

Blackblot PMTK - Agile Software Development - Guidelines



Table of Contents

1.	INTRODUCTION	2
1.1.	DOCUMENT OBJECTIVE	2
2.	AGILE OVERVIEW	2
2.1.	SECTION OBJECTIVE	2
2.2.	AGILE DEFINITION	2
2.3.	AGILE FUNDAMENTALS	2
2.4.	AGILE REPRESENTATION	2
2.5.	AGILE METHODS' SCOPE	3
3.	SCRUM METHOD	3
3.1.	SECTION OBJECTIVE	3
3.2.	SCRUM DESCRIPTION	3
3.3.	SCRUM ARTIFACTS	3
3.4.	SCRUM ROLES	3
4.	BLACKBLOT PMTK METHODOLOGY™ AND AGILE/SCRUM	4
4.1.	SECTION OBJECTIVE	4
4.2.	PMTK IMPLEMENTATION RULES	4
4.3.	SUCCESSFUL SCRUM IMPLEMENTATION RULES	4

1. Introduction

1.1. Document Objective

This document describes the parameters of an interface between the Blackblot Product Manager's Toolkit® (PMTK) product management methodology and contemporary Agile Software Development (Agile) principles.

<Comment: Agile was formulated by software developers for the purpose of software development and therefore is characterized as Agile Software Development. Phrases such as Agile Project Management, Agile Product Development, Agile Product Management, Agile Product Planning, Agile Product Engineering, Agile Product Delivery, etc. are misleading characterizations and should be avoided.>

2. Agile Overview

2.1. Section Objective

This section aims to provide a better understanding of the concepts which govern Agile Software Development.

2.2. Agile Definition

Agile is a conceptual framework for software development that is based on iterative and incremental development principles, and promotes continuous planning, development, and testing work.

<Comment: Agile principles originated from contract/custom software development. Agile is not, and was not meant to be, an experimental, discovery or research-driven software development technique. Also, with its roots firmly in software, Agile does not, and was not meant to, validate product concepts or generate market requirements. Product concepts are validated by the Product Architect, not the product developers, and market requirements are generated by the Product Planner.>

2.3. Agile Fundamentals

Agile espouses short development cycles in which the software product is developed in a flexible manner. Flexible means that the development team is self-managing and non-hierarchical, development work prioritization is adaptive, and team interaction and collaborative decision making are highly encouraged.

<Comment: Agile is meant to allow the delivery of a software product in stages, providing additional validation or fine-tuning of the software's feature set with each delivery stage.>

2.4. Agile Representation

The term "Agile" also collectively represents a group of Agile-inspired software development methods, such as Scrum, DSDM, Crystal, FDD, and XP, which all reside in the solution space (designing and building a product).

<Comment: Agile software development methods are not product management methodologies, such as the Blackblot PMTK Methodology™, since the product management domain resides in the problem space (identifying and articulating market problems from a product planning perspective).>

2.5. Agile Methods' Scope

Agile Software Development methods describe processes, deliverables and roles which relate primarily to the design, specification, implementation, and verification of a software product.

3. Scrum Method

3.1. Section Objective

This section aims to provide a better understanding of pertinent elements in the Scrum software development method and ties them to the Blackblot PMTK Methodology™.

3.2. Scrum Description

Scrum is a software development method which is based on Agile principles and values.

3.3. Scrum Artifacts

3.3.1. Product Backlog

The Product Backlog is a prioritized list of Product Backlog Items (PBI).

<Comment: Scrum does not define what a Product Backlog Item (PBI) is nor what it should represent. Consequently the PBI is the subject of a wide range of extreme and inconsistent interpretations.>

3.3.2. Sprint Backlog

The Sprint Backlog is the set of Product Backlog Items (PBI) planned for implementation in a Sprint (iterative development period).

3.4. Scrum Roles

3.4.1. Product Owner

The Product Owner role is a managerial role that is responsible for defining, prioritizing, and maintaining Product Backlog Items (PBI) which are contained in the Product Backlog.

3.4.2. Scrum Master

The Scrum Master role is an authoritative role that is responsible for enacting Scrum values and practices.

3.4.3. Development Team

The Development Team is a cross-functional group of individuals who collectively possess the required skills to perform the required software development work.

4. **Blackblot PMTK Methodology™ and Agile/Scrum**

4.1. Section Objective

This section aims to provide guidelines which build an interface between the Blackblot PMTK Methodology™ (in the problem space) and contemporary Agile and Scrum principles (in the solution space).

<Comment: The underlying guiding principle is that a software product can be developed with any Agile method but only after the market problem has been realized, understood, and documented.>

4.2. PMTK Implementation Rules

4.2.1. The Product Planner always creates a Market Requirements Document (MRD) or any similar documentation which represents the market problem.

<Comment: The Product Planner role is part of the Blackblot Product Management Team Model. This strategic role is in the problem space and is owned by a market expert who articulates the market problem. This role is unrelated and unaffected by Agile. The Blackblot Product Planner role relates to the Product Manager role in the Scaled Agile Framework (SAFe) process framework.>

4.2.2. The Product Architect always creates a Product Requirements Document (PRD) or any similar documentation which represents the solution to the market problem.

<Comment: The Product Architect role is part of the Blackblot Product Definition Team Model. This tactical role is in the solution space and is owned by a product expert who creates a high-level design for the product. The Blackblot Product Architect role relates to the Architecture Owner role in the Disciplined Agile Delivery (DAD) process framework.>

4.2.3. When combining roles and assigning them to one person, all the roles that are being combined shall belong to one space only (either problem or solution space).

<Comment: When joining roles from the problem space with roles in the solution space, the roles in the solution space will dominate because the individual often does not have enough details on the market problem, and because the solution requires the documentation of many details and constant interaction with the engineers. Do not join roles from both the problem and solution spaces and assign them to one person. Explicitly, the Scrum Product Owner role (solution space) and the Blackblot PMTK Product Planner role (problem space) should never be joined and performed by the same person.>

4.3. Successful Scrum Implementation Rules

4.3.1. The Product Owner role resides solely in the solution space.

4.3.2. The Product Owner role manages only the Product Backlog.

4.3.3. The Product Backlog only represents a prioritized list of the required software development work.