

需求开发和管理

Requirements Development and
Management (RDM)

RDM的目的

目的：抽取需求，确保利益相关方取得一致理解，并调整需求、计划和工作产品。

价值：确保客户的需求和期望得到满足。



RDM等级和实践概述

第 1 级

RDM 1.1 记录需求。

第 2 级

RDM 2.1 抽取利益相关方的需求、期望、约束条件和接口或连接。

RDM 2.2 将利益相关方的需求、期望、约束条件、接口或连接转化为优先的客户需求。

RDM 2.3 与需求提供者就需求的含义达成一致。

RDM 2.4 获得项目参与者的承诺，即他们可以落实这些需求。

RDM 2.5 开发、记录和维护需求和活动或工作产品之间的双向可追溯性。

RDM 2.6 确保计划和活动或工作产品与需求保持一致。

第 3 级

RDM 3.1 开发并持续更新解决方案及其组件的需求。

RDM 3.2 开发操作概念和场景。

RDM 3.3 分配要落实的需求。

RDM 3.4 识别、开发并持续更新接口或连接需求。

RDM 3.5 确保需求是必要且充分的。

RDM 3.6 在利益相关方的需求和约束条件之间取得平衡。

RDM 3.7 确认需求，以确保生成的解决方案在目标环境中按照预期工作。

Level 1

1.1 记录需求

Level 2

2.1 抽取利益相关方的需求、期望、约束条件和接口或连接。

- *主动识别*所有干系人的需要、期望、约束、接口；从整个产品的生命周期考虑
- *识别不同的需求提供者，不同的需求来源*
 - 包括但不限于：客户、最终用户、高层经理、开发人员、实施人员、市场人员、竞争对手、法律法规等

Level 2

2.2 将利益相关方的需求、期望、约束条件、接口或连接转化为划分优先级的客户需求。

- 对所有引导收集的原始需求进行讨论，消除冲突与不一致，并最终形成客户需求
- 与客户讨论客户需求，最终整理成文档，并**请客户确认客户需求**；有时也会开发Demo、原型请客户试用并确认需求
- **客户需求是从客户的角度描述需求，没有考虑技术因素**
- 开发客户需求时也要把**客户的验收要求**明确下来，并通过评审和客户达成一致理解

Level 2

2.3 与需求提供者就需求的含义达成共识。

- 明确需求提供者，统一需求(变更)来源
- 建立需求评价和验收准则
- 分析需求，确保需求满足建立的评价准则
- 与需求提供者讨论需求，达成一致的理解

Level 2

2.4 获得项目参与者的承诺，即他们可以实现这些需求。

项目成员通过**需求评审**达成对需求的承诺：

- 项目参与人评价需求对自己工作的影响，并**承诺按照需求执行接下来的工作**，如：计划、设计、编码、测试等
- 当需求变更被发生后，评估对自己工作的影响，一旦变更被批准，**承诺按照变更后的需求开展接下来的工作**

承诺的文化

- 承诺必须是**自愿的**
- 承诺必须是透明的，必须是**公开的,正式的**
- 承诺及其变更必须和相关人员进行了充分协商
- 承诺必须是经过充分准备的,避免**过渡承诺**
- 承诺要随时而变，当无法实现时要提前告知
- 承诺的实现**过程要透明**
- 为了承诺要**尽力而为**

Level 2

2.5 开发、记录和维护需求和活动或工作产品之间的双向可追溯性。

#需求跟踪-1

前向跟踪与后向跟踪

- **前向跟踪**：前向跟踪意味着看需求是否在生命周期的后期阶段（如设计和编码阶段）的输出元素中得到体现
- **后向跟踪**：后向跟踪则相反，它意味着后期各个阶段的输出元素满足何种需求。后向跟踪也经常意味着跟踪到原始需求的能力

#需求跟踪-2

横向跟踪与纵向跟踪

- 横向跟踪：需求之间的相互影响关系
- 纵向跟踪：
 - 需求与派生需求
 - 需求与设计\编码\测试用例之间的实现与验证关系
 - 需求的责任分配关系

#需求跟踪-3

建立与维护跟踪矩阵

- 分阶段建立
- 专人负责
- 需求变更时更新

#需求跟踪-4

跟踪矩阵举例

客户需求	产品需求	系统测试用例	概要设计	详细设计	代码
CR1	PR1 PR2	ST1 ST2	HD1	DD1 DD2 DD3	**.c ...

#需求跟踪-5

建立需求跟踪矩阵的好处

- 在整个生命周期对需求进行跟踪，确保所有需求都被实现（**完整性**）；确保后续的工作产品都有需求来源（**必要性**）
- 分析需求变更波及的范围，**确保下游的工作产品和需求保持一致**
- 减少回归测试花费的工作量：通过需求跟踪矩阵可以**确定回归测试范围**

Level 2

2.6 确保计划和活动或工作产品与需求保持一致。

- 评审项目计划、工作产品，发现问题，并解决问题
- 需求基线变更后，评价变更对后续工作产品的影响，并识别后续工作产品需要的变更

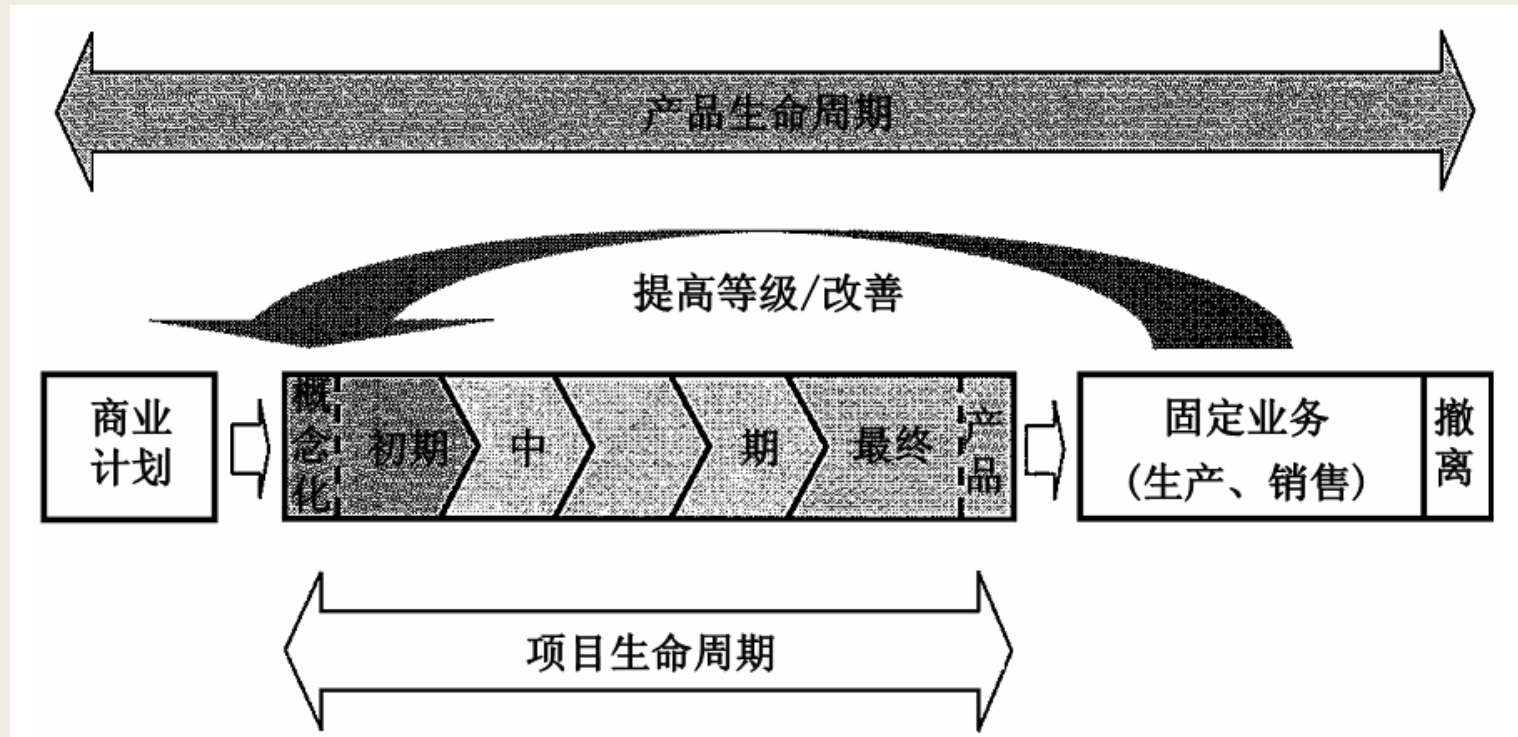
Level 3

3.1 开发并持续更新解决方案及其组件的需求。

- 从**技术的角度**描述需求形成正式的产品需求规格说明书作为设计开发的基础
 - 详细描述产品的操作场景
 - 考虑派生需求：例如，从技术约束得来的需求
 - 考虑产品整个生命周期阶段的需求
 - 考虑质量、性能需求
 - 需求描述时为需求划分优先级，例如：高中低

有时客户是比较专业的技术人员提出用户需求，此时用户需求与产品需求差别不大，可以合并撰写

#产品需求范围-产品生命周期



Level 3

3.2 开发操作概念和场景。

一般在产品需求规格说明书中建立需求的操作概念和场景。

- 场景主要指业务流程，包括：
 - 1) 用户和产品的交互
 - 2) 适用于对产品的开发、使用和维护的活动中
- 操作概念指从用户的视角考虑，在什么背景/环境下执行场景（业务流程）

操作概念和场景可以随着对产品的深入了解逐渐进行细化，应和相关干系人进行评审操作概念和场景以进一步细化和发现新的需求。

Level 3

3.3 分配要实现的需求。

- 根据系统的功能和设计约束划分产品功能模块
- 定义每个产品功能模块实现了哪些需求，因此要建立需求跟踪矩阵，建立详细的**需求与产品功能模块的对应关系**

Level 3

3.4 识别、开发并持续更新接口或连接需求。

- 接口需求可以在产品需求规格书中描述也可以有单独的接口需求文档
- 接口需求包括内外部接口需求
 - 系统构件间的接口需求
 - 系统与外部系统的接口需求
- 在技术框架设计时可能产生接口需求

Level 3

3.5 确保需求是必要且充分的。

- 分析需求找到8大问题
 - 正确
 - 清楚
 - 必要性
 - 无二义性
 - 完备性
 - 可验证性
 - 可实现
 - 优先级

Level 3

3.6 在利益相关方的需求和约束条件之间取得平衡。

- 分析需求，看是否能够满足相关干系人的期望和约束
 - 例如：性能要求、时间要求等
- 平衡需求与项目目标、资源间的冲突
- 识别需求相关的风险
 - 例如：需求稳定性风险

Level 3

3.7 确认需求，以确保生成的解决方案在目标环境中按照预期工作。

- 在项目早期和客户一起对需求进行评审，确保最终的产品是满足最终用户需求的；如果不能在早期就需求进行确认，项目后期开发及最后验收将存在风险，对这些风险要进行充分识别
- 在项目早期制作Demo、原型等和客户确认需求，并充分收集客户的反馈

问题与回答

