



1ST EDITION

# Technical Program Manager's Handbook

Empowering managers to efficiently manage technical projects and build a successful career path

**JOSHUA ALAN TETER**

Foreword by Ben Tobin  
Career and Leadership Coach, Ben Tobin Coaching  
Former Software Development Manager, Amazon



# Preface

The role of a Technical Program Manager (or TPM) has been around inside and outside of the tech industry for quite a while; yet somehow there is still quite a sense of mystery around what the role is and why it is beneficial, let alone how someone can succeed in being a TPM. This book looks to correct that by diving into what it means to be a TPM, where the role came from, and where it is headed. You'll get a look into how the TPM works and develops their career in the Big 5 – Amazon, Apple, Alphabet (Google), Meta (Facebook), and Microsoft.

I've been at Amazon for a little over 9 years now and I remember that when I first interviewed, I had a hard time remembering what TPM even stood for, let alone what they did. In my onsite interviews, I asked what the job role was and what the day-to-day was like. 9 years later and I'm asked those same questions by interviewees at least once a week. I attend conferences discussing what it means to be a TPM and have written papers on what it means to be a TPM within my own organization because as you'll see in this book, it depends on where you are as to what the role entails. However, no matter what, there are foundational principles that are followed across the industry that will set you on the right path and help you when you get stuck in a rut without a way forward.

Let's get you ready to be a successful TPM!

## Who this audiobook is for

This book is meant for TPMs at every stage of their career, including those that are considering transitioning into the role. To get the most out of this book, there is an expectation that the reader will have some basic knowledge of project management. I tend to lean into the **Project Management Professional** (or **PMP**) lingo and style but the book does not follow a specific methodology, as I don't believe a single methodology can be adequately applied to this role!

The book will cover some basic programming topics, although very little code is used except for illustrative purposes in *Chapter 11, Code Development Expectations*. Most concepts are explored using figures and text, as that fits the audience of the book the best.

To read the book, there's no expectation of a specific technical proficiency, although as you will discover, there is an expectation that you'll have that if you want to be a TPM. This book will guide you through the technical skills that are prerequisites for most TPMs.

## What this audiobook covers

**Chapter 1, *Fundamentals of a Technical Program Manager***, is an introduction to what a TPM is and where the role originated.

**Chapter 2, *Pillars of a Technical Program Manager***, sets out the three pillars of a TPM: project management, program management, and the technical toolset.

**Chapter 3, *Introduction to Program Management***, covers the key management areas that will be covered throughout the book: plan management, risk management, and stakeholder management. It also introduces a case study that will be used for all examples throughout the book.

**Chapter 4, *Driving Toward Clarity***, elaborates on the recurring trait that defines everything a TPM does: being clarity-driven.

**Chapter 5, *Plan Management***, dives deeper into the plan management best practices and goes over scenarios that are common in the tech industry.

**Chapter 6, *Risk Management***, explores the risk management best practices and goes over scenarios that are common in the tech industry.

**Chapter 7, *Stakeholder Management***, discusses the stakeholder management best practices and goes over scenarios that are common in the tech industry.

**Chapter 8, *Managing a Program***, explains the differences between managing a program and a project and how program management builds on top of project management.

**Chapter 9, *Career Paths***, examines the career paths available for a TPM using interviews and job data from across the Big 5 tech companies.

**Chapter 10, *The Technical Toolset***, is all about the three fundamental tools in a TPM's technical toolset: programming fundamentals, system design, and architectural design.

*Chapter 11, Code Development Expectations*, is an outline of the programming fundamentals that a TPM is expected to understand and draw upon.

*Chapter 12, System Design and Architectural Landscape*, clarifies the system and architectural design patterns and principles that are useful to a TPM.

*Chapter 13, Enhancing Management Using Your Technical Toolset*, covers the technical toolset and dives deeper into how and where in a TPM's day-to-day work it can be used to enhance their career.

## Download the color images

We also provide a PDF file that has color images of the screenshots and diagrams used in this book. You can download it here: <https://packt.link/yfEY>.

# Chapter 1

## Figures

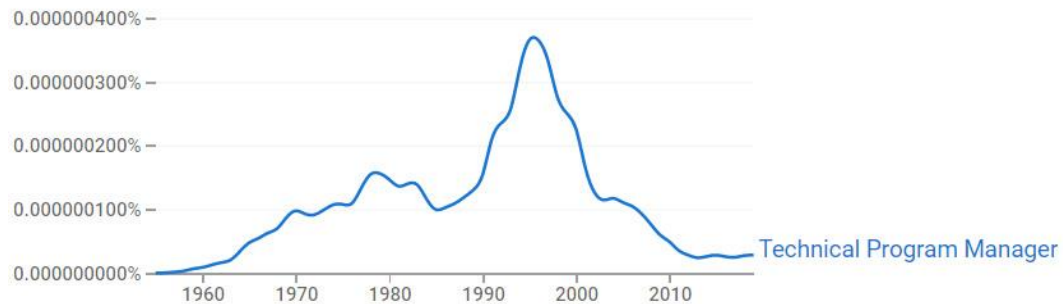


Figure 1.1 – Google Ngram Viewer results of the occurrence of the term “Technical Program Manager” from 1955 to 2019 with a smoothing of 3

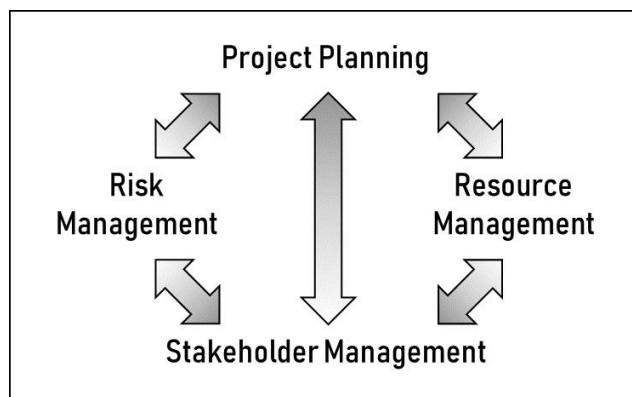
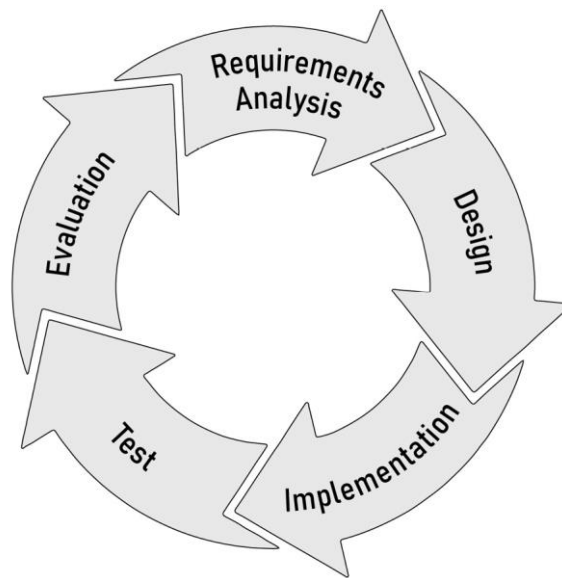


Figure 1.2 – Key management areas



**Figure 1.3 – The SDLC**

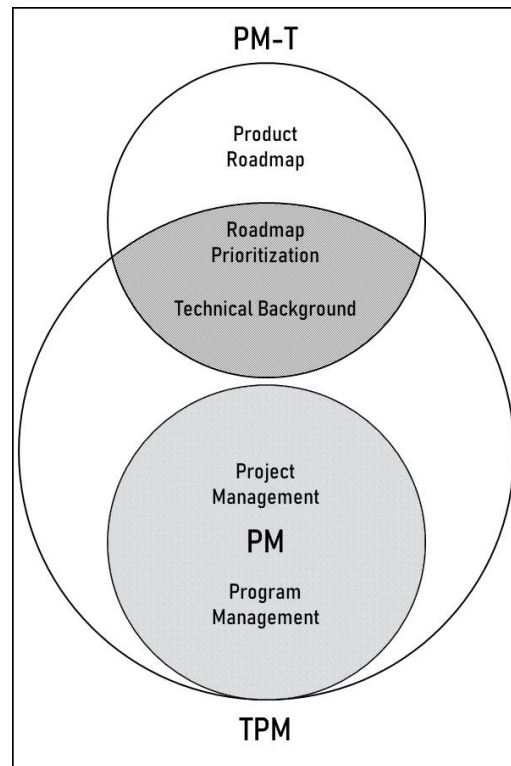


Figure 1.4 – A Venn diagram showing the PM, TPM, and PM-T

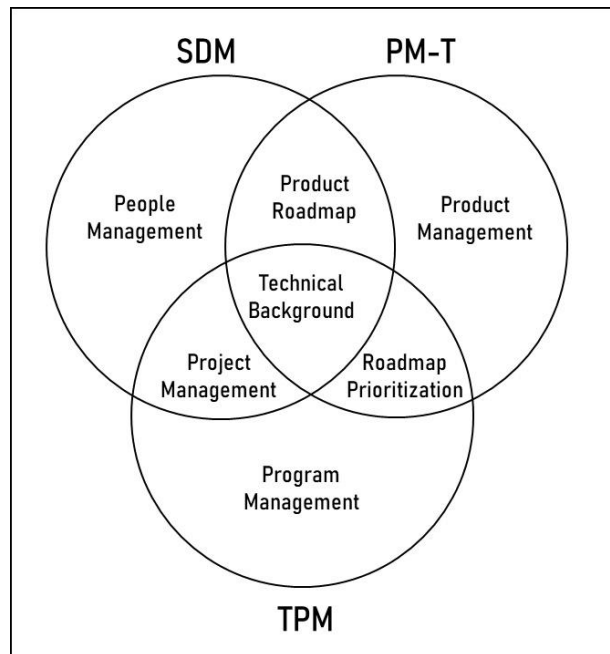


Figure 1.5 – A Venn diagram showing the TPM, SDM, and PM-T

## Tables

| Normalized Level | Company   | Qualifications                                      | Education  |
|------------------|-----------|---|--|
| Entry            | Apple     | SDLC<br>PM  | CS or Comparable                                     |
|                  | Google    | Align across multiple teams<br>PM/SDLC              | CS or Comparable                                     |
|                  | Microsoft | PM/SDLC<br>Influence without authority<br>Biz Intel | CS or Comparable<br>Equivalent Work Experience (EWE) |
| Industry         | Amazon    | PM/SDLC<br>Remove ambiguity                         | CS or Comparable<br>EWE                              |



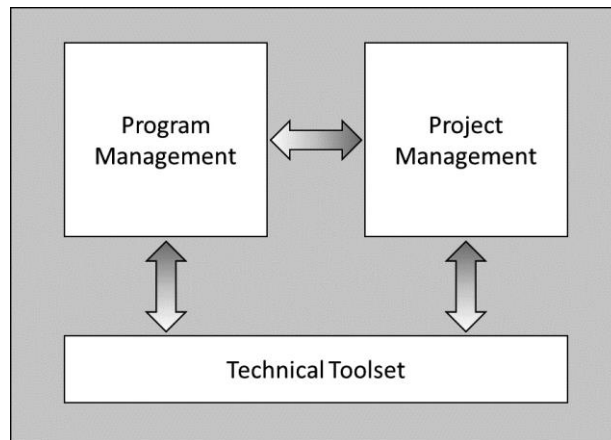
|                  |           |  |                           |
|------------------|-----------|--|---------------------------|
|                  |           | Thought leadership   |                           |
|                  | Apple     | Leading a team<br>Established PM/SDLC<br>Communication<br>Strategy and Program<br>Delivery | BS or MS<br>EWE           |
|                  | Google    | E2E Delivery<br>System Design<br>Data Analysis   | CS or Comparable          |
|                  | Meta      | PM/SDLC<br>Works with other TPMs   | CS or Comparable<br>EWE   |
|                  | Microsoft | Exp. Writing code<br>Defines program goals<br>PM/SDLC                                      | CS or Comparable<br>EWE   |
| <b>Principal</b> | Amazon    | PM/SDLC<br>Remove ambiguity<br>Thought leadership  | CS or Comparable<br>EWE   |
|                  | Microsoft | Proven PM<br>Strong technical<br>proficiency<br>Excellent communication                    | BS/MS in CS or Comparable |

**Table 1.1 – A functional comparison of job roles across the tech industry**

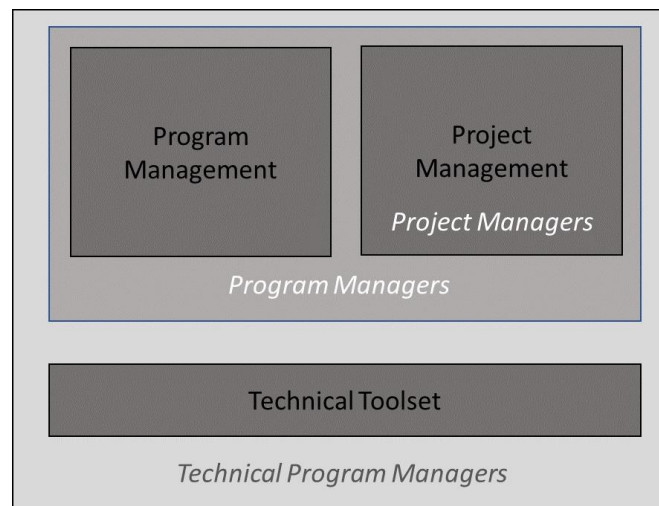
| Level     | Company   | Focus  | Years of Experience |
|-----------|-----------|--|---------------------|
| Entry     | Apple     | SDLC/PM Fundamentals                             | None                |
|           | Google    | SDLC/PM Fundamentals                             | 0 to 1              |
|           | Microsoft | SDLC/PM Fundamentals<br>Influence w/o authority  | None                |
| Industry  | Apple     | Strategy and Program Deliver                     | 5 to 10             |
|           | Amazon    | SDLC/PM Fundamentals<br>Remove ambiguity         | 3 to 8              |
|           | Google    | End-to-End Delivery<br>System Design             | 2 to 6              |
|           | Meta      | SDLC/PM Fundamentals<br>Cross-team collaboration | 4 to 7              |
|           | Microsoft | Program definition<br>SDLC/PM proven record      | 6 to 10             |
| Principal | Amazon    | Program delivery<br>Identify ambiguous problems  | Over 8              |
|           | Microsoft | Proven PM/SDLC<br>Strong technical proficiency   | 8 +                 |

**Table 1.2 – The TPM progression through three job levels**

## Chapter 2



**Figure 2.1 – The pillars of a TPM**



**Figure 2.2 – The functional overlap between the role of program managers and TPMs**



Figure 2.3 – The project management triangle

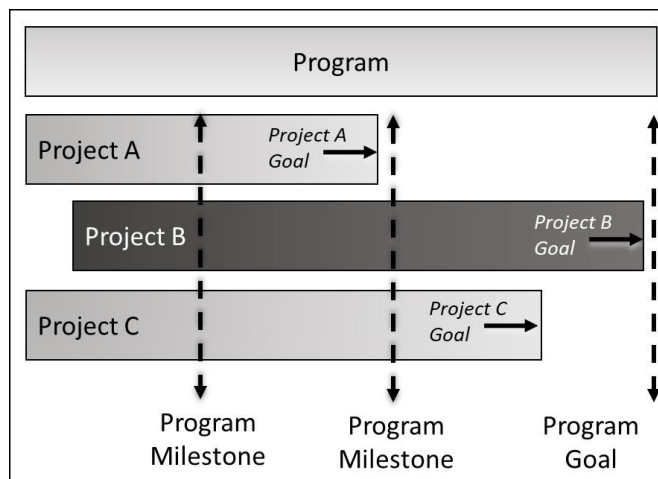
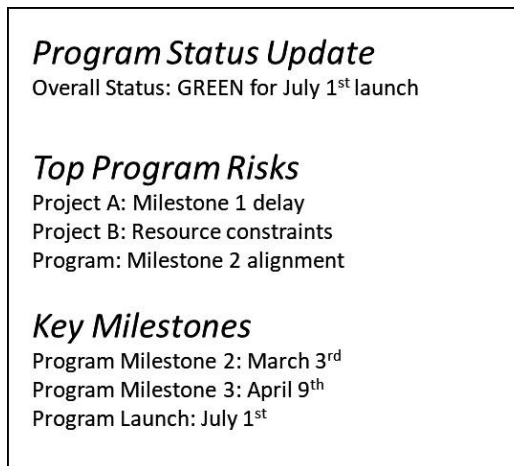
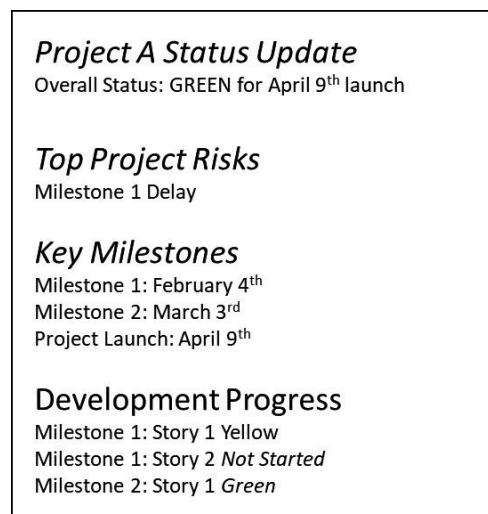


Figure 2.4 – A project versus a program



**Figure 2.5 – A program status update**



**Figure 2.6 – A status update for Project A**

# Chapter 3

## Figures

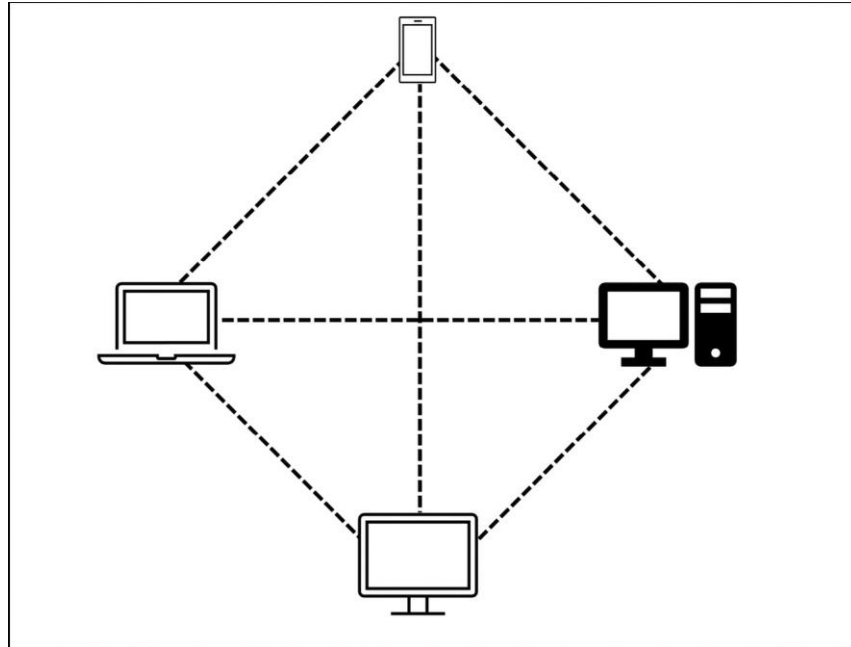


Figure 3.1 – A simple P2P network diagram

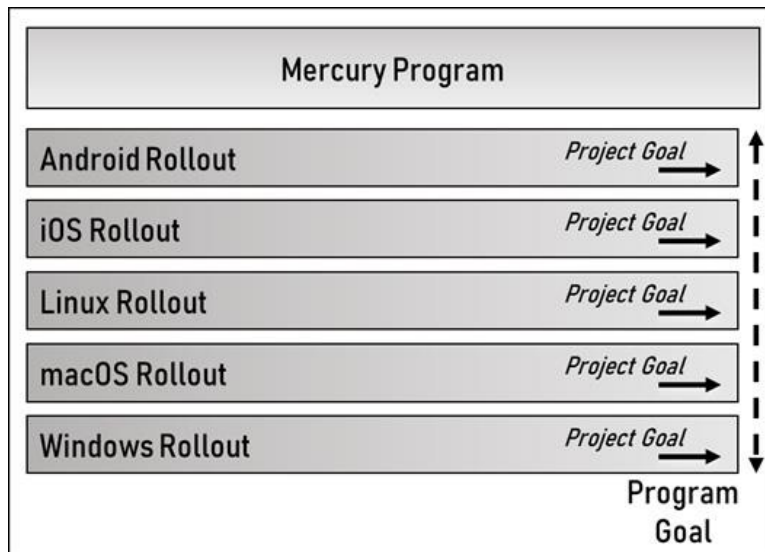


Figure 3.2 – The Mercury program roadmap

## Tables

| Project/program name    | Description                     | Goal  | How does it help the program?                                   |
|-------------------------|---------------------------------|---|---|
| Mercury program         | Program for P2P messaging app   | Build and deploy a P2P messaging app to 90% of the user base. | N/A   |
| Android rollout project | Project for the P2P Android app | Deploy the app for the Android ecosystem.                     | Android represents 72% of the worldwide market share on mobile. |
| iOS rollout project     | Project for the P2P iOS app     | Deploy the app for the iOS ecosystem.                         | iOS represents 26% of the worldwide market share on mobile.     |
| Linux rollout project   | Project for the P2P Linux app   | Deploy the app for the Linux ecosystem.                       | Linux represents 2.5% of the worldwide                          |

|                         |                                 |   |   |
|-------------------------|---------------------------------|---|---|
|                         |                                 |   | desktop market share.   |
| macOS rollout project   | Project for the P2P macOS app   | Deploy the app for the macOS ecosystem.   | macOS represents 15% of the worldwide desktop market share.   |
| Windows rollout project | Project for the P2P Windows app | Deploy the app for the Windows ecosystem. | Windows represents 76% of the worldwide desktop market share. |

**Table 3.1 – The Mercury project structure**

| ID | Milestone            | Predecessor | Effort (weeks) |
|----|----------------------|-------------|----------------|
| 1  | P2P Subsystem Ready  |             | 8              |
| 2  | User Interface Ready | 1           | 16             |
| 3  | End-to-End (E2E)     | 2           | 4              |

**Table 3.2 – A simplified view of the Windows project plan**



| Project /  |                                       | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug |
|------------|---------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|
| ID         | Milestone                             |     |     |     |     |     |     |     |     |
| 1 Windows  | P2P Subsystem Ready                   |     |     |     |     |     |     |     |     |
| 2 Windows  | UI Ready                              |     |     |     |     |     |     |     |     |
| 3 Windows  | E2E Testing Complete                  |     |     |     |     |     |     |     |     |
| 4 macOS    | P2P Subsystem Ready                   |     |     |     |     |     |     |     |     |
| 5 macOS    | UI Ready                              |     |     |     |     |     |     |     |     |
| 6 macOS    | E2E Testing Complete                  |     |     |     |     |     |     |     |     |
| 7 Linux    | P2P Subsystem Ready                   |     |     |     |     |     |     |     |     |
| 8 Linux    | UI Ready                              |     |     |     |     |     |     |     |     |
| 9 Linux    | E2E Testing Complete                  |     |     |     |     |     |     |     |     |
| 10 Android | P2P Subsystem Ready                   |     |     |     |     |     |     |     |     |
| 11 Android | UI Ready                              |     |     |     |     |     |     |     |     |
| 12 Android | E2E Testing Complete                  |     |     |     |     |     |     |     |     |
| 13 iOS     | P2P Subsystem Ready                   |     |     |     |     |     |     |     |     |
| 14 iOS     | UI Ready                              |     |     |     |     |     |     |     |     |
| 15 iOS     | E2E Testing Complete                  |     |     |     |     |     |     |     |     |
| 16 Mercury | Windows to macOS Integration Testing  |     |     |     |     |     |     |     |     |
| 17 Mercury | Windows to Anroid Integration Testing |     |     |     |     |     |     |     |     |
| 18 Mercury | Windows to Linux Integration Testing  |     |     |     |     |     |     |     |     |
| 19 Mercury | Windows to iOS Integration Testing    |     |     |     |     |     |     |     |     |
| 20 Mercury | macOS to Linux Integration Testing    |     |     |     |     |     |     |     |     |
| 21 Mercury | macOS to Android Integration Testing  |     |     |     |     |     |     |     |     |
| 22 Mercury | macOS to iOS Integration Testing      |     |     |     |     |     |     |     |     |
| 23 Mercury | Linux to Android Integration Testing  |     |     |     |     |     |     |     |     |
| 24 Mercury | Linux to iOS Integration Testing      |     |     |     |     |     |     |     |     |
| 25 Mercury | Android to iOS Integration Testing    |     |     |     |     |     |     |     |     |

Table 3.3 – The program plan with the Gantt chart

| Project /  |                                       | Jul | Aug |
|------------|---------------------------------------|-----|-----|
| ID         | Milestone                             |     |     |
| 16 Mercury | Windows to macOS Integration Testing  |     |     |
| 17 Mercury | Windows to Anroid Integration Testing |     |     |
| 18 Mercury | Windows to Linux Integration Testing  |     |     |
| 19 Mercury | Windows to iOS Integration Testing    |     |     |
| 20 Mercury | macOS to Linux Integration Testing    |     |     |
| 21 Mercury | macOS to Android Integration Testing  |     |     |
| 22 Mercury | macOS to iOS Integration Testing      |     |     |
| 23 Mercury | Linux to Android Integration Testing  |     |     |
| 24 Mercury | Linux to iOS Integration Testing      |     |     |
| 25 Mercury | Android to iOS Integration Testing    |     |     |

Table 3.4 – A closeup of the Gantt chart, focusing on the integration testing

| ID | Project/program         | Risk                     | Strategy                           |
|----|-------------------------|--------------------------|------------------------------------|
| 1  | Windows To-Do App       | Network testing failures | Acceptance – 3-week milestone slip |
| 2  | Linux To-Do App         | Distribution regressions | Mitigation                         |
| 3  | Android Project Tracker | App approval delays      | Acceptance – 2-week addition       |
| 4  | iOS Project Tracker     | App approval delays      | Acceptance – 3-week addition       |

**Table 3.5 – A risk register for the Mercury program company**

| ID | Project / Program | Stakeholder              | Development Report | Monthly Review | Quarterly Review |
|----|-------------------|--------------------------|--------------------|----------------|------------------|
| 1  | macOS             | TPM Manager              | Yes                | Yes            | Yes              |
| 2  | Windows           | Dev. Team Manager        | Yes                | Yes            | No               |
| 3  | Linux             | Lead Engineer            | Yes                | No             | No               |
| 4  | Android           | Android Division Lead    | No                 | Yes            | Yes              |
| 5  | iOS               | Director, Mobile Systems | No                 | Yes            | Yes              |
| 6  | Mercury           | VP, Productivity         | No                 | No             | Yes              |

**Table 3.6 – The stakeholder plan for the Mercury program**

# Chapter 4

## Figures

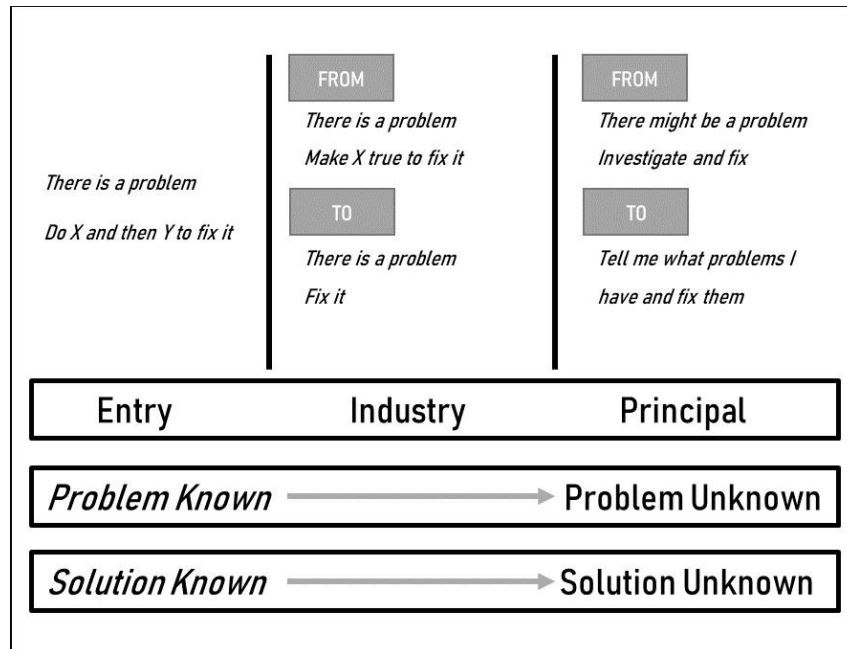


Figure 4.1 – Growth by ambiguity

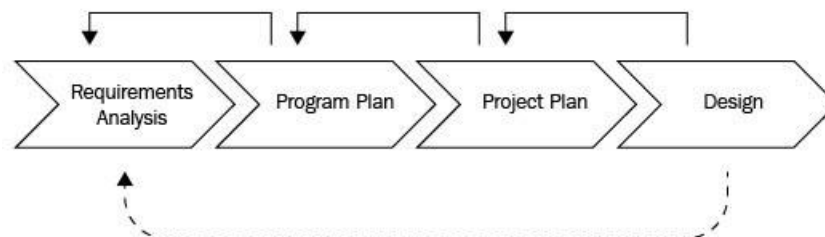
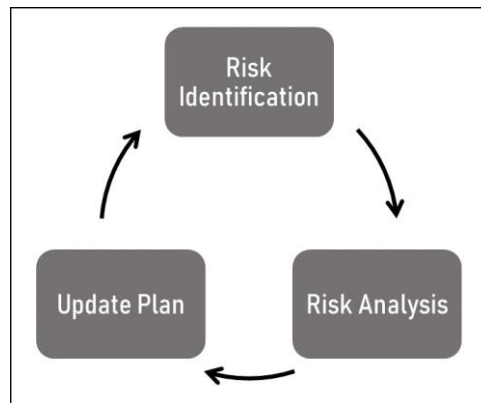
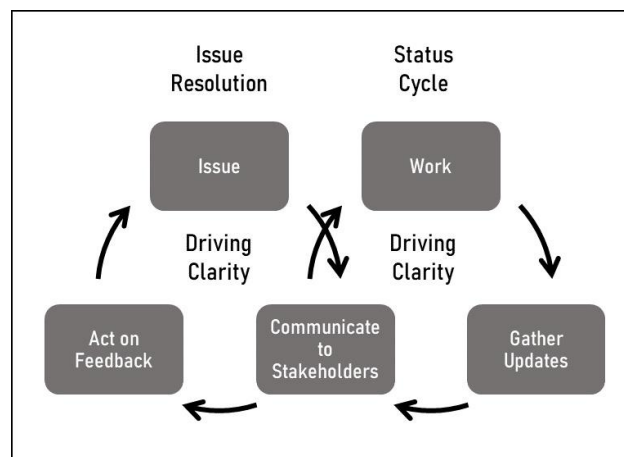


Figure 4.2 – Driving clarity in plan management



**Figure 4.3 – Driving clarity in risk management**



**Figure 4.4 – Driving clarity using stakeholder communication**

## Links

Visit <https://www.mariogerard.com/take-it-to-the-next-level-as-a-tpm/> for more information.



# Chapter 5

## Tables

| Tool       | Project Management | Portfolio Management | Resource Management | Stakeholder Management |
|------------|--------------------|----------------------|---------------------|------------------------|
| MS Project | ☑                  |                      | ☑                   | ☑                      |
| Smartsheet | ☑                  | ☑                    | ☑                   | ☑                      |
| Clarizen   | ☑                  | ☑                    | ☑                   | ☑                      |
| Asana      | ☑                  | ☑                    | ☑                   | ☑                      |
| MS Excel   |                    |                      |                     |                        |

**Table 5.1 – Comparing program management tools**

| ID | Requirement   |
|----|---|
| 1  | Create a <b>peer-to-peer</b> (or <b>P2P</b> ) messaging system  |
| 2  | System must allow sending text messages to other peers on the system  |
| 3  | Standard UX elements from other messaging apps should be available  |
| 4  | All messages sent and received for a user must be accessible to the user within the app until the user explicitly deletes a message |

**Table 5.2 – Initial requirements for the Mercury program’s Windows Rollout Project**

| ID         | Requirements  |
|------------|---|
| <b>1.0</b> | <b>A P2P messaging system, with no servers, should be created</b>                           |
| <b>2.0</b> | <b>The system must allow sending text messages to other users on the network/system</b>     |
| 2.1        | The text message should support all Unicode characters, including emojis                    |
| 2.2        | The text should support rich text formatting (bold, italic, underline, font type, and size) |
| <b>3.0</b> | <b>Standard UX elements seen in other messaging apps should be available</b>                |
| 3.1        | Address book of saved contacts  |
| 3.1.1      | Add to address book   |

|            |  |
|------------|--|
| 3.1.2      | Remove from address book   |
| 3.1.3      | Load address book entirely or a single address                       |
| 3.1.3      | Export address book entirely or a single address                     |
| 3.2        | User profile   |
| 3.2.1      | User profile image   |
| 3.2.2      | User alias should be changeable                                      |
| 3.2.3      | Short bio section including description/bio, company, title          |
| 3.3        | Presence indicator   |
| 3.3.1      | Configurable statuses  |
| 3.3.2      | Configurable locations   |
| 3.4        | Access control   |
| 3.4.1      | User should be able to accept contact requests                       |
| 3.4.2      | User should be able to block contact requests                        |
| <b>4.0</b> | <b>All messages sent and received should be visible to the user</b>  |
| 4.1        | Exception for when a message is deleted by the user                  |
| 4.2        | Messages should contain a status indicator of {sent, received, read} |

**Table 5.3 – Clarified requirements**

| ID | Requirement IDs | Use Case   |
|----|-----------------|--|
| 1  | 1.0             | As an admin, have no centralized setup or maintenance        |
| 2  | 1.0             | As a user, install and use without a central server          |
| 3  | 3.2             | As a user, create a user profile with a picture and an alias |
| 4  | 3.1             | As a user, add or remove a contact to my contact list        |
| 5  | 3.4             | As a user, block and accept a contact request                |
| 6  | 2               | As a user, send messages to a contact using rich text        |
| 7  | 4               | As a user, see new messages sent to me                       |
| 8  | 4               | As a user, see all messages both sent and received           |
| 9  | 4.1             | As a user, delete a message                                  |
| 10 | 3.3             | As a user, set presence information                          |

**Table 5.4 – Use cases**

| ID | Use Case ID | Task | Estimate (Weeks) |
|----|-------------|------|------------------|
|----|-------------|------|------------------|

|    |      |   |     |
|----|------|---|-----|
| 1  | 1, 2 | Create P2P network                                    | 12  |
| 2  | 1, 2 | Design network  | 4   |
| 3  | 1, 2 | Implement   | 6   |
| 4  | 6    | Text message send/receive API                         | 8   |
| 5  | 6    | Set up API for request/response                       | 4   |
| 6  | 7, 8 | Use an Ack tag to track message status                | 2   |
| 7  | 6    | Ensure Unicode support in API payload                 | 1   |
| 8  | 5    | Text message to new contact initiates contact request | 1   |
| 9  | 4    | Address book  | 16  |
| 10 | 4    | Add/remove API using message protocol/API             | 4   |
| 11 | 4    | Import/export of the address book                     | 4   |
| 12 | 4    | Import/export of an address entry                     | 4   |
| 13 | -    | Support search/discovery for members on network       | 4   |
| 14 | 3    | User profile object                                   | 3   |
| 15 | 3    | CRUD  | 1   |
| 16 | 3    | Alias CRUD  | 1   |
| 17 | 3    | Bio text CRUD   | 1   |
| 18 | 10   | Presence object                                       | 2.5 |
| 19 | 10   | Status key-value pair                                 | 1   |
| 20 | 10   | Location key-value pair                               | 1   |
| 21 | 10   | Support full Unicode including emojis in values       | 0.5 |
| 22 | 5    | Access control  | 1   |
| 23 | 5    | Accept or deny contact request                        | 1   |
| 24 | 4    | Message library                                       | 2.5 |
| 25 | 8    | Maintain message list for both sent and received      | 2   |
| 26 | 9    | Allow deletion of sent/received message from list     | 0.5 |

**Table 5.5 – From use cases to tasks**

| ID | ReqId | Task               | Duration | Swarm # | Predecessors | Start Date | End Date |
|----|-------|--------------------|----------|---------|--------------|------------|----------|
| 1  | 1     | Create P2P network | 16       |         |              |            |          |



|   |       |   |           |   |     |           |           |
|---|-------|---|-----------|---|-----|-----------|-----------|
| 2 | 1     | Design network  | 6         | 1 |     | 2-Jan-23  | 6-Feb-23  |
| 3 | 1     | Implement   | 10        | 2 | 2fs | 13-Feb-23 | 24-Mar-23 |
| 4 | 1.3   | Create a networkId to tie a message to a given network instance | 3         | 1 | 2fs | 27-Mar-23 | 14-Apr-23 |
| 5 | 2     | <b>Text message send/receive API</b>                            | <b>13</b> |   |     |           |           |
| 6 | 2     | Set up API for request/response                                 | 6         | 2 |     | 2-Jan-23  | 20-Jan-23 |
| 7 | 4.2   | Use an Ack tag to track message status                          | 3         | 1 | 6fs | 23-Jan-23 | 17-Feb-23 |
| 8 | 2.1   | Ensure Unicode support in API payload                           | 2         | 1 | 6fs | 23-Jan-23 | 3-Feb-23  |
| 9 | 3.4.1 | New text message to new contact initiates contact request       | 2         | 1 | 6fs | 6-Feb-23  | 17-Feb-23 |

**Table 5.6 – Excerpt of Windows Rollout Project Plan**

| Level of Ambiguity in Task | Confidence of Estimate Accuracy | Team Overhead | Buffer |
|----------------------------|---------------------------------|---------------|--------|
| Medium                     | Low                             | 10%           | 35%    |
| High                       | Low                             | 10%           | 40%    |
| Low                        | Medium                          | 10%           | 25%    |
| Medium                     | Medium                          | 10%           | 30%    |
| High                       | Medium                          | 10%           | 35%    |
| Low                        | High                            | 10%           | 20%    |

|        |      |     |     |
|--------|------|-----|-----|
| Medium | High | 10% | 25% |
|--------|------|-----|-----|

**Table 5.7 – Estimation buffer matrix**

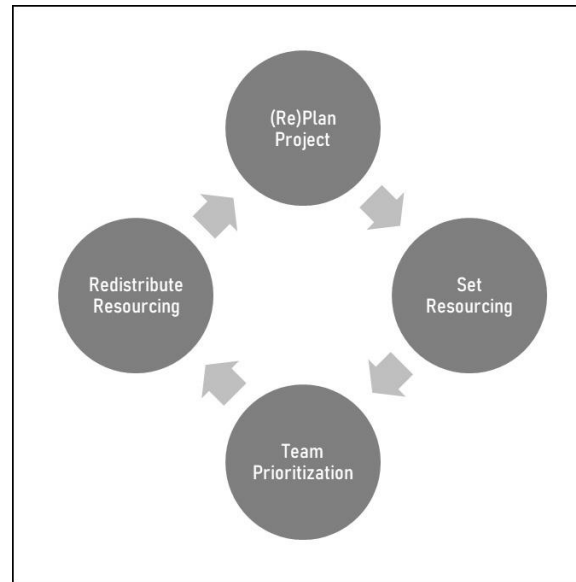
| Feature                       | Start Date | End Date  |
|-------------------------------|------------|-----------|
| Create P2P network            | 2-Jan-23   | 14-Apr-23 |
| Text message send/receive API | 2-Jan-23   | 17-Feb-23 |
| Address book                  | 17-Apr-23  | 9-Jun-23  |
| User profile object           | 20-Feb-23  | 3-Mar-23  |
| Presence object               | 27-Feb-23  | 13-Mar-23 |
| Access control                | 20-Mar-23  | 24-Mar-23 |
| Message library               | 20-Mar-23  | 7-Apr-23  |

**Table 5.8 – Feature list**

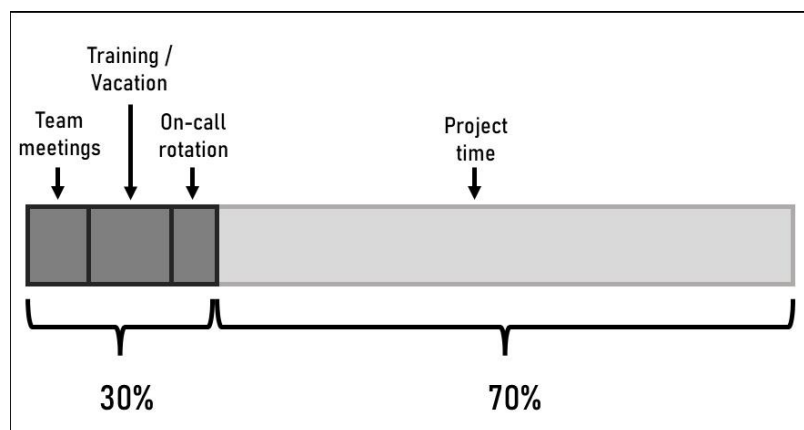
| Task               | Duration | Swarm # | Resourcing | Start Date | End Date  |
|--------------------|----------|---------|------------|------------|-----------|
| Create P2P network | 16       |         |            |            |           |
| Design network     | 6        | 1       | Arun       | 2-Jan-23   | 6-Feb-23  |
| Implement          | 10       | 2       | Arun, Bex  | 13-Feb-23  | 24-Mar-23 |
| Create network Id  | 3        | 1       | Arun       | 27-Mar-23  | 14-Apr-23 |
| Arun on-call       | 1        | 1       | Arun       | 20-Feb-23  | 24-Feb-23 |
| Bex on-call        | 1        | 1       | Bex        | 6-Mar-23   | 10-Mar-23 |

**Table 5.9 – Updated partial plan with resourcing**

## Figures



**Figure 5.1 – Cyclical prioritization**



**Figure 5.2 – Showing overhead, non-project, and project hours to determine available hours**

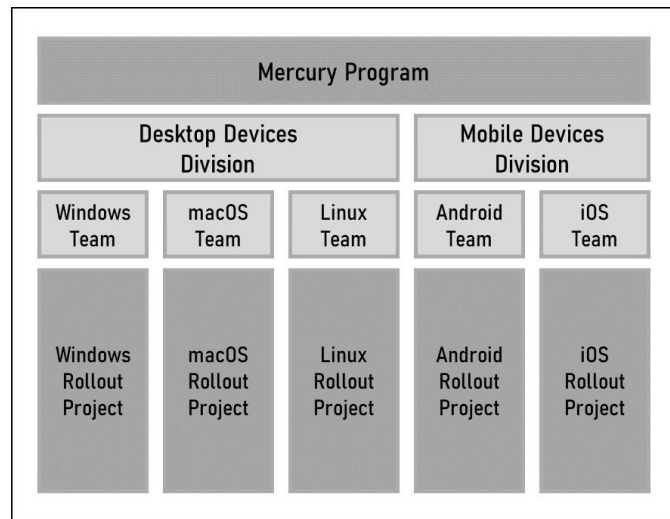


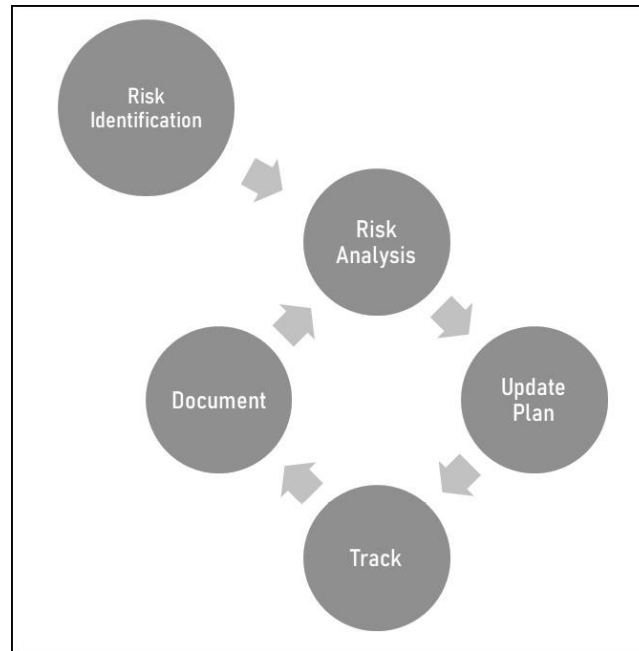
Figure 5.3 – Project and program boundaries

## Further reading

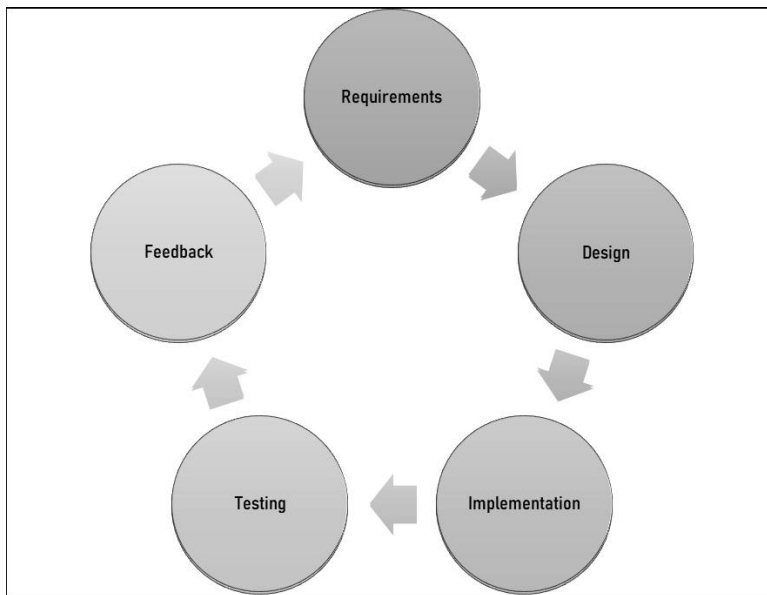
Dr. Goldratt, Eliyahu. *Critical Chain* (North River Press, 1997). This book describes the critical chain methodology that I have discussed under the *Buffers* heading in this chapter. The method is more intuitive to the way we work and give us a tangible way to handle the unknowns by assigning buffers based on complexity and ambiguity.

## Chapter 6

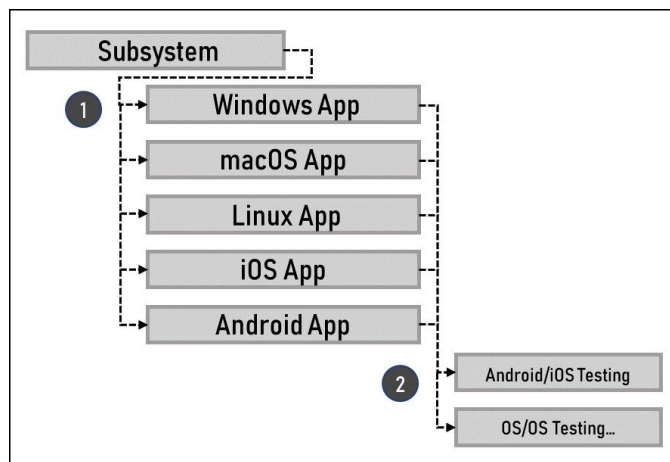
## Figures



**Figure 6.1 – Risk assessment process**



**Figure 6.2 – SDLC**



**Figure 6.3 – Cross-project risks**

## Tables

| Probability | Impact     | Risk Score |
|-------------|------------|------------|
| Low (1)     | Low (1)    | Low (2)    |
| Medium (2)  | Low (1)    | Low (3)    |
| High (3)    | Low (1)    | Medium (4) |
| Low (1)     | Medium (2) | Low (3)    |
| Medium (2)  | Medium (2) | Medium (4) |
| High (3)    | Medium (2) | Medium (5) |
| Low (1)     | High (3)   | Medium (4) |
| Medium (2)  | High (3)   | Medium (5) |
| High (3)    | High (3)   | High (6)   |

**Table 6.1 – Risk scorecard**

| ID | Risk                          | Probability | Impact | Strategy   |
|----|-------------------------------|-------------|--------|--|
| 1  | Cross-platform tooling issues | High        | High   | Acceptance: Shift timelines to account for delays<br>Mitigation: Training and crashing               |
| 2  | Tight testing timeline        | Medium      | Medium | Acceptance: Shift timelines to account for delays<br>Mitigation: Shift timelines to allow for buffer |
| 3  | App store approval delays     | Low         | Low    | Acceptance: Shift timelines to account for delays  |

**Table 6.2 – Risk log for the Mercury program**

| ID  | Risk  | Probability | Impact | Strategy   |
|-----|---|-------------|--------|--|
| 1   | Cross-platform tooling issues                   | High        | High   | Acceptance: Shift timelines to account for delays<br>Mitigation: Training and crashing |
| 1.1 | New <b>Integrated Desktop Environment (IDE)</b> | High        | High   | Mitigation: Training on a new IDE  |

|     |                                    |      |        |   |
|-----|------------------------------------|------|--------|---|
| 1.2 | New coding language for some teams | High | High   | Mitigation: Training on a new language                  |
| 1.3 | Cross-team collaboration           | High | Medium | Mitigation: Daily stand-ups, co-location, or chat rooms |

**Table 6.3 – Cross-platform IDE risk analysis**



# Chapter 7

## Tables

| Type                                   | Goal   | Recurrence  | Owner       | Distribution   |
|--|--|---|-------------|--|
| Stand-up                               | Day-to-day collaboration and unblocking progress                         | Daily   | SDE Lead    | Sprint board, in-person updates, or email progress updates |
| Status Update                          | Milestone-level project status   | Weekly ( <i>Every Tuesday</i> )                                 | Project TPM | Email  |
| <b>Monthly Business Review (MBR)</b>   | Leadership-level program status with key insights relevant to leadership | Monthly ( <i>3rd Wednesday of every month</i> )                 | TPM Lead    | Meeting and Email  |
| <b>Quarterly Business Review (QBR)</b> | Senior leadership-level program status                                   | Quarterly ( <i>3rd Wednesday of the 1st month per quarter</i> ) | PM-T Lead   | Meeting and Email  |

**Table 7.1 – Example communication plan**

| Name               | Alias      | Department               | Project               | Role     | Comm Type |
|--------------------|------------|--------------------------|-----------------------|----------|-----------|
| Josh Teter         | jteter     | Mercury                  | All                   | TPM      | N/A       |
| Arun Ardibeddi     | aardibeddi | Windows Team             | Windows Rollout       | SDM      | MBR       |
| Danielle Wednesday | dwednesday | Windows Team             | Windows Rollout       | SDE Lead | Stand-up  |
| Bob Belkan         | bbelkan    | Windows Team             | Windows Rollout       | SDE      | Stand-up  |
| Vicky Preston      | vpreston   | Desktop Devices Division | Windows, macOS, Linux | VP       | QBR       |

|                  |          |              |                 |     |     |
|------------------|----------|--------------|-----------------|-----|-----|
| Cassette Santoro | csantoro | Windows Team | Windows Rollout | TPM | MBR |
| Artem Danyluk    | adanyluk | Linux Team   | Subsystem       | TPM | MBR |

**Table 7.2 – Example stakeholder list for the Mercury program**

| Step ID | Name                    | TPM     | SDM    | PM-T | Lead SDE | SDE | Bus. |
|---------|-------------------------|---------|--------|------|----------|-----|------|
| 1       | Requirements refinement | A       | C      | C    | C        | C   | R    |
| 2       | Project planning        | R and A | C      | C    | C        | C   | C    |
| 3       | High-level design       | C       | C      | C    | A        | R   | I    |
| 4       | Low-level design        | C       | C      | C    | A        | R   | I    |
| 5       | Sprint planning         | R or A  | R or A | I    | C        | C   | I    |
| 6       | Daily stand-ups         | C       | C      | I    | A        | R   | I    |
| 7       | Status report           | R and A | C      | C    | C        | I   | I    |
| 8       | MBR                     | R       | C      | A    | C        | (I) | I    |

**Table 7.3 – RACI chart**

# Figures

## Windows Rollout Project Status: Feb-21-22

*Next Status: Feb-28-22*

### Executive Summary

**Status:** **Yellow** for Jun-5-22 launch

**Summary:** A delay in the Text Message API definition as caused a day-for-day slip of starting the User Profile Object work. With project buffer and the early stage of the project, this is expected to be recoverable.

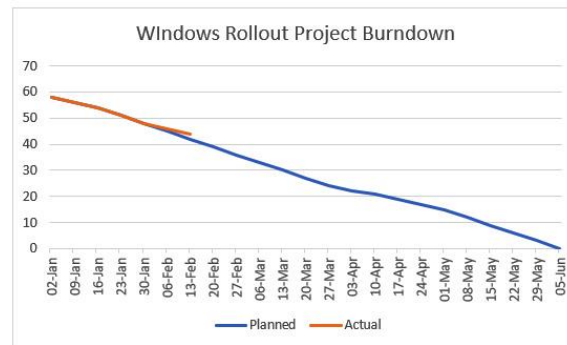
**Path to Green:** Cassette Santoro to work with Danielle's SDM to move her upcoming on-call rotation to later in the year to make up lost time. This will add buffer back into the project lost by the delay. **ETA Feb-23-22.**

### Risk Log

| Id  | Risk                                     | Probability | Impact | Strategy   |
|-----|--|-------------|--------|--|
| 1   | Cross-platform tooling issues            | High        | High   | <b>Acceptance:</b> Shift timelines to account for delays<br><b>Mitigation:</b> training and crashing |
| 1.1 | New Integrated Desktop Environment (IDE) | High        | High   | <b>Mitigation:</b> training on new IDE   |
| 1.2 | New Coding Language for some teams       | High        | High   | <b>Mitigation:</b> training on new language  |
| 1.3 | Cross-team collaboration                 | High        | Medium | <b>Mitigation:</b> daily stand ups, co-location, or chat rooms                                       |

**Figure 7.1 – Status report above the fold**

### Project Burndown



### Project Contacts

TPM: Cassette Santoro (@csantoro)  
SDM: Arun Ardibeddi (@aardibeddi)  
SDE Lead: Danielle Wednesday (@dwednesday)

### Communication Schedule

Status Archive: [Link](#)

Next Status: Feb-28-22

Next MBR: Mar-16-22

Next QBR: Apr-20-22

**Figure 7.2 – Status report below the fold**

# Chapter 8

## Figures

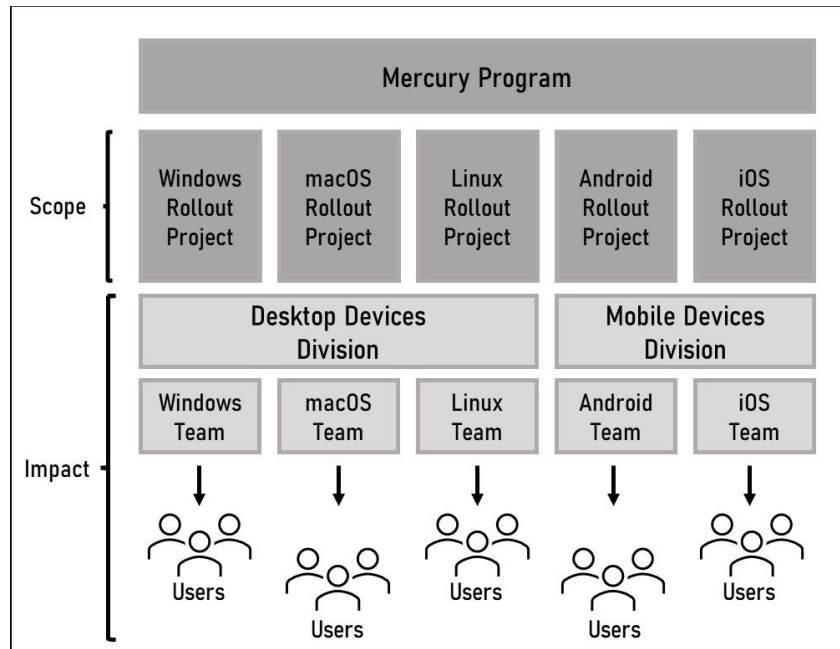


Figure 8.1 – Scope versus impact

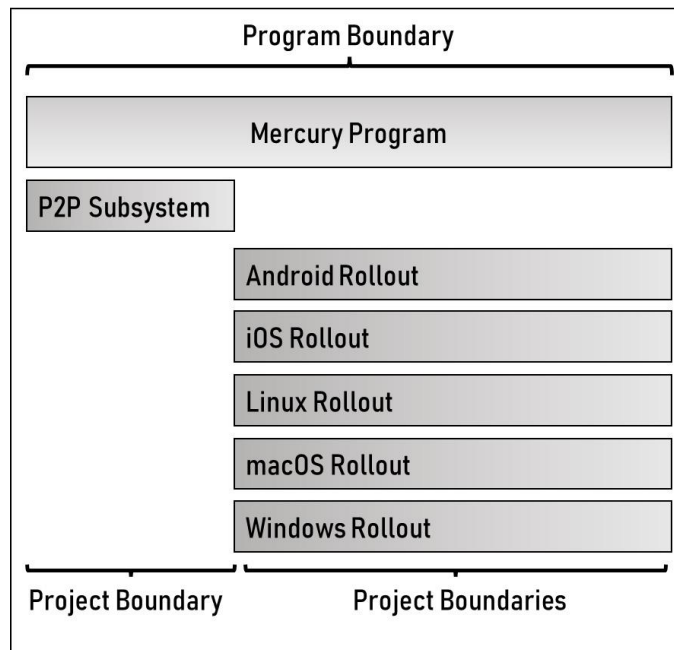


Figure 8.2 – Program versus project boundary

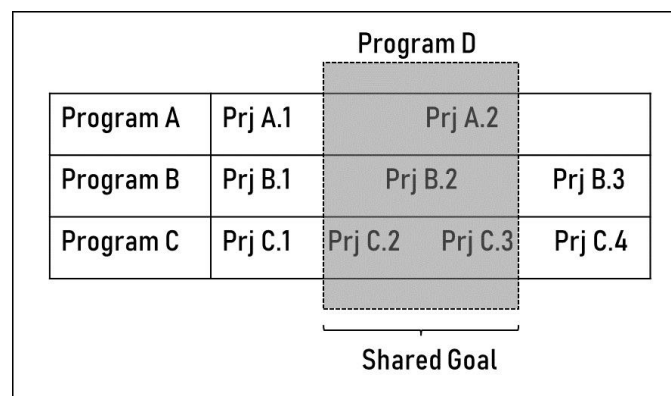
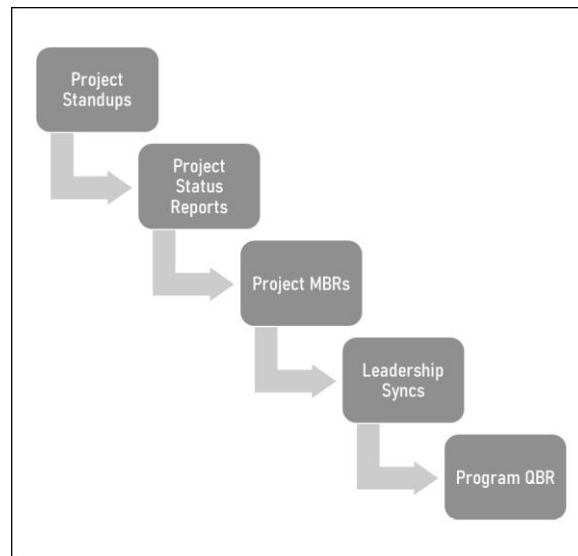


Figure 8.3 – Defining an in situ program



**Figure 8.4 – Aligning communication**

## Tables

| Step ID | Name   | Program TPM | PM-T | Project TPM | Business |
|---------|--|-------------|------|-------------|----------|
| 1       | Program Planning                               | A(/R)       | C    | C           | (R)      |
| 2       | Project Status Report                          | A           | C    | R           | C        |
| 3       | Program <b>Quarterly Business Review (QBR)</b> | R/A         | C    | C           | I        |

**Table 8.1 – Program-level roles and responsibilities**

## Chapter 9

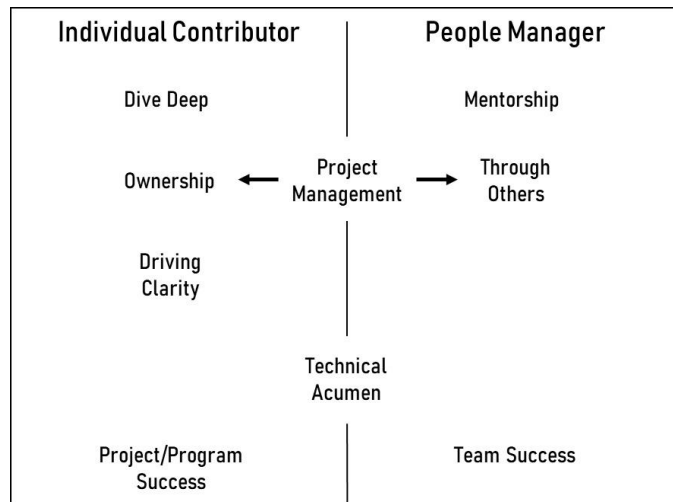


Figure 9.1 – Traits of an IC versus a people manager

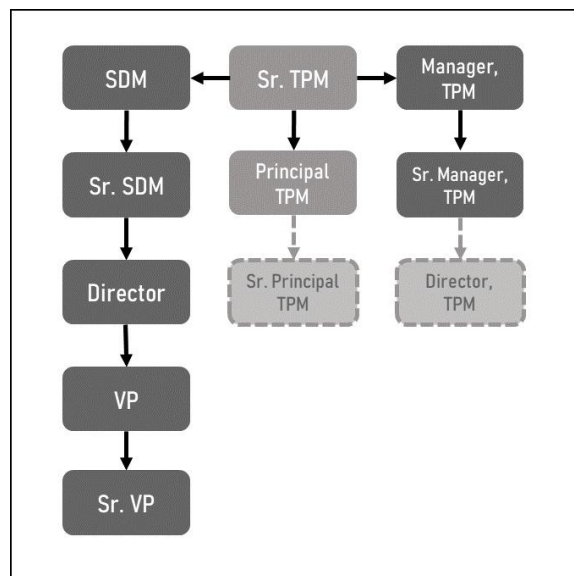
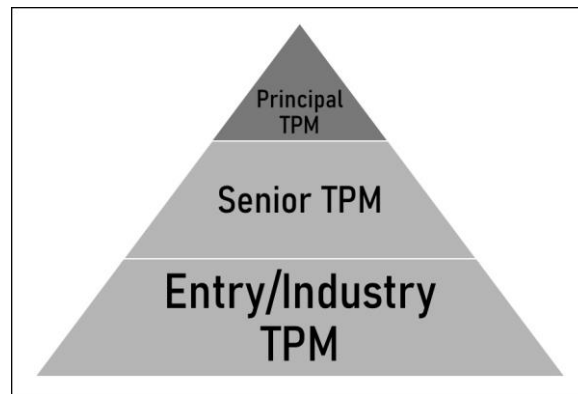


Figure 9.2 – TPM career paths





**Figure 9.3 – IC career path**

# Chapter 10

## Figures

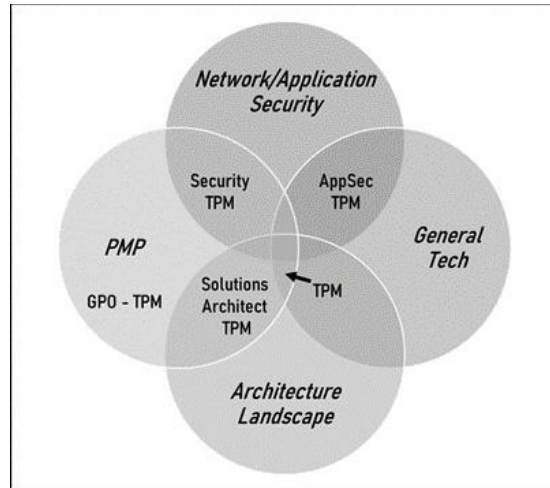
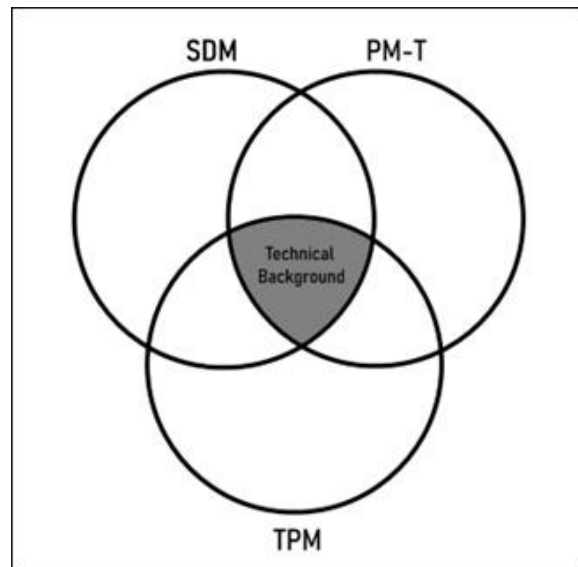


Figure 10.1 – Specialized PM overlaps



**Figure 10.2 – Technical overlap across job families**

## Tables

| Key Management Area    | Step                            | Role                    |
|------------------------|---------------------------------|-------------------------|
| Planning               | Refine requirements             | Accountable             |
|                        | Create functional specification | Accountable/Responsible |
|                        | Sprint planning                 | Responsible/Consult     |
|                        | Review designs                  | Consult                 |
| Stakeholder Management | Draft communication plan        | Accountable/Responsible |
|                        | Daily stand-ups                 | Consult                 |
|                        | Status report/meeting           | Accountable             |
|                        | Monthly business review         | Responsible             |
| Risk Management        | Risk analysis                   | Accountable             |
|                        | Risk monitoring                 | Accountable/Responsible |
|                        | Issue resolution                | Accountable             |

**Table 10.1 – Roles and responsibilities of a TPM**

# Chapter 11

## Figures

```
public void SendTextMessage(string message, MemberInfo miTo)
{
    messenger.SendMessage(new MessageInformation
    {
        Member = myInfo,
        MessageType = MessageInformation.MessageTypes.Text,
        To = miTo.ComputerName,
        Text = message
    });
}
```

Figure 11.1 – Mercury code snippet

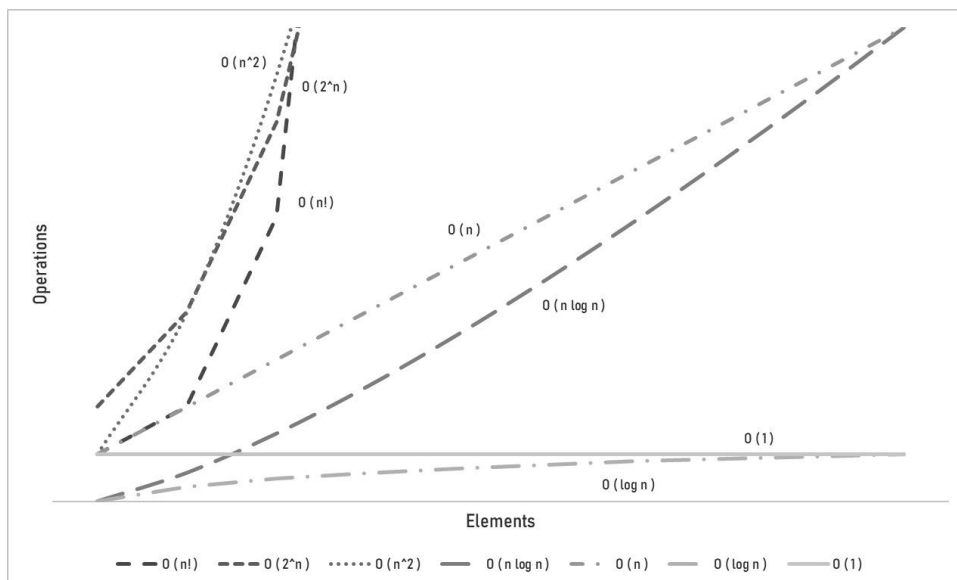
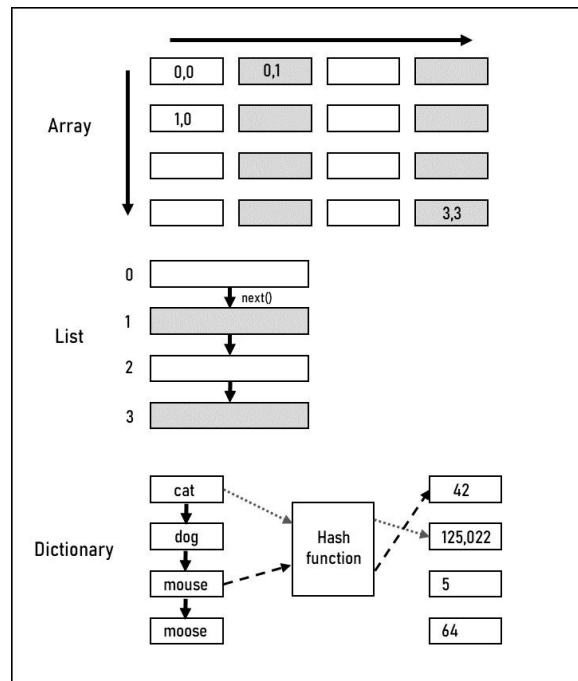


Figure 11.2 – Big O space and time complexity



**Figure 11.3 – Linear data structures**

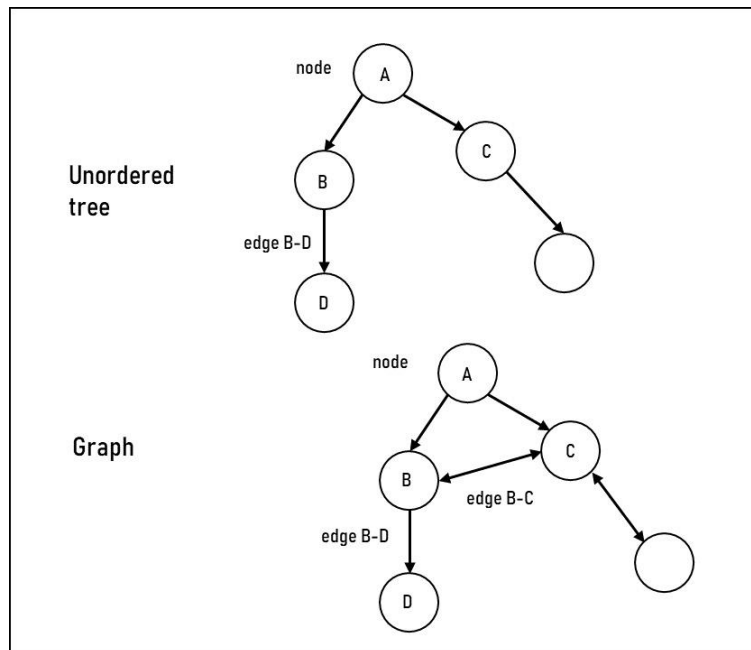
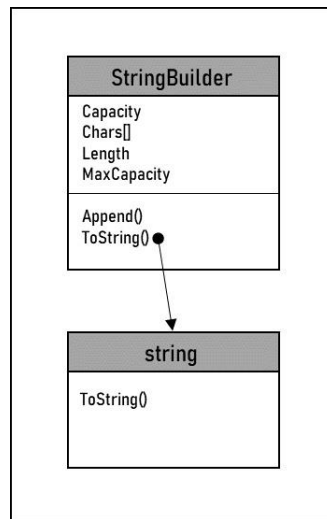
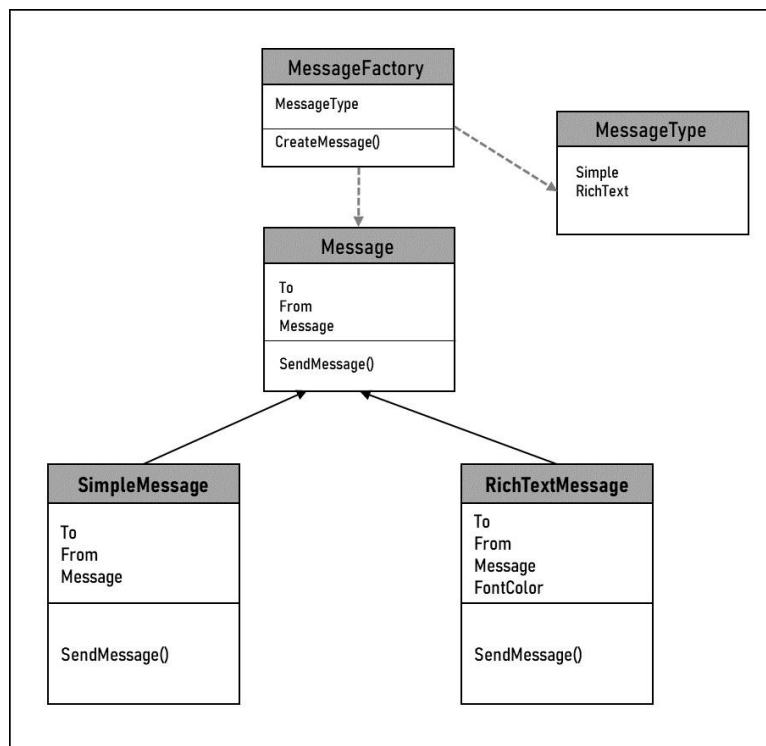


Figure 11.4 – Examples of non-linear tree and graph data structures

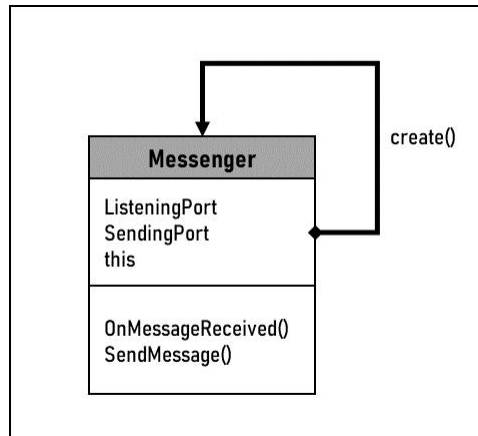


**Figure 11.5 – Builder pattern**

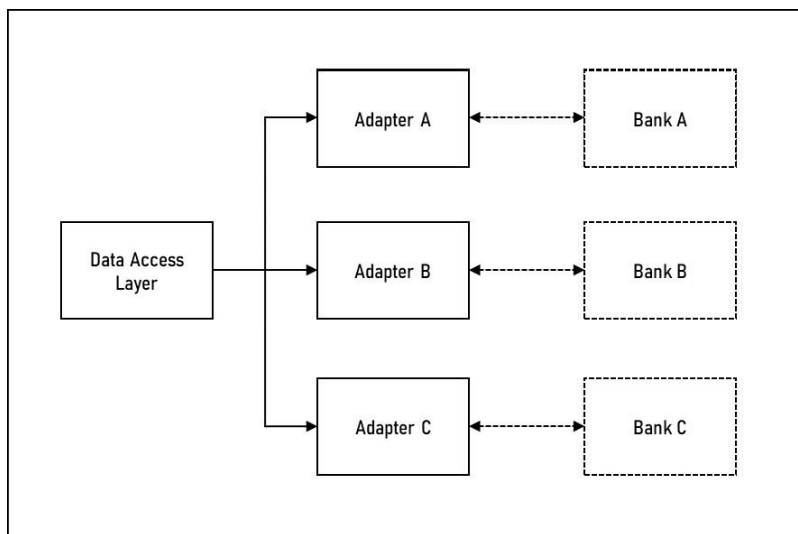




**Figure 11.6 – Simple factory**



**Figure 11.7 – Singleton**



**Figure 11.8 – Using adapters to connect with multiple currency converters**

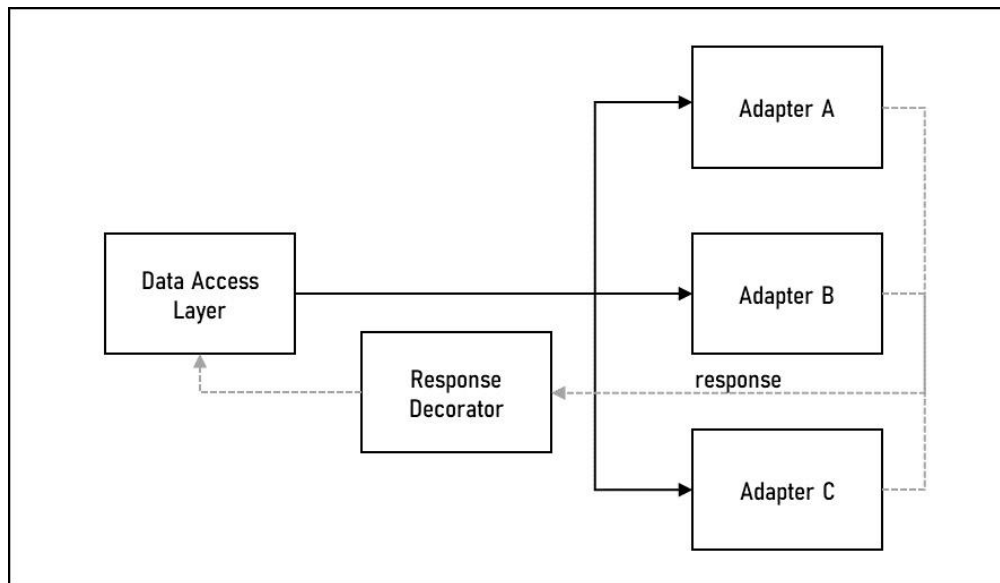


Figure 11.9 – Using a decorator to add data to a response

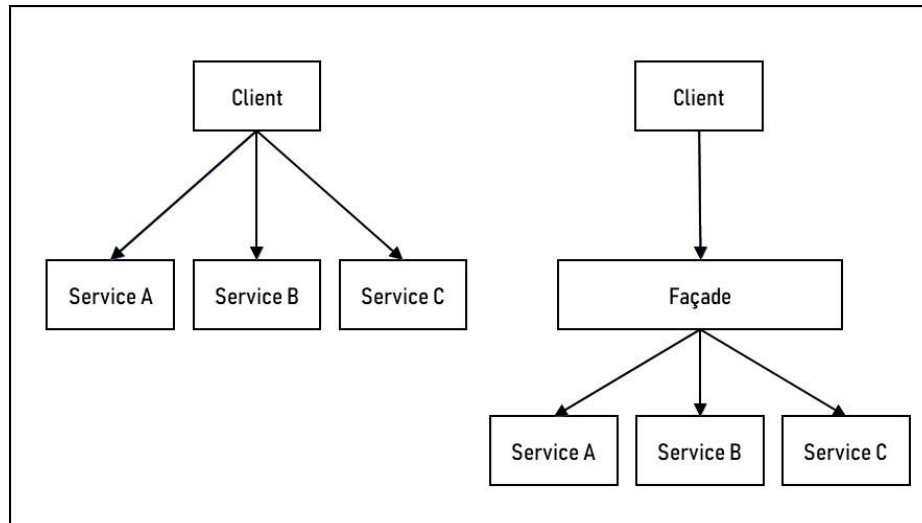


Figure 11.10 – Using a façade to simplify client interfaces

## Further reading

- *Learning Object-Oriented Programming*, by *Gaston C. Hillar*

This is a great introductory book OOP, starting with a basic real-world understanding of objects and methods. If you are unfamiliar with OOP or want an in-depth refresher, this is a good place to start.

<https://www.packtpub.com/product/learning-object-oriented-programming/9781785289637>

- *Mastering Functional Programming*, by *Anatolii Kmetiuk*

This book uses both a traditional functional language, Scala, as well as an OOP language staple, Java, to teach the foundations of functional programming. It then goes beyond the basics to get you comfortable using functional programming concepts and styles in your day-to-day programming.

<https://www.packtpub.com/product/mastering-functional-programming/9781788620796>

- *Hands-On Design Patterns with Java*, by *Dr. Edward Lavieri*

This book gives you a real hands-on approach to learning a large number of design patterns using Java. All design patterns I covered are also covered here at a greater depth, making it a good next step to dive deeper.

<https://www.packtpub.com/product/hands-on-design-patterns-with-java/9781789809770>

- *Everyday Data Structures*, by *William Smith*

This book discusses data structures as well as algorithms, a cornerstone of computer science, in great depth. It uses hands-on programming in various OOP languages to explore each data structure and algorithm. All the data structures I discussed are in this book, and I encourage you to dive deeper using this book.

<https://www.packtpub.com/product/everyday-data-structures/9781787121041>

# Chapter 12

## Figures

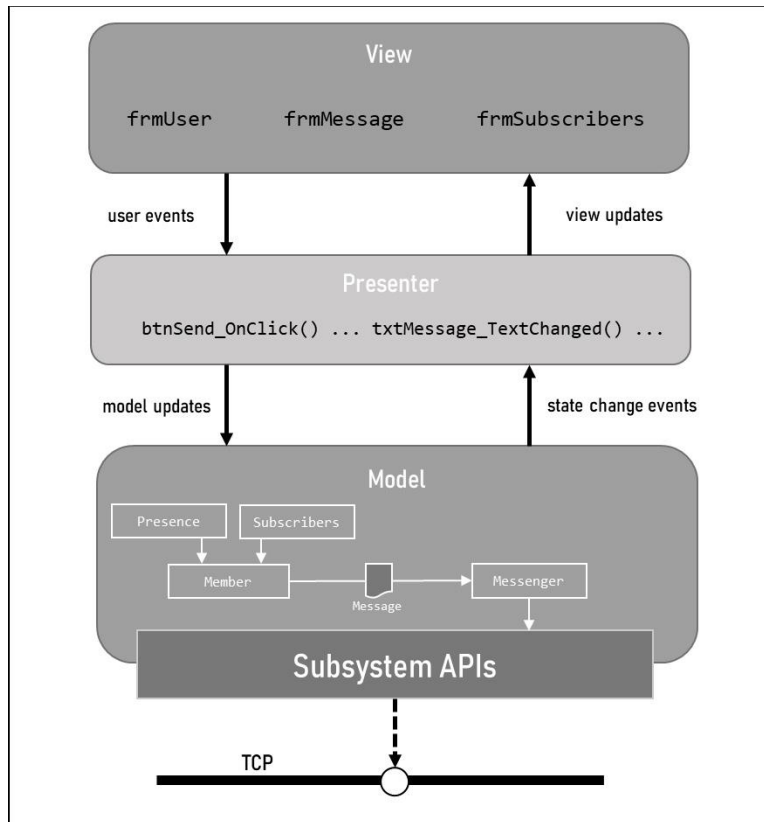


Figure 12.1 – Windows Mercury application system design

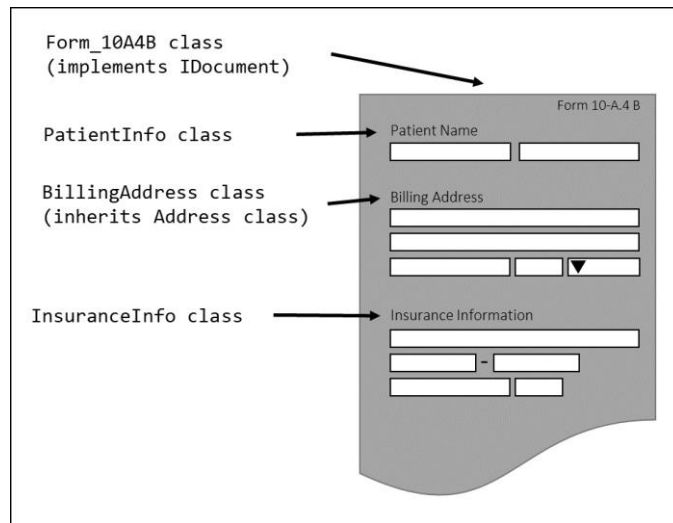


Figure 12.2 – Object-oriented architecture

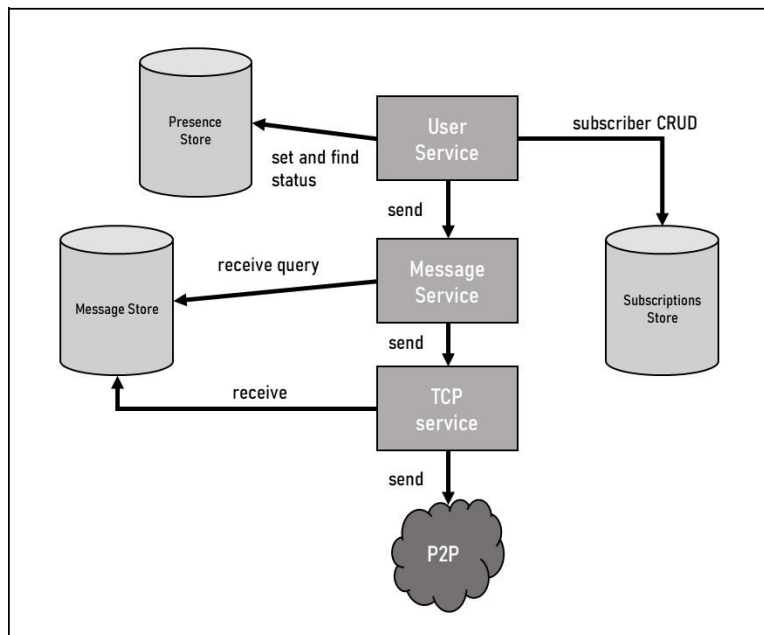
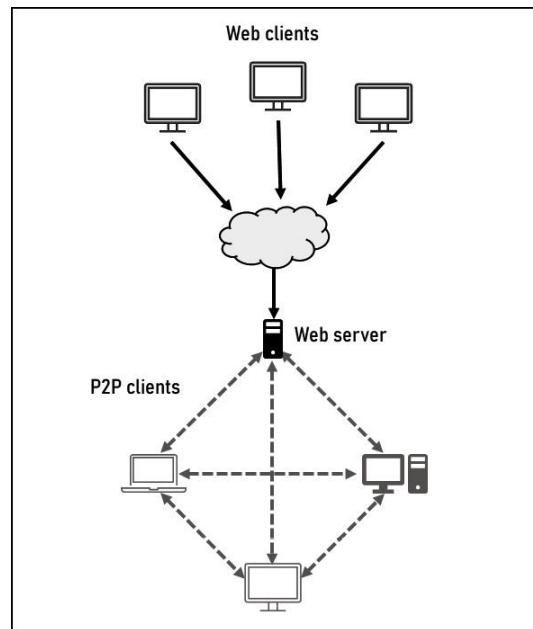
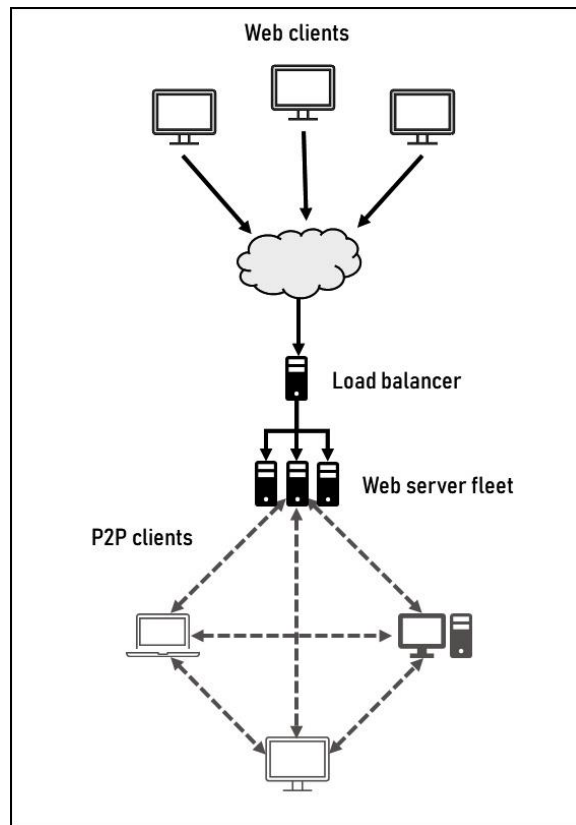


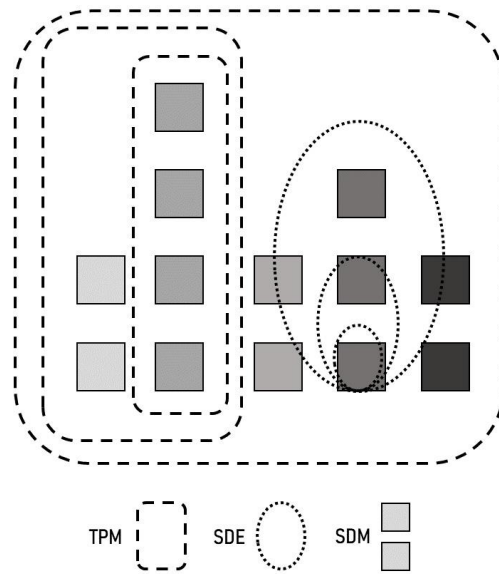
Figure 12.3 – Mercury re-imagined as an SOA



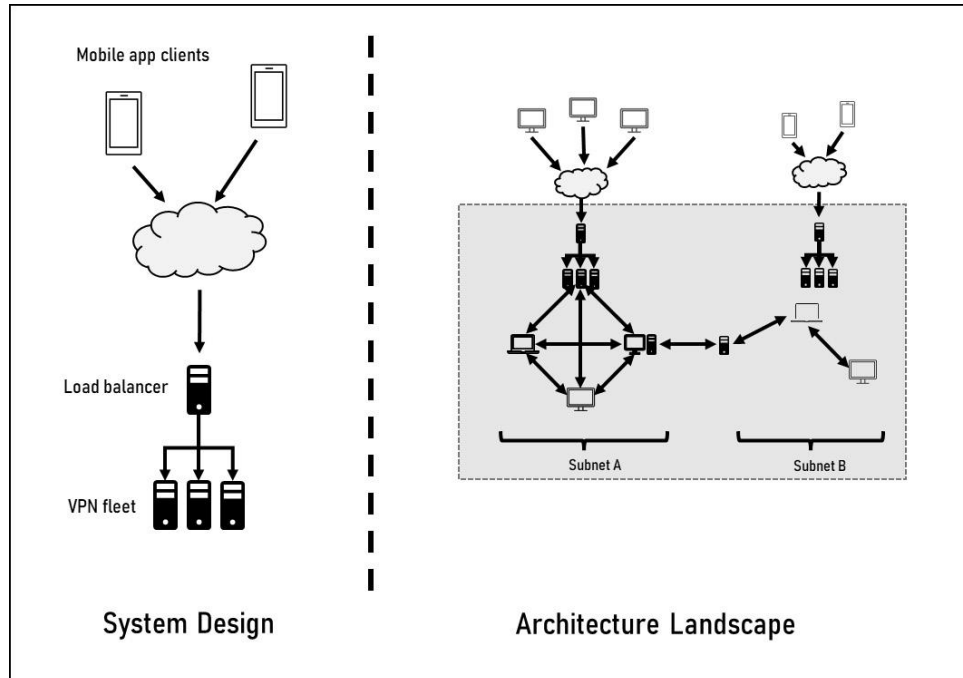
**Figure 12.4 – Web browser Mercury system design**



**Figure 12.5 – System design with availability and scalability mitigations**

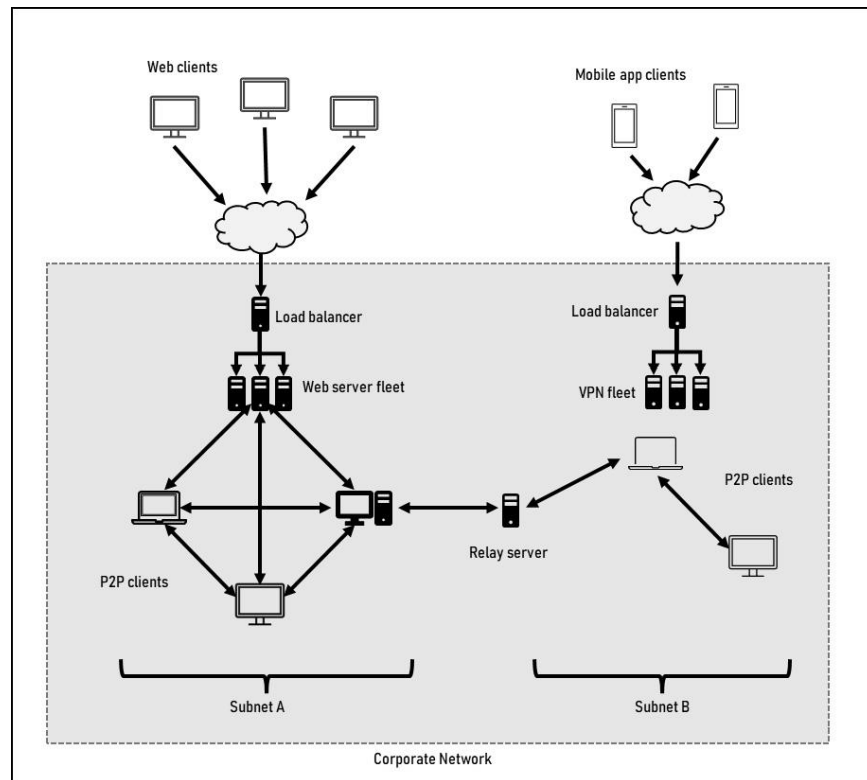


**Figure 12.6 – Areas of concern across job families**





**Figure 12.7 – System design versus architectural landscape**



**Figure 12.8 – Mercury corporate installation architectural landscape**

## Further reading

- *Architectural Patterns*, by Pethuru Raj, et al.

This book covers all of the system design patterns discussed in this chapter, as well as additional patterns. If this is an area of particular interest to you, this is a good place to start.

<https://www.packtpub.com/product/architectural-patterns/9781787287495>

- *Solutions Architect's Handbook – Second Edition*, by Saurabh Shrivastava, et al.

The work of a solutions architect is a popular field, as it focuses on moving from on-premises to the cloud. To do this, a full understanding of the current architecture is needed in order to determine the right solution for the cloud. As such, this offers a great view of understanding an entire architecture.

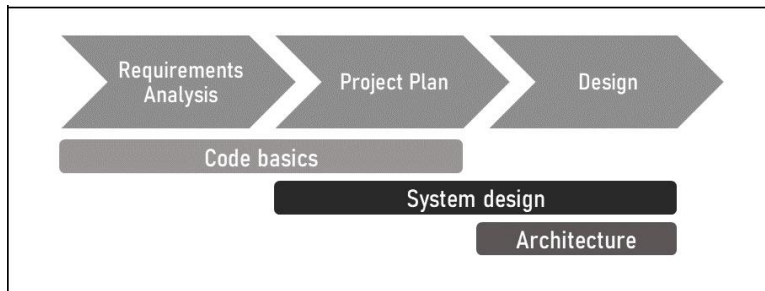
<https://www.packtpub.com/product/solutions-architect-s-handbook/9781801816618>

- *Hands-On Design Patterns with Java*, by *Dr. Edward Lavieri*

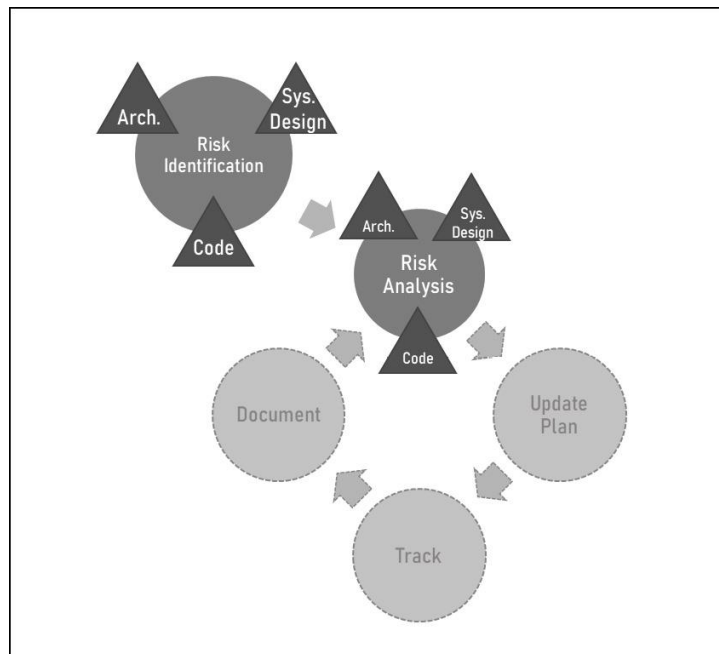
This book gives you a real hands-on approach to learning about a large number of design patterns using Java. All the design patterns I covered are also covered here in greater depth, making it a good next step to dive deeper.

<https://www.packtpub.com/product/hands-on-design-patterns-with-java/9781789809770>

# Chapter 13



**Figure 13.1 – Technical toolset during planning**



**Figure 13.2 – Technical toolset in risk management**