

## Chapter 2

# Introduction to Product Engineering

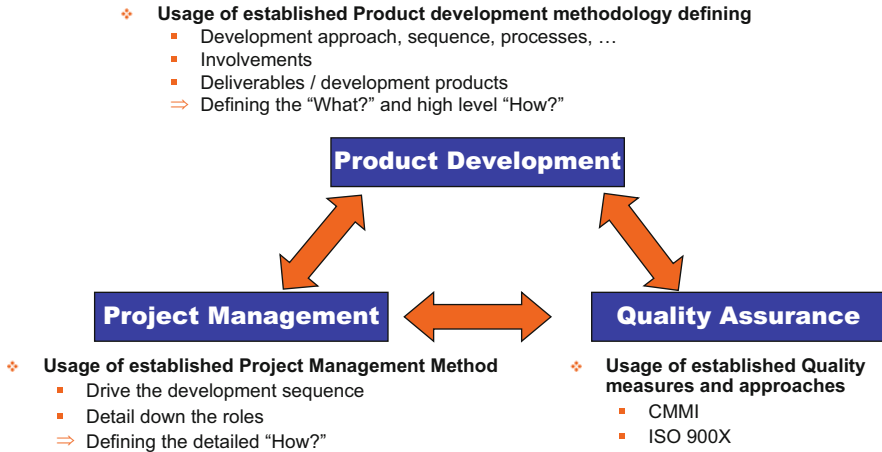
**Abstract** This chapter gives a general introduction into the authors' understanding of Product Engineering in general and defines the essentials of new product development and innovation. Several established baseline methods for product development and project management are presented and evaluated for their applicability. Therefore this chapter provides the base knowledge needed for the MEMS Product Engineering methodology presented in Part II of this book.

### 2.1 What is Product Engineering?

Before discussing the specific role of Product Engineering in the MNT industry and why customer orientation and distribution are predominant factors, it is necessary to clearly define what exactly comprises Product Engineering. In [20] Product Engineering is defined as “the discipline of driving and managing product development efforts following a predefined methodology” and that “it considers the whole realization-cycle of a product from its idea over the various development stages to a marketable product.” Besides this generic definition there are more, slightly different, definitions for various disciplines (manufacturing, software development, etc.) reported in literature [5, 22]. All of these definitions have in common that they consider Product Engineering to be the task that drives product innovation from the product idea through the whole product life cycle (design, development, manufacturing, usage, and recycling or disposal).

To understand the boundary constraints of new products and to get a clearer picture, it needs to be defined what characterizes product innovation. Drivers for innovation and for new products are technology advances, changing customer needs, etc. Another area to be investigated are the key ingredients for successful new product developments and innovation in general. As documented by Cooper et al. in [6], their analysis revealed that there are several key ingredients:

- to embrace a long-term commitment to new product innovation,



**Fig. 2.1** Interactions between product development, project management, and quality assurance

- to develop a vision, objectives, and strategy for new product development efforts driven by and linked to the business's corporate objectives and strategy,
- to install a systematic, high quality new product process.

Several of the success factors are non-systematic and depend on company politics and management decisions. This book focuses mostly on the last point to establish a systematic new product process. In this book such an approach is referred to as a Product Engineering methodology.

The next question to answer is about the necessary ingredients for a high quality product engineering process. The first major ingredient is to have a methodology to coordinate the life cycle of a product. This mostly consists of processes for the development, the improvements, and the phase-out of a product. Because a pure methodology only sets the processes how to perform the tasks, the second necessary ingredient consists of a framework of tools supporting the different activities. Both, the methodology as well as the tool framework, require a laser sharp on quality. This means the quality of business process execution and the metrics/KPIs (Key Performance Indicators) thereof as well as a tool framework assuring the quality execution of the work and the product. All these aspects require high attention to develop high quality products. This triangle of forces and ingredients is depicted in Fig. 2.1.

The major challenge to be addressed for setting up product engineering activities is to select or define a product development methodology. This is a key activity and depends on the market, the business model, and the history of the company.

## 2.2 Product Engineering Essentials

To define a product development methodology and to set up the necessary method framework, it is essential to recap the essentials for product development / innovation. The following sections are only very brief summaries and much more detail can be found in, e.g., [6, 26].

### 2.2.1 *Drivers for Innovation*

Innovation is driven by several different factors which need to be reflected in the product creation process. These factors can be roughly categorized into technology advances, changing customer needs, shortening product life cycles, and increased world competition. Additionally a new product can be new to the company developing it or even new to the market. In detail new products can be split into the categories of new to the world products, new product lines, additions to existing product lines, improvements and revisions to existing products, repositionings (products developed for one area but afterwards used in a different area, e.g., Aspirin, Viagra) and new products due to cost reductions. Because these categories have a major impact on the business case of a potential new product, an idea for a product needs to be carefully analyzed and categorized during the initial assessments.

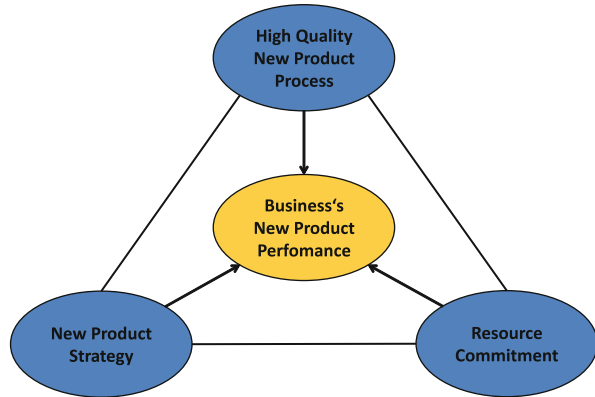
An additional aspect of innovation is that the best inventor is not necessarily the one who solves a problem best, it is often about seeing and understanding a problem. The detection and understanding of the customers' problems is key for an innovative company. Therefore the early involvement of customers in product ideas and a constant contact with the customers to detect problems is the key for future innovation. These points are supported by Davidsen et al. [7].

### 2.2.2 *Success Factors for New Product Developments*

Cooper and his colleagues have analyzed several different product development efforts and approaches. They report, in the widely accepted and supported book "Product Leadership" [6], the following key ingredients for new product success:

- Embracement of a long-term commitment to product development.
- Development of a vision, objectives, and strategy for the new product effort driven by and linked to the business's corporate objectives and strategy.
- Installation of a systematic, high quality new product process in the business and to practice discipline, following the principles of the process.
- Making available necessary resources.
- Fostering the innovation in the organization.

**Fig. 2.2** Cornerstones of new product performance (based on [6])



These findings can be summarized into the cornerstones of new product performance shown in Fig. 2.2. It presents that the resource commitment, a new product strategy, and a new product process are the major contributors to a businesses innovation performance. The cornerstones business strategy and resource commitment are beyond the scope of this book and highly depend on the business itself. The third cornerstone of a systematic, high quality new product process is addressed by the Product Engineering Methodology described in Sect. 5.2.

### 2.2.3 Success Factors for a New Product Process

Before going into the details of the methodology, the key success factors for a new product process or product development methodology should be summarized as they are reported in literature. Cooper's book [6] presents several primary and secondary success factors for a new product development methodology. From their point of view a new product process needs to address the following goals:

- Exemplary Quality of Execution
- Sharper focus, better project prioritization
- Strong market orientation
- Better up-front homework and sharp, early, and stable product definition
- Fast-paced parallel processing by different disciplines
- A true cross-functional team approach
- Products with competitive advantage
- A fast-paced and flexible process

For an *Exemplary Quality of Execution* the parameters focusing on completeness, quality, and importance should be continuously watched. This can be achieved by establishing quality-control checkpoints, by designating the leadership team to become quality controllers, by clear and consistent metrics at intermediate checkpoints, by defining activities, tasks, methods, and best practices into the development phases, by specifying visible deliverables for each phase, along with

action standards and by having effective resource allocation methods for each development phase.

Often the product engineering performance of a business suffers from too many projects with not enough resources. Therefore *sharper focus and better project prioritization* is essential for innovation to thrive. The product engineering process needs to allow to have project funnels rather than project tunnels. So there need to be tough (Go/Hold/Recycle/Kill) decisions.

To achieve a *strong market orientation* of the innovation efforts, marketing tasks need to be designed into the product engineering process. This includes to perform preliminary market assessments. Market research is to determine user needs and wants, competitive analysis, value-in-use analysis, concept testing, customer reaction and feedback during development, user test and field trials, test market or trial sell and market launch based on a solid marketing plan. Additionally the project executives and leadership team needs to ensure the execution of marketing tasks and provides the necessary marketing resource.

For commercially successful innovation *high quality up-front homework and sharp, early and stable product definitions* are essential. The up-front homework needs to define the product and its business case in a stable manner. It lays the baseline for a successful or non-successful innovation. Therefore the new product process needs to ensure the proper execution of the early development phases and stages.

Although the Stage-Gate® process by its nature is a sequential process, the practical implementation of the product engineering process needs to enable and foster parallel execution of all necessary, multifunctional tasks. *A fast-paced parallel processing by different disciplines* is essential. Therefore a sequential approach has to be discouraged and each result achieved by one group needs to be passed on to the other groups as soon as possible, although the work might not be finished yet. Exchange of intermediate results and close cooperation is essential and should be fostered by the method.

Another essential feature of the methodology is *a true cross-functional team approach*. Such an approach needs the ingredients: A cross-functional team with committed team players; defined team captain or leader, accountable for the entire project; a leader with formal authority (co-opting authority from the functional heads); fluid team structure, with new members added or dropped as work demands; a small core group responsible, committed, and accountable from beginning to end.

To develop *Products with a competitive advantage / differentiated products* the development results need to deliver unique benefits and a superior value for the customer. To achieve this in an innovation process the required ingredients are: checkpoint criteria focusing on product superiority; development phases need activities fostering superiority; teams are required to deliver superiority evidence to checkpoints.

The product engineering process needs to enable fast progression and needs to accommodate for different types of projects. Therefore *a fast-paced and flexible process* is required. This implies that smaller or less risky projects need to be able to reduce the overhead by skipping or lightening certain managerial aspects but

still execute the same process. The process must allow for flexibility within certain ranges and ensure fluidity in the process execution. For that fuzzy checkpoints need to be used, where decisions to proceed are taken although not all necessary deliverables are completely available but will be within a certain time frame.

The evaluated success factors inducing the above goals and additional factors for the business environment itself supporting the development methodology are given in [6].

## 2.3 Methods and Tools

Literature describes a plenitude of different product development methods, often targeted specifically at a certain industry like the Agile Method, the V-Model, and the Extreme Programming (XP), which are targeted specifically at the software industry. A recent and good summary of different approaches especially for user-driven product design is given in [14]. An older article summarizing several methods is [15].

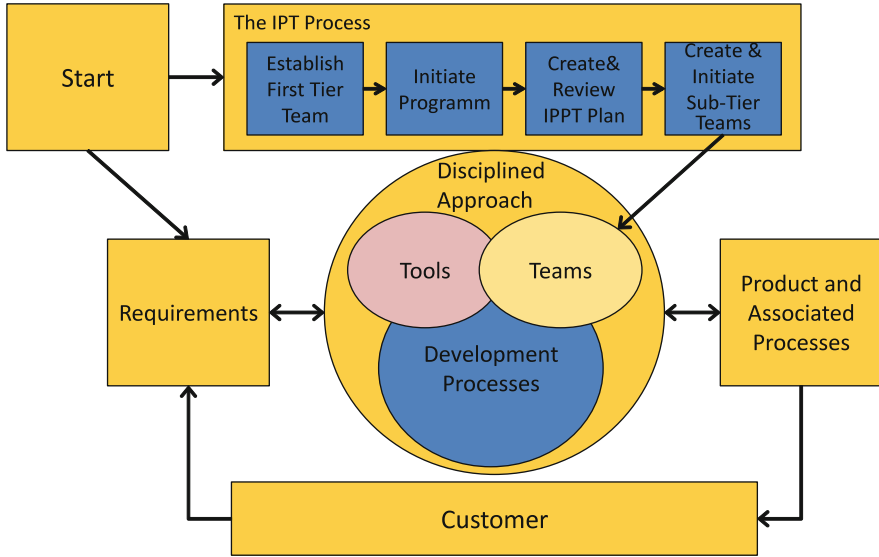
Common ground of all those different approaches is the move to multidisciplinary teams and taking the whole life cycle of the product into account. Additionally, most modern methods can be categorized into either linear staged methods based on activities or into iterative, information and decision-based methods. The linear staged methods have a relatively fixed set of Stages. The iterative methods cycle the whole development loop several times. A good example for the linear staged methods is the Stage-Gate® method introduced in [6]. The Integrated Product and Process Development (IPPD) method, introduced in [8], is an example for an iterative method.

A background article on product development methods that compares Leonardo DaVincis problem detection and solving approach with current product development approaches can be found in [16]. The modern method used for the comparison is similar to the Stage-Gate® method and fosters multidisciplinary team structures. An additional similarity is the multiplicity of available tools at hand to the development team(s). The Leonardo DaVinci approach as well as the modern development approaches like Stage-Gate® consider the problem solving or product development as a combination of art and engineering. Both product development approaches try to foster the creativity through a multiplicity of tools available at hand.

### 2.3.1 Baseline Product Development Methods

#### 2.3.1.1 IPPD/CE

The Integrated Product and Process Development (IPPD) approach is an expansion of the concurrent engineering (CE) approach. Both utilize a systematic approach to the integrated, concurrent development of a product and its associated manufacturing and sustainment processes. Their major aim is to satisfy the customer needs.



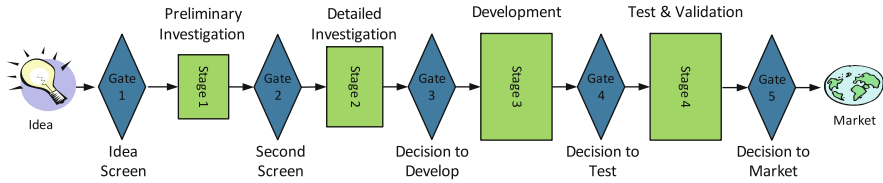
**Fig. 2.3** A generic IPPD iterative process (based on [8])

IPPD is defined as: “A management process that integrates all activities from product concept through production and field support, using a multifunctional team, to simultaneously optimize the product and its manufacturing and sustainment processes to meet cost and performance objectives.” [8]

A graphical overview of the generic IPPD management process is presented in Fig. 2.3. The input to the iterative development process shown in the figure is the requirements generated by the customer(s). They are elicited in a negotiation among many parties, each having serious and important concerns and impacts on the product. For that the customer(s) and their requirements need to be known in detail. The disciplined approach includes five general activities:

- understanding the requirements,
- outlining the approach,
- planning the effort,
- allocating resource and
- executing and tracking the plan.

Decisions made using this approach should be reevaluated as a system matures and circumstances (budgetary, threat, technology) change. The disciplined approach provides a framework for utilizing Tools, Teams, and Development Processes in a structured manner. Tools include documents, information systems, methods, and technologies that can be fit into a generic, shared framework that focuses on planning, executing, and tracking. Teams are made up of everyone who has a stake in the outcome or product of the team, including the customer and suppliers. Collectively, team members should represent the know-how needed and have the



**Fig. 2.4** The generic Stage-Gate<sup>®</sup> Process (based on [6])

ability to control the resources necessary for getting the job done. Development processes are those activities which lead to both, the end product and its associated processes. To ensure efficient use of resources, it is necessary to understand what activities are essential and how they affect the product and each other. Examples include requirements analysis, configuration management, and detailed design drawings. Product and Associated Processes include what is produced and provided to the customer. These steps are iterated if additional or changed customer requirements are encountered.

In summary IPPD and CE are development processes highly focused on the customer requirements and fulfilling them by iterative product extensions and releases. In so far the basic approach is rather similar to the Agile development method for software. An overview of the approach and its application in the acquisition process of the Department of Defence is given in [8].

### 2.3.1.2 Stage-Gate<sup>®</sup>

The Stage-Gate<sup>®</sup> process is an operational roadmap for driving new product development projects from idea to launch—a blueprint for managing the new product process improving effectiveness and efficiency. The central point of this method is a staging and gating process where each Stage is executed by a multidisciplinary team and the Gates build intermediate assessment points. Pioneered and developed by Robert G. Cooper, published in the book [6] and several papers and web sites, it is the world's most widely used new product development process. Examples for adopters of the Stage-Gate<sup>®</sup> methodology are Hewlett-Packard, Procter&Gamble, and SonyEricsson. An overview of the generic process is presented in Fig. 2.4.

A successful innovation begins with an idea and ends with the launch of a new product. The steps between these points can be viewed as a dynamic process. Stage-Gate<sup>®</sup> divides this process into a series of activities (Stages) and decision points (Gates) covering the success factors described in Sect. 2.1.

Each Stage contains a set of defined concurrent activities, incorporating industry best practices. Activities during each Stage are executed in parallel to enhance speed to market. This is further enhanced by the use of cross-functional/multidisciplinary teams all working towards the same goals and the same set of deliverables required for the next Gate. As the method is a blueprint for new product processes, the

amount of Stages can vary between different process implementations in different environments. Typical process implementations use growing sizes of Stages where each later Stage is bigger than a previous one in terms of commitment, effort, money, etc. On the other hand the risks and the uncertainties of the project need to decrease.

Gates are the decision points where the Gatekeepers decide whether to continue funding a project. The Gates consist of three main elements which are the required deliverables as input to the Gate, a set of evaluation criteria for the deliverables and the outputs. The outputs, in the positive assessment case, consist of the funding decision, approved deliverables, an agreed action plan for the next Stage and an agreement on the set of required deliverables for the next Gate. At the Gates selection criteria are applied and tough decisions are made about the projects, but the Gates are essential for the success of a new project strategy. They are the keys to distinguish between the potentially good and bad products and enable to focus scarce resources on the potentially best projects, thus establishing value for money. The outcome of the decision point is either to Go (continue developments), Kill (cancel developments), Hold (temporarily stop developments), or Recycle (redo previous Stage) the project. Gates also act as “quality control” checkpoints evaluating among other things: Have the deliverables been generated in a quality fashion? Is the project (still) attractive from an economic and business standpoint? Is the information sufficient to make a decision? Is the action plan and request for resources sound?

The Gates are executed by the Gatekeepers of the process. The Gatekeepers are a team of senior leaders who make Go/Kill/Hold/Recycle decisions at the Gates. They are from different functional areas and can commit the required resources. They use a pre-set list of criteria and rules for the Gates. By those rules and criteria they are bound so that they cannot play favorites.

The benefits of using Stage-Gate<sup>®</sup> are that it introduces discipline into an ordinarily chaotic process. It focuses attention upon quality of execution and speeds up the innovation process. The method ensures a complete process (no critical steps are omitted) and facilitates a focus on the performance of a new product. Software environments supporting and guiding developments based on this method are available as well, one example is PD-Trak [9].

### 2.3.1.3 Selection

As regularly reported in literature, there is nothing like “The method” fitting all purposes and industries. Therefore the application area of the product engineering approach / product development method needs to be carefully taken into account. The method developed in the scope of the CORONA project and this book is to be used in the MNT area. Advantages of the one or the other method may be less relevant in the considered case. As an example, certain aspects of the criticism about the staged development given in [2] about the missing integration of all life cycle aspects are less relevant in the MNT space due to the nature of the value chain and that MNT products are only a small portion of a final, tangible customer product.

Aspects claimed as advantages of the IPPD methodology like the inclusion of manufacturing aspects, distribution and inclusion of suppliers into the engineering process can easily be covered with the extensions through the T<sup>2</sup>M method as well as through the multidisciplinary teams approach in the MNT field.

Another specialty to take into account is the sequential nature of the MNT process developments (introduced in more detail in Sect. 4.3.1). During the process development efforts the steps material development, material optimization, sequencing, modularization, etc. need to be executed in this standard sequence and can hardly be changed or parallelized. Therefore the process development has a kind of predefined nature and needs to be incorporated. These efforts can be integrated more easily into the Stage-Gate<sup>®</sup> method.

Furthermore the CORONA project executed a market analysis about the currently used methodologies in the MNT field. An overwhelming majority of currently in use methods, where a formal method was in place, were based on a phase and gate-based process anyway. The independent efforts of the MEMS Industry Group (MIG) to develop the MIG Technology Development Process (TDP) template [17] resulted in exactly the same results. As highlighted in the whitepaper describing the template, their survey on common industry practices concluded that the most widely used development methodology is based on a staged and gated-based process. Actually the process proposed by the template is rather similar to the process suggested by this book.

MNT product innovation has a very high research character. Due to that the “contract” between the management and the engineering team, being the essential base in IPPD, is very difficult to achieve because of all the unknowns. Additionally, considering the engineering with different teams along the value chain, the methodology needs to cater for structure, possible management and partner involvement, team building and multi-site / multi-partner development. Therefore a highly integrated development method like IPPD requiring a co-located development team and offering only less management involvement possibilities has several disadvantages. Especially the required multi-site / multi-partner engineering team of the MNT business, the technology development needs to be done at least at one place with equipment access, requires a more structured and controlled approach.

Therefore it can be concluded that the advantages of IPPD in comparison with Stage-Gate<sup>®</sup>, as it has been discussed in [2], are less applicable. Additionally the arguments for IPPD have some drawbacks as well, especially to the MNT market. As the author suggests in the showcase, the transition from a Stage-Gate<sup>®</sup> method to an IPPD-based method is possible as well and to structure the sometimes chaotic process, a staged method is preferable. Additionally Stage-Gate<sup>®</sup> has a very good reputation, is in widespread use, and is supported by various additional authors [7, 19]. Because of that the decision was taken to base the herein proposed methodology on the Stage-Gate<sup>®</sup> approach, having the preconditions for IPPD in mind and working around some issues via the T<sup>2</sup>M extension.

### 2.3.2 *Baseline Project Management Methods*

The necessity for the addition of a Project Management (PM) method extending the Product Development method is described in the methods overview (Sect. 2.1). To not digress too much from the subject, no general introduction into project management itself is given herein. For such an introduction and the historical progression of project management, e.g., [23] can be reviewed. Especially [23] gives a good definition for what a project is, which success factors do exist, and what distinguishes a project from a product, program, and operations.

A literature study revealed that there exist a variety of different PM Methods. An exhaustive list of such methods and project maturity models can be found in [12]. An analysis of how commonly the different methods are used and their general applicability to product developments of all kinds, not limiting it to software projects, reduces the possible candidates. Additionally it must be easily possible to combine the PM method with a product development method. Applying all these criteria the remaining list consists of PMBOK [11], PRINCE2® [18], and ISO10006 [13].

To give an overview, the following sections very briefly summarize these three methods and the differences between them.

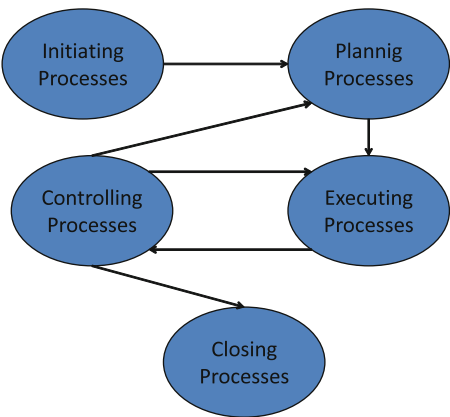
#### 2.3.2.1 **PMBOK**

The Project Management Body Of Knowledge (PMBOK) is standardized in the IEEE Std. 1490–1998 and can be considered *the* project management standard in North America. It is officially documented in [11].

Under PMBOK a project is initiated, guided, etc. by its Stakeholders. These are the people involved or affected by the project activities, e.g., project sponsor, project manager, user, suppliers, and customers. They define the needs, provide resources and so on. The PMBOK project management principles use five main PM processes which are initiating, planning, executing, controlling, and closing of the project. The processes partly overlap each other as well as they interact with each other. An overview of the interaction of the processes is presented in Fig. 2.5. PMBOK describes the processes in terms of Inputs (e.g., documents, plans, designs), tools & techniques (mechanisms applied to the inputs), and outputs (e.g., documentations, products).

Additionally the standard defines nine knowledge areas which are closely tied into the processes and the tools, techniques, and best practices. The nine knowledge areas are heading the different rows in Table 2.1 and the columns indicate the different process groups. The cross sections of the table list the corresponding processes of the group in that knowledge area and refer to the corresponding PMBOK chapter. A graphical overview of the interaction of the different knowledge areas is presented in Fig. 2.6.

**Fig. 2.5** Management process of the PMBOK (based on [11])



**Table 2.1** PMBOK overview

| Knowledge Areas & Chapter numbers    | Process Groups & Processes Within Groups<br>Process section no. within PMBOK Chapter |                                                                                                                                                |                                                  |                                                                   |                    |
|--------------------------------------|--------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|-------------------------------------------------------------------|--------------------|
|                                      | Initiation                                                                           | Planning                                                                                                                                       | Executing                                        | Monitoring & Controlling                                          | Closing            |
| Project Management Integration 4     | 1 Develop Project Charter<br>2 Develop Preliminary Project Scope Statement           | 3 Develop Project Management Plan                                                                                                              | 4 Direct and Manage Project Execution            | 5 Monitor and Control Project Work<br>6 Integrated Change Control | 7 Close Project    |
| Project Scope Management 5           |                                                                                      | 1 Scope Planning<br>2 Scope Definition<br>3 Create Work Break-down Structure (WBS)                                                             |                                                  | 4 Scope Verification<br>5 Scope Control                           |                    |
| Project Time Management 6            |                                                                                      | 1 Activity Definition<br>2 Activity Sequencing<br>3 Activity Resource Estimating<br>4 Activity Duration Estimating<br>5 Schedule Development   |                                                  | 6 Schedule Control                                                |                    |
| Project Cost Management 7            |                                                                                      | 1 Cost Estimating<br>2 Cost Budgeting                                                                                                          |                                                  | 3 Cost Control                                                    |                    |
| Project Quality Management 8         |                                                                                      | 1 Quality Planning                                                                                                                             | 2 Perform Quality Assurance                      | 3 Perform Quality Control                                         |                    |
| Project Human Resource Management 9  |                                                                                      | 1 Human Resource Planning                                                                                                                      | 2 Acquire Project Team<br>3 Develop Project Team | 4 Manage Project Team                                             |                    |
| Project Communications Management 10 |                                                                                      | 1 Communications Planning                                                                                                                      | 2 Information Distribution                       | 3 Performance Reporting<br>4 Manage Stakeholders                  |                    |
| Project Risk Management 11           |                                                                                      | 1 Risk Management Planning<br>2 Risk Identification<br>3 Qualitative Risk Analysis<br>4 Quantitative Risk Analysis<br>5 Risk Response Planning |                                                  | 6 Risk Monitoring and Control                                     |                    |
| Project Procurement Management 12    |                                                                                      | 1 Plan Purchases and Acquisitions<br>2 Plan Contracting                                                                                        | 3 Request Seller Responses<br>4 Select Sellers   | 5 Contract Administration                                         | 6 Contract Closure |

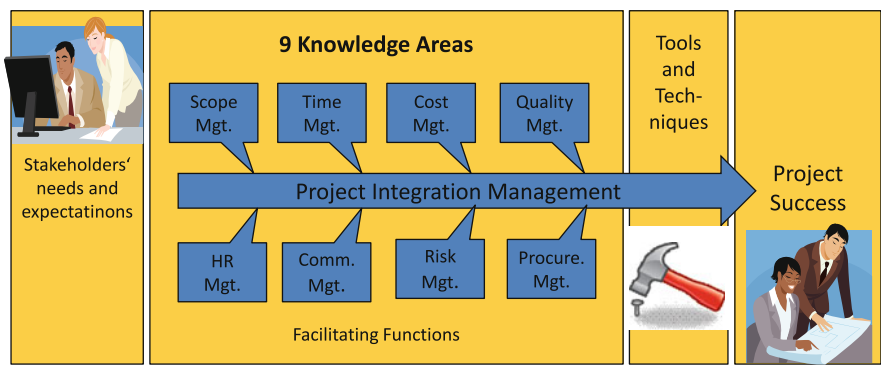


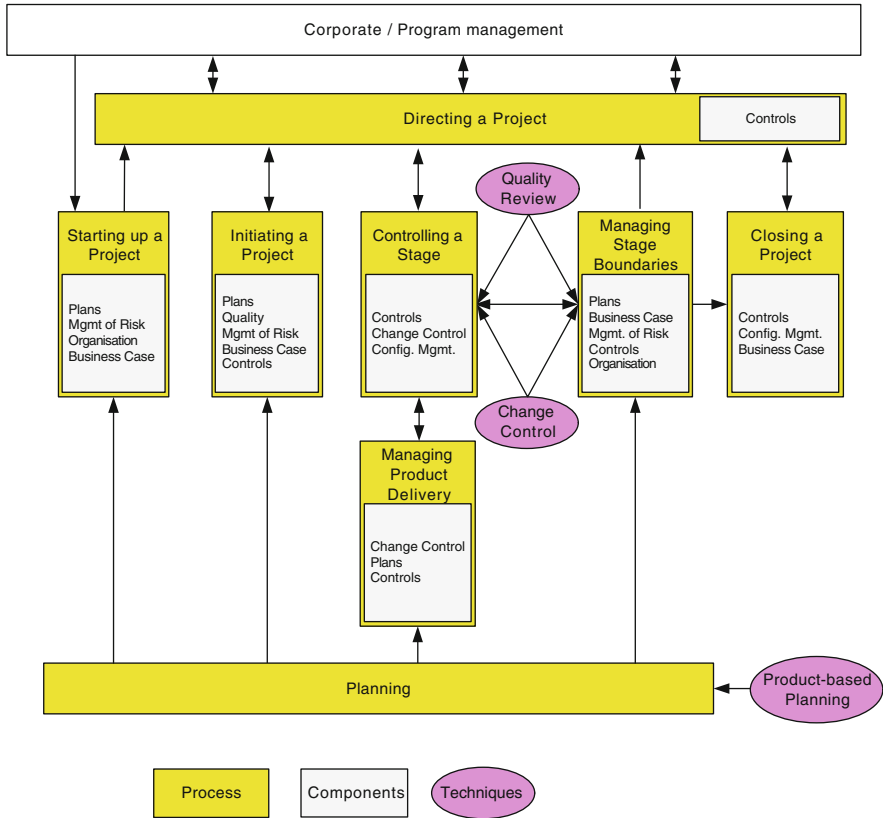
Fig. 2.6 Knowledge areas and their interaction of the PMBOK (extracted from [23])

In summary PMBOK can be characterized to be a collection of processes and knowledge areas generally accepted as best practices within PM. Its basic concepts are applicable to all projects and programs (and operations). The knowledge base provides the fundamentals of PM, irrespective of the type of project (software, construction, environmental, aerospace, etc.) and recognizes five basic process groups and nine knowledge areas typical for almost all projects.

2.3.2.2 PRINCE2®

The PRINCE2® project management method, published and maintained by the Office of Government Commerce (OGC) in the UK, is widely used and has become the standard for project management in Europe. The OGC has published several books on PRINCE2® with [18] being the most important one. Another good resource about the method is [3].

PRINCE2® is based on the approach to treat the efforts of project management as a process. It is highly product oriented by focusing on the business case and emphasizing quality aspects. The method provides a framework of components and techniques to perform the different project management processes. The controlled and systematic preparations and execution of projects in a process following PRINCE2® explicitly prevent problems of other PM methods with weaker processes. This is a central aspect and an additional benefit compared to other methods. An overview of the PRINCE2® processes, its components, and the techniques provided is presented in Fig. 2.7. The figure shows the interaction between the different processes depicted in Table 2.2. In summary PRINCE2® can be characterized to be a collection of processes, components, and techniques generally applicable to PM in diverse fields (software, construction, environmental, aerospace, etc.). Its basic concepts are applicable to all projects and programs (and operations). The method provides the fundamentals of PM, recognizing eight basic processes with sub-elements, eight components used during the different processes, and three basic techniques applied.



**Fig. 2.7** Management processes, components, and techniques of PRINCE2® (based on [3])

The PRINCE2® method benefits can be summarized as, the method

1. identifies management, specialist, and quality Products/Deliverables and helps to ensure that they are produced on time and to budget
2. focuses attention on the quality of Products/Deliverables and on the viability of the business case
3. separates the management and specialist aspects of organization, planning, and control
4. facilitates controls at all levels
5. makes the project's progress more visible to management
6. provides a communication medium for all project staff
7. ensures that work progresses in the correct sequence
8. involves senior management in the project at the right time and in the right place
9. allows the project to be stopped and, if required, re-started completely under management control, at any time in the project's life
10. is in the public domain and requires no license fee
11. has a well-established user group dedicated to the support, promotion, and strengthening of the method

**Table 2.2** PRINCE2<sup>®</sup> overview

| Process                   | Process elements                                                                                                                                                                                                                                                                                   | Components                                                          | Techniques                       |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|----------------------------------|
| Starting up a project     | SU1. Appointing a project executive and a project manager<br>SU2. Designing a project management team<br>SU3. Appointing a project management team<br>SU4. Preparing a project brief<br>SU5. Defining project approach<br>SU6. Planning initiation stage                                           | Plans<br>Mgmt. of risk<br>Organization<br>Business case             |                                  |
| Directing a project       | DP1 Authorizing initiation<br>DP2 Authorizing a project<br>DP3 Authorizing a stage or exception plan<br>DP4 Giving ad hoc direction<br>DP5 Confirming project closure                                                                                                                              |                                                                     |                                  |
| Planning                  | PL1. Designing a plan<br>PL2. Defining and analyzing products<br>PL3. Identifying activities and dependencies<br>PL4. Estimating<br>PL5. Scheduling<br>PL6. Analyzing risks<br>PL7. Completing a plan                                                                                              | Product-based planning                                              |                                  |
| Initiating a project      | IP1. Planning quality<br>IP2. Planning a project<br>IP3. Refining the business case and risks<br>IP4. Setting up project controls<br>IP5. Setting up project files<br>IP6. Assembling a project initiation document (PID)                                                                          | Plans<br>Quality<br>Mgmt. of risk<br>Business case<br>Controls      |                                  |
| Controlling a stage       | CS1. Authorizing work package<br>CS2. Assessing progress<br>CS3. Capturing project issues<br>CS4. Examining project issues<br>CS5. Reviewing stage status<br>CS6. Reporting highlights<br>CS7. Taking corrective action<br>CS8. Escalating project issues<br>CS9. Receiving completed work package | Controls<br>Change control<br>Config. Mgmt.                         | Quality review<br>Change control |
| Managing product delivery | MP1. Accepting a work package<br>MP2. Executing a work package<br>MP3. Delivering a work package                                                                                                                                                                                                   | Change control<br>Plans<br>Controls                                 | Quality review<br>Change control |
| Managing stage boundaries | SB1. Planning a stage<br>SB2. Updating a project plan<br>SB3. Updating a project business case<br>SB4. Updating the risk log<br>SB5. Reporting stage end<br>SB6. Producing an exception plan                                                                                                       | Plans<br>Business case<br>Mgmt. of risk<br>Controls<br>Organization |                                  |
| Closing a project         | CP1. Decommissioning a project<br>CP2. Identifying follow-on actions<br>CP3. Project evaluation review                                                                                                                                                                                             | Controls<br>Config. Mgmt.<br>Business case                          |                                  |

### 2.3.2.3 ISO10006

Another standard in the context of project management is ISO10006 which is documented in [13]. Up to now the standard has been released in two revisions, ISO10006:1997 and ISO10006:2003 [4]. The initial version of this standard is based on a draft of PMBOK and is therefore relatively similar to PMBOK. But the ISO standard addresses project management from the quality aspects and targets more to quality assurance in project management. On top of that it is less detailed than PMBOK and offers less help to project managers. Being based on PMBOK it shares the weakness of being more a collection of best practices, knowledge areas, and required skills. The ISO10006 focuses less on an explicit project management process and has only a very abstract and loose project execution process. Therefore it is less practical and provides more a kind of a checklist for quality-driven project management than a project management process. An additional weak point of the ISO10006 is that it misses or only very briefly covers subjects like scope, cost, time, and risk. Literature comparing PMBOK and ISO10006 [25] or analyzing ISO10006 itself [21] discourages the use of ISO10006. The later article even states that the application of ISO10006 could result in the opposite effects because of potential misinterpretations of some guidelines having no clear context. Because of these weaknesses and the very limited usage of ISO10006 in commercial environments this method is disregarded from further investigation.

### 2.3.2.4 Selection

Only PMBOK and PRINCE2® remain for consideration as a baseline project management method, after having eliminated the ISO10006 standard because of its weaknesses and its focus on quality management in projects. Investigating these two possibilities further it can be noted that PMBOK is less process centric, although it defines the five basic processes. PMBOK puts a lot of emphasis on the different knowledge areas a project manager needs to handle and approaches project management from the project manager's required skill set. This approach makes the PMBOK method very valuable for teaching project management skills. In contrast PRINCE2® is focused more on the practical guidance of a particular project. This point of view is shared by [27, 28]. This process and guidance orientation, together with the inherent management of the project in stages, makes PRINCE2® a better fit with the Stage-Gate® method than PMBOK.

An additional difference between PMBOK and PRINCE2® is that the latter emphasizes the business case and the project brief and PMBOK does not even mention one of them. The business case is a very central vehicle for the PRINCE2® method. The method enforces the constant update according to gained knowledge and changing conditions as well as its constant validation by the project board after each development stage. Therefore PRINCE2® has a higher applicability for product engineering than PMBOK. This is even emphasized further by the product, rather than task-centric planning method of PRINCE2®. A product in that context

can be any kind of outcome of a project, a tangible one, like an MNT device, or an intangible one like a redefinition of a business process. On the other hand PMBOK contains knowledge areas like human resources (HR) and procurement management which are not covered in PRINCE2®.

Another distinguishing aspect is that PRINCE2® is designed for a variety of customer and supplier scenarios while PMBOK primarily considers that the project is run by one supplier for one customer. This aspect makes the PRINCE2® method more flexibly applicable, especially for scenarios where the project is run by an instance in a sandwich position having customers and external suppliers.

Furthermore it can be noticed that PRINCE2® puts more emphasis on documentation than PMBOK. It provides a plentitude of templates or guidelines for project templates helping to become the knowledge pool for a project. Between those are the definitions of an issue log, a risk log, and acceptance criteria that are not addressed by PMBOK. On the other hand this strong emphasis on documentation can burden a project and creates a significant management overhead for small projects. This can be prevented by using PRINCE2® as a framework where a project can select the applicable standards depending on risk and size.

Summing up and concluding, it can be said that PRINCE2® is a better extending match for the Stage-Gate® method than PMBOK. This is due to its advantages in its process model, its product and business case-centric definition, and its more flexible project environment scenario. Its drawbacks of missing out certain areas, which are covered by PMBOK, can be circumvented by adding not covered ground from the PMBOK and using its more detailed documentation of best practices. This combined approach is suggested by [24, 27] as well. The documentation overhead can be reduced by relaxing the constraints for small and minor risk projects. On the other hand the focus on documentation builds a good baseline for quality certification.

### 2.3.3 *T<sup>2</sup>M*

As noted in the last section an increased involvement of the customer in the Product Engineering process is necessary. On the other hand there is also an increased need for modularity and flexibility. Every company has developed its own business processes and cannot be forced to fully adapt to external processes introduced by random customers. The goal is therefore to devise a highly customizable methodology for customer interaction that doesn't disrupt or damage established processes.

The overall approach of CORONA on Product Engineering is deliverable driven instead of task driven: "you can't control what you can't manage, and you can't manage what you can't specify" [1]. Meaning that the progress of the project is measured based on concrete artefacts (deliverables) created instead of workload/money spent. For these reasons the CORONA approach supports a more fine-granular handling of customer interaction and collaborative engineering. The Task Transition

Methodology (T<sup>2</sup>M) provides the means for organizing the engineering work within the Stages. T<sup>2</sup>M distinguishes between three basic entities:

- Tasks—describing product creation activities which are implementing
- Deliverables—which are documents or other artefacts that are checked at
- Transitions—serving as synchronization points between tasks

In short Tasks generate Deliverables that are checked at Transitions that again lead to subsequent Tasks. At first glance T<sup>2</sup>M looks quite similar to Stage-Gate®. It is definitely not the same as it will become clear when defining the concrete semantics of Tasks, Deliverables, and Transitions.

### 2.3.3.1 Tasks

In T<sup>2</sup>M Tasks represent activities that implement Deliverables or to put it in less abstract words: Tasks are activities whose results are called Deliverables, e.g., the Deliverable of a Task “Market Analysis” would be a “Market Analysis Report.” Therefore the most basic Task consists of a name that describes the overall activity, a list of Deliverables that have to be implemented, and last but not least the budget that has been assigned to perform the Task.

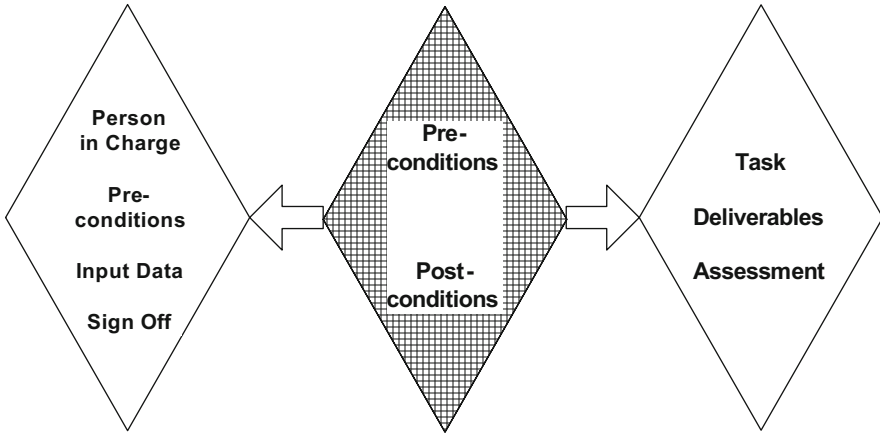
Additionally, T<sup>2</sup>M Tasks may serve as container entities that contain other (Sub-) Tasks and Transitions. This allows building up a hierarchy, which could, for example, be used to define methodology templates. It is not intended to use the hierarchy for micro managing. Upcoming issues within a Task are not escalated outside of the Task. To the outside every Task is strictly a black box.

The size and scope of each Task depends on the preferences of the involved partners. Very small Tasks lead to larger overhead and limited flexibility and should generally be avoided in projects where innovation is important. Too large Tasks are harder to control and may commence in unwanted directions. As a rule of thumb Tasks should be split at significant milestones or if the executing personnel changes.

While it is in principle possible that more than one company is involved in a single Task it is unadvisable with regard to two of the main challenges in MNT Product Engineering: different locations and IPR. It is therefore advisable to split a Task into sub-tasks till every sub-task can be performed at a single location/company and the same personnel involved.

### 2.3.3.2 Deliverables

The Deliverable of T<sup>2</sup>M is the actual outcome of a Task. Like in Stage-Gate® this has to be something tangible like a document, a computer program, or a physical artefact. Spent money or wasted time may be the main outcome of some Tasks but do NOT count as Deliverables. An additional restriction of T<sup>2</sup>M is that it forbids Deliverables that create overhead. There may never be a Deliverable that is created



**Fig. 2.8** Transition with pre- and post-conditions

only for the sake of a Transition. A Deliverable should be created to advance the project and not the process.

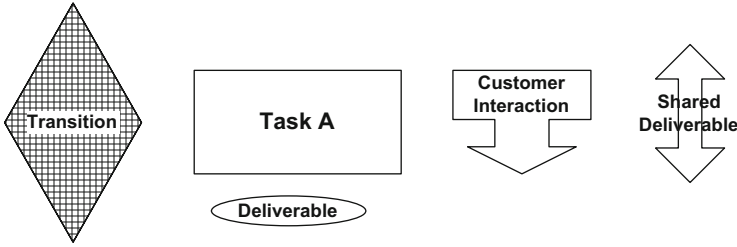
Each Deliverable has a specification, an implementation, and a rating. The specification describes the requirements and constraints of the Deliverable. The implementation is the resulting Deliverable, and the rating describes how the implementation matches the specification. In its most simple form this could be “accepted” or “rejected.”

A Deliverable is always implemented during a single Task. While under practical conditions it is possible to share an incomplete Deliverable with other Tasks the responsibility for completing the Deliverable stays at the original Task.

### 2.3.3.3 Transitions

The Transition serves as a synchronization point between Tasks. Here the results (Deliverables) of the preceding Task are collected and handed over to the subsequent Tasks. A Transition is activated if all required Deliverables are accepted. After activation of a Transition all preceding Tasks are closed and all subsequent Tasks are activated. This procedure and their pre- and post-conditions are presented in Fig. 2.8. Like Janus, the Roman god of Gates, each transition has two faces. One that looks back at the previous Tasks and checks if all specified Deliverables are accepted and another one that looks at the upcoming Task to specify the next Deliverables.

The Transition is controlled by the persons in charge of the subsequent Task(s) and, depending on individual agreements, the customer. This is reasonable as those persons are also the ones that ought to use the Deliverables of the last Task and have sufficient knowledge of the Task to provide an implementable specification.



**Fig. 2.9** 5 Symbols used in T<sup>2</sup>M diagrams

It should be noted, however, that the actual specification of the Deliverables does not take place during project execution but during project planning. The Transition provides only an opportunity for last minute changes to the specification! A Transition is also the only place where customer involvement is allowed (if the customer is not actively involved in the Task anyway). Typical contributions of the customer are result assessments and specification updates.

As mentioned earlier, Transitions are similar but not equivalent to the Gates of Stage-Gate<sup>®</sup>. In contrast to Gates, Transitions are meant to be lightweight synchronization points between Tasks with possible customer feedback. There is no detailed assessment of the Deliverables, no hard decisions, and no budget approvals. Ideally there is not even a personal meeting. Should any issues arise the transition stays inactive till the issues are resolved by the PRINCE2<sup>®</sup> processes.

### 2.3.3.4 Modeling Processes with T<sup>2</sup>M

T<sup>2</sup>M processes are modeled by concatenating Tasks and Transitions. A T<sup>2</sup>M diagram consists mostly of the four basic elements shown in Fig. 2.9.

The symbol labeled “Customer Interaction” can be added to Transitions to indicate that at this transitions the customer is involved. The actual nature of this interaction should be reflected in the symbol’s label (e.g., “notification,” “sign-off,” “review”). The double-sided arrow labeled “Shared Deliverable” is used when two parallel Tasks are so closely related that a synchronization via Transitions is too inconvenient. In this case unfinished Deliverables or other documents may be shared on mutual agreement.

A very basic T<sup>2</sup>M process with three Tasks and three Transitions is shown in Fig. 2.10. Task 2a and 2b run in parallel and are synchronized by Trans2 and Trans3. The Tasks either are used as a container for Sub-Tasks (e.g., other T<sup>2</sup>M modules) or are handled as a black box. Whatever happens inside a black box, the persons in charge for that Task has the full responsibility for that Task, including potential Sub-Tasks. The first Transition of a module is called Entry-Transition and ensures that all preconditions of the module are met. Like any other Transition it is controlled by the persons responsible for the immediately following Tasks.

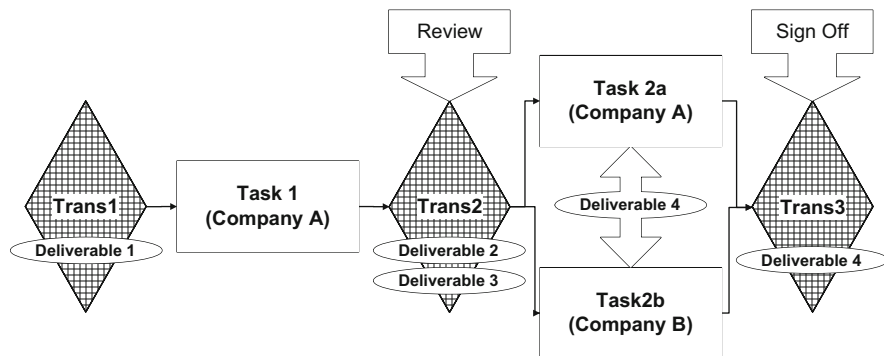


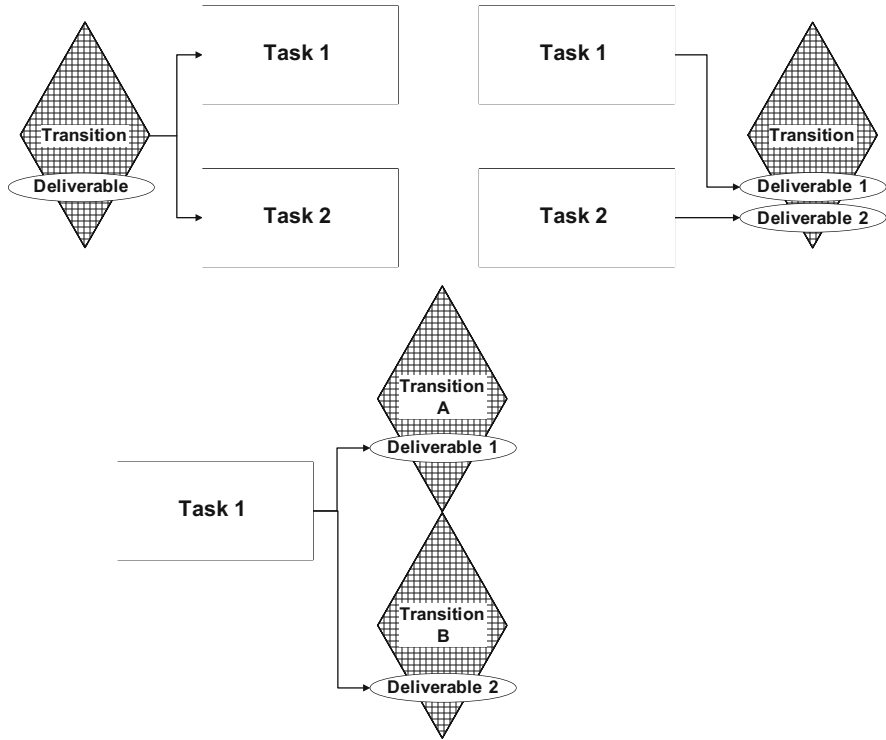
Fig. 2.10 Small example T<sup>2</sup>M

The results (Deliverables) of a Task are collected in the Transition that directly follows the Task. Here the consistency of specification and implementation of each Deliverable are checked. Only if all Deliverables of a task are accepted the Transition is activated.

As noted in Sect. 2.3.3.3 a Transition is also the only place for customer interaction in a T<sup>2</sup>M process. In a T<sup>2</sup>M diagram customer interaction is modeled by placing a customer interaction symbol right above the Transition in question. The actual nature of the customer interaction depends on individual agreements and may range from simple status reports to full Transition control. In the example diagram shown in Fig. 2.10 this has been done for Trans2 and Trans3. The customer interaction in Trans2 would be a simple review of the deliverables, so the customer has the opportunity to assess the current status and to give input to the following Tasks. The customer interaction for Trans3 is labeled “sign-off” which essentially means that a sign-off of some or all of the Deliverables by the customer is required. As a result the Transition can only be activated if the customer allows it.

Besides process control and customer interaction the main objective of the Transitions in a T<sup>2</sup>M process is Task synchronization. The synchronization aspect happens if multiple Tasks start or end at a single Transition. Some example scenario of these task synchronization aspects are presented in Fig. 2.11. More than one Task starting at a single Transition, as indicated in the upper left of Fig. 2.11, makes sense if the Tasks are closely related. Per definition a Transition is controlled by the persons responsible for the immediately following Tasks. Therefore there has to be a consensual agreement about the acceptance of the preceding Deliverables and the specification of the upcoming Deliverables. Otherwise the Transition wouldn’t be activated and the Tasks wouldn’t start. If there is a disagreement on the assessment which cannot be resolved between the people directly involved in the transition, the topic becomes escalated to the project management, according to the PRINCE2® processes.

If more than one Task ends at a single Transition, as indicated in the upper right of Fig. 2.11, the situation is similar. The Transition cannot be activated until all the

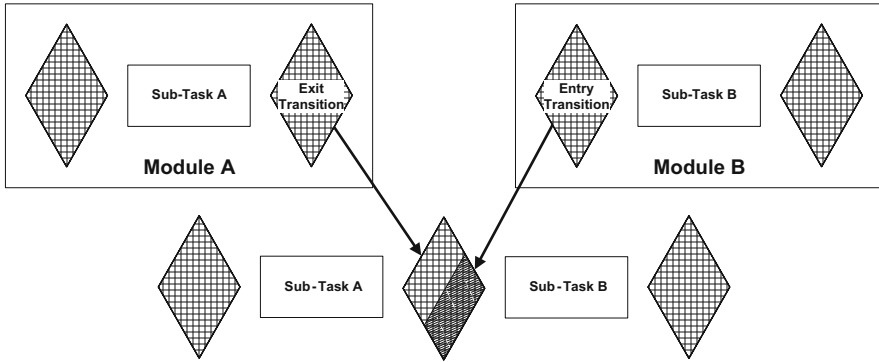


**Fig. 2.11** T<sup>2</sup>M task synchronization

Deliverables of *all* Tasks are accepted. As a result neither Task is closed, so work may continue till all interdependencies between the parallel Tasks have been sorted out. Otherwise one of the Tasks may end and the team disbanded, while the other Task still needs input.

Another possible situation is multiple Transitions following a single Task, as indicated in the bottom of Fig. 2.11. Such a situation becomes feasible if the Deliverables of that Task are used in completely independent follow-up Tasks. As a result some of the Tasks may already start while others are still waiting for their input Deliverables to be accepted. Of course the Task preceding the transition may only be closed when all Transitions have been activated.

Ideally every Deliverable is implemented during a single Task and only used by other Tasks after being accepted. In a real Product Engineering process such a clean and definite approach is often not feasible. Especially in parallel development tasks, e.g., MEMS- and IC-components, a need for very fine granular information exchange may arise. In a T<sup>2</sup>M diagram such a tight coupling between two Tasks is represented by a “shared deliverable” symbol. A shared Deliverable means that while the responsibility for implementing the Deliverable still remains at the original Task, intermediate states of the Deliverable are shared with the other Tasks. As this



**Fig. 2.12** Combining T<sup>2</sup>M modules

happens without T<sup>2</sup>M synchronization mechanisms, information consistency has to be provided by other technical or nontechnical means, e.g., by an electronic project binder.

### 2.3.3.5 T<sup>2</sup>M Modularization

This book aims to provide “Best Practices” for certain organizational situations that may occur during cooperative Product Engineering. To be able to model some of these practices using T<sup>2</sup>M processes a modularization concept is needed. Fortunately, modularization with T<sup>2</sup>M is quite straightforward following a few simple rules. A T<sup>2</sup>M module is basically a Task that contains at least two dedicated Transitions: The first Transition of a module is called Entry-Transition and ensures that all preconditions of the module are met. Like any other Transition it is controlled by the persons responsible for the immediately following Tasks. The main difference to “normal” Transitions is that there are no Deliverables to be accepted, which is quite logical as there is no preceding Task to implement them. The functionality of the Entry-Transition is therefore limited to the planning of the subsequent Tasks. The Exit-Transition, on the other hand, has no subsequent Task. Therefore the Exit-Transition lacks all aspects concerned with planning of subsequent Tasks. Modeling T<sup>2</sup>M processes from T<sup>2</sup>M modules therefore means combining the Exit- and Entry-Transitions of two neighboring modules into a full Transition as shown in Fig. 2.12.

Overall, because the T<sup>2</sup>M method is embedded into the Stage-Gate<sup>®</sup> method a quite hierarchical approach can be modeled in a consistent and comprehensive way. The Stage-Gate<sup>®</sup> method provides a general framework for innovation, guided by PRINCE2<sup>®</sup> project management process realized through breaking down the Stages into a product/project-specific T<sup>2</sup>M model for realization.

**Table 2.3** Responsibilities in the preliminary investigation stage (example)

|                                          | R&D management | Development management | Manufacturing management | Design team | Process team | Package team | Processing engineers | Operators | M&S | Legal | Financial | Project management |
|------------------------------------------|----------------|------------------------|--------------------------|-------------|--------------|--------------|----------------------|-----------|-----|-------|-----------|--------------------|
| Preliminary technical feasibility report | A              | I                      |                          | R           | R            | R            | S                    |           |     |       |           | I                  |
| Preliminary market assessment report     | C              |                        |                          |             |              |              |                      |           | R   |       |           | I                  |
| Preliminary prospects report             | C              |                        |                          |             |              |              |                      |           | R   |       |           | I                  |
| Preliminary partner list                 | C              |                        |                          |             |              |              |                      |           | R   |       |           | I                  |
| Preliminary legal assessment report      | C              |                        |                          | C           | C            | C            |                      |           |     | R     |           | I                  |
| Draft project mandate                    | C              |                        |                          | C           | C            | C            |                      |           |     |       |           | R                  |
| Project initialization stage plan        | C              |                        |                          | C           | C            | C            |                      |           |     |       |           | R                  |
| Req. for authorization to proceed        |                |                        |                          |             |              |              |                      |           |     |       |           | R                  |

2.3.4 Responsibility Assignment Matrix

One of the most important tasks in modeling an innovation process is the assignment of responsibilities. Naturally in a strictly hierarchical work environment there is a kind of pyramid structure with a small number of high ranked executives at the top (CEO, CTO, etc.) and a larger number of project workers with a limited decision range at the bottom. For cooperative distributed engineering projects the situation is not that easy as the responsibility is shared among the various partner companies and the customer. The customer is in a special situation as, while on top of the hierarchy in our PE approach, he/she has basically no responsibility on the project results.

In our PE approach the responsibility is assigned and executed at decision points (Gates, Transitions) and connected with Deliverables. A common method to describe and identify responsibilities is the responsibility assignment matrix (RAM) or RACI-matrix, which describes the involvement of project partners in project Tasks and Deliverables. There are several ways to describe this involvement. Most common is the RACI method which classifies the involvement into one of the four categories *responsible (R)*, *accountable (A)*, *consulted (C)*, and *informed (I)*. It is described in more detail in [10, 23]. Responsible is used to describe the parties who are doing the work to achieve the Task. Accountable is the party who is ultimately accountable for the results of the work. There can be only one party accountable for any given Task or Deliverable. Consulted stands for parties who contribute with their experience or knowledge but are not directly involved. Informed stands for parties that are just informed about the current state or the results of the work. In contrast to consulted they are not expected to generate feedback. An example of an RACI matrix as used in Sect. 5.3.3 is given in Table 2.3.

There are several alternative approaches to RACI that extend the participation model and can be used to provide a more fine granular distinction of responsibilities. Most notably are RASCI (documented in, e.g., [10]), which adds a *support* (*S*) category that describes parties who are assigned to *responsible* (*R*) to assist in completing the work, and CAIRO, which adds a *omitted* (*O*) category for parties that are explicitly not to be involved.

Due to the complex nature of cooperative distributed engineering we distinguish between three hierarchical levels. The top-level describes how the responsibilities are shared among the project partners. The matrix documents the role each partner has at the project management board, the individual Gates and the top-level Deliverables (e.g., IC, MEMS, and package). This is clearly a level where no individuals but company departments appear on the matrix. The matrix distinguishes between the four partner types of customer, developer, supplier, and user. The middle level is mostly about individual executive responsibility. Here the contact-persons of each company/department are assigned to roles within project management, Gates and T<sup>2</sup>M Transitions. The third level deals with the responsibilities of the individual developer that are linked to the implementation of concrete Deliverables.

## References

1. Allen, P., Higgins, S., McRae, P., Schlamann, H.: Service Orientation: Winning Strategies and Best Practices, 1st edn. Cambridge University Press, Cambridge (2006)
2. Anderson, R.E.: Phased product development: friend or foe? *Business Horizons* **39**(6), 30–36 (1996)
3. Bentley, C.: PRINCE2 a Practical Handbook, 3rd edn. ButterworthHeinemann, Oxford (2009)
4. British Standard Institute: ISO 10006:2003: Quality management systems – Guidelines for quality management in projects (2003)
5. Buxton, W.: Performance by design: the role of design in software product development. In: Proceedings of the Second International Conference on Usage-Centered Design, pp. 1–15. Constantine & Lockwood Ltd., Rowley (2003)
6. Cooper, R.G.: Product Leadership – Creating and Launching Superior New Products, 1st edn. Perseus Books, New York (1998)
7. Davidsen, B.A.: Innovation and Product Development - Methods and Tools. *Elektronikk* **100**(2) (2004)
8. Department of Defense: Dod Guide to Integrated Product and Process Development. Office of the Secretary of Defense (OSD), Washington (1996)
9. DRM Associates, PD-TRAK® NPD Process Tools, <http://www.npd-solutions.com/pdtrak.html> (2011). Accessed 24 Jan 2013
10. Hightower, R.: Internal Controls, Policies and Procedures, 1st edn. Wiley, Hoboken (2008)
11. Institute, P.M.: A Guide to the Project Management Body of Knowledge, 3rd edn. Project Management Institute, Newtown Square (2004)
12. Institute, P.M.: Organizational Project Management Maturity Model (OPM3), 2nd edn. Project Management Institute, Newtown Square (2013)
13. International Organization for Standardization: ISO 10006: Quality management systems – Guidelines for quality management in projects (2003)
14. Janhager, J.: User Consideration in Early Stages of Product Development - Theories and Methods. PhD Thesis, KTH Machine Design (2005)

15. Karbhari, V.M.: Product and process development methods. JTEC Panel Adv. Manuf. Technol. Polym. Compos. Struct. Jpn. (1994)
16. Lövgren, R.: Product development methods: Leonardo da vinci versus modern engineering education. In: The Renaissance Engineer of Tomorrow: Firenze, 8–11 September 2002: Proceedings of the 30th SEFI Annual Conference. Edizioni Polistampa, Firenze (2002)
17. MEMSIndustryGroup: Technology development process template. [http://www.memsindustrygroup.org/i4a/doclibrary/getfile.cfm?doc\\_id=373](http://www.memsindustrygroup.org/i4a/doclibrary/getfile.cfm?doc_id=373) (2012). Accessed 10 March 2013
18. OGC: Managing Successful Projects with PRINCE2, 4th edn. The Stationery Office Ltd, Norwich (2009)
19. Olson, E.L., Bakke, G.: A comparison of new product development practices in North America and Norway. *Teletronikk* **100**(2), 120–125 (2004)
20. Ortloff, D.: Product engineering for silicon based MEMS IP. PhD Thesis, University of Siegen (2006)
21. Project Management Partners: ISO 10006: risky business. <http://www.pmpartners.com/resources/iso10006.html> (2005). Accessed 27 May 2013
22. Pugh, S.: Total Design: Integrated Methods for Successful Product Engineering, 1st edn. Addison-Wesley, Reading (1991)
23. Schwalbe, K.: PRINCE2 a Practical Handbook, 4th edn. Kathy Schwalbe LLC, Minneapolis, Minnesota (2013)
24. Siegelau, Jay: PMBOK and PRINCE2. <http://www.slideshare.net/sundong/prince2-vs-pmbok> (2005). Accessed 24 January 2013
25. THE WELCH COMPANY: Reviewed ISO 10006 on project management, compared PMBOK. <http://www.welchco.com/sd/08/00101/02/95/07/21/134937.HTM> (1995). Accessed 24 January 2013
26. Ulrich, K.T.: Product Design and Development, 1st edn. McGraw Hill, Columbus, OH (2003)
27. Verpoorten, J.H.: Projectmanagement – PRINCE2 versus PMBOK. [http://www.cs.uu.nl/docs/vakken/pm/sheets/college7\\_1.pdf](http://www.cs.uu.nl/docs/vakken/pm/sheets/college7_1.pdf) (2005). Accessed 22 Aug 2013
28. Wideman, M.: Comparing PRINCE2 with PMBOK. AEW Services, Vancouver (2002)

MEMS Product Engineering

Handling the Diversity of an Emerging Technology. Best  
Practices for Cooperative Development

Ortloff, D.; Schmidt, T.; Hahn, K.; Bieniek, T.; Janczyk, G.;  
Brück, R.

2014, XXVI, 195 p. 63 illus. in color., Hardcover

ISBN: 978-3-7091-0705-8