Programming Methodology Framework aka PMF
draft-dulaunoy-programming-methodology-framework-00

Abstract

This document describes the Programming Methodology Framework also known under the PMF methodology. The methodology is based on the manifesto written by Zed A. Shaw [PROGRAMMING-MF-MANIFESTO] which describes a natural approach to software engineering with a strong focus on the act of programming. The PMF methodology uses a soft naming to allow for a non-partisan reference to official engineering or project documents describing one of the most used software engineering methodologies.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 13, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must
Table of Contents

1. Introduction .................................................. 2
   1.1. Management and PMF methodology ......................... 3
   1.2. Conventions and Terminology ............................. 3
2. Security Considerations ........................................ 3
3. Acknowledgements .............................................. 4
4. References ..................................................... 4
   4.1. Normative References .................................... 4
   4.2. Informative References .................................. 4
Authors’ Addresses ................................................ 4

1. Introduction

In 2011, Zed A. Shaw published a blog post which describes:

I think I’m going to create the ultimate software development methodology. It’ll be revolutionary compared to others because it will focus on the one thing that gets software done. Its entire focus will be this one, glorious, completely useful activity for writing software. It’s name even embodies the perfection of this programming methodology.

The PMF methodology was published as a manifesto later [PROGRAMMING-MF-MANIFESTO]. The manifesto clearly describes the focus on programming to avoid the surrounding management overhead and pivot towards the delivery of the software. The [THE-TAO-OF-PROGRAMMING] describes similar methodologies which strongly focus on coding, scripting and programming.

The overall concept of PMF methodology follows the following process:

- Compose a list of features to be implemented and use an iterative programming process.
- Enhance said features using trial and error programming.
- Orchestrate your testing and integration processes using pragmatic programming.

Repeat the above process until the software is delivered.

A simplified overview of the process can be described as follow:
1.1. Management and PMF methodology

A simple management process is a requirement of the PMF methodology. The management process is meant to be complementary instead of interfering with the programming aspect and solely serves to support the PMF methodology.

- Collect requirements of the users/customers.
- Provide programmers with the desired requirements.
- Review whether the software to be delivered matches the requirements.

1.2. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Security Considerations

Secure and defensive programming can only come by practicing programming and this also includes the act of simplifying or removing code to reduce the attack surface.
3. Acknowledgements

The authors wish to thank all the programmers who program.

4. References

4.1. Normative References


4.2. Informative References


Authors’ Addresses

Alexandre Dulaunoy
Computer Incident Response Center Luxembourg
16, bd d’Avranches
Luxembourg L-1160
Luxembourg

Phone: +352 247 88444
Email: alexandre.dulaunoy@circl.lu

Andras Iklody
Computer Incident Response Center Luxembourg
16, bd d’Avranches
Luxembourg L-1160
Luxembourg

Phone: +352 247 88444
Email: andras.iklody@circl.lu