

## 第3章

# 软件项目任务分解



### 3.1 任务分解定义

项目章程确定了甲乙双方项目经理,乙方项目经理根据标书、投标书和合同并结合以往的项目经验,对整个项目任务进行分解,形成任务分解结构图,安排相应的人员来完成相应的任务;安排合适的时间来启动相应任务;分配合适的经费到相应任务。任务分解结构图经过甲方项目经理认可,集中双方人员进行开会评估,形成一致意见之后,就可以按照任务分解结构图开始实施。

#### 3.1.1 任务分解结构图

当需要解决的问题过于复杂时,可以将问题进行分解,直到分解后的子问题容易解决,然后分别解决这些子问题。规划项目时,也应该从任务分解开始,将一个项目分解为更多的工作子项目,使项目变得更小、更易管理、更易操作,这样可以提高估算成本、时间和资源的准确性,使工作变得更易操作,责任分工更加明确。完成项目本身是一个复杂的过程,必须采取分解的手段把主要的可交付成果分成更容易管理的单元才能一目了然,最终得出项目的任务分解结构。任务分解是对需求的进一步细化,是最后确定项目所有任务范围的过程。任务分解的结果是任务分解结构图。任务分解结构图是面向可交付成果的对项目元素的分组,组织并定义了整个项目的范围。不包括在任务分解结构图中的工作不是该项目的工作,只有在任务分解结构图中的工作才属于该项目的工作范围。任务分解结构图是一个分级的树形结构,展示了对项目由粗到细的分解过程,如图 3-1 所示。

“自顶向下,逐步求精;删繁化简,分而治之”是自古以来解决复杂问题的思想,对于软件项目来讲,就是将大的项目划分成几个小项目来做,将周期长的项目分成几个明确的阶段。项目越大,对项目组的管理人员、开发人员的要求越高,参与的人员越多,需要协调沟通的渠道越多,周期越长,开发人员也容易疲劳。将大项目拆分成几个小项目,可以降低对项

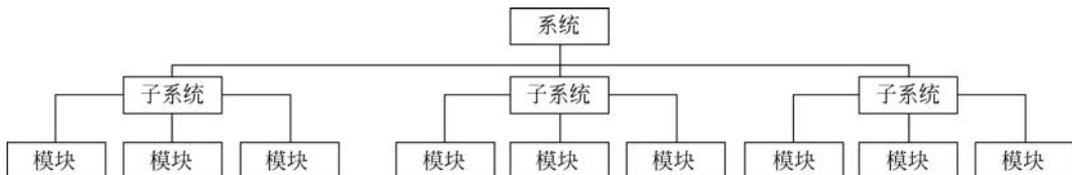


图 3-1 任务分解结构图

目管理人员的要求,减少项目的管理风险,而且能够充分下放项目管理的权力,充分调动相关人员的积极性,目标比较具体明确,易于取得阶段性的成果,使开发人员有成就感。项目范围、项目工期和项目规模可以看成是支持项目成功的三大支柱。

任务分解结构图的建立对于项目来说意义非常重大,它使得原来看起来非常笼统、模糊的项目目标清晰起来,使得项目管理有依据,项目团队的工作目标清楚明了。如果没有一个完善的任务分解结构图或者范围定义不明确,就会不可避免地出现变更,很可能造成返工、延长工期、降低团队士气等一系列不利的后果。

制定好一个任务分解结构图的指导思想是逐层深入。首先确定项目成果框架,然后每层再进行工作分解,这种方式的优点是结合进度划分,更直观,时间感强,评审中容易发现遗漏或多出的部分,也更容易被大多数人理解。

任务分解结构图中的每一个具体子项通常指定唯一的编码,这对有效地控制整个项目系统非常重要,项目组成元素的编码对于所有人来说应当是有共同的认知。因此,任务分解结构图的编码设计与结构设计应该有一一对应关系,即结构的每一层次代表编码的某一个位数,同时项目各组成元素的编码都是唯一的。图 3-2 所示为确定了任务分解结构图编码的任务分解结果。



图 3-2 有编码的任务分解结构图

### 3.1.2 工作包

工作包是任务分解结构图的最低层次的可交付成果,是任务分解结构图的最小元素。工作包应当由唯一主体负责,可以分配给另外一位项目经理通过子项目的方式完成。我们可以对工作包进行进度安排、风险分析以及跟踪控制。工作包还将被进一步分解为项目进度中的活动,每个被分配的活动都应该与一个工作包相关。当工作包被项目的活动创建后,所有工作包的总和等于项目范围,当项目范围完成时,它将满足产品的范围,所有这些构成一个完整的项目。

### 3.1.3 任务分解的形式

一般来说,进行任务分解时,可以采用清单或者图表的形式来表达任务分解的结果。

## 1. 清单形式

采用清单的任务分解方式是将任务分解的结果以清单的表述形式进行层层分解。下面以一个项目为例进行说明。这个项目是“代码生成器”，代码生成器是生成界面与数据库表之间进行交互的 C# 代码，可以快速生成规范的代码，提高软件开发速度和质量，降低开发成本。针对这个项目，采用清单形式进行任务分解如下。

### 1 代码生成器

#### 1.1 界面存向数据库表的代码

1.1.1 文本框值赋给数据库表

1.1.2 文本框值赋给 DataSet

1.1.3 DataSet 值赋给数据库表

#### 1.2 数据库表读向界面的代码

1.2.1 数据库表值读向 DataSet

1.2.2 DataSet 读向文本框

#### 1.3 界面读向界面的代码

1.3.1 一组文本框读向 DataSet

1.3.2 DataSet 读向另一组文本框

#### 1.4 生成 PageLoad 的代码

#### 1.5 生成查询列的代码

#### 1.6 生成删除记录的代码

## 2. 图表形式

采用图表的任务分解过程是进行任务分解时采用图表的形式进行层层分解。例如，对于上面的“代码生成器”项目，采用图表形式的分解结果如图 3-3 所示。

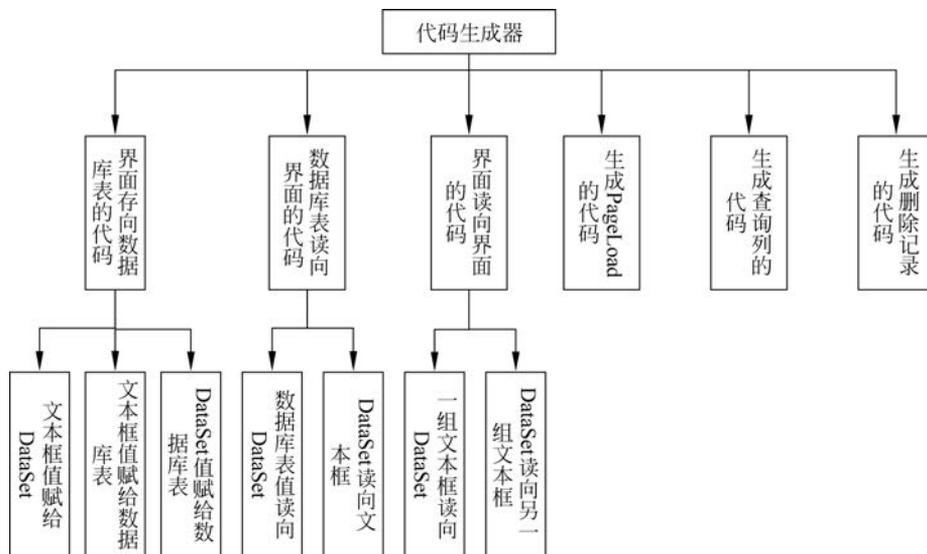


图 3-3 图表分解形式

## 3.2 任务分解过程

进行任务分解应该采取一定的步骤,并且分解过程中保持唯一的分解标准。任务分解的基本过程如图 3-4 所示。

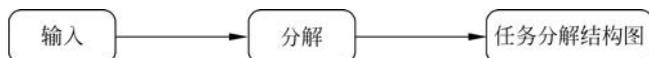


图 3-4 任务分解过程

任务分解应该根据需求分析的结果和项目相关的要求,同时参照以往的项目分解结果进行,其最终结果是任务分解结构图。

### 3.2.1 任务分解的基本步骤

分解意味着分割主要工作子项,使它们变成更小、更易操作的要素,直到工作子项被明确、详细地界定,这有助于未来项目的具体活动(规划、评估、控制和选择)的开展。一般地,进行任务分解的基本步骤如下。

(1) 确认并分解项目的主要组成要素。通常,项目的主要要素是这个项目的工作子项。以项目目标为基础,作为第一级的整体要素。项目的组成要素应该用有形的、可证实的结果来描述,目的是使绩效易检测。当确定要素后,这些要素就应该用项目工作怎样开展、在实际中怎样完成来定义。有形的、可证实的结果既包括服务,也包括产品。

(2) 确定分解标准,按照项目实施管理的方法分解,而且分解的标准要统一。分解要素是根据项目的实际管理而定义的,不同的要素有不同的分解层次。例如,项目生存期的阶段可以当作第一层次来划分,把第一层次中的项目子项在第二阶段继续进行划分。

(3) 确认分解是否详细,是否可以作为费用和时间估计的标准,明确责任。

(4) 确定项目交付成果。交付成果应当是有衡量标准的,以此检查交付成果。

(5) 验证分解正确性。验证分解正确后,建立一套编号系统。

图 3-3 所示的分解过程是按照功能组成进行分解的,分解后的编码系统如图 3-5 所示。

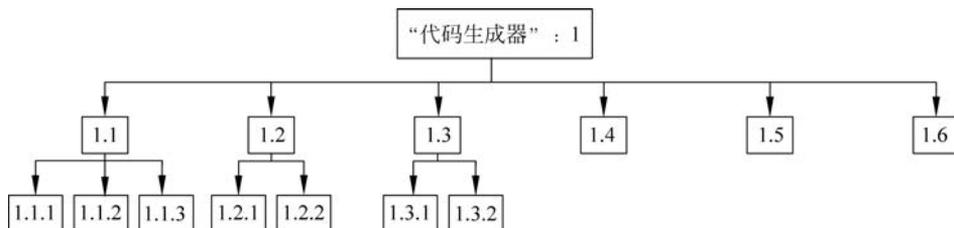


图 3-5 “代码生成器”项目分解后的编码系统

### 3.2.2 任务分解的标准

进行任务分解的标准应该统一,不能有双重标准。选择一种项目分解标准之后,在分解过程中应该统一使用此标准,避免使用不同标准导致混乱。例如,可以以生存期阶段为标

准,以功能(产品)组成为标准,或者采用其他标准等。对于“代码生成器”项目,其任务分解采用两种标准进行,结果如下。

“代码生成器”项目按照功能组成标准进行分解的结果如下。

- (1) 界面存向数据库表的代码。
- (2) 数据库表读向界面的代码。
- (3) 界面读向界面的代码。
- (4) 生成 PageLoad 的代码。
- (5) 生成查询列的代码。
- (6) 生成删除记录的代码。

“代码生成器”项目按照生存期阶段标准进行分解的结果如下。

- (1) 规划。
- (2) 需求。
- (3) 设计。
- (4) 编码。
- (5) 测试。
- (6) 提交。

如果同时使用这两个标准进行任务分解,可能出现的结果如下。

- (1) 界面存向数据库表的代码。
- (2) 数据库表读向界面的代码。
- (3) 界面读向界面的代码。
- (4) 生成 PageLoad 的代码。
- (5) 生成查询列的代码。
- (6) 生成删除记录的代码。
- (7) 规划。
- (8) 需求。
- (9) 设计。
- (10) 编码。
- (11) 测试。
- (12) 提交。

因此,同时使用多种标准进行任务分解会导致混乱,使任务出现重叠。所以,进行任务分解时应该使用统一的标准。

### 3.2.3 任务分解结构图字典

任务分解结构图具体工作要素的阐述通常收集在任务分解结构图字典中。一个典型的任务分解结构图字典既包括对工作包的阐述,也可以包括其他信息,如对进度表的日期、成本预算和员工分配等问题的阐述。通过任务分解结构图字典可以明确任务分解结构图的组件是什么。表 3-1 是一个任务分解结构图字典的例子。

表 3-1 任务分解结构图字典实例

|            |  |
|------------|--|
| 任务分解结构图标识号 | BSM-LBL  |
| 名称         | BSN 事件日志管理系统   |
| 主题目标       | 网管的安全管理系统  |
| 描述         | 1. 存储事件数据：记录对应时间<br>2. 设置时间滤波：对所有事件提供浏览功能<br>3. 浏览事件日志：对所有事件提供浏览功能<br>4. 规划 BSN 事件日志<br>5. 生成历史数据：可生成历史事件报告<br>6. 管理 BSN 事件日志：可以调整 BSN 事件的配置参数 |
| 完成的任务      | 1、2、3 已经完成   |
| 责任者        | × × ×  |
| 完成的标志      | 通过质量保证部的验收报告   |
| 备注         |  |

### 3.3 任务分解方法

任务分解有很多具体方法,如模板参照、类比、自顶向下和自底向上等方法。

#### 1. 模板参照方法

许多应用领域都有标准或半标准的任务分解结构图,它们可以当作模板参考使用。例如,图 3-6 是某软件企业进行项目分解的任务分解结构图模板,本图仅作为参照示例,不代表任何特定项目的具体分解标准,而且也不是唯一的参照模板。

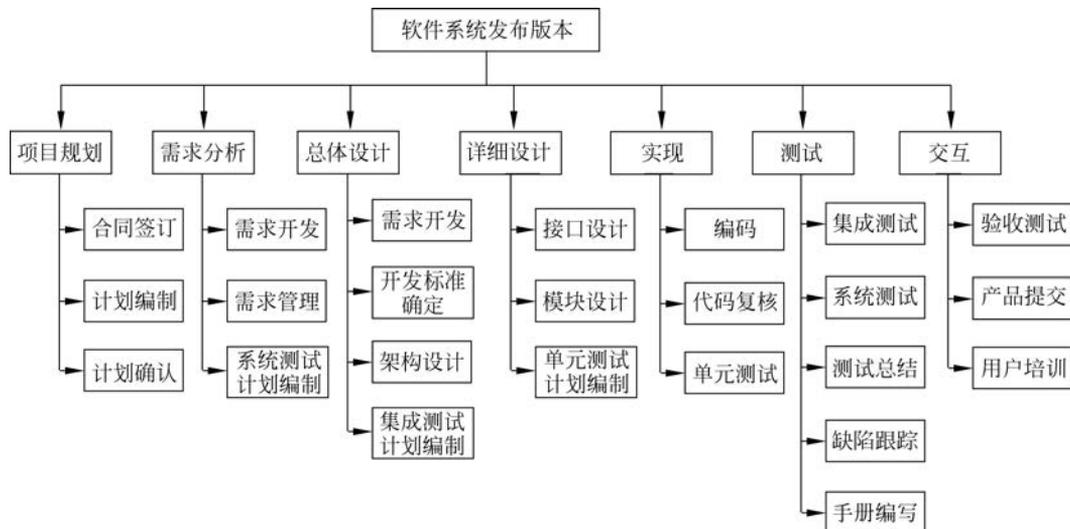


图 3-6 任务分解结构图模板

有些企业有一些任务分解结构图分解的指导说明和模板,项目人员应该通过分析和处理相应的信息来编制项目的任务分解结构图。

## 2. 类比方法

虽然每个项目是唯一的,但是任务分解结构图经常被“重复使用”,有些项目在某种程序上是具有相似性的。例如,从每个阶段看,许多项目有相同或相似的周期和因此而形成的相同或相似的工作子项要求。有些企业会保存一批项目的任务分解结构图和项目文档,以便为其他项目的开发提供参照。很多项目管理工具提供了一些任务分解结构图的实例作为参考,因此可以选择一些类似的项目作为参考来开发任务分解结构图。

## 3. 自顶向下方法

自顶向下方法采用演绎推理方法,沿着从一般到特殊的方向进行,从项目的大局着手,然后逐步分解子项,将项目变为更细、更完善的部分,如图 3-7 所示。

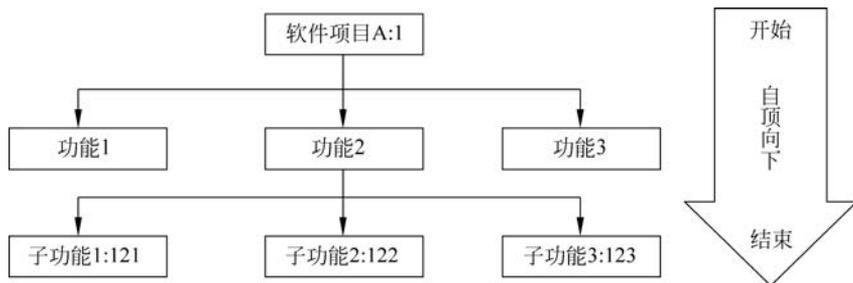


图 3-7 自顶向下方法

自顶向下方法需要有更多的逻辑和结构,它也是创建任务分解结构图的常用方法。使用自顶向下方法来生成任务分解结构图,首先要确定每一个解决方案,然后将该方案划分成能够实际执行的若干步骤。在日常生活中,你可能已经不自觉地使用过自顶向下的工作方法。例如,当你决定要购买一辆小汽车时,需要首先确定买哪种类型的汽车——运动型多用途车、轿车、小型货车,然后考虑能够买得起什么车、什么颜色等,这个思维过程就是一个从主要问题逐渐细化到具体问题的过程。

如果任务分解结构图开发人员对项目比较熟悉或者对项目大局有把握,就可以使用自顶向下方法。应用自顶向下方法开发任务分解结构图时,可以采用下面的操作:首先确定主要交付成果或者阶段,将它们分别写在便条上,然后按照一定的顺序将它们贴在白板上,接下来开始考察第 1 个交付成果或者第 1 阶段,将这些部件分解为更小的交付成果,然后继续分解这些交付成果,直到分解为比较容易管理的工作包,即任务分解结构图的最小单元。

分解项目交付成果需要一定的技巧。可能一开始不能将任务划分得太细,但一定要考虑将合适的时间和资源分配给每个阶段中必须完成的活动。只要把握大方向,然后给团队成员分配他们应该完成的工作即可,而不必详细描述具体的工作机制。

完成了第 1 个主要交付成果或者完成第 1 阶段以后,就可以进行第 2 个交付成果或者第 2 阶段的工作,以此类推,不断重复上述过程。所有的交付成果被分解成工作包后,白板上一定已经贴满了便条,实际上这已经清楚地表达了项目的执行过程。

## 4. 自底向上方法

自顶向下方法从一般到特殊的方向进行,而自底向上方法是从特殊向一般方向进行的。自底向上方法首先定义项目的一些特定任务,然后将这些任务组织起来,形成更高级别的任

务分解结构图层,如图 3-8 所示。

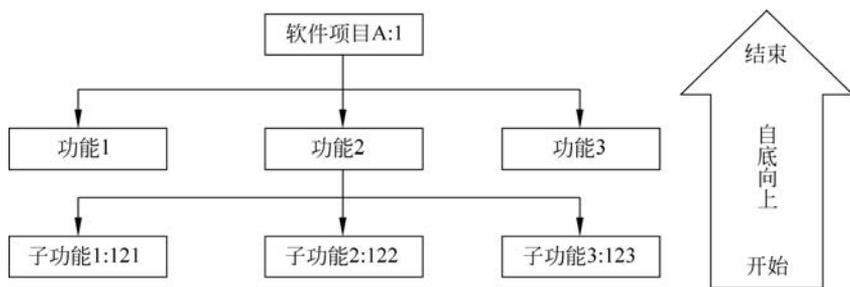


图 3-8 自底向上方法

采用自底向上方法开发任务分解结构图时,可以将可能的任务都写在便条上,然后将它们粘在白板上,这样有利于观察和研究任务之间的关系,接着按照逻辑关系层层组合,形成最后的任务分解结构图。

自底向上方法是一种理想的发挥创造力的解决问题方法。试想现在有一个项目团队正设法找到一个廉价的连接北京和上海网络的解决方案。自底向上方法将会设法寻找一个针对该问题的独特方案,而不对能解决该类问题的每一个方案都进行详细研究。这种方法可能会研究新软件的使用、新服务提供商或者某些实际执行的情况,这些工作尚未执行,还有待讨论。

对于项目人员来说,如果这个项目是一个崭新的项目,就可以考虑采用自底向上方法编制任务分解结构图。

## 3.4 任务分解结果

### 3.4.1 任务分解结果的检验

任务分解后,需要核实分解的正确性。

(1) 明确并识别出项目的各主要组成部分,即明确项目的主要可交付成果。一般来讲,项目的主要组成部分包括项目的可交付成果和项目管理本身。这一步需要解答的问题是:要实现项目的目标需要完成哪些主要工作? 更低层次的子项是否必要和充分? 如果不必要或者不充分,就必须重新修正(增加子项、减少子项或修改子项)这个组成要素。

(2) 确定每个可交付成果的详细程度是否已经达到了足以编制恰当的成本和历时估算。“恰当”的含义可能会随着项目的进程而发生一定的变化,因为对将来产生的一项可交付成果进行分解也许是不大可能的。如果最底层要素存在重复现象,就应该重新分解。

(3) 确定可交付成果的组成元素。组成元素应当用切实的、可验证的结果来描述,以便进行绩效考核。组成元素应该根据项目工作实际是如何组织和完成的来定义。切实、可验证的结果既可包括产品,又可包括服务。这一步要解决的问题是:要完成上述各组成部分,有哪些更具体的工作要做? 每个子项都有明确的、完整的定义吗? 如果不是,这种描述需修正或扩充。

(4) 核实分解的正确性,需要明确如下问题。

① 最底层项对于项目分解来说是否是必要且充分的？如果不是，则必须修改、添加、删除或重新定义组成元素。

② 每项定义是否清晰完整？如果不完整，则需要修改或扩展。

③ 每项是否都能够恰当地编制进度和预算？如果不能，则需要做必要的修改，以便提供合适的管理控制。

(5) 与相关人员对任务分解结构图结果进行评审。

对于实际的项目，特别是对于较大的项目而言，在进行任务分解的时候，要注意以下几点。

(1) 要清楚地认识到，确定项目的分解结构就是将项目的产品或服务、组织和过程这三种不同的结构综合为项目分解结构的过程。任务分解的规模和数量因项目而异。项目经理和项目的工作人员要善于将项目按照产品或服务的结构进行划分、按照项目的阶段划分以及按照项目组织的责任进行划分等有机地结合起来。

(2) 项目最底层的工作要非常具体，任务分解的结果必须有利于责任分配，而且要完整地分配给项目内外的不同个人或者组织，以便明确各个工作块之间的界面，并保证各工作块的负责人能够明确自己的具体任务、努力的目标和所承担的责任。同时，工作划分得具体，也便于项目的管理人员对项目的执行情况进行监督和业绩考核。

(3) 实际上，逐层分解项目或其主要的可交付成果的过程，也就是给项目的组织人员分派各自角色和任务的过程。先分大块任务，然后细分小的任务，最底层是可控的和可管理的任务，避免分解过细，最好不要超过7层。注意收集与项目相关的所有信息，注意参看类似项目的任务分解结果，与相关人员讨论。

(4) 对于最底层的工作块，一般要有全面、详细和明确的文字说明，定义任务完成的标准。对于项目，特别是较大的项目来说，也许会有许多工作块，因此常常需要把所有的工作块的文字说明汇集到一起，编成一个项目任务分解结构字典。

(5) 并非任务分解结构图中所有的分支都必须分解到同一水平，各分支中的组织原则可能会不同。任何分支最底层的子项叫作工作包。按照软件项目的平均规模来说，推荐任务分解时至少拆分到一周的工作量(40小时)。工作包是完成一项具体工作所要求的一个特定的、可确定的、可交付以及独立的工作单元，需为项目控制提供充分而合适的管理信息。任何项目并不是只有唯一正确的任务分解结构图。

成功完成的任务分解结构图是对项目总范围的组织和界定。根据情况，任务分解中可以包括诸如管理、质量等任务的分解。当然也可以在后续的活动分解时，再分解出相应的管理、质量等活动。

### 3.4.2 任务分解的重要性

任务分解结构图提供了项目范围基线，是范围变更的重要输入。通过任务分解，项目经理可以集中注意力到项目的目标上，不必考虑细节问题。同时任务分解为开发项目提供了一个实施框架，其中明确了责任，为评估和分配任务提供具体的工作包，是进行估算和编制项目进度的基础，对整个项目成功的集成和控制起到非常重要的作用。具体来说，任务分解结构图的重要性体现在以下几个方面。

(1) 任务分解结构图明确了完成项目所需的工作。它是项目的充分必要条件,即明确了必须完成的任务,可以只需完成的任务。任务分解结构图可以保证项目经理对项目完成所必备的可交付成果了如指掌。

(2) 任务分解结构图建立了时间观念。通过创建任务分解结构图,项目经理及其团队可以一起为项目的可交付成果而努力。

(3) 任务分解结构图提供了一种控制手段。任务分解结构图能以可视化的方式提供项目中每个任务的状态和进展情况。

(4) 任务分解结构图是范围基线的重要组成部分。任务分解结构图提供了范围基线的核心部分,批准的需求规格以及任务分解结构图字典也是基线的组成部分。

(5) 任务分解结构图可以及时提示是否变更。任务分解结构图基于项目的范围和需求确定了所有的项目可交付成果,可以监督范围变化。

## 习题

### 一、填空题

1. 任务分解是将一个项目分解为更多的工作细目或者\_\_\_\_\_,使项目变得更小、更易管理、更易操作。

2. 一般来说,进行任务分解时,可以采用\_\_\_\_\_或\_\_\_\_\_两种形式来表达任务分解的结果。

3. WBS的全称是\_\_\_\_\_。

4. 任务分解结构图最底层次的可交付成果是\_\_\_\_\_。

### 二、判断题

1. 任务分解结构图提供了项目范围基线。( )

2. 一个工作包可以分配给另一个项目经理去完成。( )

3. 如果开发人员对项目比较熟悉或者对项目大局有把握,开发任务分解结构图时最好采用自底向上方法。( )

4. 对于一个没有做过的项目,开发任务分解结构图时可以采用自底向上方法。( )

5. 在任务分解结果中,最底层的要素必须是实现项目目标的充分必要条件。( )

6. 任务分解是将一个项目分解为更多的工作细目或者子项目,使项目变得更小、更易管理和操作。( )

7. 一个工作包应当由唯一主题负责。( )

8. 任务分解结构图的最高层次的可交付成果是工作包。( )

9. 对任务的分解只能是自上而下的。( )

10. 任务分解结构图的最底层任务是能分配到一个人完成的任务。( )

### 三、选择题

1. 以下未体现任务分解结构图重要性的是( )。

A. 帮助组织工作

B. 防止遗漏工作

C. 为项目估算提供依据

D. 确定团队成员责任

2. 任务分解结构图中的每一个具体细目通常都指定唯一的( )。  
A. 编码                      B. 地点                      C. 功能模块                      D. 提交截止日期
3. 下列不是创建任务分解结构图的方法的是( )。  
A. 自顶向下                      B. 自底向上                      C. 控制方法                      D. 模板参照
4. 任务分解时,( )方法从特殊到一般的方向进行,首先定义一些特殊的任务,然后将这些任务组织起来,形成更高级别的层次。  
A. 模板参照                      B. 自顶向下                      C. 类比                      D. 自底向上
5. 下列关于任务分解结构图的说法,不正确的是( )。  
A. 任务分解结构图是任务分解的结果  
B. 不包括在任务分解结构图中的任务就不是该项目的工作  
C. 可以采用清单或者图表的形式标识任务分解结构图的结果  
D. 如果项目是一个崭新的项目,最好采用自顶向下方法开发任务分解结构图
6. 检验任务分解结果的标准不包括( )。  
A. 最底层的要素是否是实现目标的充分必要条件  
B. 分解的层次不少于3层  
C. 最底层元素是否有重复  
D. 最底层要素是否有清晰完整的定义
7. 任务分解结构图是对项目由粗到细的分解过程,它的结构是( )。  
A. 分层的集合结构                      B. 分级的树形结构  
C. 分层的线性结构                      D. 分级的图状结构
8. 任务分解时,( )方法从一般到特殊的方向进行,从项目的大局着手,然后逐步分解子细目,将项目变为更细、更完善的部分。  
A. 模板参照                      B. 自顶向下                      C. 类比                      D. 自底向上

#### 四、简答题

1. 试写出任务分解的方法和步骤。
2. 当项目过于复杂时,可以对项目进行任务分解,这样做的好处是什么?
3. 检验任务分解结果的标准是什么?