composition of small independent services.

Micro Frontends

Micro Frontend architecture is a design approach to decompose a frontend app into mini-apps.

Moore's Law

The power of computers per unit cost doubles every 24 months. The most popular version states: The number of transistors on an integrated circuit will double in about 18 months.

MTBF - Mean Time Between Failures

MTBF is the average time between repairable failures, the metric tracks the availability and reliability of a product.

MTTA - Mean Time To Acknowledge

MTTA is the average time it takes from when an alert is triggered to when work begins on the issue, the metric tracks the team responsiveness.

MTTF - Mean Time To Failure

MTTF is the average time between non-repairable failures.

MTTR - Mean Time To Recovery

MTTR is the average time it takes to recover a system failure.

Mutation Testing

Mutation Testing is a kind of testing where certain statement in the code are mutated to check whether the test cases are covering those changes.

N

Ninety-ninety rule

The first 90% of the code takes 10% of the time. The remaining 10% takes the other 90% of the time.

Norvig's Law

Any technology that surpasses 50% penetration will never double again (in any number of months).

O

Observability

Observability is a technical solution that enables to actively debug a system by analysing their properties and patterns.

OLAP - Online Analytical Processing

OLAP systems have the primary objective of data analysis and not data processing.

OLTP - Online Transaction Processing

OLTP systems have the primary objective is data processing and not data analysis.

P

PaaS - Platform as a Service

PaaS is a complete cloud environment where develop and deploy software onto.

Pareto Principle

For many phenomena, 80% of consequences stem from 20% of the causes.

Pipelines

A CI/CD pipeline automates the software delivery process: builds the code, runs the tests, and deploys a new version of the application.

R

Refactoring

Refactoring is the technique for restructuring an existing code and altering its internal structure without changing its external behaviour.

RPO - Recovery Point Objective

An RPO describes the interval of time during a disruption before the quantity of data lost exceeds the tolerance defined.

RTO - Recovery Time Objective

An RTO is the duration of time within which a business process must be restored after a disaster to avoid unacceptable consequences.

RUM - Real User Monitoring

Real User Monitoring is a passive monitoring technique that collects and analyses user interactions to improve the end-user experience.

S

SaaS - Software as a Service

Software as a Service is an on-demand software hosted by the provider who gives access to the product usually via a subscription model.

Secret Management

Secrets management refers to tools and methods for managing digital authentication credentials (secrets, such as password and API keys).

Serverless

Serverless computing is a cloud model where the cloud provider is responsible for executing a piece of code by dynamically allocating the resources, sometimes referred to as FaaS.

Service Mesh

A service mesh is an infrastructure layer designed to handle a high volume of network communications among application services, generally, it provides service discovery, load balancing, encryption, observability, traceability, authentication & authorization, and support for the circuit breaker pattern.

SLA - Service Level Agreement

An SLA is an agreement between the provider and the client about measurable metrics.

SLI - Service Level Indicator

An SLI is the actual measurement of an SLO.

SLO - Service Level Objective

An SLO is an agreement about a specific metric within an SLA.

T

TDD - Test Driven Development

TDD is a process that relies on the repetition of a short development cycle: turn the requirements into specific test cases, the minimal code will be written so that the tests pass, then (optionally) the code will be refactored.

Technical Debt

Technical debt refers to the consequences due to poorly written code and compromises during the development, taking in consideration the effort that has to be done to "repay" the debt to go back to acceptable levels.

U

Uptime

The website uptime is the time that a website or web service is available to the users over a given period.

V

Velocity

The velocity, in an "agile" iteration, is the sum of all the story points associated with each completed user stories during that iteration.

W

Wirth's law

Software gets slower faster than hardware gets faster.

Y

Yak Shaving

Yak shaving is a term that refers to a series of nested (never-ending) tasks that need to be performed before a project can progress to its next stage.

Yoda Conditions

The Yoda conditions is a programming style where the variable and constant will be inverted in a conditional expression.

Week 15: DDD, Functional & CQRS/ES

Time needed: 17 hours

Books

1. Domain-Driven Design: Tackling Complexity in the Heart of Software[305]

[305] https://www.amazon.com/Domain-Driven-Design-Tackling-Complexity-Software/dp/0321125215

Notes

(free space)

Notes

(free space)

Week 16: DDD, Functional & CQRS/ES

Time needed: 11 hours

DDD

1. DDD Reference[306]
   Domain-Driven Design Reference: Definitions and Pattern Summaries - Domain Language
2. Summary of a four days DDD training[307]
   Summary of a four days DDD training | by Thomas Ferro | Medium
3. The beginner's guide to BDD (behaviour-driven development)[308]
   The beginner's guide to BDD (behaviour-driven development)

[306] https://domainlanguage.com/product/domain-driven-design-reference/
[307] https://medium.com/@t.ferro184/summary-of-a-four-days-ddd-training-74103a6d99a1
[308] https://inviqa.com/blog/bdd-guide

Functional

1. Benefits of Functional Programming by Example[309]
   Benefits of Functional Programming by Example | by Nick McCurdy | Medium
2. Don't Be Scared Of Functional Programming[310]
   Donâ€™t Be Scared Of Functional Programming â€" Smashing Magazine Clear Search Back to top
3. So You Want to be a Functional Programmer (Part 1)[311]
   So You Want to be a Functional Programmer (Part 1) | by Charles Scalfani | Medium
4. So You Want to be a Functional Programmer (Part 2)[312]
   So You Want to be a Functional Programmer (Part 2) | by Charles Scalfani | Medium
5. So You Want to be a Functional Programmer (Part 3)[313]
   So You Want to be a Functional Programmer (Part 3) | by Charles Scalfani | Medium
6. So You Want to be a Functional Programmer (Part 4)[314]
   So You Want to be a Functional Programmer (Part 4) | by Charles Scalfani | Medium
7. So You Want to be a Functional Programmer (Part 5)[315]
   So You Want to be a Functional Programmer (Part 5) | by Charles Scalfani | Medium

[309] https://medium.com/@nickmccurdy/benefits-of-functional-programming-by-example-76f1135b0b18
[310] https://www.smashingmagazine.com/2014/07/dont-be-scared-of-functional-programming/
[311] https://medium.com/@cscalfani/so-you-want-to-be-a-functional-programmer-part-1-1f15e387e536
[312] https://medium.com/@cscalfani/so-you-want-to-be-a-functional-programmer-part-2-7005682cec4a
[313] https://medium.com/@cscalfani/so-you-want-to-be-a-functional-programmer-part-3-1b0fd14eb1a7
[314] https://medium.com/@cscalfani/so-you-want-to-be-a-functional-programmer-part-4-18fbe3ea9e49
[315] https://medium.com/@cscalfani/so-you-want-to-be-a-functional-programmer-part-5-c70adc9cf56a

8. So You Want to be a Functional Programmer (Part 6)[316]
   So You Want to be a Functional Programmer (Part 6) | by Charles Scalfani | Medium

CQRS

1. CQRS: What? Why? How?. CQRS is a useful pattern to reasonâ€¦ | by StÃ©phane Derosiaux | Medium[317]
   CQRS: What? Why? How?. CQRS is a useful pattern to reasonâ€¦ | by StÃ©phane Derosiaux | Medium
2. CQRS pattern - Azure Architecture Center | Microsoft Docs[318]
   CQRS pattern - Azure Architecture Center | Microsoft Docs
3. CQRS[319]
   CQRS

Event Sourcing

1. Event Sourcing[320]
   Event Sourcing
2. Event Sourcing and CQRS - Event Store Blog[321]
   Event Sourcing and CQRS - Event Store Blog
3. What they donâ€™t tell you about event sourcing | by Hugo Rocha | Medium[322]
   What they donâ€™t tell you about event sourcing | by Hugo Rocha | Medium

[316] https://medium.com/@cscalfani/so-you-want-to-be-a-functional-programmer-part-6-db502830403
[317] https://medium.com/@sderosiaux/cqrs-what-why-how-945543482313
[318] https://docs.microsoft.com/en-us/azure/architecture/patterns/cqrs
[319] https://martinfowler.com/bliki/CQRS.html
[320] https://martinfowler.com/eaaDev/EventSourcing.html
[321] https://www.eventstore.com/blog/event-sourcing-and-cqrs
[322] https://medium.com/@hugo.oliveira.rocha/what-they-dont-tell-you-about-event-sourcing-6afc23c69e9a

Videos

1. Pim Elshoff: Technically DDD Video @ DPC2018[323] | Slides[324]
2. Greg Young - CQRS and Event Sourcing - Code on the Beach 2014 - YouTube[325]

Books

1. The InfoQ eMag: Domain-Driven Design in Practice[326]
   The InfoQ eMag: Domain-Driven Design in Practice
2. Domain Driven Design Quickly[327]
   Domain Driven Design Quickly
3. Domain-Driven Design Object-Orientation Done Right[328]
   Domain-Driven Design - Dzone Refcardz
4. SOA Patterns Service-Orient Your Enterprise[329]
   SOA Patterns - Dzone Refcardz
5. The Anatomy Of Domain-Driven Design - Booklet[330]
   Anatomy Ofâ€¦ by Scott Millett et al. [PDF/iPad/Kindle]

[323] https://www.youtube.com/watch?v=JpcNeeetijo
[324] https://speakerdeck.com/pelshoff/technically-ddd-v3
[325] https://www.youtube.com/watch?v=JHGkaShoyNs
[326] https://www.infoq.com/minibooks/emag-domain-driven-design/
[327] https://www.infoq.com/minibooks/domain-driven-design-quickly/
[328] https://dzone.com/refcardz/getting-started-domain-driven
[329] https://dzone.com/refcardz/soa-patterns
[330] https://leanpub.com/theanatomyofdomain-drivendesign

Notes

(free space)

Notes

(free space)

Week 17: Networking

Time needed: 17 hours

HTTP

1. HTTP for servers[331]
   HTTP for servers

IP

1. Understanding IP Addressing and CIDR Charts[332]
   Understanding IP Addressing and CIDR Charts â€" RIPE Network Coordination Centre
2. CIDR.xyz[333]
   CIDR.xyz
3. Understanding IP Addressing: Everything You Ever Wanted To Know[334]

[331] http://www.and.org/texts/server-http
[332] https://www.ripe.net/about-us/press-centre/understanding-ip-addressing
[333] https://cidr.xyz/
[334] http://pages.di.unipi.it/ricci/501302.pdf

Time

1. UTC is enough for everyone…right?[335]
   UTC is Enough for Everyone, Right?

Books

1. The DevOps Handbook[336]

[335] https://zachholman.com/talk/utc-is-enough-for-everyone-right

[336] https://www.amazon.com/Devops-Handbook-World-Class-Reliability-Organizations/dp/1942788002

Notes

(free space)

Notes

(free space)

Week 18: DevOps

Time needed: 11 hours

Cloud

1. How to Succeed with Cloud-native Applications[337]
   How to Succeed with Cloud-native Applications | by Tijl Dullers | FAUN | Medium
2. Don't get locked up into avoiding lock-in[338]
   Don't get locked up into avoiding lock-in

Deployments

1. Production deployment guides[339]
   Production deployment guides
2. Production Readiness Checklist[340]
   Production Readiness Checklist

[337] https://medium.com/faun/how-to-succeed-with-cloud-native-applications-f222ecd3f746
[338] https://martinfowler.com/articles/oss-lockin.html
[339] https://gruntwork.io/guides
[340] https://gruntwork.io/devops-checklist/

Infrastructure

1. 5 Common Server Setups For Your Web Application[341]
   5 Common Server Setups For Your Web Application | DigitalOcean DigitalOcean home DigitalOcean Homepage

MVP

1. Your ultimate guide to Minimum Viable Product (+great examples)[342]
   Your ultimate guide to Minimum Viable Product (+great examples) | Fast Monkeys â€“ Official Blog
2. How To Create a Minimum Viable Product[343]
   How To Create a Minimum Viable Product
3. Spikes, POCs, Prototypes and the MVP[344]
   Spikes, POCs, Prototypes and the MVP | by Leigh Garland | STUDIO ZERO | Medium

SRE

1. School of SRE[345]
   SchoolOfSRE

[341] https://www.digitalocean.com/community/tutorials/5-common-server-setups-for-your-web-application
[342] https://blog.fastmonkeys.com/2014/06/18/minimum-viable-product-your-ultimate-guide-to-mvp-great-examples/
[343] https://code.tutsplus.com/articles/how-to-create-a-minimum-viable-product--cms-22245
[344] https://medium.com/studio-zero/spikes-pocs-prototypes-and-the-mvp-5cdffa1b7367
[345] https://linkedin.github.io/school-of-sre/

Tools

1. htop explained[346]
   htop explained | peteris.rocks
2. Linux Performance Analysis in 60,000 Milliseconds[347]
   Linux Performance Analysis in 60,000 Milliseconds | by Netflix Technology Blog | Netflix TechBlog
3. The Ultimate DevOps Tool Chest[348]
   The Ultimate DevOps Tool Chest | Digital.ai
4. A tcpdump Tutorial with Examples â€" 50 Ways to Isolate Traffic[349]
   A tcpdump Tutorial with Examples â€" 50 Ways to Isolate Traffic | Daniel Miessler search mail mail mail

Mix

1. Practical DevOps Learning Path â€" from where should i start ?[350]
   Practical DevOps Learning Path â€" from where should i start ? | by Abdennour Toumi | Jan, 2021 | Medium
2. DORA research program[351]
   DORA research program
3. CALMS Framework | Atlassian[352]
   CALMS Framework | Atlassian"

---

[346] https://peteris.rocks/blog/htop/
[347] https://netflixtechblog.com/linux-performance-analysis-in-60-000-milliseconds-accc10403c55
[348] https://xebialabs.com/the-ultimate-devops-tool-chest/
[349] https://danielmiessler.com/study/tcpdump/
[350] https://abdennoor.medium.com/practical-devops-learning-path-from-where-should-i-start-9d536a5a7250
[351] https://www.devops-research.com/research.html
[352] https://www.atlassian.com/devops/frameworks/calms-framework

Books

1. The Cynefin Mini-Book[353]
   The Cynefin Mini-Book
2. Foundations of RESTful Architecture[354]
   Foundations of RESTful Architecture - Dzone Refcardz
3. The InfoQ eMag: Tech Ethics[355]
   The InfoQ eMag: Tech Ethics
4. Continuous Integration Patterns and Anti-Patterns[356]
   Continuous Integration - Dzone Refcardz
5. Continuous Delivery Patterns and Anti-Patterns in the Software Lifecycle[357]
   Continuous Delivery - Dzone Refcardz

[353] https://www.infoq.com/minibooks/cynefin-mini-book/
[354] https://dzone.com/refcardz/rest-foundations-restful
[355] https://www.infoq.com/minibooks/emag-tech-ethics/
[356] https://dzone.com/refcardz/continuous-integration
[357] https://dzone.com/refcardz/continuous-delivery-patterns

Notes

(free space)

Notes

(free space)

Week 19: Security

Time needed: 19 hours

SSL/TLS

1. Everything you need to know about HTTP security headers[358]
   Appcanary - Everything you need to know about HTTP security headers
2. TLS/SSL Explained: TLS/SSL Terminology and Basics[359]
   TLS/SSL Explained: TLS/SSL Terminology and Basics - DZone Security

Mix

1. I'm harvesting credit card numbers and passwords from your site. Here's how.[360]
   Iâ€™m harvesting credit card numbers and passwords from your site. Hereâ€™s how. | by David Gilbertson | Hacker-Noon.com | Medium

[358] https://blog.appcanary.com/2017/http-security-headers.html
[359] https://dzone.com/articles/tlsssl-terminology-and-basics
[360] https://medium.com/hackernoon/im-harvesting-credit-card-numbers-and-passwords-from-your-site-here-s-how-9a8cb347c5b5

2. Part 2: How to stop me harvesting credit card numbers and passwords from your site[361]
   Part 2: How to stop me harvesting credit card numbers and passwords from your site | by David Gilbertson | HackerNoon.com | Medium

---

Books

1. [](https://www.amazon.com/Certified-Ethical-Hacker-Study-Guide/dp/1119533198)
   CEH v10 Certified Ethical Hacker Study Guide: 9781119533191: Computer Science Books @ Amazon.com

[361] https://medium.com/hackernoon/part-2-how-to-stop-me-harvesting-credit-card-numbers-and-passwords-from-your-site-844f739659b9

Notes

(free space)

Notes

(free space)

Week 20: Architecture

Time needed: 16 hours

Distributed

1. Patterns of Distributed Systems[362]
   Patterns of Distributed Systems

Documentation

1. Agile software architecture documentation[363]
   Agile software architecture documentation - Coding the Architecture

Enterprise

1. Catalog of Patterns of Enterprise Application Architecture[364]
   Catalog of Patterns of Enterprise Application Architecture

[362] https://martinfowler.com/articles/patterns-of-distributed-systems/

[363] http://www.codingthearchitecture.com/2016/05/31/agile_software_architecture_documentation.html

[364] https://martinfowler.com/eaaCatalog/index.html

2. Software Architecture Monday - Enterprise Architecture Lessons[365]
   Software Architecture Monday | Developer to Architect | Mark Richards

Event-Driven

1. Software Architecture Monday - Event-Driven Architecture Lessons[366]
   Software Architecture Monday | Developer to Architect | Mark Richards

Hexagonal

1. DDD, Hexagonal, Onion, Clean, CQRS, â€¦ How I put it all together[367]
   DDD, Hexagonal, Onion, Clean, CQRS, â€¦ How I put it all together â€“ @hgraca

Microservices

1. Microservices[368]
   Microservices
2. Seven Microservices Anti-patterns[369]
   Seven Microservices Anti-patterns

[365] https://www.developertoarchitect.com/lessons-enterprise.html
[366] https://www.developertoarchitect.com/lessons-eda.html
[367] https://herbertograca.com/2017/11/16/explicit-architecture-01-ddd-hexagonal-onion-clean-cqrs-how-i-put-it-all-together/
[368] https://martinfowler.com/articles/microservices.html
[369] https://www.infoq.com/articles/seven-uservices-antipatterns/

3. Software Architecture Monday - Microservices Lessons[370]
   Software Architecture Monday | Developer to Architect | Mark Richards

The Twelve Factors App

1. I. Codebase[371]
   The Twelve-Factor App
2. II. Dependencies[372]
   The Twelve-Factor App
3. III. Config[373]
   The Twelve-Factor App
4. IV. Backing services[374]
   The Twelve-Factor App
5. V. Build, release, run[375]
   The Twelve-Factor App
6. VI. Processes[376]
   The Twelve-Factor App
7. VII. Port binding[377]
   The Twelve-Factor App
8. VIII. Concurrency[378]
   The Twelve-Factor App
9. IX. Disposability[379]
   The Twelve-Factor App

[370] https://www.developertoarchitect.com/lessons-microservices.html
[371] https://www.12factor.net/codebase
[372] https://www.12factor.net/dependencies
[373] https://www.12factor.net/config
[374] https://www.12factor.net/backing-services
[375] https://www.12factor.net/build-release-run
[376] https://www.12factor.net/processes
[377] https://www.12factor.net/port-binding
[378] https://www.12factor.net/concurrency
[379] https://www.12factor.net/disposability

10. X. Dev/prod parity[380]
    The Twelve-Factor App
11. XI. Logs[381]
    The Twelve-Factor App
12. XII. Admin processes[382]
    The Twelve-Factor App
13. 12 Fractured Apps[383]
    12 Fractured Apps. Over the years Iâ€™ve witnessed more andâ€¦ | by Kelsey Hightower | Medium

---

Books

1. Clean Architecture[384]
   Clean Architecture: A Craftsman's Guide to Software Structure and Design (Robert C. Martin Series): Martin, Robert: 9780134494166: Amazon.com: Books

[380] https://www.12factor.net/dev-prod-parity
[381] https://www.12factor.net/logs
[382] https://www.12factor.net/admin-processes
[383] https://medium.com/@kelseyhightower/12-fractured-apps-1080c73d481c
[384] https://www.amazon.com/Clean-Architecture-Craftsmans-Software-Structure/dp/0134494164

Notes

(free space)

Notes

(free space)

Week 21: Architecture

Time needed: 19 hours

Mix

1. Software Architecture Monday - General Architecture Lessons[385]
   Software Architecture Monday | Developer to Architect | Mark Richards

---

Videos

1. Simon Brown: Software Architecture vs. Code Video @ GOTO 2014[386] | Slides[387]
2. The Frustrated Architect[388]
   The Frustrated Architect

[385] https://www.developertoarchitect.com/lessons-general.html
[386] https://www.youtube.com/watch?v=GAFZcYlO5S0
[387] http://gotocon.com/dl/goto-amsterdam-2014/slides/SimonBrown_SoftwareArchitectureVsCode.pdf
[388] https://www.infoq.com/presentations/The-Frustrated-Architect/

3. GOTO 2017 â€¢ The Many Meanings of Event-Driven Architecture â€¢ Martin Fowler[389]
   GOTO 2017 â€¢ The Many Meanings of Event-Driven Architecture â€¢ Martin Fowler - YouTube

Books

1. Software Architecture for Developers[390]
   Software Architecture for Developers
2. Building Evolutionary Architectures[391]
3. 97 Things Every Software Architect Should Know[392]

[389] https://www.youtube.com/watch?v=STKCRSUsyP0
[390] https://softwarearchitecturefordevelopers.com/
[391] https://www.amazon.com/Building-Evolutionary-Architectures-Support-Constant/dp/1491986360
[392] https://www.amazon.com/Things-Every-Software-Architect-Should/dp/059652269X

Notes

(free space)

Notes

(free space)

Week 22: Jobs

Time needed: 12 hours

Freelancing

1. 20 Reasons To Say “No” to Freelancing[393]
   20 Reasons To Say “No” to Freelancing - Hongkiat Facebook Twitter Instagram Pinterest LinkedIn Google+ Youtube Reddit Dribbble Behance Github CodePen Whatsapp Email

Job Offer

1. How To Prepare For A Salary Negotiation: A Check List[394]
   How To Prepare For A Salary Negotiation: A Check List - Adobe 99U Adobe-full-color Adobe-white Adobe-black logo-white Adobe-full Adobe Behance arrow-down arrow-down 2 arrow-right arrow-right 2 Line close-tablet-03 close-tablet-05 comment dropdown-close dropdown-open facebook instagram linkedin logo rss search share twitter

[393] https://www.hongkiat.com/blog/reasons-not-to-freelance/

[394] https://99u.adobe.com/articles/61016/how-to-prepare-for-a-salary-negotiation-a-check-list

2. How To Turn Down A Job Offer[395]
   How To Turn Down A Job Offer

Ladder

1. Sharing Our Engineering Ladder[396]
   Sharing Our Engineering Ladder â€" RTR Dress Code
2. The Career Ladder Isn't In The Office[397]
   The Career Ladder Isnâ€™t In The Office | by Sean Johnson | HackerNoon.com | Medium

Preparation

1. Reverse interview[398]
   GitHub - viraptor/reverse-interview: Questions to ask the company during your interview
2. 34 Crucial Tips For Your Next Job Interview[399]
3. 7 Free Career Aptitude Tests You Can Take Online Today[400]
   7 Free Career Aptitude Tests You Can Take Online Today Logo - Full (Color)

Questions

1. Interview questions[401]

[395] https://www.forbes.com/sites/jacquelynsmith/2013/08/13/how-to-turn-down-a-job-offer-2
[396] http://dresscode.renttherunway.com/blog/ladder
[397] https://medium.com/hackernoon/the-career-ladder-isnt-in-the-office-43cfe5e3b066
[398] https://github.com/viraptor/reverse-interview
[399] https://www.lifehack.org/articles/work/34-crucial-tips-for-your-next-job-interview.html
[400] https://blog.hubspot.com/marketing/career-aptitude-tests
[401] https://github.com/odino/interviews

GitHub - odino/interviews: Random questions to ask during interviews.

2. 30 Smart Answers To Tough Interview Questions[402]
   30 Smart Answers To Tough Interview Questions - Business Insider Business Insider logo Close icon Loading Menu icon Search icon Business Insider logo Account icon Account icon Business Life News Reviews Search icon Insider logo Close icon Business Life News All Account icon World globe Facebook Icon Twitter icon LinkedIn icon YouTube icon Instagram icon Business Insider logo Close icon Chevron icon Chevron icon Facebook Icon Email icon Link icon Twitter icon LinkedIn icon Fliboard icon More icon Close icon Loading Close icon
3. The 20 Toughest Job Interview Questions Heard At Apple, Google, Amazon And Others[403]
   Toughest Job Interview Questions - Business Insider Business Insider logo Close icon Loading Menu icon Search icon Business Insider logo Account icon Account icon Business Life News Reviews Search icon Insider logo Close icon Business Life News All Account icon World globe Facebook Icon Twitter icon LinkedIn icon YouTube icon Instagram icon Business Insider logo Close icon Chevron icon Chevron icon Facebook Icon Email icon Link icon Twitter icon LinkedIn icon Fliboard icon More icon Close icon Loading Close icon

Quit

1. Is It Better To Quit Or Get Fired?[404]
   Is It Better To Quit Or Get Fired?

[402] https://www.businessinsider.com/30-smart-answers-to-tough-interview-questions-2013-8
[403] https://www.businessinsider.com/toughest-job-interview-questions-2013-7
[404] https://www.forbes.com/sites/deborahljacobs/2013/07/31/is-it-better-to-quit-or-get-fired

2. Programmers: Before you turn 40, get a plan B[405]
   Programmers: Before you turn 40, get a plan B | Improving Software
3. 14 Signs It's Time To Leave Your Job[406]
   14 Signs It's Time To Leave Your Job
4. How to Tell If You're In a Dead End Job (and What You Can Do About It)[407]
   How to Tell If You're In a Dead End Job (and What You Can Do About It)

Remote

1. Quick, work remote! A guide on how to set up your remote working strategy[408]
   Quick, work remote! A guide on how to set up your remote working strategy Â· Intense Minimalism

Resume

1. 20 Critical Skills to Add to Resume (For All Types of Jobs)[409]
2. How To Botox Your Resume To Land A Job[410]
   How To Botox Your Resume To Land A Job
3. The 5-Step Editing Process for a Perfect Resume[411]
   The 5-Step Editing Process for a Perfect Resume

[405] https://improvingsoftware.com/2009/05/19/programmers-before-you-turn-40-get-a-plan-b/
[406] https://www.forbes.com/sites/jacquelynsmith/2013/09/04/14-signs-its-time-to-leave-your-job
[407] https://lifehacker.com/how-to-tell-if-youre-in-a-dead-end-job-and-what-you-ca-910478489
[408] https://intenseminimalism.com/2020/quick-work-remote/
[409] https://www.lifehack.org/836615/resume-skills
[410] https://www.forbes.com/sites/nextavenue/2013/08/28/how-to-botox-your-resume-to-land-a-job
[411] https://mashable.com/2014/03/15/editing-resume/

4. When Should You Lie on Your Resume?[412]
   When Should You Lie on Your Resume?
5. How To Craft The Perfect Web Developer RÃ©sumÃ©[413]
   How To Craft The Perfect Web Developer RÃ©ÂsuÂmÃ© â€”
   Smashing Magazine Clear Search Back to top

Books

1. 97 Things Every Programmer Should Know[414]

[412] https://lifehacker.com/when-should-you-lie-on-your-resume-955825518
[413] https://www.smashingmagazine.com/2018/06/web-developer-resume/
[414] https://www.amazon.com/Things-Every-Programmer-Should-Know/dp/0596809484

Notes

(free space)

Notes

(free space)

Level: Advanced

Level Definition

> The individual now looks at the bigger picture. Their focus falls onto understanding the essentials of the framework and often experience frustration when documentation is oversimplified. Proficiency is defined by the self-improvement skills which each person in the stage has. Not only does the proficient practitioner learn from his own mistakes, he observes others as well, anything could be a vital source of information.
>
> - https://www.360pmo.com/the-five-dreyfus-model-stages/

Duration: 7 weeks (∼2 months)Â
Average per week: ∼16 hours

Schedule

- Week 23: Build Up Dictionary
- Week 24: Productivity
- Week 25: Standards & Best Practices

- Week 26: Standards & Best Practices
- Week 27: Management
- Week 28: Management
- Week 29: Management

Week 23: Build Up Dictionary

C

Chaos Engineering

> Chaos Engineering is a disciplined approach to identify failures in advance by experimenting on a system to understand how it will respond under certain conditions.

Conflict Resolutions

> Conflict resolution skills are the methods and processes involved in facilitating the peaceful ending of a conflict.

Conway's Law

> Any piece of software reflects the organizational structure that produced it. Organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations.

Critical Thinking

> Critical thinking is the ability to engage in reflective and independent thinking that allows making the best decisions possible.

D

Delegation

Delegation is an important management skill, it is the shifting of authority/responsibility for a particular task or decision from one person to another.

F

Four Key Metrics

From the 2014 State of DevOps report have been identified the four key metrics for software delivery performance: lead time, deployment frequency, mean time to restore (MTTR), and change fail percentage.

Functional Reactive Programming

FRP is a combination of functional and reactive paradigms by integrating time flow and compositional events into functional programming.

L

Leadership

Leadership is the art of motivating a group of people to act toward achieving a common goal, usually by directing colleagues with a strategy to meet the company's needs.

P

Postel's Law

Be conservative in what you send, be liberal in what you accept.

Probabilistic Programming

Probabilistic Programming is a tool for statistical modelling, it borrows programming concepts and applies them to the problems of designing and using statistical models.

T

The Peter Principle

In a hierarchy, every employee tends to rise to his level of incompetence.

Threat Modeling

Threat modelling is the practice of identifying and understanding threats and mitigations to protect confidential data or intellectual property.

Three Rs of Enterprise Security

The Three Rs of Enterprise Security: Rotate credentials, Repave every server and application from a known good state and Repair vulnerable operating systems and application.

Time Management

Time management a the process for organizing and planning your time between specific activities by working smarter and not harder so that more can be done in less time.

Week 24: Productivity

Time needed: 18 hours

Cognitive Biases

1. Cognitive Biases in Programming[415]
   Cognitive Biases in Programming. As developers, weâ€™re familiar with theâ€¦ | by Yash Ranadive | HackerNoon.com | Medium

Critical Thinking

1. How to Improve Critical Thinking[416]
   How to Improve Critical Thinking | Scott H Young

Decision Making

1. A Checklist for Making Faster, Better Decisions[417]
   A Checklist for Making Faster, Better Decisions Navigation Menu Account Menu Search Menu Close menu Search

[415] https://medium.com/hackernoon/cognitive-biases-in-programming-5e937707c27b
[416] https://www.scotthyoung.com/blog/2019/03/07/improve-critical-thinking/
[417] https://hbr.org/2016/03/a-checklist-for-making-faster-better-decisions

2. The Decision Matrix: How to Prioritize What Matters[418]
   The Decision Matrix: How to Prioritize What Matters
3. 7 mental models you should know for smarter decision making[419]
   The Science Behind Smarter Decision Making: 7 Mental Models To Know

Destructive Approaches

1. Jobs contribute to workaholism, insomnia, divorce, death[420]
   Jobs contribute to workaholism, insomnia, divorce, death - Business Insider Business Insider logo Close icon Loading Menu icon Search icon Business Insider logo Account icon Account icon Business Life News Reviews Search icon Insider logo Close icon Business Life News All Account icon World globe Facebook Icon Twitter icon LinkedIn icon YouTube icon Instagram icon Business Insider logo Close icon Chevron icon Chevron icon Facebook Icon Email icon Link icon Twitter icon LinkedIn icon Fliboard icon More icon Close icon Valuable Not valuable Loading Chevron icon
2. KPIs, Velocity, and Other Destructive Metrics[421]
   KPIs, Velocity, and Other Destructive Metrics | Allen Holub

Stress

1. 30 Free Or Cheap Ways To Reduce Stress And To Refresh Yourself[422]

[418] https://fs.blog/2018/09/decision-matrix/
[419] https://thenextweb.com/lifehacks/2016/08/01/989517/
[420] https://www.businessinsider.com/disturbing-facts-about-your-job-2011-2
[421] https://holub.com/kpis-velocity-and-other-destructive-metrics/
[422] https://www.lifehack.org/articles/money/30-free-cheap-ways-reduce-stress-and-refresh-yourself.html

Time Management

1. Time Management[423]
   Time Management Skills and Training from MindTools.com

Mix

1. Strategy vs. Tactics: What's the Difference and Why Does it Matter?[424]
   Strategy vs. Tactics: Why the Difference Matters
2. 16 Tips for Getting 90 Percent of Your Work Done Before Lunch[425]
   16 Tips for Getting 90 Percent of Your Work Done in the Morning | Inc.com logo navigation logo Combined Shape Group 5 Group 3 Fill 1 Group 3 Group 3 Group 5 Group 3 Fill 1 Group 3 Group 3 logo logo navigation logo Combined Shape Shape
3. 26 Time Management Hacks I Wish I'd Known at 20[426]
   26 Time Management Hacks I Wish I'd Known at 20
4. 44 ways to be more productive[427]
   44 ways to be more productive
5. 13 Tech CEOs And Founders Reveal Their Favorite Productivity Hacks To Help You Get More Done[428]
   Tech CEOs Favorite Productivity Hacks - Business Insider Business Insider logo Close icon Loading Menu icon Search icon Business Insider logo Account icon Account icon Business Life News Reviews Search icon Insider logo Close icon

[423] https://www.mindtools.com/pages/main/newMN_HTE.htm
[424] https://fs.blog/2018/08/strategy-vs-tactics/
[425] https://www.inc.com/neil-patel/16-tips-for-getting-90-of-your-work-done-in-the-morning.html
[426] https://www.slideshare.net/egarbugli/26-time-management-hacks-i-wish-id-known-at-20
[427] https://www.stl-training.co.uk/sharing/16-44-ways-be-more-productive.html
[428] https://www.businessinsider.com/tech-ceos-favorite-productivity-hacks-2013-8

Business Life News All Account icon World globe Facebook Icon Twitter icon LinkedIn icon YouTube icon Instagram icon Business Insider logo Close icon Chevron icon Chevron icon Facebook Icon Email icon Link icon Twitter icon LinkedIn icon Fliboard icon More icon Close icon Loading Chevron icon Close icon

6. Pretend Your Time is Worth $1,000/Hour and You'll Become 100x More Productive[429]
   Pretend Your Time is Worth $1,000/Hour and Youâ€™ll Become 100x More Productive | by Anthony Moore | The Startup | Medium
7. Things I Learnt The Hard Way (in 30 Years of Software Development)[430]
   Julio Biason .Net 4.1 | Things I Learnt The Hard Way (in 30 Years of Software Development)

Videos

1. GOTO 2017 â€¢ Forget Velocity, Let's Talk Acceleration â€¢ Jessica Kerr[431]
   GOTO 2017 â€¢ Forget Velocity, Let's Talk Acceleration â€¢ Jessica Kerr - YouTube

Books

1. 97 Things Every Project Manager Should Know[432]

[429] https://medium.com/swlh/pretend-your-time-is-worth-1-000-hour-and-youll-become-100x-more-productive-f04628bb3e6d
[430] https://blog.juliobiason.me/thoughts/things-i-learnt-the-hard-way/
[431] https://www.youtube.com/watch?v=Lbcyyu8XB_Y
[432] https://www.amazon.com/Things-Every-Project-Manager-Should/dp/0596804164

2. Making Work Visible[433]

[433] https://www.amazon.com/Making-Work-Visible-Exposing-Optimize/dp/1942788150

Notes

(free space)

Notes

(free space)

Week 25: Standards & Best Practices

Time needed: 22 hours

Architecture

1. Terraform best practices[434]
   Welcome - Terraform Best Practices
2. Your guide to Kubernetes best practices[435]
   A guide to our top Kubernetes posts | Google Cloud Blog

Cloud

1. AWS Well-Architected[436]
   AWS Well-Architected - Build secure, efficient cloud applications
2. AWS Well-Architected - Framework Overview[437]
   AWS Well-Architected Framework - AWS Well-Architected

[434] https://www.terraform-best-practices.com/
[435] https://cloud.google.com/blog/products/containers-kubernetes/your-guide-kubernetes-best-practices
[436] https://aws.amazon.com/architecture/well-architected/
[437] https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html

Framework AWS Well-Architected Framework - AWS Well-Architected Framework

3. AWS Well-Architected - Operational Excellence Pillar[438]
   Operational Excellence Pillar - AWS Well-Architected Framework - Operational Excellence Pillar Operational Excellence Pillar - AWS Well-Architected Framework - Operational Excellence Pillar
4. AWS Well-Architected - Security Pillar[439]
   Security Pillar - AWS Well-Architected Framework - Security Pillar Security Pillar - AWS Well-Architected Framework - Security Pillar
5. AWS Well-Architected - Reliability Pillar[440]
   Reliability Pillar - AWS Well-Architected Framework - Reliability Pillar Reliability Pillar - AWS Well-Architected Framework - Reliability Pillar
6. AWS Well-Architected - Performance Efficiency Pillar[441]
   Performance Efficiency Pillar - AWS Well-Architected Framework - Performance Efficiency Pillar Performance Efficiency Pillar - AWS Well-Architected Framework - Performance Efficiency Pillar
7. AWS Well-Architected - Cost Optimization Pillar[442]
   Cost Optimization Pillar - AWS Well-Architected Framework - Cost Optimization Pillar Cost Optimization Pillar - AWS Well-Architected Framework - Cost Optimization Pillar
8. Cloud Design Patterns[443]
   Cloud design patterns - Azure Architecture Center | Microsoft Docs

[438] https://docs.aws.amazon.com/wellarchitected/latest/operational-excellence-pillar/welcome.html
[439] https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/welcome.html
[440] https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/welcome.html
[441] https://docs.aws.amazon.com/wellarchitected/latest/performance-efficiency-pillar/welcome.html
[442] https://docs.aws.amazon.com/wellarchitected/latest/cost-optimization-pillar/welcome.html
[443] https://docs.microsoft.com/en-us/azure/architecture/patterns/

Data

1. Data Management patterns[444]
   Data Management patterns - Cloud Design Patterns | Microsoft Docs

Design

1. Design and Implementation patterns[445]
   Design and Implementation patterns - Cloud Design Patterns | Microsoft Docs

HA

1. Availability patterns[446]
   Reliability patterns - Cloud Design Patterns | Microsoft Docs
2. Resiliency patterns[447]
   Reliability patterns - Cloud Design Patterns | Microsoft Docs

Observability

1. Management and Monitoring patterns[448]
   Management and Monitoring patterns - Cloud Design Patterns | Microsoft Docs

[444] https://docs.microsoft.com/en-us/azure/architecture/patterns/category/data-management
[445] https://docs.microsoft.com/en-us/azure/architecture/patterns/category/design-implementation
[446] https://docs.microsoft.com/en-us/azure/architecture/patterns/category/availability
[447] https://docs.microsoft.com/en-us/azure/architecture/patterns/category/resiliency
[448] https://docs.microsoft.com/en-us/azure/architecture/patterns/category/management-monitoring

Scalability

1. Performance and Scalability patterns[449]
   Reliability patterns - Cloud Design Patterns | Microsoft Docs

Security

1. Security patterns[450]
   Reliability patterns - Cloud Design Patterns | Microsoft Docs
2. Secure Programming HOWTO[451]
3. WHAT DO WE REALLY NEED TO ENCRYPT. CHEAT-SHEET[452]
   What Do We Really Need to Encrypt. Cheatsheet
4. Security architecture anti-patterns[453]

[449] https://docs.microsoft.com/en-us/azure/architecture/patterns/category/performance-scalability
[450] https://docs.microsoft.com/en-us/azure/architecture/patterns/category/security
[451] https://dwheeler.com/secure-programs/Secure-Programs-HOWTO.html
[452] https://www.cossacklabs.com/blog/what-we-need-to-encrypt-cheatsheet.html
[453] https://www.ncsc.gov.uk/whitepaper/security-architecture-anti-patterns

Notes

(free space)

Notes

(free space)

Week 26: Standards & Best Practices

Time needed: 18 hours

Books

1. How to Monitoring the SRE Golden Signals[454]
   How to Monitoring the SRE Golden Signals (E-Book)
2. AWS Well-Architected Framework - Operational Excellence Pillar[455]
3. AWS Well-Architected Framework - Security Pillar[456]
4. AWS Well-Architected Framework - Reliability Pillar[457]
5. AWS Well-Architected Framework - Performance Efficiency Pillar[458]
6. AWS Well-Architected Framework - Cost‘ Optimization Pillar[459]

[454] https://www.slideshare.net/OpsStack/how-to-monitoring-the-sre-golden-signals-ebook

[455] https://docs.aws.amazon.com/wellarchitected/latest/operational-excellence-pillar/wellarchitected-operational-excellence-pillar.pdf

[456] https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/wellarchitected-security-pillar.pdf

[457] https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/wellarchitected-reliability-pillar.pdf

[458] https://docs.aws.amazon.com/wellarchitected/latest/performance-efficiency-pillar/wellarchitected-performance-efficiency-pillar.pdf

[459] https://docs.aws.amazon.com/wellarchitected/latest/cost-optimization-pillar/wellarchitected-cost-optimization-pillar.pdf

7. Framework for Improving Critical Infrastructure Cybersecurity Version 1.1[460]
8. Handbook for Computer Security Incident Response Teams (CSIRTs)[461]
9. INTERPOL Global Guidelines for Digital Forensics Laboratories[462]
10. ISO 31000:2018 Risk management â€" Guidelines[463]
    ISO - ISO 31000:2018 - Risk management â€" Guidelines
11. ISO/IEC 27000:2018 Information technology â€" Security techniques[464]
    ISO - ISO/IEC 27000:2018 - Information technology â€" Security techniques â€" Information security management systems â€" Overview and vocabulary

[460] https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf
[461] https://resources.sei.cmu.edu/asset_files/Handbook/2003_002_001_14102.pdf
[462] https://www.interpol.int/content/download/13501/file/INTERPOL_DFL_GlobalGuidelinesDigitalForensicsLaboratory.pdf
[463] https://www.iso.org/standard/65694.html
[464] https://www.iso.org/standard/73906.html

Notes

(free space)

Notes

(free space)

Week 27: Management

Time needed: 13 hours

Approaches

1. Radical Candor â€” The Surprising Secret to Being a Good Boss[465]
   Radical Candor â€” The Surprising Secret to Being a Good Boss | First Round Review
2. The Future of Management Is Teal[466]
   The future of management is teal

Leadership

1. 30 Outdated Leadership Practices Holding Your Company Back[467]
   30 Outdated Leadership Practices Holding Your Company Back

[465] https://firstround.com/review/radical-candor-the-surprising-secret-to-being-a-good-boss/

[466] https://www.strategy-business.com/article/00344

[467] https://www.forbes.com/sites/mikemyatt/2013/07/28/30-outdated-leadership-practices-holding-your-company-back

Product/Project Management

1. 15 Ways to Screw Up an IT Project[468]
   15 Ways to Screw Up an IT Project | CIO
2. 15 Project Management Quotes That Will Help You Stay Motivated[469]
3. 20 Product Prioritization Techniques: A Map and Guided Tour[470]
   20 Product Prioritization Techniques: A Map and Guided Tour
4. Why Companies Need Full-Time Product Managers (And What They Do All Day)[471]
   Why Companies Need Full-Time Product Managers (And What They Do All Day) â€" Smashing Magazine Clear Search Back to top
5. Classic Mistakes Enumerated[472]
   Classic Mistakes Enumerated

Public Speaking

1. 20 tips for better conference speaking[473]
   20 tips for better conference speaking ~ Authentic Boredom
2. The Secret Activity Behind A Successful Speaker[474]
   The Secret Activity Behind A Successful Speaker

[468] https://www.cio.com/article/2384088/15-ways-to-screw-up-an-it-project.html
[469] https://www.lifehack.org/articles/work/15-project-management-quotes-that-will-help-you-stay-motivated.html
[470] https://foldingburritos.com/product-prioritization-techniques/
[471] https://www.smashingmagazine.com/2014/09/why-companies-need-full-time-product-managers/
[472] https://web.archive.org/web/20170707001328/http://www.stevemcconnell.com/rdenum.htm
[473] http://cameronmoll.com/archives/2009/02/20_tips_better_conference_speaking/
[474] https://www.forbes.com/sites/nickmorgan/2013/08/22/the-secret-activity-behind-a-successful-speaker

Roles

1. The Five Flavors of Being a CTO[475]
   The Five Flavors of Being a CTO
2. The Role of the CTO: Four Models for Success[476]

Quit

1. 17 REASONS NOT TO BE A MANAGER[477]
   17 Reasons NOT To Be A Manager â€“ charity.wtf
2. Career Break Or Sabbatical? How To Decide What Is Right For You[478]
   Career Break Or Sabbatical? How To Decide What Is Right For You | Careershifters
3. Career alternatives for a burnt-out developer?[479]
   Career alternatives for a burnt-out developer? - programmer burntout | Ask MetaFilter caret-down clock comment email facebook feed go-to-bottom go-to-top heart log-out moon pencil search-white twitter cog list user mefi-shirt bracketed-plus down-arrow html-bracket-left html-bracket-right slash two-lines bold close hyperlink icon_19502 icon_248 icon_299 italic media1 media2 media4 media5 media7 media8 music-note hide show
4. Surviving being senior (tech) management.[480]
   Surviving being senior (tech) management. | by kellan | Medium

[475] https://www.linkedin.com/pulse/five-flavors-being-cto-matt-tucker/
[476] http://www.brixtonspa.com/Career/The_Role_of_the_CTO_4Models.pdf
[477] https://charity.wtf/2019/09/08/reasons-not-to-be-a-manager/
[478] https://www.careershifters.org/expert-advice/career-break-or-sabbatical-how-to-decide-what-is-right-for-you
[479] https://ask.metafilter.com/124950/Career-alternatives-for-a-burntout-developer
[480] https://medium.com/@kellan/surviving-being-senior-tech-management-aa6654efd027

Mix

1. Your first 90 days as CTO or VP Engineering.[481]
   Your first 90 days as CTO or VP Engineering.
2. The Joel Test For Programmers (The Simple Programmer Test)[482]
   The Joel Test Updated For Programmers
3. Learnings from 80 startup CTOs[483]
   Learnings from 80 startup CTOs. I participated in a startup CTO meet-upâ€¦ | by Javier Escribano | Medium

---

Books

1. The Manager's Path[484]

[481] https://lethain.com/first-ninety-days-cto-vpe/
[482] https://simpleprogrammer.com/joel-test-programmers-simple-programmer-test/
[483] https://medium.com/@fesja/learnings-from-80-startup-ctos-88ddb5f9c024
[484] https://www.amazon.com/Managers-Path-Leaders-Navigating-Growth/dp/1491973897

Notes

(free space)

Notes

(free space)

Week 28: Management

Time needed: 18 hours

Books

1. Accelerate[485]
2. 97 Things Every Engineering Manager Should Know[486]
   Amazon.com: 97 Things Every Engineering Manager Should Know: Collective Wisdom from the Experts (9781492050902): Fournier, Camille: Books

[485] https://www.amazon.com/Accelerate-Building-Performing-Technology-Organizations/dp/1942788339

[486] https://www.amazon.com/Things-Every-Engineering-Manager-Should/dp/1492050903

Notes

(free space)

Notes

(free space)

Week 29: Management

Time needed: 15 hours

Videos

1. Andrea Provaglio: Rethinking Leadership Video @ GOTO 2017[487] | Slides[488]
2. Michael Lopp: The New Manager Death Spiral Video @ #LeadDevNewYork 2018[489] | Slides[490]
3. Patrick Kua: The Constant Life of a Tech Lead Video @ The Lead Developer UK 2017[491] | Slides[492]
4. Roy Osherove: Ten Mistakes Team Leaders Make Video @ Skills Matter 2011[493] | Slides[494]
5. Dan North: Patterns of Effective Teams Video @ GOTO 2017[495] | Slides[496]

[487] https://www.youtube.com/watch?v=A04Pu5LlzHw

[488] https://files.gotocon.com/uploads/slides/conference_7/273/original/GOTO%20Berlin%20-%20Rethinking%20Leadership-2.pdf

[489] https://www.youtube.com/watch?v=pAbU3WJ-NBw

[490] https://speakerdeck.com/calibrate/9-new-manager-death-spiral

[491] https://www.youtube.com/watch?v=9jd_vpcLK50

[492] https://www.slideshare.net/patkua/constant-life-of-a-tech-lead

[493] https://www.youtube.com/watch?v=qhjXc6niO3k

[494] https://www.slideshare.net/royosherove/ten-mistakes-software-team-leaders-make-by-roy-osherove-5whyscom

[495] https://www.youtube.com/watch?v=lvs7VEsQzKY

[496] https://files.gotocon.com/uploads/slides/conference_3/62/original/Patterns_of_Effective_Teams%20PDF.pdf

Books

1. Managing Humans[497]

[497] https://www.amazon.com/Managing-Humans-Humorous-Software-Engineering/dp/1484221575

Notes

(free space)

Notes

(free space)

Appendices

Appendix A: Extra Learning Paths

1. Startup Playbook[498]
   Startup Playbook
2. Awesome lists[499]
   GitHub - sindresorhus/awesome: ðŸ˜Ž Awesome lists about all kinds of interesting topics
3. Blockchain Learning Path[500]
   GitHub - protofire/blockchain-learning-path: A suggested learning path for blockchain development
4. Developer Roadmap[501]
   GitHub - luuductrung1234/dev-roadmap: the learning path and resource collections to become software developer
5. Kubernetes Learning Path v2.0[502]
   Kubernetes Learning Path | Microsoft Azure
6. Starway to Orione: the Orione Team Learning Path[503]
   GitHub - xpeppers/starway-to-orione: The Orione Team Learning Path
7. The of Secret Knowledge[504]
   GitHub - trimstray/the-book-of-secret-knowledge: A collection of inspiring lists, manuals, cheatsheets, blogs, hacks, one-liners, cli/web tools and more.
8. Virgilio[505]
   GitHub - virgili0/Virgilio: Your new Mentor for Data Science

[498] https://playbook.samaltman.com/
[499] https://github.com/sindresorhus/awesome
[500] https://github.com/protofire/blockchain-learning-path
[501] https://github.com/luuductrung1234/dev-roadmap
[502] https://azure.microsoft.com/en-us/resources/kubernetes-learning-path/
[503] https://github.com/xpeppers/starway-to-orione
[504] https://github.com/trimstray/the-book-of-secret-knowledge
[505] https://github.com/virgili0/Virgilio

E-Learning.

9. hacker-laws[506]
   GitHub - dwmkerr/hacker-laws: ðŸ'»ðŸ"– Laws, Theories, Principles and Patterns that developers will find useful. #hackerlaws
10. ShowPath.tech[507]
    GitHub - PJijin/Show-Path: ðŸ'¨â€�ðŸ'»Learning Path for Programmers https://roadmap.now.sh
11. Frontend Development[508]
    GitHub - dypsilon/frontend-dev-bookmarks: Manually curated collection of resources for frontend web developers.

[506] https://github.com/dwmkerr/hacker-laws
[507] https://github.com/PJijin/Show-Path
[508] https://github.com/dypsilon/frontend-dev-bookmarks

Appendix B: Exercises

Week 2

1. Let's Echo[509]
2. Looping and Skipping[510]
3. A Personalized Echo[511]
4. The World of Numbers[512]
5. Comparing Numbers[513]
6. Getting started with conditionals[514]
7. More on Conditionals[515]
8. Cut #1[516]
9. Cut #2[517]
10. Cut #3[518]
11. Cut #4[519]
12. Cut #5[520]
13. Cut #6[521]
14. Cut #7[522]

[509] https://www.hackerrank.com/challenges/bash-tutorials-lets-echo/problem
[510] https://www.hackerrank.com/challenges/bash-tutorials---looping-and-skipping/problem
[511] https://www.hackerrank.com/challenges/bash-tutorials---a-personalized-echo/problem
[512] https://www.hackerrank.com/challenges/bash-tutorials---the-world-of-numbers/problem
[513] https://www.hackerrank.com/challenges/bash-tutorials---comparing-numbers/problem
[514] https://www.hackerrank.com/challenges/bash-tutorials---getting-started-with-conditionals/problem
[515] https://www.hackerrank.com/challenges/bash-tutorials---more-on-conditionals/problem
[516] https://www.hackerrank.com/challenges/text-processing-cut-1/problem
[517] https://www.hackerrank.com/challenges/text-processing-cut-2/problem
[518] https://www.hackerrank.com/challenges/text-processing-cut-3/problem
[519] https://www.hackerrank.com/challenges/text-processing-cut-4/problem
[520] https://www.hackerrank.com/challenges/text-processing-cut-5/problem
[521] https://www.hackerrank.com/challenges/text-processing-cut-6/problem
[522] https://www.hackerrank.com/challenges/text-processing-cut-7/problem

15. Cut #8[523]
16. Cut #9[524]

Week 4

1. The Hurdle Race[525]
2. A Very Big Sum[526]
3. Designer PDF Viewer[527]
4. Viral Advertising[528]
5. Solve Me First[529]
6. Correctness and the Loop Invariant[530]
7. Breaking the Records[531]
8. Intro to Tutorial Challenges[532]
9. Plus Minus[533]
10. Staircase[534]
11. CamelCase[535]
12. Cats and a Mouse[536]
13. Bill Division[537]
14. Utopian Tree[538]
15. Divisible Sum Pairs[539]

[523] https://www.hackerrank.com/challenges/text-processing-cut-8/problem
[524] https://www.hackerrank.com/challenges/text-processing-cut-9/problem
[525] https://www.hackerrank.com/challenges/the-hurdle-race
[526] https://www.hackerrank.com/challenges/a-very-big-sum
[527] https://www.hackerrank.com/challenges/designer-pdf-viewer
[528] https://www.hackerrank.com/challenges/strange-advertising
[529] https://www.hackerrank.com/challenges/solve-me-first
[530] https://www.hackerrank.com/challenges/correctness-invariant
[531] https://www.hackerrank.com/challenges/breaking-best-and-worst-records
[532] https://www.hackerrank.com/challenges/tutorial-intro
[533] https://www.hackerrank.com/challenges/plus-minus
[534] https://www.hackerrank.com/challenges/staircase
[535] https://www.hackerrank.com/challenges/camelcase
[536] https://www.hackerrank.com/challenges/cats-and-a-mouse
[537] https://www.hackerrank.com/challenges/bon-appetit
[538] https://www.hackerrank.com/challenges/utopian-tree
[539] https://www.hackerrank.com/challenges/divisible-sum-pairs

16. Service Lane[540]
17. Alternating Characters[541]
18. Insertion Sort - Part 2[542]
19. Sequence Equation[543]
20. Counting Sort 2[544]
21. Maximizing XOR[545]
22. Birthday Cake Candles[546]
23. The Love-Letter Mystery[547]
24. Find Digits[548]
25. Lonely Integer[549]
26. Grading Students[550]
27. Beautiful Days at the Movies[551]
28. Jumping on the Clouds: Revisited[552]
29. Marc's Cakewalk[553]
30. Flipping bits[554]

Week 5

1. Fizz Buzz Kata[555]
   Fizz Buzz Kata

[540] https://www.hackerrank.com/challenges/service-lane
[541] https://www.hackerrank.com/challenges/alternating-characters
[542] https://www.hackerrank.com/challenges/insertionsort2
[543] https://www.hackerrank.com/challenges/permutation-equation
[544] https://www.hackerrank.com/challenges/countingsort2
[545] https://www.hackerrank.com/challenges/maximizing-xor
[546] https://www.hackerrank.com/challenges/birthday-cake-candles
[547] https://www.hackerrank.com/challenges/the-love-letter-mystery
[548] https://www.hackerrank.com/challenges/find-digits
[549] https://www.hackerrank.com/challenges/lonely-integer
[550] https://www.hackerrank.com/challenges/grading
[551] https://www.hackerrank.com/challenges/beautiful-days-at-the-movies
[552] https://www.hackerrank.com/challenges/jumping-on-the-clouds-revisited
[553] https://www.hackerrank.com/challenges/marcs-cakewalk
[554] https://www.hackerrank.com/challenges/flipping-bits
[555] https://kata-log.rocks/fizz-buzz-kata

2. Roman Numerals Kata[556]
   Roman Numerals Kata
3. String Calculator Kata[557]
   String Calculator Kata
4. Task List Kata[558]
   Task List Kata

Week 6

1. Tell Don't Ask Kata[559]
   Tell Don't Ask Kata
2. Game of Life Kata[560]
   Game of Life Kata
3. Banking Kata[561]
   Banking Kata
4. Gossiping Bus Drivers Kata[562]
   Gossiping Bus Drivers Kata

Week 11

1. Race Car Katas - Leaderboard[563]
   Race Car Katas - Leaderboard
2. Mars Rover Kata[564]
   Mars Rover Kata

[556] https://kata-log.rocks/roman-numerals-kata
[557] https://kata-log.rocks/string-calculator-kata
[558] https://kata-log.rocks/task-list-kata
[559] https://kata-log.rocks/tell-dont-ask-kata
[560] https://kata-log.rocks/game-of-life-kata
[561] https://kata-log.rocks/banking-kata
[562] https://kata-log.rocks/gossiping-bus-drivers-kata
[563] https://kata-log.rocks/race-car-katas-leaderboard
[564] https://kata-log.rocks/mars-rover-kata

Week 12

1. Japan Population[565]
2. Population Density Difference[566]
3. Revising Aggregations - Averages[567]
4. Weather Observation Station 16[568]
5. Employee Names[569]
6. Japanese Cities' Names[570]
7. Select By ID[571]
8. Select All[572]
9. Japanese Cities' Attributes[573]
10. Revising Aggregations - The Sum Function[574]
11. Revising Aggregations - The Count Function[575]
12. Revising the Select Query II[576]
13. Stand out from the crowd[577]
14. Weather Observation Station 14[578]
15. Employee Salaries[579]
16. Average Population[580]
17. Weather Observation Station 1[581]
18. Weather Observation Station 13[582]

[565] https://www.hackerrank.com/challenges/japan-population
[566] https://www.hackerrank.com/challenges/population-density-difference
[567] https://www.hackerrank.com/challenges/revising-aggregations-the-average-function
[568] https://www.hackerrank.com/challenges/weather-observation-station-16
[569] https://www.hackerrank.com/challenges/name-of-employees
[570] https://www.hackerrank.com/challenges/japanese-cities-name
[571] https://www.hackerrank.com/challenges/select-by-id
[572] https://www.hackerrank.com/challenges/select-all-sql
[573] https://www.hackerrank.com/challenges/japanese-cities-attributes
[574] https://www.hackerrank.com/challenges/revising-aggregations-sum
[575] https://www.hackerrank.com/challenges/revising-aggregations-the-count-function
[576] https://www.hackerrank.com/challenges/revising-the-select-query-2
[577] https://www.hackerrank.com/challenges/weather-observation-station-14
[578] https://www.hackerrank.com/challenges/salary-of-employees
[579] https://www.hackerrank.com/challenges/average-population
[580] https://www.hackerrank.com/challenges/weather-observation-station-1
[581] https://www.hackerrank.com/challenges/weather-observation-station-13
[582] https://www.hackerrank.com/challenges/weather-observation-station-10

19. Weather Observation Station 10[583]
20. African Cities[584]

Week 15

1. Update List[585]
2. Reverse a List[586]
3. Filter Array[587]
4. List Length[588]
5. Solve Me First FP[589]
6. Filter Positions in a List[590]
7. Sum of Odd Elements[591]

[583] https://www.hackerrank.com/challenges/african-cities
[584] https://www.hackerrank.com/challenges/weather-observation-station-2
[585] https://www.hackerrank.com/challenges/fp-update-list/problem
[586] https://www.hackerrank.com/challenges/fp-reverse-a-list/problem
[587] https://www.hackerrank.com/challenges/fp-filter-array/problem
[588] https://www.hackerrank.com/challenges/fp-list-length/problem
[589] https://www.hackerrank.com/challenges/fp-solve-me-first/problem
[590] https://www.hackerrank.com/challenges/fp-filter-positions-in-a-list/problem
[591] https://www.hackerrank.com/challenges/fp-sum-of-odd-elements

Appendix C: More IT Books

1. Freely available programming books[592]
   GitHub - EbookFoundation/free-programming-books: Freely available programming books
2. Sitepoint Library[593]
   Library - SitePoint Premium

[592] https://github.com/EbookFoundation/free-programming-books
[593] https://www.sitepoint.com/premium/library/

Appendix D: IT Trends

1. Master Technology Trends, Digital Trends & Digital Business[594]
   Master Technology Trends, Digital Trends & Digital Business
2. Technology Radar[595]
   Technology Radar | An opinionated guide to technology frontiers | ThoughtWorks
3. Stack Overflow Annual Developer Survey[596]
   Stack Overflow Developer Survey 2020
4. Exploit Database - Exploits for Penetration Testers, Researchers, and Ethical Hackers[597]
5. The 2020 State of DevOps Report is here![598]
   2020 State of DevOps Report | presented by Puppet, & CircleCi
6. The Global CTO Survey 2020 Report[599]
   The Global CTO Survey 2020 Report
7. State Of Remote Work[600]
   State of Remote Work 2019 | Buffer

[594] https://www.gartner.com/en/information-technology/insights/trends-predictions
[595] https://www.thoughtworks.com/radar
[596] https://insights.stackoverflow.com/survey/2020
[597] https://www.exploit-db.com/
[598] https://puppet.com/resources/report/2020-state-of-devops-report
[599] https://www.stxnext.com/resources/cto-survey-2020
[600] https://buffer.com/state-of-remote-work-2019